

MendNet: Restoration of Fractured Shapes Using Learned Occupancy Functions

N. Lamb , S. Banerjee , and N. K. Banerjee

Clarkson University, USA

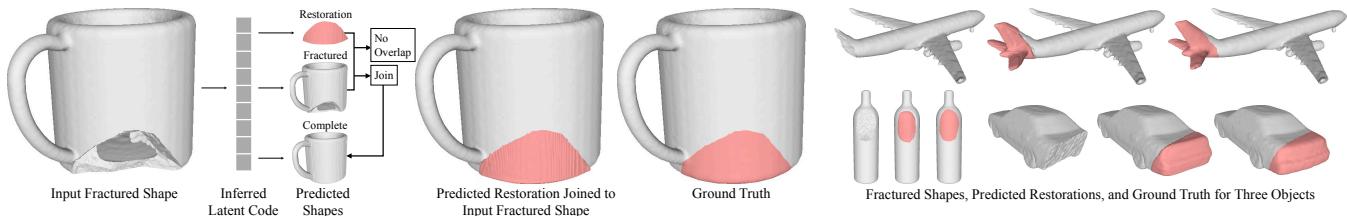


Figure 1: Given an input fractured shape, we provide a novel approach to automatically infer a latent code and predict a restoration from the code. Our approach encourages that the predicted restoration avoid overlap with the fractured shape regenerated using the latent code, and that the predicted restoration and fractured shapes join to form a complete shape estimated from the code. Our approach provides restorations that show a close join to the fractured shape.

Abstract

We provide a novel approach to perform fully automated generation of restorations for fractured shapes using learned implicit shape representations in the form of occupancy functions. Our approach lays the groundwork to perform automated object repair via additive manufacturing. Existing approaches for restoration of fractured shapes either require prior knowledge of object structure such as symmetries between the restoration and the fractured object, or predict restorations as voxel outputs that are impractical for repair at current resolutions. By leveraging learned occupancy functions for restoration prediction, our approach overcomes the curse of dimensionality with voxel approaches, while providing plausible restorations. Given a fractured shape, we fit a function to occupancy samples from the shape to infer a latent code. We apply a learned transformation to the fractured shape code to predict a corresponding code for restoration generation. To ensure physical validity and well-constrained shape estimation, we contribute a loss that models feasible occupancy values for fractured shapes, restorations, and complete shapes obtained by joining fractured and restoration shapes. Our work overcomes deficiencies of shape completion approaches adapted for repair, and enables consumer-driven object repair and cultural heritage object restoration. We share our code and a synthetic dataset of fractured meshes from 8 ShapeNet classes at: <https://github.com/Terascale-All-sensing-Research-Studio/MendNet>.

CCS Concepts

- Computing methodologies → Shape analysis; Neural networks;

1. Introduction

There is a wealth of examples in the real world of objects with functional, historical, or sentimental value that have undergone damage and are in need of repair. Damage occurs in many forms. During normal use an object may be dropped, causing it to fracture. Objects may disassembled and their parts mislaid resulting in part loss. Objects may undergo heavy use or suffer damage from the elements causing deformation and weathering. Fractured objects may be reassembled if their parts can be found, though

in some cases part loss also occurs, e.g. the broken fish in Figure 2. The fracturing process may crush parts, creating fragments that are too small to be reassembled and leaving gaps between larger parts, as occurs for the sugar jar in Figure 2. Objects with weathering damage must be manually rebuilt using other materials. Users are likely to attempt to repair objects with sentimental value or with functional use if the repair process requires minimal effort. However, the majority of current work in shape repair requires a user to generate a repair part manually, e.g. for



Figure 2: Examples of objects that have undergone fracture damage. Image links and licensing information are provided in the Appendix. Fractured objects may have sentimental value to the owner, such as the plate, cup, and grave marker. Parts may be lost or destroyed when the fracture takes place, as occurs for the fish, marble table, flowerpot, and sugar jar. Objects may be fractured and eroded, as occurs for the grave marker and Calyx Krater. Specialized users with modeling experience may attempt to repair damaged objects manually with other materials, e.g. the Calyx Krater.

a terracotta artifact [SCC*11], a silver crown [SAF*18], a fruit bowl [AES*11], a dinosaur vertebra [SJW*14], a dental prosthesis [RMV*10], or the Calyx Krater [Mus22] shown in the bottom right of Figure 2, which requires expertise in 3D modeling or artistic design that is out of the scope of the average consumer. Ad hoc object repairs by users without specialized experience are unlikely to be sustainable. Some approaches propose domain-specific repair pipelines, e.g. for aerospace components [GCGY08; ZLC06], or medical implants [HHN07; WKCF16], which are unlikely to generalize to objects outside of that domain. The scientific study of damage is largely restricted to microscale analysis in material sciences [ANZ06; ZQ18; SMZG18; FLF*19], while consumer-oriented repair of damaged objects remains an understudied field. A gap remains in the macroscopic analysis of shape to enable rapid consumer-driven repair of damaged objects.

In this work we present MendNet, a fully automated approach that generates restoration 3D models for 3D shapes, as shown on the right of Figure 1, corresponding to 3D models that have undergone fracture-based damage by learning the relationship between fractured, restoration, and complete shapes. The restoration of fractured shapes has significant impact in fields such as industrial recycling [ZLC06], cultural heritage [LKL*20; PSA*17; SCC*11; SAF*18], paleontology [SJW*14], medical applications such as dentistry, orthopedics and reconstructive surgery [HHN07; RMV*10; WKCF16], and consumer-focused repair.

A related though fundamentally different problem than fractured shape repair that has seen substantial prior work is incomplete shape completion [PFS*19; YWC*21; TTG*20; ZYDL21; DWM*22; LWL20; HASB20; CZ19; BLRW16; SGF16; WZX*16; SM17; DRB*18; FMJB16; YWW*17; SG20; LPS*16; SG18; MHLZ20; MON*19; PNM*20; CAP20; GCV*19; GCS*20]. Approaches to perform incomplete shape completion assume that the

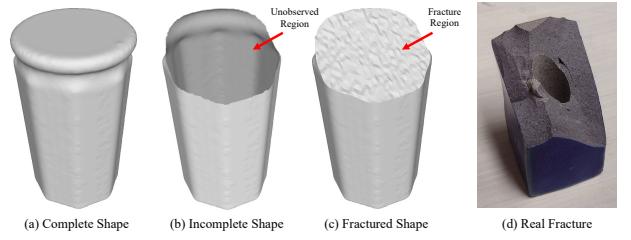


Figure 3: (a) Compared to the complete shape, (b) the incomplete shape is a open manifold that is a strict subset of the complete shape. (c) The fractured shape includes novel geometry at the fracture region not present in the complete shape. We use synthetic fractures that mimic the geometry of (d) real fractures.

input to the approach is an incomplete shape observation, such as a single image, a depth map, or an open manifold such as the one shown in Figure 3(b), which are strict subsets of the corresponding complete shape, shown in Figure 3(a). Our approach addresses the challenge that fractured shapes introduce a novel fracture region, shown in Figure 3(c), that is not present in the corresponding complete shape, by learning to infer a restoration shape code from an input fractured shape, as shown on the left of Figure 1. Though a small body of prior work exists to generate restorative parts for fractured objects, these approaches depend on prior knowledge of shape structure, e.g., a precise pre-existing 3D replica [LBB19] that may not always exist, or require that the object be symmetrical and the damage occur asymmetrically [SLL*08; GSP*14; PSA*17]. The work of Hermoza and Sipiran [HS18] is the only known approach to generate repair parts for damaged shapes without structural constraints. The approach uses voxel representations with outputs that are too coarse for repair. MendNet generates high-resolution restorations by leveraging learned continuous domain representations that implicitly represent shape surfaces, i.e., occupancy functions, using deep networks.

The first novel contribution of our work in restoration generation is the formalization of logical relationships that (a) ensure the restoration is physically valid, i.e., that it does not intersect with the fractured shape, and that (b) impose bounds on restoration shapes learned from small datasets by leveraging the complete shape, whose space is easier to span with limited data. We impose boundedness by encouraging that the occupancy set of a complete shape is the union of occupancy sets of fractured and restoration shapes. To represent the relationship between fractured and restoration, and fractured and complete shapes, our second contribution is a set of deep networks that map the fractured shape code inferred using learned occupancy functions to latent codes for the restoration and complete shapes. We use the occupancy function to represent shapes, instead of the more common signed distance function (SDF), to apply logical relationships of validity and boundedness to the restoration shape. Our third contribution is a set of loss functions that simultaneously estimate parameters for occupancy and latent code transformation networks, and that encode point occupancy feasibility for predicted restoration shapes using a T-norm [GQ91] relaxation of our formalized logical relationships.

Collecting a dataset of real-world fractured and corresponding complete shapes is challenging due to the time-consuming nature of 3D scanning and the difficulty of aligning fractured and complete shapes. Since no large-scale real-world dataset exists, we create a novel dataset of synthetically generated fractured shapes spanning 8 classes from the ShapeNet dataset [CFG*15]. We generate synthetic fractures, e.g. Figure 3(c), that mimic the texture of fractures that occur for brittle materials such as ceramic or clay, e.g. Figure 3(d). We use the mugs, jars, and bottles classes to represent common household objects that can suffer damage, and the cars, planes, sofas, chairs, and tables classes to capture objects of diverse geometry. We demonstrate results of using a ShapeNet-trained network to generate restoration shapes for synthetically fractured models from the QP cultural heritage dataset [KPA*09]. We provide an evaluation of our approach against 3D-ORGAN [HS18] and baseline approaches based on shape completion and subtraction using DeepSDF [PFS*19] and PoinTr [YRW*21] with ConvONet [PNM*20]. We provide a quantitative analysis of the failure cases of our approach, and an ablation study justifying our loss functions. We share our synthetic dataset and code at: <https://github.com/Terascale-All-sensing-Research-Studio/MendNet>.

2. Related Work

2.1. Automated Fractured Shape Restoration

Despite its impact in repair, limited work exists on automated restoration of fractured shapes. Some approaches restore fractured shapes by leveraging complete shapes generated automatically using symmetries [SLL*08; GSP*14; PSA*17]. They fail if damaged parts of the shape are non-symmetric to intact regions of the shape. The 3D-ORGAN approach of Hermoza and Sipiran [HS18] uses a generative adversarial network to predict voxelized complete shapes that may be used to restore fractured shapes. However, the voxel grids in their approach lack the resolution required to produce physically realizable restoration shapes. The implicit shape representation in our work removes reliance on voxel space and enables tractable high-resolution mesh generation.

2.2. Incomplete Shape Completion

Unlike fractured shapes whose fracture surface distinguishes them from their complete counterparts, incomplete shapes consist of open surfaces that are subsets of complete shapes. Most voxel-based incomplete shape completion methods [BLRW16; SGF16; WZX*16; SM17; DRB*18; FMJB16; YWW*17; SG20; LPS*16; SG18] struggle to predict high-resolution outputs due to the curse of dimensionality. Though some advancements have been made in computational efficiency of voxel-based methods using hierarchical models [DDN20; DRB*18] or sparse convolutions [DDN20; YGF21], these approaches still require a pre-discretization of the domain, and even using scalable voxel-based approaches the level of resolution required to accurately represent the fracture region may be infeasible. While more efficient, point cloud approaches [ADMG18; SLK19; LSY*20; SK20; LCL18; YRW*21] require mesh generation to be usable for repair. The resulting meshes may not satisfy non-intersection with the fractured shape.

Learned implicit shape representations, such as deep networks for signed distance functions (SDFs) [PFS*19; SCT*20; YWC*21; TTG*20; ZYDL21; LWL20; MHLZ20], occupancy functions [MON*19; CZ19; PNM*20; JK20; CAP20; LESP21; LDG18], level sets [GCV*19; GCS*20], signed directional functions [ZA21] and unsigned distance functions [TLX*21] have gained traction due to the compactness of their representations compared to voxels, and their capture of bulk shape in contrast to point clouds. Park et al. [PFS*19] provide an autodecoder architecture to represent SDF in terms of latent shape codes. They provide shape completion by inferring a latent code based on point samples of an incomplete shape. Later approaches improve autodecoder inference speed using meta-learning [SCT*20] and use novel samples to tune latent codes and network parameters [YWC*21]. Hierarchical approaches combine intermediate implicit shape representations, representing object regions with global functions that represent shapes as structured sets of elements [GCS*20; GCV*19], or represent shapes as a function of multiple deformable patches [TTG*20]. Approaches using learned occupancy functions [MON*19; CZ19; PNM*20; LWL20] reconstruct complete shapes from point clouds, voxel grids, or images.

While restoration approaches apart from the low-fidelity method of 3D-ORGAN are non-existent, one may imagine a multi-stage restoration generation approach, where the fracture region is removed from the fractured shape and the fractured shape is fed to a shape completion approach or where a shape completion approach is trained to generate complete shapes from fractured shapes, and a restoration shape is generated by subtracting the fractured shape from the predicted complete shape. However, as we demonstrate in Section 7, deviations in the inferred complete shape from the input fractured shape introduce small artifacts on the surface of the fractured shape. Artifacts cannot be avoided unless the complete shape is predicted such that it represents the fractured shape in the non-fractured region with perfect accuracy, which is computationally intractable. Precise cleanup of artifacts requires manual intervention, or relies on heuristics such as thresholds for small component removal. Manual cleanup techniques are impractical for propagation to consumer spaces where users lack 3D shape manipulation expertise, and approaches based on heuristics are unlikely to generalize. We provide a fully automated approach that leverages shape relationships to generate artifact-free restorations.

3. Functional Shape Representation

We define a shape as a closed set \mathcal{S} of points in 3D space. For a 3D point $\mathbf{x} \in \mathbb{R}^3$, the occupancy $o_{\mathcal{S}}(\mathbf{x}) \in \{0, 1\}$ at point \mathbf{x} is 1 if \mathbf{x} is in the interior of \mathcal{S} and 0 otherwise. $\mathcal{S} \in \{\mathcal{F}, \mathcal{R}, \mathcal{C}\}$ where \mathcal{F} , \mathcal{R} and \mathcal{C} are the fractured, restoration, and complete shapes respectively. We use the autodecoder architecture of DeepSDF [PFS*19] to represent the occupancy $o_{\mathcal{S}}(\mathbf{x})$ as a function, i.e., as

$$o_{\mathcal{S}}(\mathbf{x}) = f_{\Theta}(\mathbf{z}_{\mathcal{S}}, \mathbf{x}), \quad (1)$$

where f_{Θ} is parametrized on the point \mathbf{x} and a latent code $\mathbf{z}_{\mathcal{S}}$ representing the shape \mathcal{S} . We represent f_{Θ} using a multilayer perceptron (MLP) with weights Θ based on the autodecoder architecture of DeepSDF. The autodecoder architecture encodes a shape by performing optimization during inference over observations of

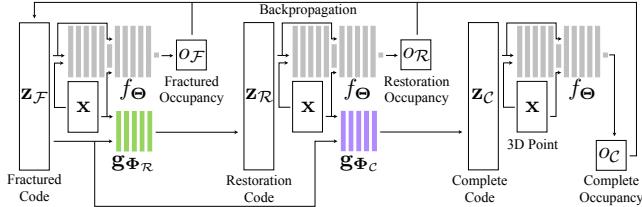


Figure 4: Network architecture. The occupancy function f_Θ is trained to output an occupancy value o_F , o_R , and o_C from the 3D point \mathbf{x} and the latent codes \mathbf{z}_F , \mathbf{z}_R , and \mathbf{z}_C respectively. The networks \mathbf{g}_{Φ_R} and \mathbf{g}_{Φ_C} transform \mathbf{z}_F to \mathbf{z}_R and \mathbf{z}_C . Backpropagation updates the gradients of the network weights during training, and the gradients of \mathbf{z}_F during training and inference.

a shape to obtain a latent code. We use the MLP autodecoder architecture, instead of e.g. a convolutional architecture, as it allows our approach to apply constraints to the restoration shape during the optimization process at inference time, as discussed in Section 5. Our motivation to set the output of f_Θ to occupancy instead of SDF as in DeepSDF is to represent the occupancy relationships between \mathcal{F} , \mathcal{R} , and \mathcal{C} discussed in Section 4. Similar to DeepSDF, we infer $\mathbf{z}_F \in \mathbb{R}^p$ using observations of the input fractured shape \mathcal{F} where p represents the dimension of \mathbf{z}_F . Since \mathcal{R} and \mathcal{C} are not known for an input \mathcal{F} , we express the latent codes of \mathcal{R} and \mathcal{C} , $\mathbf{z}_R \in \mathbb{R}^p$ and $\mathbf{z}_C \in \mathbb{R}^p$, using transformation functions \mathbf{g}_{Φ_R} and \mathbf{g}_{Φ_C} , where

$$\mathbf{z}_R = \mathbf{g}_{\Phi_R}(\mathbf{z}_F, \mathbf{x}), \text{ and } \mathbf{z}_C = \mathbf{g}_{\Phi_C}(\mathbf{z}_F, \mathbf{x}). \quad (2)$$

Similar to f_Θ , we represent \mathbf{g}_{Φ_R} and \mathbf{g}_{Φ_C} using MLPs with weights Φ_R and Φ_C . Though \mathbf{z}_R and \mathbf{z}_C represent global shape descriptors, we find that conditioning them on the input point \mathbf{x} yeilds a lower chamfer distance. Figure 4 shows our network architecture. Network f_Θ provides the latent code and the 3D point \mathbf{x} to 11 dense layers. Layers 1 to 5 and 7 to 11 contain 512 units each and layer 6 contains $509 - p$ units with a skip connection. We use the leaky rectified linear unit (ReLU) for intermediate layer activations, and sigmoid for the last layer. Networks \mathbf{g}_{Φ_R} and \mathbf{g}_{Φ_C} each have 5 dense layers with 512 units per layer and leaky ReLU for activation.

4. Occupancy Relationships

Shape sets \mathcal{F} , \mathcal{R} , and \mathcal{C} are related by the conditions that \mathcal{F} and \mathcal{R} should not intersect, i.e., $\mathcal{F} \cap \mathcal{R} = \emptyset$ and that \mathcal{C} is a union of \mathcal{F} and \mathcal{R} , i.e., $\mathcal{C} = \mathcal{F} \cup \mathcal{R}$. At a point \mathbf{x} , the set conditions impose that feasible options for o_F , o_R , and o_C are that they are either all 0, or one of o_F and o_R is 1 for mutual exclusion, in which case o_C is 1. Figure 5(a) illustrates feasible relationships for various points in the space containing the three shapes. As shown in Figure 5(b), all other choices of values for o_F , o_R , and o_C are physically impossible. Figure 5(c) shows the truth table for the feasibility of each choice of o_F , o_R , and o_C as a binary value. To leverage the feasibility conditions in a loss function, we obtain the Boolean expression corresponding to the feasibility captured by Figure 5(c) as

$$\begin{aligned} & (\neg o_F \wedge \neg o_R \wedge \neg o_C) \vee (o_F \wedge \neg o_R \wedge o_C) \vee \\ & (\neg o_F \wedge o_R \wedge o_C) = 1, \end{aligned} \quad (3)$$

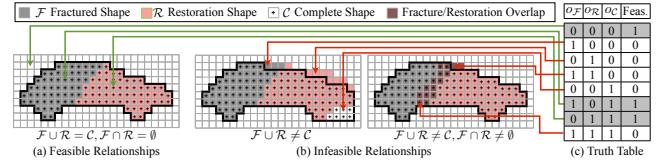


Figure 5: (a) Occupancy values for feasible relationships between the fractured \mathcal{F} , restoration \mathcal{R} , and complete \mathcal{C} shapes. The non-intersection and union conditions impose that occupancy values are either all 0, or one of \mathcal{F} and \mathcal{R} have occupancy of 1 in which case the occupancy of \mathcal{C} is 1. (b) Occupancy values for infeasible relationships. Overlap causes \mathcal{F} and \mathcal{R} to have occupancy of 1. When the union condition is not met, occupancy of \mathcal{C} is opposite of that for feasible triplets with occupancies of \mathcal{F} and \mathcal{R} . (c) Truth table showing feasibility (Feas.). Feasible choices are in gray.

where operators \wedge , \vee , and \neg represent logical and, or, and not. To use the expression in Equation (3) with the continuous values provided as the output of f_Θ in Equation (1), we relax the logical relationship using the product T-norm [GQ91], as

$$h_{\text{feas}}(o_F, o_R, o_C) = ((1 - o_F)(1 - o_R)(1 - o_C)) + (o_F(1 - o_R)o_C) + ((1 - o_F)o_R o_C). \quad (4)$$

We predict occupancy values for \mathcal{F} , \mathcal{R} , and \mathcal{C} so that we can use Equation (4) to constrain the restoration shape during inference.

5. Network Optimization and Inference

We use a dataset of training tuples, $(\mathcal{F}, \mathcal{R}, \mathcal{C})$, of fractured, restoration, and complete shapes to estimate the weights of the networks f_Θ , \mathbf{g}_{Φ_R} , and \mathbf{g}_{Φ_C} , and the fractured shape codes. We use the Adam optimizer [KB14] to optimize the loss

$$\mathcal{J} = \sum_{\mathbf{z}_F \in \mathcal{Z}} (\mathcal{J}_{\text{data}} + \mathcal{J}_{\mathcal{RC}} + \lambda_{\text{feas}} \mathcal{J}_{\text{feas}} + \lambda_{\text{reg}} \mathcal{J}_{\text{reg}}) \quad (5)$$

over the set \mathcal{Z} of latent codes for all training fractured shapes and over Θ , Φ_R , and Φ_C . In Equation (5), $\mathcal{J}_{\text{data}}$, represented as

$$\mathcal{J}_{\text{data}} = \sum_{\mathbf{x} \in \mathcal{X}} l(f_\Theta(\mathbf{z}_F, \mathbf{x}), o_F(\mathbf{x})), \quad (6)$$

models the loss of reconstructing the ground truth occupancy of \mathcal{F} from a sampling of points \mathcal{X} around each shape. $\mathcal{J}_{\mathcal{RC}}$, given as

$$\mathcal{J}_{\mathcal{RC}} = \sum_{\mathbf{x} \in \mathcal{X}} \sum_{\mathcal{S} \in \{\mathcal{R}, \mathcal{C}\}} l(f_\Theta(\mathbf{g}_{\Phi_S}(\mathbf{z}_F, \mathbf{x}), \mathbf{x}), o_S(\mathbf{x})), \quad (7)$$

models the data loss of reconstructing \mathcal{R} and \mathcal{C} . Function l represents the binary cross entropy loss, and $o_S(\mathbf{x})$ represents the ground truth occupancy for shape $S \in \{\mathcal{F}, \mathcal{R}, \mathcal{C}\}$. $\mathcal{J}_{\text{feas}}$ in Equation (5) induces the feasibility function h_{feas} from Equation (4) to be 1 to satisfy the feasibility condition expressed by Equation (3). We replace the occupancy values in the expression for h_{feas} in Equation (4) with the functional representations from Equations (1) and (2), and set up the feasibility loss as

$$\begin{aligned} \mathcal{J}_{\text{feas}} = & \sum_{\mathbf{x} \in \mathcal{X}} l(h_{\text{feas}}(f_\Theta(\mathbf{z}_F, \mathbf{x}), \\ & f_\Theta(\mathbf{g}_{\Phi_R}(\mathbf{z}_F, \mathbf{x}), \mathbf{x}), f_\Theta(\mathbf{g}_{\Phi_C}(\mathbf{z}_F, \mathbf{x}), \mathbf{x})), 1). \end{aligned} \quad (8)$$

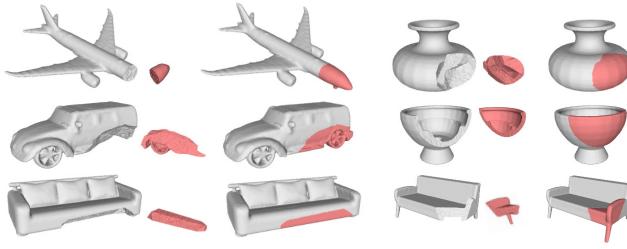


Figure 6: Ground truth fractured and restoration meshes, in gray and red, using simple ceramic fractures.

The regularizer, \mathcal{J}_{reg} , forces latent encodings to be concentrated according to a zero mean Laplacian distribution as

$$\mathcal{J}_{\text{reg}} = \|\mathbf{z}_{\mathcal{F}}\|_1 + \sum_{S \in \{\mathcal{R}, \mathcal{C}\}} \|\mathbf{g}_{\Phi_S}(\mathbf{z}_{\mathcal{F}}, \mathbf{x})\|_1. \quad (9)$$

Hyperparameters λ_{feas} and λ_{reg} represent the weights on the feasibility loss and regularizer.

During inference, given a fractured shape \mathcal{F} and a point sampling \mathcal{X} , we infer the latent code $\mathbf{z}_{\mathcal{F}}$ by optimizing the loss

$$\mathcal{J} = \mathcal{J}_{\text{data}} + \lambda_{\text{feas}} \mathcal{J}_{\text{feas}} + \lambda_{\text{reg}} \mathcal{J}_{\text{reg}} + \lambda_{\text{nzer}} \mathcal{J}_{\text{nzer}} \quad (10)$$

in $\mathbf{z}_{\mathcal{F}}$. In some cases we find that all occupancy values for the restoration are predicted as zero if the predicted complete and fractured shapes are similar to each other, or if the optimization process fails to converge. We include a non-zero loss $\mathcal{J}_{\text{nzer}}$,

$$\mathcal{J}_{\text{nzer}} = \sum_{\mathbf{x} \in \mathcal{X}} l(f_{\Theta}(\mathbf{g}_{\Phi_{\mathcal{R}}}(\mathbf{z}_{\mathcal{F}}, \mathbf{x}), \mathbf{x}), 1), \quad (11)$$

that discourages all zero restorations by encouraging restoration occupancy values to be high. Hyperparameter λ_{nzer} represents the weight on the non-zero loss. After inferring $\mathbf{z}_{\mathcal{F}}$, we generate $\mathbf{z}_{\mathcal{R}}$ and $\mathbf{z}_{\mathcal{C}}$ using Equation (2). To obtain predicted fractured, restoration, and complete meshes we use f_{Θ} to estimate occupancy values for 256³ points sampled uniformly in a 1.1 unit cube, apply Marching Cubes [LC87] to the predicted values, and extract the 0.5 isosurface. Latent code inference and mesh reconstruction take 53 and 8.8 seconds respectively on an 8-core Intel desktop with 1 NVIDIA RTX 3090. We discuss our implementation in the supplementary.

6. Datasets and Data Preparation

Since no ground truth dataset of real-world fractured object models exists, we generate a dataset of synthetically fractured shapes. We use the method described by Lamb et al. [LWL*21] to synthetically fracture a mesh by subtracting a subdivided and randomized geometric primitive from the mesh so that between n_{\min} and n_{\max} vertices are removed, which we term simple ceramic fracture meshes. We use a cube or icosahedron with edges subdivided 0, 1, or 2 times for the geometric primitive. To create low-frequency surface variation, we apply a random translation to the vertices of the primitive. To create high frequency surface variation, we subdivide the edges to be less than 0.025 units and apply a second random translation to the vertices. We perform a retention test where, if the mesh cannot be fractured such that between n_{\min} and n_{\max} vertices are removed after 15 attempts, we discard it. Before fracturing we

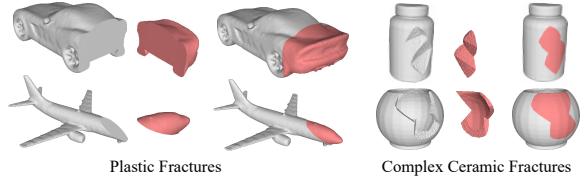


Figure 7: Ground truth fractured and restoration meshes, in gray and red, using hard plastic and complex ceramic fractures.

waterproof and normalize each mesh with respect to the unit cube. For fracturing we set n_{\min} and n_{\max} to 5% and 20% of all vertices. We show example ground truth fractured meshes in Figure 6.

We test the generalizability of our approach by training on simple ceramic fractures and testing on synthetic hard plastic fractures and complex ceramic fractures. To simulate hard plastic fractures that break cleanly without surface roughness, we generate fractured shapes using a geometric primitive without high-frequency surface variation, i.e. with low-frequency surface variation only. We simulate complex ceramic fractures by subtracting two slightly offset geometric primitives from the complete shape. Subtracting two primitives generates fractured shapes with concavities and often generates multiple fractured surfaces. We show example hard plastic and complex ceramic fractured meshes in Figure 7.

We generate fractured, restoration, and complete meshes by fracturing 3D models from two publicly available datasets.

1. **ShapeNet.** To evaluate our approach on household objects that are susceptible to damage we use 3D models from the bottles, jars, and mugs classes from the ShapeNet [CFG*15] dataset. To demonstrate restoration results on objects of more complex geometry, we include 5 additional classes, i.e., planes, cars, chairs, sofas, and tables. As the jars, bottles, and mugs classes from ShapeNet contain fewer than 600 complete meshes, we fracture meshes from these classes 3, 3, and 7 times respectively. We fracture meshes from all other classes once. We generate simple ceramic fractures for 1,788 jars, 1,494 bottles, 1,498 mugs, 4,045 planes, 6,000 chairs, 3,514 cars, 6,000 tables, and 3,173 sofas from ShapeNet. Of the models fractured, 1,534 jars, 1,376 bottles, 1,074 mugs, 3,938 planes, 5,555 chairs, 3,307 cars, 5,614 tables, and 3,051 sofas pass the retention test giving us a total of 25,449 out of 27,512 meshes. We generate hard plastic and complex ceramic fractures for the test set of each ShapeNet class.
2. **QP Cultural Heritage Dataset.** The QP cultural heritage dataset [KPA*09] contains 408 artist-designed 3D meshes in the style of ancient Greek pottery. We fracture each mesh once with simple ceramic fractures. We retain 317 of the 408 meshes.

We associate each fractured-restoration pair with its corresponding original complete shape to create tuples for training and evaluation with respect to ground truth. We use the implementation of Mescheder et al. [MON*19] to compute ground truth occupancy values. For shapes from the ShapeNet dataset, we perform an 80%-20% training and testing split. We train one network per class.

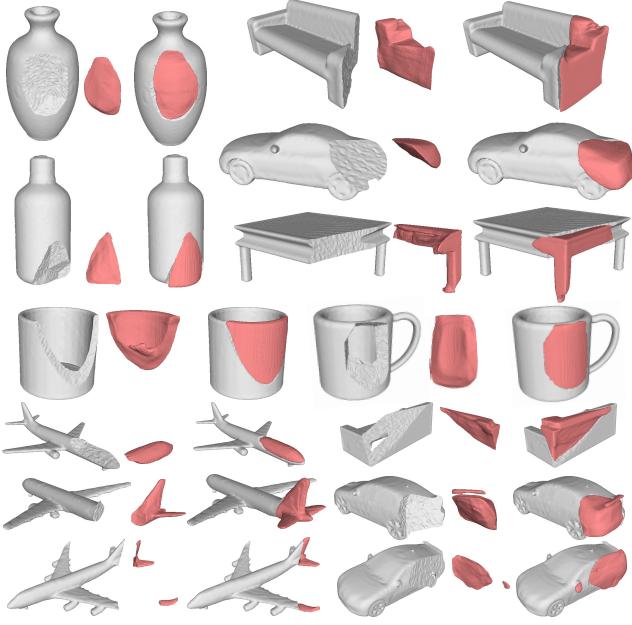


Figure 8: Ground truth input fractured shape in gray, restoration predicted using MendNet in red, and predicted restoration joined to ground truth fractured shape for a variety of objects spanning 8 classes from the ShapeNet dataset.

7. Results

We show input fractured shapes, restoration shapes predicted by MendNet opened to reveal the surface in common with the fractured shape, and the restorations joined to the fractured shapes in Figure 8. Our approach generates restoration shapes that match closely to the fracture region of the fractured shape. Predicted restorations do not show connected or disconnected artifacts, making restoration shapes generated using our approach ideal for physical object repair. As shown by the opened restorations, our approach is able to generate plausible fracture surface counterparts on the restoration, enabling smooth joins. Our approach reproduces the overall shape of complex missing components, e.g., airplane tail fins in the bottom left of Figure 8, and produces restoration parts even when the fractured object has elongated components broken off, e.g., the sofa in the bottom right of Figure 8. We generate feasible restoration shapes when multiple parts are broken off, e.g., the plane and the car on the bottom of Figure 8. Our approach is able to generate restorations for fractured shapes that have asymmetrical fractures, such as the bottom sofa, the planes, and the cars in Figure 8. Existing approaches based on symmetry [SLL*08; GSP*14; PSA*17] are unable to handle asymmetrical restorations.

In Figure 9, we show restoration shapes generated for the QP cultural heritage dataset [KPA*09] using MendNet trained using the bottles class from ShapeNet. We synthesize fractures for QP meshes using the approach discussed in Section 6. Our approach is able to generalize to cultural heritage objects even though the network is trained without cultural heritage samples. As shown in Figure 9, restoration shapes generated using our approach fit closely

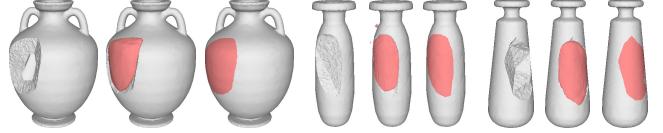


Figure 9: Input fractured shapes of a Greek amphora and two alabastrons from the QP dataset, restorations predicted using our approach trained on bottles, and ground truth restoration.

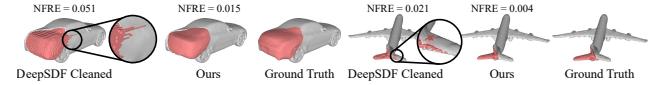


Figure 10: Non-fractured region error (NFRE) for predicted restoration shapes joined to input fractured shapes using DeepSDF with cleanup and using our approach.

to the fractured cultural heritage objects and restore the fractured shapes to their original structure. Restoration shapes generated using MendNet may be used to aid in the repair of historical artifacts, e.g., the Calyx Krater shown at the bottom of Figure 2.

7.1. Metrics

We introduce two metrics based on the intersection over union (IoU) to evaluate predicted restoration shapes. The **Intersection Score (IS)** captures the physical validity of a predicted restoration shape by measuring the degree of overlap with the fractured shape. We compute the IS as the IoU of the predicted restoration and the input fractured shape. The **Union Score (US)** captures constrained shape estimation via the complete shape by measuring its overlap with the fractured and restoration shapes. We compute US by performing a union of the predicted restoration and input fractured shape, and taking the IoU of the union with the ground truth complete shape. For physical validity the IS should be low while the US should be high. The **non-zero percentage (NZ%)** indicates the percentage of restoration shapes that are generated by a given approach, i.e. the percentage that are not predicted as all zero.

When comparing our approach to 3D-ORGAN [HS18] and the baseline approaches, we use the **chamfer distance (CD)** as a measure of geometric similarity. The CD is defined as the average distance from every point in one mesh to the closest point in another mesh in both directions. Approaches based on subtraction produce restoration shapes that exhibit protruding artifacts on the surface of the fractured shape. To quantify protrusion geometry we use the **non-fractured region error (NFRE)**, which measures the percentage of the non-fractured surface of the fractured shape that is covered by artifacts. We compute the NFRE by sampling n points on the non-fractured region of the fractured mesh, the predicted restoration mesh, and the ground truth restoration mesh, and computing the percentage of points on the non-fractured region of the fractured mesh that have a nearest neighbor in the predicted restoration closer than η , and a nearest neighbor in the ground truth restoration farther than η . Predicted restorations that demonstrate artifacts on the surface of the fractured mesh outside of the fractured region will have a high NFRE. For computing the NFRE we

Fracture	Hard Plastic		Complex Ceramic		Simple Ceramic	
Metric	NZ%	CD	NZ%	CD	NZ%	CD
bottles	81.8	0.078	71.0	0.122	78.4	0.081
cars	57.3	0.035	56.6	0.032	57.9	0.024
chairs	97.7	0.168	97.3	0.176	96.7	0.171
jars	96.6	0.118	97.0	0.158	98.0	0.128
mugs	79.8	0.052	71.7	0.098	78.6	0.074
planes	92.8	0.066	93.6	0.074	93.9	0.091
sofas	78.4	0.173	73.5	0.199	72.7	0.188
tables	96.5	0.185	95.7	0.206	94.3	0.208
Mean	85.1	0.109	82.1	0.133	83.8	0.121

Table 1: Percentage of non-zero restorations (NZ%), and chamfer distance (CD) for different fracture types.

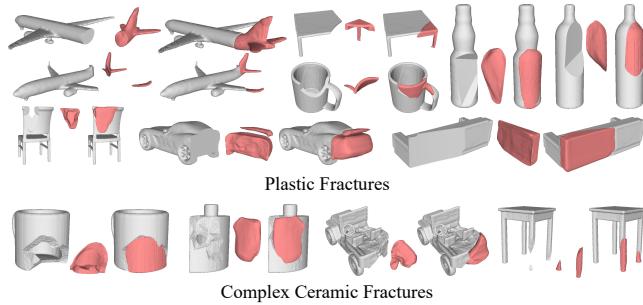


Figure 11: Input fractured shapes and predicted restorations, in gray and red, for hard plastic and complex ceramic fractures.

use $\eta = 0.02$ and $n = 30,000$. We show restoration shapes and corresponding NFRE values generated using DeepSDF with cleaning and generated using MendNet in Figure 10.

7.2. Restoring Hard Plastic and Complex Ceramic Fractures

In Figure 11 we show results for training our approach on simple ceramic fractures and testing our approach on hard plastic and complex ceramic fractures. As shown in Figure 11, though our approach is trained on fractures that mimic ceramic, MendNet is able to generate closely fitting restoration shapes for hard plastic fractures, including for shapes that cannot be repaired using symmetry such as the mug, car, planes, chair, and sofa. We also show feasible and artifact-free restorations for more complex ceramic fractures, including for multi-component fractures, e.g., the table. Table 1 shows that at 0.133 the CD for complex ceramic fractures is higher than for simple ceramic fractures at 0.121. As we train our approach on simple ceramic fractures with few concave regions, MendNet sometimes struggles to generate restorations that fit closely to the fracture region for highly concave fractures, such as the mug shown in Figure 11. However, using mesh deformation techniques [SCL*04], the restoration may be automatically expanded to fit the surface of the fracture yielding a closer fit.

7.3. Evaluation of Inference Losses

We study the impact of the feasibility and non-zero losses introduced in Section 5, i.e., $\mathcal{J}_{\text{feas}}$ and $\mathcal{J}_{\text{nzer}}$, using the intersection score

Method	$\mathcal{J}_{\text{data}}$			$\mathcal{J}_{\text{data}}, \mathcal{J}_{\text{nzer}}$			$\mathcal{J}_{\text{data}}, \mathcal{J}_{\text{feas}}$			$\mathcal{J}_{\text{data}}, \mathcal{J}_{\text{nzer}}, \mathcal{J}_{\text{feas}}$		
	NZ%	IS	US	NZ%	IS	US	NZ%	IS	US	NZ%	IS	US
bottles	76.3	0.02	0.95	85.6	0.21	0.80	77.0	0.00	0.95	78.4	0.03	0.92
cars	50.8	0.00	0.95	71.4	0.20	0.84	40.1	0.00	0.96	57.9	0.00	0.95
chairs	84.4	0.01	0.87	98.8	0.14	0.16	56.4	0.00	0.90	96.7	0.03	0.48
jars	98.3	0.02	0.92	99.7	0.13	0.76	92.2	0.01	0.93	98.0	0.04	0.84
mugs	79.5	0.01	0.95	82.7	0.05	0.90	75.9	0.00	0.95	78.6	0.00	0.95
planes	79.7	0.01	0.94	99.2	0.14	0.16	61.7	0.00	0.95	93.9	0.01	0.78
sofas	57.8	0.01	0.92	89.4	0.27	0.41	36.9	0.00	0.93	72.7	0.02	0.79
tables	80.3	0.01	0.87	99.3	0.13	0.16	58.2	0.00	0.90	94.3	0.03	0.53
Mean	75.9	0.01	0.92	90.8	0.16	0.53	62.3	0.00	0.93	83.8	0.02	0.78

Table 2: Percentage of non-zero restorations (NZ%), intersection score (IS), and union score (US) for various loss function combinations presented in our work. Bolded values are best across all loss choices for corresponding metrics. Bold-italicized metric values correspond to where $\mathcal{J}_{\text{data}}, \mathcal{J}_{\text{nzer}}, \mathcal{J}_{\text{feas}}$ performs second best.

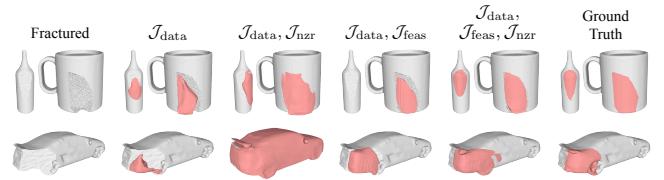


Figure 12: Input fractured shape in gray, together with estimated restoration shapes in red when using $\mathcal{J}_{\text{data}}$ alone, using $\mathcal{J}_{\text{data}}$ and $\mathcal{J}_{\text{nzer}}$, using $\mathcal{J}_{\text{data}}$ and $\mathcal{J}_{\text{feas}}$, and using all losses.

(IS) and union score (US). Table 2 summarizes IS and US values over combinations of $\mathcal{J}_{\text{data}}$, $\mathcal{J}_{\text{nzer}}$, and $\mathcal{J}_{\text{feas}}$. The table also shows the percentage of non-zero restorations (NZ%). Using $\mathcal{J}_{\text{data}}$ alone generates restorations with a low NZ% at 75.9%. As shown in the second column of Figure 12, using $\mathcal{J}_{\text{data}}$ alone sometimes produces restoration shapes in the correct vicinity that lack connectivity and protrude outside the complete shape boundary. Using $\mathcal{J}_{\text{nzer}}$ with $\mathcal{J}_{\text{data}}$ raises NZ% to 90.8%. However, it provides a lower US of 0.53, and a higher IS of 0.16, indicating that $\mathcal{J}_{\text{nzer}}$ produces restorations that protrude into the fractured shape. Sometimes the restoration may encompass the fractured shape, e.g., the car in Figure 12.

Using $\mathcal{J}_{\text{data}}$ and $\mathcal{J}_{\text{feas}}$ induces the restoration shape to remain inside the complete shape, and produces restoration shapes with the lowest IS and highest US. However, it reduces NZ% to 62.3%. For instance, a restoration is not generated for the bottle in the fourth column of Figure 12. Using all three losses, our approach predicts restoration shapes that remain within the complete shape and join smoothly with the fractured shape as shown in the fifth column of Figure 12. Though using all three losses shows a lower US than using $\mathcal{J}_{\text{data}}$ alone, we observe that complete shapes predicted using all three losses are usually slightly larger than when using $\mathcal{J}_{\text{data}}$ alone, as shown in Figure 12, resulting in slightly larger restoration shapes that are disproportionately penalized by the IoU, which is based on volume. As shown in Table 2, we show higher NZ% than when $\mathcal{J}_{\text{data}}$ is used alone or with $\mathcal{J}_{\text{feas}}$, at 83.8%, and a lower mean IS and higher mean US than when $\mathcal{J}_{\text{nzer}}$ is used.

Method	$\mathcal{J}_{\text{data}}$			$\mathcal{J}_{\text{data}}, \mathcal{J}_{\text{nzs}}$			$\mathcal{J}_{\text{data}}, \mathcal{J}_{\text{nint}}$			$\mathcal{J}_{\text{data}}, \mathcal{J}_{\text{nzs}}, \mathcal{J}_{\text{nint}}$		
	NZ%	IS	US	NZ%	IS	US	NZ%	IS	US	NZ%	IS	US
bottles	75.2	0.04	0.95	78.8	0.22	0.76	55.4	0.01	0.95	53.2	0.00	0.95
cars	47.0	0.01	0.95	68.8	0.27	0.78	27.2	0.00	0.94	41.0	0.00	0.95
chairs	69.9	0.01	0.90	99.7	0.19	0.21	31.6	0.00	0.90	75.9	0.00	0.85
jars	88.8	0.02	0.94	94.6	0.14	0.78	65.1	0.00	0.94	73.9	0.00	0.93
mugs	72.3	0.01	0.94	74.1	0.05	0.91	68.2	0.00	0.93	70.5	0.00	0.94
planes	79.3	0.01	0.94	99.2	0.19	0.25	53.1	0.00	0.93	88.1	0.00	0.90
sofas	54.4	0.01	0.92	97.7	0.35	0.48	17.6	0.00	0.91	47.1	0.00	0.91
tables	77.3	0.00	0.89	98.8	0.15	0.23	50.2	0.00	0.89	85.0	0.00	0.74
Mean	70.5	0.01	0.93	89.0	0.19	0.55	46.1	0.00	0.93	66.8	0.00	0.89

Table 3: Percentage of non-zero restorations (NZ%), intersection score (IS), and union score (US) with loss functions for our approach that predicts the restoration directly. Best values are in bold.



Figure 13: Restoration shapes generated using our approach that predicts the restoration directly, $Ours_{\mathcal{F}R}$, and our approach that predicts all three shapes, $Ours_{\mathcal{F}RC}$, both with all three losses.

7.4. Evaluation of Predicting Restoration Shape Directly

We can predict the restoration shape directly by removing the complete latent transformation branch of the network, \mathbf{g}_{Φ_C} , and replacing $\mathcal{J}_{\text{feas}}$ with a non-intersection constraint, $\mathcal{J}_{\text{nint}}$, which penalizes intersection of the fractured and restoration shapes, i.e. encodes the relationship $(o_{\mathcal{F}} \wedge o_{\mathcal{R}}) = 0$. However, we find that disabling the network from predicting values for the complete shape, and by extension removing the union condition of $\mathcal{J}_{\text{feas}}$, shown in Figure 5(a), increases the frequency of all zero restoration shapes. To train the network variation we use the loss $\mathcal{J} = \mathcal{J}_{\text{data}} + \mathcal{J}_{\mathcal{R}} + \mathcal{J}_{\text{reg}}$, where $\mathcal{J}_{\mathcal{R}}$ is derived from Equation (7), computed over \mathcal{R} instead of over \mathcal{R} and \mathcal{C} . We use $\mathcal{J}_{\text{nint}} = \sum_{\mathbf{x} \in \mathcal{X}} l(f_{\Theta}(\mathbf{z}_{\mathcal{F}}, \mathbf{x}) f_{\Theta}(\mathbf{g}_{\Phi_R}(\mathbf{z}_{\mathcal{F}}, \mathbf{x}), \mathbf{x}), 0)$, computed over set of inference-time sample points \mathcal{X} . We show results using combinations of $\mathcal{J}_{\text{data}}$, \mathcal{J}_{nzs} , and $\mathcal{J}_{\text{nint}}$ in Table 3. We refer to our approach that predicts the restoration directly as $Ours_{\mathcal{F}R}$ and our approach that predicts all shapes as $Ours_{\mathcal{F}RC}$.

Using $Ours_{\mathcal{F}R}$ with $\mathcal{J}_{\text{data}}$ alone shows similar IS and US scores to $Ours_{\mathcal{F}RC}$ with $\mathcal{J}_{\text{data}}$ and a lower NZ% at 70.5%. Adding the non-zero loss increases the NZ%, though it produces restorations with a higher IS and lower US than using $\mathcal{J}_{\text{data}}$ alone. Using $\mathcal{J}_{\text{nint}}$ with $\mathcal{J}_{\text{data}}$ has little effect on the IS or US and decreases NZ% to 46.1% compared to using $\mathcal{J}_{\text{data}}$ alone, as $\mathcal{J}_{\text{nint}}$ forces the restoration away from the fractured shape. As shown on the right of Table 3, using all three losses with $Ours_{\mathcal{F}R}$ still shows a low NZ% compared to using $\mathcal{J}_{\text{data}}$ alone, and may predict restoration shapes that do not fully join to the fractured shape, as occurs with the mug in Figure 13. Though $\mathcal{J}_{\text{feas}}$ and $\mathcal{J}_{\text{nint}}$ both penalize intersection of the predicted restoration and fractured shape, $\mathcal{J}_{\text{feas}}$ also encourages intersection of the complete shape and restoration shape, providing a reduced space for the restoration to occupy and aiding in latent code convergence, giving a high NZ%. As shown by the rightmost super-

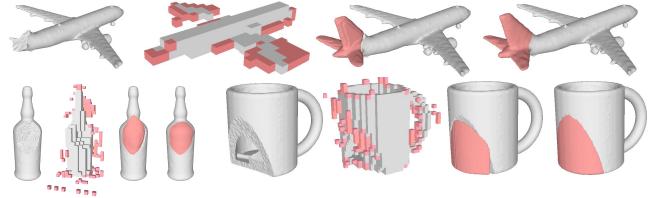


Figure 14: For each object, from left to right, we show the input fractured shape, voxel restoration using 3D-ORGAN, restoration using our approach, and the ground truth restoration.

columns of Table 2 and Table 3, with all losses $Ours_{\mathcal{F}RC}$ achieves a majority of physically valid restoration shapes with a high NZ% of 83.8%, while $Ours_{\mathcal{F}R}$ shows a NZ% of 66.8%.

7.5. Comparison to 3D-ORGAN

We compare our work to the voxel-based approach of 3D-ORGAN [HS18], the only prior method to perform shape completion using fractured shapes as input. 3D-ORGAN takes a voxelized fractured shape as input and predicts a corresponding complete shape. The density of voxel grids required to acquire high-resolution restoration outputs prohibits tractable learning using modern hardware, and requires a quantity of training data that is unreasonable in real-world environments. We use the original voxel grid resolution used by 3D-ORGAN, which takes 32^3 fractured voxel grids as input and produces 32^3 complete voxel grids as output. We adapt 3D-ORGAN to generate restoration meshes by subtracting the input fractured voxel grid from the complete voxel grid predicted by the network during inference and applying Marching Cubes to the subtracted voxel grid to reconstruct a mesh. During inference we run 3D-ORGAN for two iterations by feeding the predicted complete voxel grid from the first iteration as input to the second iteration, as recommended by the authors.

As shown in first row of Table 4, we use the CD and NFRE to compare our approach to 3D-ORGAN (3DO). We summarize metrics over all shapes with non-zero restorations over all approaches. Our approach outperforms 3D-ORGAN overall with a mean CD of 0.122 and NFRE of 0.076 compared to 0.241 and 0.160 respectively, outperforms for all classes individually in terms of CD, and for all classes except cars and tables in terms of NFRE. While the NFRE for our approach on cars is higher than for 3D-ORGAN, we find that 3D-ORGAN is successful at generating non-zero restorations for only 21.73% of car shapes. We observe that 3D-ORGAN produces physically unrealizable restorations with a high number of surface artifacts, as shown in Figure 14. 3D-ORGAN often predicts restoration shapes that are in the wrong location or are sparse and have many disconnected voxel elements, making them unusable for shape repair.

7.6. Comparison to Baseline Approaches using Shape Completion followed by Subtraction

Since 3D-ORGAN produces low-fidelity results, we sought to compare our method to high-resolution restoration generation as a baseline for fair evaluation. For the baselines, we predict a complete

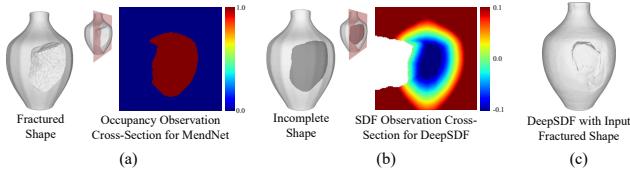


Figure 15: Input (a) fractured shape and (b) incomplete shape and cross-sections showing point observations for MendNet and DeepSDF. For DeepSDF, we remove sample points with nearest neighbors on the fractured shape in the fracture region. (c) Predicted complete shape with elements of the fracture that results from using a fractured shape as input for DeepSDF.

shape using a shape completion approach and generate the restoration using subtraction. We use the SDF based DeepSDF [PFS^{*}19], and the point cloud based PoinTr [YRW^{*}21] followed by the occupancy function based ConvONet [PNM^{*}20] as baselines. For the baselines we obtain the restoration by subtracting the input fractured shape from the predicted complete shape in occupancy space.

DeepSDF. DeepSDF operates on open incomplete shapes, and is not structured to work directly on closed fractured shapes. The autodecoder framework that DeepSDF uses is similar to an autoencoder, i.e. using the fractured shape as input directly would cause the network to reconstruct the input fractured shape instead of predicting a complete shape, as occurs for Figure 15(c). To perform inference on fractured shapes, we obtain an incomplete shape for each test fractured shape by removing the fracture surface, thereby emulating an error-free fracture detection classifier. As shown in Figure 15(b), we do not input sample points that have a nearest neighbor in the fractured shape belonging to the fracture region. To obtain a complete shape during inference we use DeepSDF to estimate a latent code given incomplete observations of the incomplete shape. To convert to occupancy, we give negative SDF values an occupancy of 1 and all other values an occupancy of 0.

PoinTr+ConvONet (PTR+CON). We also create a baseline that performs point cloud completion for an input fractured point cloud followed by surface reconstruction using the occupancy function. We use PoinTr for point cloud completion and ConvONet for surface reconstruction. We train PoinTr to predict complete point clouds from input fractured point clouds. We train on simple ceramic fractures for 800 epochs. To obtain occupancy values we input the complete point cloud predicted by PoinTr to ConvONet. For ConvONet we use pre-provided weights obtained by training on planes, benches, cabinets, cars, chairs, displays, lamps, loudspeakers, rifles, sofas, tables, telephones, and vessels from ShapeNet.

For each baseline we obtain the restoration shape using Marching Cubes over 256^3 sample points. To mitigate artifacting, we perform an automated post-subtraction cleanup for the baseline approaches by removing mesh components whose volume falls below a threshold of ϵ , where we set ϵ to 0.01. If all disconnected components have a volume less than ϵ , we retain the largest component. We provide an example of an alternative method of removing artifacts based on distance thresholding in the supplementary.

We compare MendNet to PTR+CON in the second and third

Method	Metric	bottles	cars	chairs	jars	mugs	planes	sofas	tables	Mean
3DO	CD	0.15	0.28	0.18	0.26	0.23	0.17	0.32	0.33	0.24
	NFRE	0.07	0.00	0.59	0.04	0.05	0.19	0.20	0.14	0.16
PTR+CON	CD	0.09	0.13	0.15	0.11	0.14	0.09	0.12	0.15	0.12
	NFRE	0.41	0.54	0.21	0.34	0.38	0.50	0.39	0.25	0.38
PTR+CON Cleaned	CD	0.07	0.09	0.20	0.11	0.10	0.05	0.09	0.15	0.11
	NFRE	0.11	0.25	0.10	0.15	0.17	0.11	0.17	0.12	0.15
DeepSDF	CD	0.08	0.12	0.15	0.12	0.13	0.10	0.13	0.18	0.13
	NFRE	0.59	0.73	0.64	0.71	0.55	0.83	0.48	0.47	0.62
DeepSDF Cleaned	CD	0.02	0.04	0.08	0.10	0.05	0.02	0.04	0.08	0.05
	NFRE	0.07	0.20	0.18	0.16	0.04	0.05	0.09	0.15	0.12
MendNet (Ours)	CD	0.08	0.04	0.17	0.13	0.07	0.09	0.19	0.21	0.12
	NFRE	0.04	0.03	0.14	0.03	0.01	0.07	0.08	0.20	0.08

Table 4: Chamfer distance (CD) and non-fractured region error (NFRE) comparing our approach against 3D-ORGAN (3DO) and subtraction-based restoration generation using PoinTr with ConvONet (PTR+CON) and DeepSDF. We summarize metrics over all non-zero shapes for all approaches. Best values are in bold.

rows and to DeepSDF in the fourth and fifth rows of Table 4. Our approach performs comparably to DeepSDF and PTR+CON before cleanup in terms of CD and outperforms for all classes and overall in terms of NFRE. Though we observe that DeepSDF with cleanup shows a CD score lower than ours, since the cleanup does not eliminate protrusions that are connected to the restoration, our approach outperforms the baseline in terms of the NFRE on 6 of the 8 classes, and outperforms in terms of the NFRE overall, with a value of 0.076 compared to DeepSDF with 0.116, and PTR+CON with 0.146. As shown in Figure 16, while disconnected components are reduced, for both baselines thin complex protrusions emanating from the restoration on the exterior surface of the fractured shape still remain. If the restoration is thin, i.e. the volume is low, automated cleanup may also discard elements of the restoration, e.g. for the PTR+CON mug in Figure 16. MendNet produces restorations that restore missing components and do not exhibit artifacts. Though some restorations, e.g., for the plane, do not join exactly to the fractured shape, gaps may be filled in with glue or other adhesive without requiring manual editing. Artifacts generated by the baseline approaches, e.g. by DeepSDF for the plane, car, mug, and bottle and by PTR+CON for the plane and car in Figure 16, must be manually removed, causing inconvenience to the end user and preventing the baseline approaches from being used at scale.

7.7. Latent Code Interpolation

To explore the space of latent codes learned by our approach, we interpolate between codes from different test shapes in Figure 17. We obtain intermediate codes by performing linear interpolation of the values between two fractured codes, and reconstruct the restoration, fractured, and complete meshes for the intermediate codes. In Figure 17(a), we interpolate between two fractured shapes that are derived from the same complete shape. We observe that while the complete mesh, shown on the bottom, does not change, the predicted restoration interpolates smoothly, suggesting that the transformation network for fractured to complete codes, \mathbf{g}_{Φ_C} , has learned to map multiple fractured shapes to the single complete

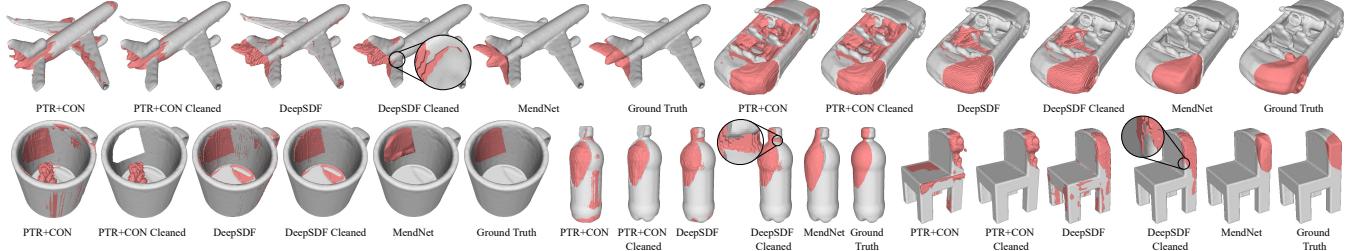


Figure 16: For each shape, from left to right, we show restorations obtained using PoinTr with ConvONet (PTR+CON) with and without cleaning, using DeepSDF with and without cleaning, using our approach, and the ground truth. The baseline approaches generate artifacts that are not eliminated by cleanup, as shown by the insets. Restorations generated using our approach lack small artifacts.

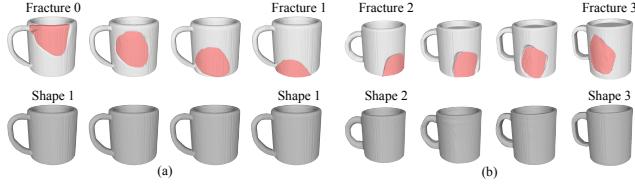


Figure 17: Meshes generated by interpolating latent codes (a) across two fractures within the same complete shape, and (b) across two fractures within different complete shapes.

shape. We also observe the fractured and restoration meshes correspond closely to the same fracture, indicating that \mathbf{g}_{Φ_R} has learned to map the input fractured shape to the singular corresponding restoration shape. When we interpolate between codes for two different shapes, e.g. Figure 17(b), we observe that our approach has learned an embedding that maps similar objects closely in latent space, evidenced by the smooth transition between meshes.

7.8. Evaluating Failure Cases

As shown in Table 2, our approach predicts restorations as all zero for 16.2% of shapes. In Table 5 we classify predicted fractured, restoration, and complete shape tuples, i.e., \mathcal{R}_p , \mathcal{F}_p , and \mathcal{C}_p into one of 5 possible cases in terms of their relationship to the ground truth fractured, restoration, and complete shapes, i.e., \mathcal{R}_{gt} , \mathcal{F}_{gt} , \mathcal{C}_{gt} .

- Case 1: The restoration is zero, the predicted fractured shape is more similar to the ground truth fractured shape than the ground truth complete shape, and the predicted complete shape is more similar to the ground truth complete shape than the ground truth fractured shape, i.e. $\mathcal{R}_p = 0$, $\mathcal{F}_p \approx \mathcal{F}_{gt}$, $\mathcal{C}_p \approx \mathcal{C}_{gt}$.
- Case 2: The restoration is zero, the predicted fractured shape is more similar to the ground truth complete shape than the ground truth fractured shape, and the predicted complete shape is more similar to the ground truth complete shape than the ground truth fractured shape, i.e. $\mathcal{R}_p = 0$, $\mathcal{F}_p \approx \mathcal{C}_{gt}$, $\mathcal{C}_p \approx \mathcal{C}_{gt}$.
- Case 3: The restoration is zero, the predicted fractured shape is more similar to the ground truth fractured shape than the ground truth complete shape, and the predicted complete shape is more similar to the ground truth fractured shape than the ground truth complete shape, i.e. $\mathcal{R}_p = 0$, $\mathcal{F}_p \approx \mathcal{F}_{gt}$, $\mathcal{C}_p \approx \mathcal{F}_{gt}$.

Case	Case 1	Case 2	Case 3	Case 4	Case 5
bottles	7.9%	13.7%	0.0%	0.0%	78.4%
cars	24.5%	16.8%	0.8%	0.0%	57.9%
chairs	1.0%	2.3%	0.0%	0.0%	96.7%
jars	0.3%	1.7%	0.0%	0.0%	98.0%
mugs	4.1%	16.4%	0.5%	0.5%	78.6%
planes	4.5%	1.6%	0.0%	0.0%	93.9%
sofas	15.5%	11.4%	0.3%	0.0%	72.7%
tables	1.0%	4.6%	0.0%	0.1%	94.3%
Mean	7.4%	8.6%	0.2%	0.1%	83.8%

Table 5: Percentage of shape tuples belonging to each case as defined in Section 7.8. Cases that occur most often are bolded.

- Case 4: The restoration is zero, the predicted fractured shape is more similar to the ground truth fractured shape than the ground truth complete shape, and the predicted complete shape is more similar to the ground truth fractured shape than the ground truth complete shape, i.e. $\mathcal{R}_p = 0$, $\mathcal{F}_p \approx \mathcal{F}_{gt}$, $\mathcal{C}_p \approx \mathcal{F}_{gt}$.
- Case 5: The restoration is non-zero, i.e. $\mathcal{R}_p \neq 0$.

We compute shape similarity using the CD, e.g. if $CD(\mathcal{F}_p, \mathcal{F}_{gt}) < CD(\mathcal{F}_p, \mathcal{C}_{gt})$ then the predicted fractured shape is more similar to the ground truth fractured shape than the ground truth complete shape. We show the frequency of each case in Table 5.

Cases 1 and 2 occur for 7.4% and 8.6% of shapes. Though predicted shape tuples belonging to case 2 are incorrect, they correspond to a configuration that fulfills the feasibility requirement, as $\mathcal{F}_p \approx \mathcal{C}_{gt}$, which will force $\mathcal{R}_p = 0$. In case 2, the network fails to optimize for a fractured code \mathbf{z}_f that accurately represents the fractured shape, though it still predicts a feasible output. In case 1, the network fails to optimize for a restoration code $\mathbf{g}_{\Phi_R}(\mathbf{z}_f, \mathbf{x})$ that accurately represents the fractured shape, and produces an infeasible output. Occurrences of case 1 are likely attributable to the network needing to optimize for three shapes simultaneously, and to optimize for a \mathbf{z}_f that converges with respect to all three shapes in order to produce a high quality result. Though requiring the network to converge with respect to all three shapes produces restoration shapes with high confidence, the complexity of the multi-objective optimization problem makes convergence difficult in some cases. Cases 1 and 2 may be recovered from using subtraction of the in-

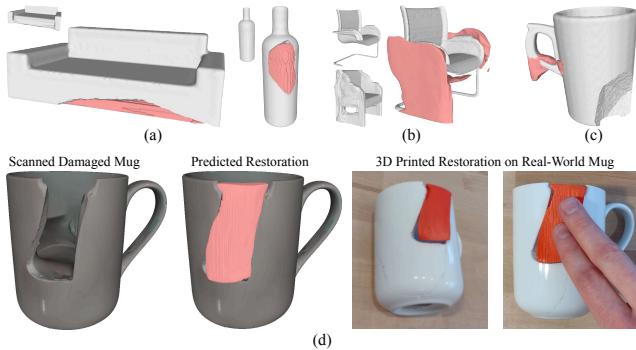


Figure 18: (a) Since our approach estimates complete shape, if restorations are predicted as zero, we can fall back to generating restorations using a subtraction. (b) For a fractured shape with thin structures (top-left), the approach may predict bulky restorations (right) by leveraging voluminous complete shape (bottom-left). (c) Restoration generated away from fracture surface. (d) Physical constraints and printing expansion generate 3D prints that do not fit in the damaged region without modification.

put fractured shape from the predicted complete shape. Cases 3 and 4 cannot be recovered from, as a legitimate complete shape is not generated. However, cases 3 and 4 occur in only 0.3% of shapes.

8. Limitations and Future Work

The primary limitation of our approach is that, as shown in Table 2, restorations are predicted as all zero for 16.2% of shapes. As discussed in Section 7.8, restorations are most often predicted as zero when the fractured or complete shapes are predicted as being similar to the ground truth complete shape, i.e. cases 1 and 2. However, in both cases, since the complete shape is generated with high accuracy, our method enables fallback to using subtraction to generate the restoration shape, as shown in Figure 18(a). Though non-recoverable cases do occur, i.e. cases 3 and 4, these cases are infrequent, only occurring 0.3% of the time. A second limitation is that the non-zero loss may cause restorations to be predicted as bulky, e.g., for shapes with fine structures in Figure 18(b). Bulky restorations are encouraged by the binary occupancy function that pushes network outputs toward 0 and 1. In future, we will encode shape relationships using continuous functions, i.e. the SDF, to mitigate all zero and bulky restorations. We will also explore high-fidelity implicit shape representations to improve restoration resolution, e.g. using normal vectors [VSG*20] and predicting at multiple resolutions using hierarchical decoders [CZG*21].

To perform restoration generation for real-world objects, a number of challenges need to be surmounted. We observe that canonical misalignments, e.g., the mug shown in Figure 18(c), cause restorations to be predicted in the incorrect location. As misalignment may occur frequently for real objects due to the difficulty of precisely orienting the fractured shape after scanning, in future we will improve the robustness of our approach to misalignment by randomizing the orientation of training and testing shapes. To study the geometry of real fractures we damaged and scanned a real-world

ceramic mug, shown in Figure 18(d). We observe that the fracture edges are smooth, have a curved profile, and demonstrate sharp cusps and turns. The smooth transitions between fracture and intact regions may cause fracture-surface classifiers to fail, i.e., rendering methods based on incomplete shape completion and subtraction unusable. When we use our approach trained on synthetic fractures to predict a restoration shape, while our approach creates a restoration that fills most of the fracture region it struggles to fill in the sharp curve at the bottom left. As a further challenge, when we 3D print the generated restoration, it cannot be ‘snapped’ into the damaged region, due to the physical constraints imposed by the negative space and the expansion of the restoration part during 3D printing. As part of future work, we will investigate ways to repair tight-fitting fractures using the feasibility condition as a hard constraint and adopting a sculpting approach that accommodates printer tolerance to refine the restoration in a post-step. We will also explore performing jigsaw carving of the restoration geometry to enable repair of narrow damage regions, such as the one in Figure 18(d).

Currently the biggest challenge to progress in the field of fracture repair is the dearth of real data, with the largest existing dataset of real fractured objects with complete counterparts containing only 22 samples [LBB19]. The generalizability of our approach to real fractures may be limited by the simple structure of objects from ShapeNet, which are derived from computer aided design (CAD) models. Our fracturing approach also simulates the subset of possible fractures that can be generated using subtraction with a randomized geometric primitive, i.e., it does not fully capture the variety and complexity of real fractures. Prior work in physics-based fracturing is focused on providing visually appealing fractures [BHTF07; CCL*22; WFL*19; WDG*19], is overly simplistic [ESW20; LWL*21; GBS*15], or is focused on micro-scale analysis of fracture from a material science perspective [ANZ06; ZQ18; SMZG18; FLF*19]. To capture the complexity of real fractures it is necessary to obtain a dataset of real fractured objects.

As part of ongoing work, we are actively collecting a dataset of real fractured objects with ground truth complete counterparts, going well beyond the object, material, and fracture-type sets in Lamb et al. [LBB19]. We are generating fractures for common materials and fracture types, e.g., ceramic chipping, wood splintering, wood shearing along grains, and large fractures for ceramic, clay, hard plastic, and concrete. Our dataset will enable restoration evaluation when significant portions of the object are eroded away as may occur with, e.g., wood or concrete. One challenge that we expect to face is that despite expansion efforts, datasets of real-world fractured objects are unlikely to approach the scale of synthetic datasets, due to the time consuming nature of 3D scanning—often taking 1 to 2 hours per sample—and the infeasibility of comprehensively spanning the fracture space even for a single object. In future, we will contribute probabilistic generative models that enable sampling fracture surface parameters to generate realistic synthetic fractures from traditional synthetic and real-world non-fractured datasets, facilitating continued research in learning-driven realistic fractured object restoration.

9. Conclusion

In this paper, we provide a fully automated approach to generate a restoration directly from a learned occupancy representation of a fractured shape by leveraging transformations between latent shape encodings. Our approach provides a single-step estimation of fractured shape code, and does not require knowledge of the fracture geometry. We show results for a variety of shapes spanning 8 object classes in ShapeNet, some of which, such as mugs, bottles, and jars represent objects that traditionally undergo damage. By generating restorations using occupancy functions that facilitate physically plausible mesh reconstruction, our work overcomes the disadvantages of existing voxel-based methods, and plays an important role in enabling rapid consumer-driven object repair, restoration of cultural heritage articles, and minimization of environmental waste.

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1945954. We thank Clarkson University and its Office of Information Technology for the use of the ACRES GPU server with 4 NVIDIA V100s sponsored by the National Science Foundation under Grant No. 1925596. We also thank Holly Rossmann and Rosalina Delwiche for their help with 3D scanning.

References

- [ADMG18] ACHLIOPAS, PANOS, DIAMANTI, OLGA, MITLIAGKAS, IOANNIS, and GUIBAS, LEONIDAS. “Learning representations and generative models for 3d point clouds”. *International conference on machine learning*. Vol. 80. 35. PMLR. 2018, 40–49 [3](#).
- [AES*11] ANTLEJ, KAJA, ERIC, MIRAN, SAVNIK, MOJCA, et al. “Combining 3D Technologies in the Field of Cultural Heritage: Three Case Studies.” *VAST (Short and Project Papers)*. 2011 [2](#).
- [ANZ06] ALAVA, MIKKO J, NUKALA, PHANI KVV, and ZAPPERI, STEFANO. “Statistical models of fracture”. *Advances in Physics* 55.3-4 (2006), 349–476 [2](#), [11](#).
- [BHTF07] BAO, ZHAOSHENG, HONG, JEONG-MO, TERAN, JOSEPH, and FEDKIW, RONALD. “Fracturing rigid materials”. *IEEE Transactions on Visualization and Computer Graphics* 13.2 (2007), 370–378 [11](#).
- [BLRW16] BROCK, ANDREW, LIM, THEODORE, RITCHIE, JAMES M, and WESTON, NICK. “Generative and discriminative voxel modeling with convolutional neural networks”. *arXiv preprint arXiv:1608.04236* 1.1 (2016), 1–9 [2](#), [3](#).
- [CAP20] CHIBANE, JULIAN, ALLDIECK, THIEMO, and PONS-MOLL, GERARD. “Implicit functions in feature space for 3d shape reconstruction and completion”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2020, 6970–6981 [2](#), [3](#).
- [CCL*22] CAO, YADI, CHEN, YUNUO, LI, MINCHEN, et al. “An Efficient B-Spline Lagrangian/Eulerian Method for Compressible Flow, Shock Waves, and Fracturing Solids”. *ACM Transactions on Graphics (TOG)* 41.5 (2022), 1–13 [11](#).
- [CFG*15] CHANG, ANGEL X., FUNKHOUSER, THOMAS, GUIBAS, LEONIDAS, et al. *ShapeNet: An Information-Rich 3D Model Repository*. Tech. rep. arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015 [3](#), [5](#).
- [CZ19] CHEN, ZHIQIN and ZHANG, HAO. “Learning implicit fields for generative shape modeling”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2019, 5939–5948 [2](#), [3](#).
- [CZG*21] CHEN, ZHANG, ZHANG, YINDA, GENOVA, KYLE, et al. “Multiresolution Deep Implicit Functions for 3D Shape Representation”. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 13087–13096 [11](#).
- [DDN20] DAI, ANGELA, DILLER, CHRISTIAN, and NIESSNER, MATTHIAS. “Sg-nn: Sparse generative neural networks for self-supervised scene completion of rgb-d scans”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 849–858 [3](#).
- [DRB*18] DAI, ANGELA, RITCHIE, DANIEL, BOKELOH, MARTIN, et al. “Scanscomplete: Large-scale scene completion and semantic segmentation for 3d scans”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2018, 4578–4587 [2](#), [3](#).
- [DWM*22] DUGGAL, SHIVAM, WANG, ZIHAO, MA, WEI-CHIU, et al. “Mending Neural Implicit Modeling for 3D Vehicle Reconstruction in the Wild”. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, 1900–1909 [2](#).
- [ESW20] ECHAVARRIA, KARINA RODRIGUEZ, SAMAROUDI, MYRSINI, and WEYRICH, TIM. “Fracturing artefacts into 3D printable puzzles to enhance audience engagement with heritage collections”. *Journal on Computing and Cultural Heritage (JOCCH)* 13.1 (2020), 1–22 [11](#).
- [FLF*19] FANG, GUODONG, LIU, SHUO, FU, MAOQING, et al. “A method to couple state-based peridynamics and finite element method for crack propagation problem”. *Mechanics Research Communications* 95 (2019), 89–95 [2](#), [11](#).
- [FMJB16] FIRMAN, MICHAEL, MAC AODHA, OISIN, JULIER, SIMON, and BROSTOW, GABRIEL J. “Structured prediction of unobserved voxels from a single depth image”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2016, 5431–5440 [2](#), [3](#).
- [GBS*15] GREGOR, ROBERT, BAUER, DANNY, SIPIRAN, IVAN, et al. “Automatic 3D Object Fracturing for Evaluation of Partial Retrieval and Object Restoration Tasks-Benchmark and Application to 3D Cultural Heritage Data.” *3DOR@ Eurographics*. 2015, 7–14 [11](#).
- [GCS*20] GENOVA, KYLE, COLE, FORRESTER, SUD, AVNEESH, et al. “Local deep implicit functions for 3d shape”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2020, 4857–4866 [2](#), [3](#).
- [GCV*19] GENOVA, KYLE, COLE, FORRESTER, VLASIC, DANIEL, et al. “Learning shape templates with structured implicit functions”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2019, 7154–7164 [2](#), [3](#).
- [GCYG08] GAO, JIAN, CHEN, XIN, YILMAZ, OGUZHAN, and GINDY, NABIL. “An integrated adaptive repair solution for complex aerospace components through geometry reconstruction”. *The International Journal of Advanced Manufacturing Technology* 36.11-12 (2008), 1170–1179 [2](#).
- [GQ91] GUPTA, MADAN M and QI, J11043360726. “Theory of T-norms and fuzzy inference methods”. *Fuzzy sets and systems* 40.3 (1991), 431–450 [2](#), [4](#).
- [GSP*14] GREGOR, ROBERT, SIPIRAN, IVAN, PAPAIOANNOU, GEORGIOS, et al. “Towards Automated 3D Reconstruction of Defective Cultural Heritage Objects.” *GCH*. Geneva, Switzerland: EUROGRAPHICS, 2014, 135–144 [2](#), [3](#), [6](#).
- [HASB20] HAO, ZEKUN, AVERBUCH-ELOR, HADAR, SNAVELY, NOAH, and BELONGIE, SERGE. “Dualsdf: Semantic shape manipulation using a two-level representation”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 7631–7641 [2](#).
- [HHN07] HARRYSSON, OLA LA, HOSNI, YASSER A, and NAYFEH, JAMIL F. “Custom-designed orthopedic implants evaluated using finite element analysis of patient-specific computed tomography data: femoral-component case study”. *BMC musculoskeletal disorders* 8.1 (2007), 1–10 [2](#).
- [HS18] HERMOZA, RENATO and SIPIRAN, IVAN. “3D reconstruction of incomplete archaeological objects using a generative adversarial network”. *Proceedings of Computer Graphics International*. New York, NY: ACM, 2018, 5–11 [2](#), [3](#), [6](#), [8](#).

- [JK20] JIA, MENG and KYAN, MATTHEW. "Learning Occupancy Function from Point Clouds for Surface Reconstruction". *arXiv preprint arXiv:2010.11378* 1.1 (2020), 1–11 3.
- [KB14] KINGMA, DIEDERIK P and BA, JIMMY. "Adam: A method for stochastic optimization". *Proc. ICLR*. La Jolla, CA: International Conference on Representation Learning, 2014, 1–15 4.
- [KPA*09] KOUTSOUDIS, ANESTIS, PAVLIDIS, GEORGE, ARNAOUTOGLOU, FOTIS, et al. "Qp: A tool for generating 3D models of ancient Greek pottery". *Journal of Cultural Heritage* 10.2 (2009), 281–295 3, 5, 6.
- [LBB19] LAMB, NIKOLAS, BANERJEE, SEAN, and BANERJEE, NATASHA KHOLGADE. "Automated reconstruction of smoothly joining 3D printed restorations to fix broken objects". *Proc. SCF*. New York, NY: ACM, 2019, 1–12 2, 11.
- [LC87] LORENSEN, WILLIAM E and CLINE, HARVEY E. "Marching cubes: A high resolution 3D surface construction algorithm". *ACM SIGGRAPH Computer Graphics* 21.4 (1987), 163–169 5.
- [LCL18] LI, JIAJIN, CHEN, BEN M, and LEE, GIM HEE. "So-net: Self-organizing network for point cloud analysis". *Proc. CVPR*. Piscataway, NJ: IEEE, 2018, 9397–9406 3.
- [LDG18] LIAO, YIYI, DONNE, SIMON, and GEIGER, ANDREAS. "Deep marching cubes: Learning explicit surface representations". *Proc. CVPR*. Piscataway, NJ: IEEE, 2018, 2916–2925 3.
- [LESP21] LIONAR, STEFAN, EMTSEV, DANIIL, SVILARKOVIC, DUSAN, and PENG, SONGYOU. "Dynamic Plane Convolutional Occupancy Networks". *Proc. WACV*. Piscataway, NJ: IEEE, 2021, 1829–1838 3.
- [LKL*20] LENGAUER, STEFAN, KOMAR, ALEXANDER, LABRADA, ARNIEL, et al. "A sketch-aided retrieval approach for incomplete 3D objects". *Computers & Graphics* 87 (2020), 111–122 2.
- [LPS*16] LI, YANGYAN, PIRK, SOEREN, SU, HAO, et al. "Fpnn: Field probing neural networks for 3d data". *Proc. NeurIPS* 29 (2016), 307–315 2, 3.
- [LSY*20] LIU, MINGHUA, SHENG, LU, YANG, SHENG, et al. "Morphing and sampling network for dense point cloud completion". *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. Menlo Park, CA: AAAI, 2020, 11596–11603 3.
- [LWL*21] LAMB, NIKOLAS, WIEDERHOLD, NOAH, LAMB, BENJAMIN, et al. "Using Learned Visual and Geometric Features to Retrieve Complete 3D Proxies for Broken Objects". *Proc. SCF*. New York, NY: ACM, 2021, 1–15 5, 11.
- [LWL20] LIN, CHEN-HSUAN, WANG, CHAOYANG, and LUCEY, SIMON. "SDF-SRN: Learning Signed Distance 3D Object Reconstruction from Static Images". *arXiv preprint arXiv:2010.10505* 1.1 (2020), 1–17 2, 3.
- [MHLZ20] MA, BAORUI, HAN, ZHIZHONG, LIU, YU-SHEN, and ZWICKER, MATTHIAS. "Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces". *arXiv preprint arXiv:2011.13495* 1.1 (2020), 1–12 2, 3.
- [MON*19] MESCHEDER, LARS, OECHSLE, MICHAEL, NIEMEYER, MICHAEL, et al. "Occupancy networks: Learning 3d reconstruction in function space". *Proc. CVPR*. Piscataway, NJ: IEEE, 2019, 4460–4470 2, 3, 5.
- [Mus22] MUSEUM, GETTY VILLA. *Case Study: A Mixing Vessel*. https://www.getty.edu/art/exhibitions/fragment_to_vase_crater.html. Accessed: 2022-04-08. 2022 2.
- [PFS*19] PARK, JEONG JOON, FLORENCE, PETER, STRAUB, JULIAN, et al. "Deepsdf: Learning continuous signed distance functions for shape representation". *Proc. CVPR*. Piscataway, NJ: IEEE, 2019, 165–174 2, 3, 9.
- [PNM*20] PENG, SONGYOU, NIEMEYER, MICHAEL, MESCHEDER, LARS, et al. "Convolutional occupancy networks". *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III* 16. Berlin, Germany: Springer, 2020, 523–540 2, 3, 9.
- [PSA*17] PAPAIOANNOU, GEORGIOS, SCHRECK, TOBIAS, ANDREADIS, ANTHOUSIS, et al. "From reassembly to object completion: A complete systems pipeline". *Journal on Computing and Cultural Heritage* 10.2 (2017), 1–22 2, 3, 6.
- [RMV*10] RENGER, FABIAN, MEHNDIRATTA, AMIT, VON TENGGEN-KOBLIGK, HENDRIK, et al. "3D printing based on imaging data: review of medical applications". *International journal of computer assisted radiology and surgery* 5.4 (2010), 335–341 2.
- [SAF*18] SEIXAS, MARIA LUIZA, ASSIS, PAULO SANTOS, FIGUEIREDO, JOÃO CURA D'ARS, et al. "The use of rapid prototyping in the joining of fractured historical silver object". *Rapid Prototyping Journal* (2018) 2.
- [SCC*11] SCOPIGNO, ROBERTO, CALLIERI, MARCO, CIGNONI, PAOLO, et al. "3D models for cultural heritage: Beyond plain visualization". *Computer* 44.7 (2011), 48–55 2.
- [SCL*04] SORKINE, OLGA, COHEN-OR, DANIEL, LIPMAN, YARON, et al. "Laplacian surface editing". *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. 2004, 175–184 7.
- [SCT*20] SITZMANN, VINCENT, CHAN, ERIC R, TUCKER, RICHARD, et al. "Metasdf: Meta-learning signed distance functions". *arXiv preprint arXiv:2006.09662* 1.1 (2020), 1–17 3.
- [SG18] STUTZ, DAVID and GEIGER, ANDREAS. "Learning 3d shape completion from laser scan data with weak supervision". *Proc. CVPR*. Piscataway, NJ: IEEE, 2018, 1955–1964 2, 3.
- [SG20] STUTZ, DAVID and GEIGER, ANDREAS. "Learning 3d shape completion under weak supervision". *IJCV* 128.5 (2020), 1162–1181 2, 3.
- [SGF16] SHARMA, ABHISHEK, GRAU, OLIVER, and FRITZ, MARIO. "Vconv-dae: Deep volumetric shape learning without object labels". *ECCV*. Berlin, Germany: Springer, 2016, 236–250 2, 3.
- [SJW*14] SCHILLING, RENÉ, JASTRAM, BENJAMIN, WINGS, OLIVER, et al. "Reviving the dinosaur: virtual reconstruction and three-dimensional printing of a dinosaur vertebra". *Radiology* 270.3 (2014), 864–871 2.
- [SK20] SON, HYEONTAE and KIM, YOUNG MIN. "SAUM: Symmetry-Aware Upsampling Module for Consistent Point Cloud Completion". *Proc. ACCV*. Berlin, Germany: Springer, 2020, 1–17 3.
- [SLK19] SARMAD, MUHAMMAD, LEE, HYUNJOO JENNY, and KIM, YOUNG MIN. "Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion". *Proc. CVPR*. Piscataway, NJ: IEEE, 2019, 5898–5907 3.
- [SLL*08] SINGARE, SEKOU, LIU, YAXIONG, LI, DICHEH, et al. "Individually prefabricated prosthesis for maxilla reconstruction". *Journal of Prosthodontics* 17.2 (2008), 135–140 2, 3, 6.
- [SM17] SMITH, EDWARD J and MEGER, DAVID. "Improved adversarial systems for 3d object generation and reconstruction". *Conference on Robot Learning*. Cambridge, UK: PMLR, 2017, 87–96 2, 3.
- [SMZG18] SHOJAEI, A, MOSSAIBY, F, ZACCARIOTTO, M, and GALVANETTO, U. "An adaptive multi-grid peridynamic method for dynamic fracture analysis". *International Journal of Mechanical Sciences* 144 (2018), 600–617 2, 11.
- [TLX*21] TANG, JIAPENG, LEI, JIABAO, XU, DAN, et al. "Sign-Agnostic CONet: Learning Implicit Surface Reconstructions by Sign-Agnostic Optimization of Convolutional Occupancy Networks". *arXiv preprint arXiv:2105.03582* 1.1 (2021), 1–16 3.
- [TTG*20] TRETSCHK, EDGAR, TEWARI, AYUSH, GOLYANIK, VLADISLAV, et al. "PatchNets: Patch-based generalizable deep implicit 3D shape representations". *Proc. ECCV*. Berlin, Germany: Springer, 2020, 293–309 2, 3.
- [VSG*20] VENKATESH, RAHUL, SHARMA, SARTHAK, GHOSH, AUROBARTA, et al. "DUDE: Deep Unsigned Distance Embeddings for Hi-Fidelity Representation of Complex 3D Surfaces". *arXiv preprint arXiv:2011.02570* 1.1 (2020), 1–9 11.

- [WDG*19] WANG, STEPHANIE, DING, MENGYUAN, GAST, THEODORE F, et al. “Simulation and visualization of ductile fracture with the material point method”. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2.2 (2019), 1–20 11.
- [WFL*19] WOLPER, JOSHUAH, FANG, YU, LI, MINCHEN, et al. “CD-MPM: continuum damage material point methods for dynamic fracture animation”. *ACM Transactions on Graphics (TOG)* 38.4 (2019), 1–15 11.
- [WKCF16] WITEK, LUKASZ, KHOURI, KIMBERLY S, COELHO, PAULO G, and FLORES, ROBERTO L. “Patient-specific 3D models for autogenous ear reconstruction”. *Plastic and Reconstructive Surgery—Global Open* 4.10 (2016), e1093 2.
- [WZX*16] WU, JIAJUN, ZHANG, CHENGKAI, XUE, TIANFAN, et al. “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling”. *Proc. NeurIPS*. San Diego, CA: Neural Information Processing Systems, 2016, 82–90 2, 3.
- [YGF21] YI, LI, GONG, BOQING, and FUNKHOUSER, THOMAS. “Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 15363–15373 3.
- [YRW*21] YU, XUMIN, RAO, YONGMING, WANG, ZIYI, et al. “Pointr: Diverse point cloud completion with geometry-aware transformers”. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 12498–12507 3, 9.
- [YWC*21] YANG, MINGYUE, WEN, YUXIN, CHEN, WEIKAI, et al. “Deep optimized priors for 3d shape modeling and reconstruction”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2021, 3269–3278 2, 3.
- [YWW*17] YANG, BO, WEN, HONGKAI, WANG, SEN, et al. “3d object reconstruction from a single depth view with adversarial learning”. *Proc. ICCV Workshops*. Piscataway, NJ: IEEE, 2017, 679–688 2, 3.
- [ZBEIDI21] ZOBEIDI, EHSAN and ATANASOV, NIKOLAY. “A Deep Signed Directional Distance Function for Object Shape Representation”. *arXiv preprint arXiv:2107.11024* 1.1 (2021), 1–23 3.
- [ZLC06] ZHENG, JIANMING, LI, ZHONGGUO, and CHEN, XI. “Worn area modeling for automating the repair of turbine blades”. *The International Journal of Advanced Manufacturing Technology* 29.9 (2006), 1062–1067 2.
- [ZQ18] ZHANG, HENG and QIAO, PIZHONG. “A state-based peridynamic model for quantitative fracture analysis”. *International Journal of Fracture* 211.1 (2018), 217–235 2, 11.
- [ZYDL21] ZHENG, ZERONG, YU, TAO, DAI, QIONGHAI, and LIU, YEBIN. “Deep implicit templates for 3D shape representation”. *Proc. CVPR*. Piscataway, NJ: IEEE, 2021, 1429–1439 2, 3.

“Who’s gonna mend my broken plate?”, cropped, by eltpics is licensed under CC BY-NC 2.0. <https://www.flickr.com/photos/eltpics/7357094714>

“Broken Fish”, cropped, by the justified sinner is licensed under CC BY-NC-SA 2.0. https://www.flickr.com/photos/the_justified_sinner/3377076803/

“Broken T2 cup - devastated!”, cropped, by Doug Beckers is licensed under CC BY-SA 2.0. <https://www.flickr.com/photos/dougbeckers/49993537707/in/photostream/>

Broken Flowerpot, cropped, provided by Wikimedia is in the public domain. <https://commons.wikimedia.org/wiki/File:BrokenClayFlowerpot.JPG>

“broken grave marker”, cropped, by Ser Amantio di Nicolao is licensed under CC BY-SA 4.0. https://commons.wikimedia.org/wiki/File:Old,_broken_grave_marker,_Westover_Church.jpg

Reconstructed Calyx Krater, cropped, provided by the Getty Museum is in the public domain. <https://www.getty.edu/art/collection/object/103V8P?altImage=6757feae-fe48-47a9-b673-c9570e3a8896#full-artwork-details>

“Broken Marble Table”, cropped, by Earthworm is licensed under CC BY-NC-SA 2.0. <https://www.flickr.com/photos/earthworm/26970777735>