# Supplementary for 'DeepMend: Learning Occupancy Functions to Represent Shape for Repair'

Nikolas Lamb, Sean Banerjee, and Natasha Kholgade Banerjee

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

## 1 Point Sampling Method

To obtain sample points $\mathcal{X}$ for training and inference, referenced in Section 3 of the main paper, for each shape $S \in \{C, F, R\}$, we sample $\frac{n}{5}$ points from a uniform distribution within a cube with side lengths of 1.1 units that surrounds $S$, and $\frac{4n}{5}$ points near the surface of $S$. To obtain the surface points, we densely sample the surface of the mesh corresponding to $S$ and perturb the sampled points by adding a random translation from a normal distribution with $\sigma = 0.02$ units. During training we merge sampled points for $C$, $F$, and $R$ to obtain $\mathcal{X}$. For training we use $n = 5,461$ for $F$ and $R$, and $n = 5,462$ for $C$. When obtaining a set of sample points for a given tuple of training shapes, after merging the sampled points we ensure that of the $n$ points, at least $m$ points are sampled from the interior of each shape and at least $m$ points are sampled from the exterior of each shape, where we use $m = \frac{n}{6}$. During inference, we use the same sampling approach except we only sample points from $F$, and use $n = 8,000$. When obtaining the sample points we ensure that half of the points belong to the interior of $F$ and half of the points belong to the exterior of $F$.

## 2 Network Implementation Details for DeepMend

As discussed in Section 3 of the main paper, we represent the occupancy function for the complete shape and the break shape with neural networks $f_{\boldsymbol{\Theta}}$ and $g_{\boldsymbol{\Phi}}$ respectively. Network $f_{\boldsymbol{\Theta}}$ has 8 dense layers where layers 1 to 4 and 6 to 8 contain 512 units each and layer 5 contains 509 - $p$ units with a skip connection to the input. Network $g_{\boldsymbol{\Phi}}$ has 5 dense layers with 512 units each. We use the leaky rectified linear unit (ReLU) for all intermediate dense layers, and use the sigmoid as the activation function for the final layers of $f_{\boldsymbol{\Theta}}$ and $g_{\boldsymbol{\Phi}}$ to output the probability that point $\mathbf{x}$ is inside the shape. During training we apply dropout with a probability of 0.2 and weight normalization [2] to each layer of $f_{\boldsymbol{\Theta}}$ and $g_{\boldsymbol{\Phi}}$. For training we use the Adam optimizer [1] with a learning rate of $5e-4$ for the network parameters $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$ and with a learning rate of $1e-3$ for the latent codes. We train for 2,000 epochs. For inference we use the Adam optimizer with a learning rate of $5e-3$ for the latent codes, and perform optimization for 1600

**Table 1.** Chamfer (CH) distance and non-fracture region error (NFRE) for our approach without the break loss ('No $\mathcal{L}_B$') and with the break loss ('$\mathcal{L}_B$'), during training. During inference for both approaches we use $\mathcal{L}_F + \mathcal{L}_{\text{prox}} + \mathcal{L}_{\text{ner}}$ to obtain the latent codes. Bold values correspond to the best performing metric value within a class.

| | Metric | airplanes | bottles | cars | chairs | jars | mugs | sofas | tables | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| No | CD | 0.108 | 0.047 | **0.064** | 0.112 | 0.091 | 0.110 | 0.097 | 0.152 | 0.097 |
| $\mathcal{L}_B$ | NFRE | 0.028 | 0.032 | 0.041 | 0.028 | 0.024 | 0.027 | 0.030 | 0.024 | 0.029 |
| $\mathcal{L}_B$ | CD | **0.037** | **0.018** | 0.092 | **0.089** | **0.064** | **0.037** | **0.052** | **0.130** | **0.065** |
| | NFRE | **0.008** | **0.011** | **0.018** | **0.009** | **0.007** | **0.008** | **0.011** | **0.012** | **0.011** |

epochs. We set $p$, the length of the complete shape latent code, to 128, and $q$, the length of the break surface latent code, to 64. As discussed in Section 3 of the main paper, for training we set the scalar multiplier on the regularization loss $\mathcal{L}_{\text{reg}}$ to $1e-4$. For inference, we set the scalar multipliers for the non-zero loss $\mathcal{L}_{\text{ner}}$, the proximity loss $\mathcal{L}_{\text{prox}}$, and the regularization loss $\mathcal{L}_{\text{reg}}$, to $1e-5$, $5e-3$, and $1e-4$ respectively. We perform a grid search for $\mathcal{L}_{\text{ner}}$ and $\mathcal{L}_{\text{prox}}$ on the jars, mugs, cars, and sofas datasets to determine the values for the scalar multipliers that give the lowest chamfer distance for predicted restoration shapes.

For the mugs dataset with 1,376 training samples and a batch size of 38, our approach takes 40 hours to train on a machine with a 40-Core Intel Xeon CPU and two NVIDIA 3090s. During inference, our approach takes 75 seconds to perform optimization to obtain the latent codes for the complete shape $\mathbf{z}_C$ and the break surface $\mathbf{z}_B$, and 8 seconds to reconstruct the restoration shape.

### 2.1 Effect of Break Loss During Training

We observe that if the break loss, $\mathcal{L}_B$, is not included during training, our approach may learn an unconstrained encoding of the break surface, as the gradient with respect to $\mathbf{z}_B$ and $\boldsymbol{\Phi}$ for sample points outside the complete shape will be negligible. As shown in Equation (3) in the main paper, by removing $\mathcal{L}_B$ the gradient for $\mathbf{z}_B$ and $\boldsymbol{\Phi}$ is computed using Equation (4) and Equation (7), which both have the output value for the break surface inside of a product with output value for the complete shape. During optimization, outside of the complete shape, the predicted value for the complete shape will trend towards 0, causing the gradient for $\mathbf{z}_B$ and $\boldsymbol{\Phi}$ to do the same. As the trending of the gradient towards 0 in areas outside of the complete shape cannot be avoided during inference, it is essential to learn a well-constrained encoding for the break surface during training.

To understand the effect of training without the break loss, we provide results with respect to the chamfer distance and the NRFE for our approach with and without $\mathcal{L}_B$ the in Table 1. We observe that training with the break loss produces restoration shapes that are more geometrically accurate those produced without the break loss, with a decrease in the chamfer distance from 0.097 to 0.065 as shown in the last column of Table 1. Training without the break loss also causes more artifacting, shown by the NFRE of 0.029 compared to 0.011 with the break

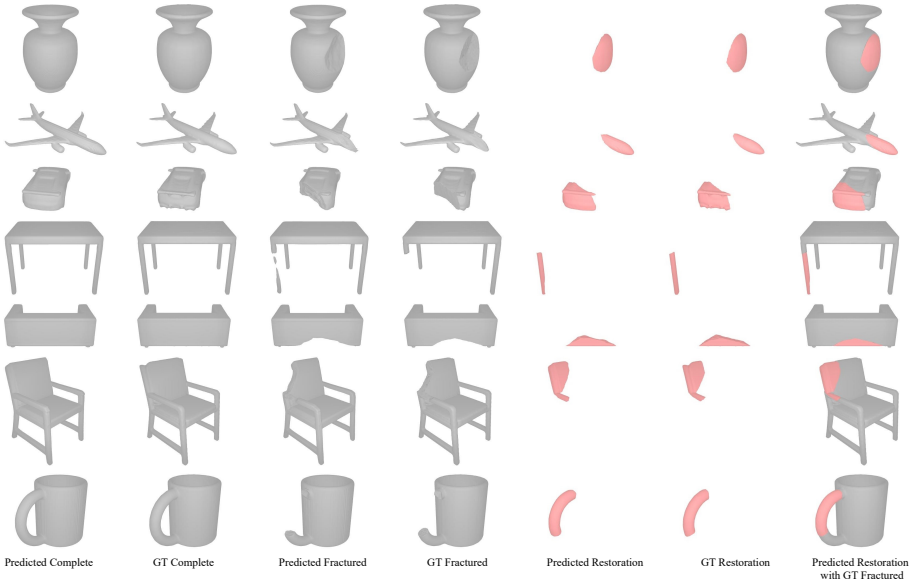


**Fig. 1.** Predicted and ground truth (GT) complete, fractured, and restoration shapes for our approach, and predicted restoration shapes shown joined to ground truth fractured shapes, corresponding to shapes shown in Figure 1 of the main paper. Restoration shapes are shown in red, all other shapes are shown in grey.

loss, as the break surface may intersect multiple times with the complete shape outside of the fracture region or be predicted as too large when the break surface encoding is under-constrained.

## 3 Inputs and Outputs for Results in Main Paper

Figure 1 shows the predicted and ground truth complete, fractured, and restoration shapes, and predicted restoration shapes joined to ground truth fractured shapes, for examples shown in Figure 1 of the main paper. In Figure 2 we show predicted and ground truth complete, fractured, and restoration shapes, and predicted restoration shapes joined to ground truth fractured shapes, for examples shown in Figure 5(b) of the main paper. In Figure 3 we show predicted and ground truth complete and fractured shapes, and restoration joined to ground truth fractured shapes for each of the loss configurations discussed in Section 5.1 'Effect of Penalties on Restoration Shape During Inference' and shown in Figure 7 of the main paper. In Figure 3, from left to right we show results for our approach with $\mathcal{L}_{\text{inf}}$, with $\mathcal{L}_{\text{inf}}$ and $\mathcal{L}_{\text{nerp}}$, with $\mathcal{L}_{\text{inf}}$ and $\mathcal{L}_{\text{ner}}$, with $\mathcal{L}_{\text{inf}}$ and $\mathcal{L}_{\text{proxp}}$, with $\mathcal{L}_{\text{inf}}$, $\mathcal{L}_{\text{proxp}}$, and $\mathcal{L}_{\text{ner}}$, with $\mathcal{L}_{\text{inf}}$ and $\mathcal{L}_{\text{prox}}$ and with $\mathcal{L}_{\text{inf}}$, $\mathcal{L}_{\text{ner}}$, and $\mathcal{L}_{\text{prox}}$, and we show the ground truth shape on the far right.

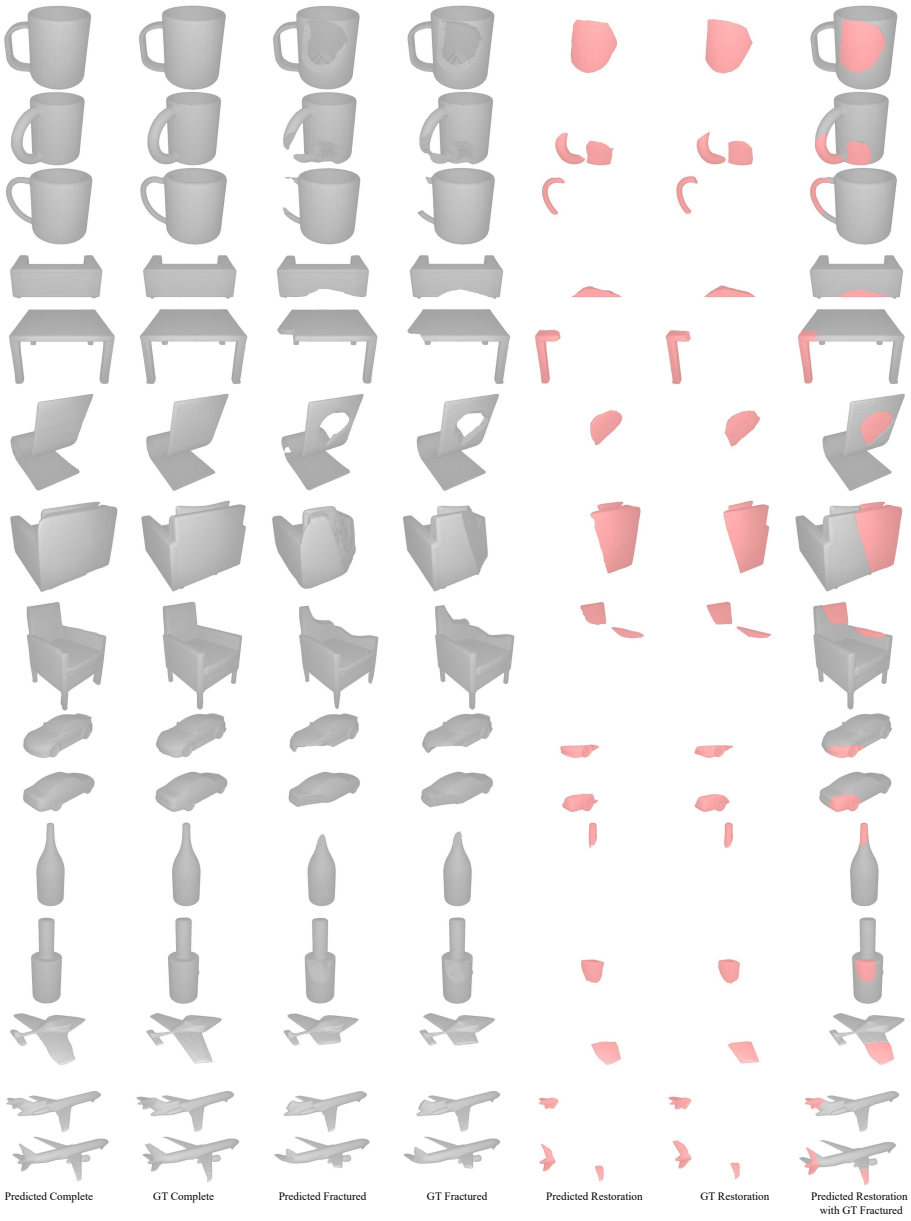


**Fig. 2.** Predicted and ground truth (GT) complete, fractured, and restoration shapes for our approach, and predicted restoration shapes shown joined to ground truth fractured shapes, corresponding to shapes shown in Figure 5(b) of the main paper. Restoration shapes are shown in red, all other shapes are shown in grey.

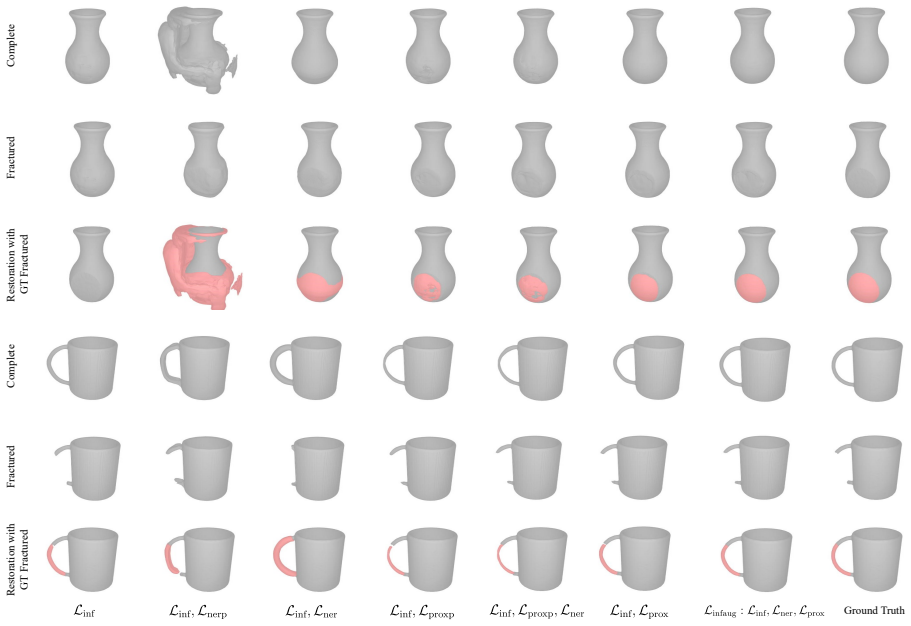


**Fig. 3.** Predicted and ground truth (GT) complete and fractured shapes, and predicted and ground truth restoration shapes shown joined to ground truth fractured shapes, corresponding to shapes shown in Figure 7 of the main paper. Restoration shapes are shown in red, all other shapes are shown in grey.

## References

1. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. ICLR. pp. 1–15. International Conference on Representation Learning, La Jolla, CA (2014)
2. Salimans, T., Kingma, D.P.: Weight normalization: A simple reparameterization to accelerate training of deep neural networks. Advances in neural information processing systems **29** (2016)