Beyond Reassembly: Fractured Object Recovery with Missing Parts

Supplementary Material

This supplementary material comprises additional details and results that complement our main paper. More specifically, we include the detailed description of our baseline methods for comparison, provide comprehensive results on fractured objects with missing parts across different categories, and elaborate the ablated method that utilizes implicit signed distance function (SDF) instead of point cloud for missing part prediction.

8. Details of Baseline Methods

Multi-view ICP We make a comparison with an ICP-based multi-view registration method [18], which was mainly used for registering 3D scans with overlaps in between.

AdaPoinTr We selected the State-of-the-Art (SOTA) point cloud completion method, AdaPoinTr [75], to serve as the final reassembled completion model for all baseline methods. We retrained the model using its official implementation, taking the incomplete assembled fragments as input.

Jigsaw We benchmark our approach against Jigsaw [36], a method that achieves shape assembly by utilizing a primal-dual descriptor. We retrained their model using its official implementation under our specific experimental settings.

PF++ We benchmark our approach against PF++ [60]. This approach achieves shape assembly by autogressively adjusting fragment poses between denoiser and verifier. We retrained their model using its official implementation under our specific experimental settings.

GARF We benchmark our approach against GARF [33], which learns fracture segmentation features in advance and achieves pose estimation based on rectified flow method. We retrained their model using its official implementation under our specific experimental settings.

Ours-TwoStage In this model, we separate pose estimation and shape prediction into two stages. For shape estimation, we directly utilize the transformer to predict the poses of the existing parts (see Fig. 5). We train this model using all the parts (without masked values) from fractured objects while applying it to testing objects with missing parts. Additionally, we train a similar shape prediction network as in our model, while taking reassembled parts as the conditional input to predict the shapes of the missing parts.

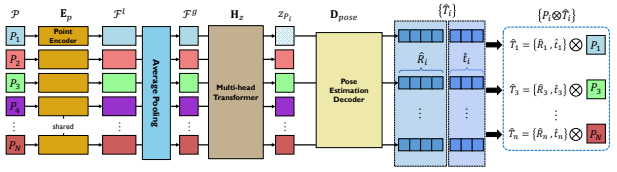

Figure 5. **Baseline model with transformer.** This model is an ablation of ours. It only estimates poses for the existing parts without predicting the shape of missing parts.

Subsequently, we employ a pre-trained DeepSDF network [47] to generate the missing parts of the model. Note that we consider the missing parts as a whole rather than individuals.

9. Results on Different Object Categories

We also showcase our results across different categories, as detailed in Table 3. Our observations indicate that simpler shapes like bottles, bowls, and vases tend to be learned more effectively during training. These shapes exhibit geometric similarities and have less variation in their structural components. Additionally, the higher abundance of training data for these three categories contributes to their better performance. In contrast, complex shapes like Toyfigures yield less satisfactory results. The network faces challenges in generalizing for this category. The relatively worse performance may be attributed to the intricacies of individual shapes.

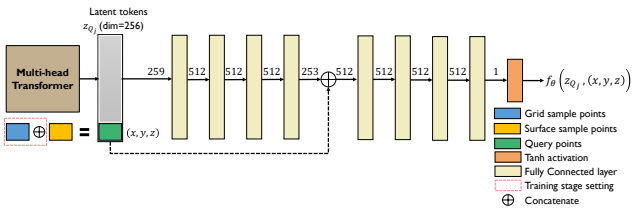

Figure 6. **SDF prediction network.** The SDF prediction decoder $\mathbf{D}_{Shape}$ comprises 8 fully connected layers. The input is the latent feature z_{Q_j} (dim 256) concatenated with the coordinates of the sample point x_k (dim 3). The output is the predicted signed distance value at x_k, approximating the function value of the ground truth signed distance field of Q_j.

Category	$\mathscr{E}_{rot} \downarrow$	$\mathscr{E}_{trans} \downarrow$	$\mathscr{E}_{missing} \downarrow$	Category	$\mathscr{E}_{rot} \downarrow$	$\mathscr{E}_{trans} \downarrow$	$\mathscr{E}_{missing} \downarrow$
Bottle	**13.71**	4.88	**34.17**	WineGlass	17.83	5.24	37.21
Bowl	**14.58**	**4.71**	37.82	DrinkBottle	16.71	4.96	39.94
Cookie	20.37	4.99	38.41	DrinkingUtensil	17.58	4.67	36.29
Spoon	18.91	5.69	40.29	Mirror	23.52	5.42	38.73
Teacup	18.62	**4.76**	35.36	BeerBottle	15.71	5.03	36.94
Teapot	14.93	4.82	38.12	Mug	19.82	4.95	40.32
ToyFigure	22.36	5.73	43.78	PillBottle	16.33	5.07	41.82
Vase	**13.83**	4.85	**34.71**	Plate	20.22	5.42	35.94
WinBottle	18.28	**4.77**	41.46	Ring	21.43	5.73	43.53
Cup	18.34	5.21	**34.23**	Statue	19.50	5.56	42.02

Table 3. **Quantitative comparisons across various categories.** The bolded values the minimum (three) observed results among all categories.

10. Ablated Shape Prediction with SDF

We also compare with a variant of our model that exploits signed distance field (SDF) as the implicit representation (instead of point cloud) for missing part prediction. The SDF prediction decoder $\mathbf{D}_{SDF}$ learns the underlying SDF of the missing parts $\{Q_j\}$ by taking as input its latent token z_{Q_j} and a query position x. The network outputs f_θ, approximating the signed distance from x to the missing shape: $f_\theta\left(z_{Q_j}, x\right) \approx \mathrm{SDF}(x)$. Here, $\mathrm{SDF}(x)$ denotes the signed distance at x, and θ represents the network parameters of $\mathbf{D}_{SDF}$. We adopt the network architecture resembling DeepSDF [47], comprising eight fully connected layers (see Fig. 6). The dimension of each intermediate fully connected layer is 512, and the final layer outputs a 1-dimensional value representing the predicted signed distance function value for each sampled point. It is worth noting that the input dimension of the first layer is 259, which is the concatenation of the 256-dimensional latent token z_{Q_j} and the coordinates of the sampled point x. Following the DeepSDF strategy, we concatenate the input to the fourth fully connected layer, enhancing the network's overall implicit field fitting performance.

For the SDF Prediction Network, we utilize a loss similar to [47] to approximate the signed distance function of missing parts as:

$$\begin{aligned} \mathscr{L}_{SDF} &= \sum_{j=1}^{M} \left\| f_\theta - \mathrm{SDF} \right\|_1 \\ &= \sum_{j=1}^{M} \sum_{k=1}^{K} \left| f_\theta(z_{Q_j}, x_k) - \mathrm{SDF}(x_k) \right| \end{aligned} \tag{10}$$

where $f_\theta(z_{Q_j}, x_k)$ represents the predicted SDF value at query location x_k using the latent code z_{Q_j}, and $\mathrm{SDF}(x_k)$ represents the ground truth SDF value. The $L1$ loss measures the absolute difference between the predicted and ground truth SDF values, which allows a more robust and accurate shape estimation compared with MSE loss according to our experience.

Please note that due to the inherent inability to directly compare SDF with point cloud, we have opted to exclude the consistency loss ($\mathscr{L}_{consistency}$) applied in cases where such a direct comparison is not feasible. The total loss function here is:

$$\mathscr{L} = \mathscr{L}_{trans} + 0.5\mathscr{L}_{rot} + 0.5\mathscr{L}_{pose} + \mathscr{L}_{SDF}. \tag{11}$$