SHARD: Global Shape-Aware Reassembly of 3D Fractured Objects

A Thesis Submitted

to the University of Texas at Austin

for the Turing Scholars program

by

Benson Ngai

to

Department of Computer Science
College of Natural Sciences
University of Texas at Austin

May 2025

ACKNOWLEDGEMENTS

I would like to thank Dr. Qixing Huang and Jiaxin Lu for introducing me to computer vision and advising me through courses, projects, and my thesis during the past two years. I'm extremely grateful to have worked with Jiaxin, whose reassembly paper got me into 3D vision into the first place and inspired me to tackle this problem; I've learned more than I could've imagined about this research topic.

ABSTRACT

Learning to automate the geometric reassembly of broken 3D objects is a complex task recently tackled by the scientific community with various practical applications, such as bone and artifact restoration. Recent works have shown that learning-based methods can reassemble fractured objects using local geometric information without requiring knowledge of the original global shape prior. These approaches struggle with objects involving complex fractures involving many or tiny fragments that lack sufficiently descriptive local geometric information, and also often assume that all fragments are present, which does not reflect real world scenarios where fragments may be missing. This paper introduces SHARD, a template-based reassembly framework that aims to reconstruct broken objects by registering fragments to an inferred global shape prior. Our local-global alignment framework mainly consists of four components: 1) a shared global-fragment geometric point cloud superpoint and feature extractor, 2) an efficient high-order convolution layer to refine and compare features of pairs to be matched, 3) superpoint matching to identify reliable coarse-level correspondences between refined global and fragment superpoint features, and 4) a fine-level registration step for improved fragment registration alignment. We evaluate SHARD on the Breaking Bad dataset and achieve promising reconstruction results through this template-guided framework, and explore potential improvements and tradeoffs in reassembly accuracy and generalizability.

1 Introduction

Reassembling fractured objects has wide practical applications in archaeological, bone, and artifact restoration. Archaeological reconstruction involves assembling fragments from historic sites to restore the original object, while bone reconstruction involves reassembling fractured bone pieces to plan for surgery. Across these applications, the overall task is constant: accurately determining original poses and spatial configurations of the provided fragments to restore the unbroken original shape. This remains a fundamental yet challenging problem in 3D vision due to incomplete fragment sets, fractures causing small fragments that lack sufficient geometric detail, and repetitive geometric patterns across fragments leading to ambiguous matches. The search space of possible alignments grows combinatorically with the number of fragments to recover; as such, brute force methods are impractical.

Past works have utilized various classical and learned approaches to tackle this task. Traditional methods utilize manually created local descriptors such as SHOT, Spin Images, FPFH, which are interpretable and don't require data to train. These descriptors are used in early pairwise-matching approaches to perform rigid fragment-fragment alignments [4], [11], and also in template-guided reassembly approaches where providing a complete global shape template and performing fragment-global matching improved robustness for reassembling thin-shelled fragments [21]. However, they don't generalize robustly across objects with topological and geometric variations, only capture local information, and are underexpressive compared to deep-learning approaches. Data-driven assembly approaches [6] [8] have shown promising results by learning feature representations directly from fractured surfaces and are capable of learning robust segmentation and multi-piece matching modules simulta-

neously. Though they generalize better across reassembly diverse object categories, they struggle with complex fractures as they solely rely on fractured surface geometries and do not consider the original object's shape (global shape prior), tackling reassembly in a bottom-up manner. Recent breakthroughs like Jigsaw++ [7] address this limitation by leveraging the reconstruction outcome of any existing reassembly method to infer and generate the complete original global shape. Incorporating this global shape prior provides several critical advantages, as it

1. Provides a strong structural prior that constrains the combinatorially large search space of possible fragment pose configurations. Reassembly can be guided by the global shape to localize each fragment to its most plausible region, significantly reducing ambiguity.

2. Identifies missing fragments. By matching existing fragments to corresponding regions on the global shape, unmatched regions indicate the absence of certain fragments, highlighting exactly which parts are missing.

3. Enforces global consistency to ensure that all fragment-region matches collectively conform to a coherent global shape, which is difficult in bottom-up pairwise approaches.

However, learned template-guided approaches have yet to be explored. Motivated by these advantages, we introduce **SHARD** (Global **SH**ape **A**ware **R**eassembly of 3**D** Fractured Objects), a learned local-global alignment framework that reassembles broken 3D objects. Given a set of fragments and the inferred global shape prior, both represented as 3D point clouds, SHARD uses the global shape as a template to guide reconstruction and compute rigid transformations to align (register) each

fragment to its corresponding region on the global shape, producing an accurate restoration of the original object. Our approach can be interpreted as a low-overlap 3D registration problem between a set of provide fragments and a global shape. Our framework comprises of four main components:

1. **Shared global-fragment superpoint and feature extractor**, refined using self and cross attention to respectively introduce intra-object context and fragment-global context.

2. **Efficient second-order feature convolution** to enable pairwise feature comparison between fragment and global superpoints.

3. **Coarse-level superpoint matching** to identify rough correspondences between fragments and regions on the global shape, roughly recovering each piece's global pose onto the original shape.

4. **Fine-level registration refinement** to finalize alignment of each fragment to its corresponding region on the unbroken global shape.

We evaluate our model on Breaking Bad [17], a dataset for geometric fracture reassembly, showing promising reassembly results for diverse object and fracture types.

2 Related Work

Early Fracture Reassembly Approaches

Early works relied heavily on handcrafted geometric feature descriptors and pairwise alignment algorithms [4, 11]. [11] proposed a complete automated system for

reassembling 3D fragments for archaeological artifacts. It extracted geometric information, such as bumpiness and curvature, of complementary fragment surface meshes, and found potential surface matches that minimized surface matching errors. The fragments were geometrically arranged optimally based on either a greedy strategy prioritizing smallest matching errors first, or a global strategy that minimized total cumulative error. This work assumed that fragment faces were nearly planar; as such, further works to handle arbitrarily shaped surfaces were developed. [4] proposed a more robust reassembly pipeline involving multiscale integral invariants for segmentation and feature extraction, introducing various global and local matching techniques. These methods ensured local and global geometric/registration consistency checks to ensure promising matching results, allowing the approach to handle partial matches and arbitrary-shaped surfaces on input fragments.

Template-Based Approaches

Another strategy is template-guided reassembly, where along with the fragments, a reference shape is provided to introduce global context for the reconstruction process, as fragment-fragment matching is improved with fragment-template matching. [21] integrated guidance from both a global reference shape and matching of adjacent fragments' fractured regions, producing many potential matching configurations that were refined to select those that maximized global matching consistency. This approach was particularly effective for thin-shell objects such as bones, which lack expressive geometric features. As such, the use of a global template disambiguates false fragment matching configurations. Even if small individual pieces lack information to perform robust pairwise matching, their placement can be estimated by how they best conform to the template's surface, ensuring that the multi-piece reassembly is globally correct and not simply locally correct.

Learned Approaches

However, these handcrafted approaches do not generalize well to complex object categories, and more importantly, cannot easily encode shape priors or reason about the global shape structure. With massive advances in deep learning within the last decade, newer, data-driven, learned approaches have been used to tackle the geometric assembly problem. New datasets specifically for geometric fracture reassembly, such as Breaking Bad [17], contain a diverse set of object meshes that have 80 fracture simulations applied to each object, resulting in 1M+ unique breakdown patterns across various object types. Rather than relying on handcrafted features, embeddings and matching functions are directly learned from this data itself. 3D point clouds are often used to represent objects and fragments due to their simplicity, flexibility and existence of powerful feature extraction architectures such as [12], [13], [18], and [20] to extract robust features for matching. Various recent learned approaches have shown promising results incorporating unique model architectures to perform multi-piece matching on these learned features. To reassemble two fragments, [1] proposed a self-supervised approach to generate pairwise 3D shape-mating data, using transformers and adversarial learning to reason about the fit of the two fragments and predict poses to tightly "mate" and reassemble them together. For reassembling multiple fragments, Jigsaw [8] proposed a promising framework to jointly learn fracture surface segmentation and matching between fragments using local fracture geometric features. Another piece-wise matching approach, Proxy Match Transform (PMT) [6], proposed an sub-quadratic feature convolution, focusing on feature matching to reliably and efficiently establish local correspondences on mating surfaces. These piece-matching frameworks learned expressive geometric cues to perform multi-piece matching and generalized well across diverse object categories, but suffered from re-

assembling complex fractures or those with missing and misaligned pieces. Jigsaw++ [7] addressed these limitations by learning category-agnostic generative shape priors to reconstruct complete shapes from partially assembled reassembly outputs (such as Jigsaw), and found improved global reconstruction accuracy and robustness. With these advances, our work focuses on using the generated shape prior provided by Jigsaw++ to learn a template-guided framework and reassemble objects by realigning their fragments to this global shape prior.

Point Cloud Registration

SHARD can be interpreted as a 3D point cloud registration problem, where the task is to align two or more point clouds into a single unified coordinate system. Specifically, our framework involves aligning multiple fragment clouds to the global shape prior that represents our unified scene. Recent registration works show promising results to learn classifiers that determine overlapped sections and predict point matching between fragments, but typically require approximately 30% overlap [3] [23] [14]. When registering fractured surfaces, this is considered as extreme low-overlap registration, where overlap between pieces could fall below 4%. When registering fragments to the global shape template, fractures involving numerous pieces could easily cause for particular fragment overlap ranges to fall below $10 - 20\%$.

3 Global Shape-Aware Reassembly

Given a set of 3D broken fragment pieces $F = \{F_1, F_2, ..., F_n\}$ with unknown orientations and an inferred 3D global shape prior G, the goal of reassembly is to estimate and recover the poses (orientation and translation) of each fragment F_i to recon-

struct the original, intact shape GT. Concretely, we aim to determine the 6-DoF rigid transformations $T_i \in SE(3)$ for each fragment F_i, such that the union of all pose-estimated fragments approximate the global shape:

$$T_1(F_1) \cup T_2(F_2) \cup ... \cup T_n(F_n) \approx G$$

We assume that the inferred global shape accurately represents the ground truth shape GT. Thus, reassembling fragments F to match G directly restores GT; hence, $G \equiv GT$. Each fragment transformation is restricted to the $SE(3)$ group, representing all rigid-body transformations composed of a rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$, expressed as a 4x4 matrix:

$$T = \begin{bmatrix} R_{3\times3} & t_{3\times1} \\ 0 & 1 \end{bmatrix}$$

Fragments and their corresponding global shape are represented as 3D point clouds, unordered sets of (x, y, z). Each fragment point cloud contains points from the unfractured and fractured surfaces, whereas the global shape point cloud only contains points from the original, unbroken surface.

We propose SHARD, a global shape-aware reassembly framework for 3D fractured objects. Instead of relying solely on local geometric matching between fragment fracture surface geometries, SHARD incorporates a global template shape prior G as an anchor to guide fragment alignment, and fragments are reassembled by aligning (registering) each piece to G. This local-global alignment framework leverages global context to better reason about semantic relationships between fragments and their corresponding regions on the global shape. Incorporating semantic meaning into

our matching framework through local geometric features between fragments and the global shape helps SHARD understand which fragments should correspond to where, and hence reassembles fragments back together with reduced ambiguity and improved reconstruction accuracy.

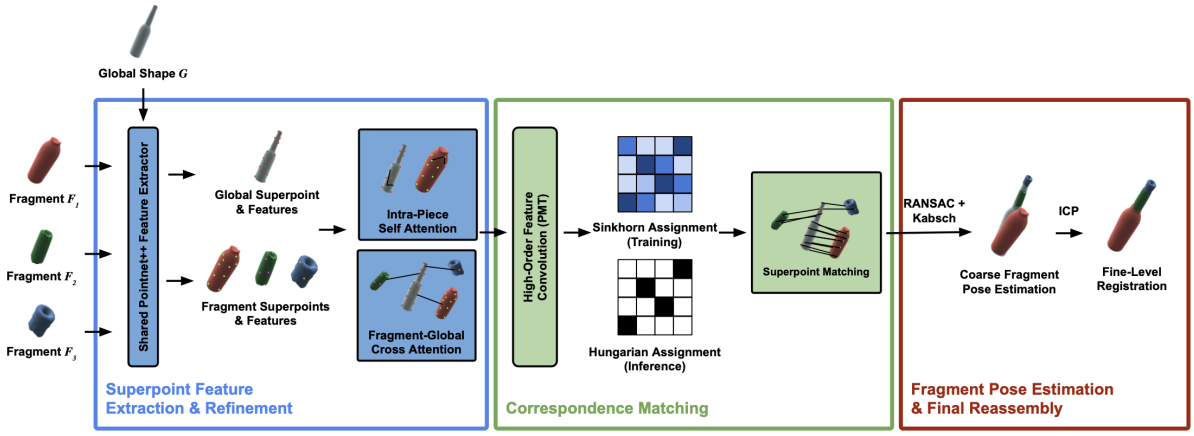

Figure 1: End-to-end Reassembly Pipeline for SHARD

Figure 1 illustrates SHARD's overall pipeline. First, the framework performs surface segmentation and extracts global and local geometric features from each fragment and the global shape, summarizing local neighborhoods into superpoints with associated embeddings. These embeddings are refined using self and cross-attention layers to incorporate intra-piece context and fragment-global context. An efficient second-order convolution correlates local geometric patches between fragment and global superpoints to identify likely correspondences between fragment and global regions. One-to-one assignment optimal correspondences, providing initial coarse-level pose estimates. A final fine-level registration step refines these poses, resulting in a precise and accurate alignment of fragments onto the global shape. The subsequent sections detail each stage of SHARD's pipeline.

3.1 Fractured Surface Segmentation

SHARD begins by preprocessing the input fragments $F = \{F_1, F_2, \ldots, F_n\}$. Each point $p_{ij} \in \mathbb{R}^3$ denotes the j^{th} point in fragment i, where $F_i = \{p_{i1}, p_{i2}, \ldots, p_{i|F_i|}\}$. Each fragment point cloud contains points p_{ij} classified as fractured (points on breakage surfaces) or unfractured (points from the original unbroken surface).

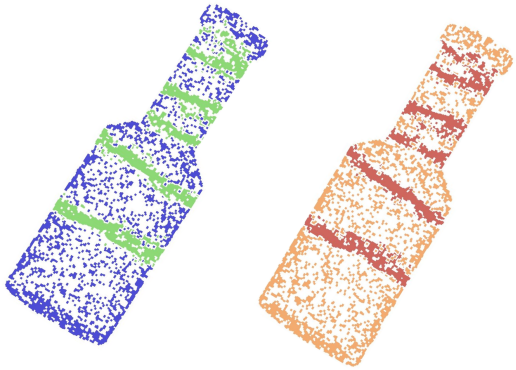

Figure 2: Visualization of fractured & unfractured Points on fragments of a sample object [8].

We cannot extract geometric features without first segmenting out fractured points, as local geometric descriptors capture fine-grained details like curvature and surface normals from point neighborhoods. Since fragment alignment involves matching unfractured fragment patches to unbroken patches on the global shape, fractured points introduce noise and lead to incorrect correspondences, as the global shape contains no fractured surfaces. We require surface segmentation to filter out fractured points, which is defined by the binary classification task to learn

$$Seg(p_{ij}) = \begin{cases} 1 & \text{if } p_{ij} \text{ is on a fractured surface} \\ 0 & \text{if } p_{ij} \text{ not on fractured surface} \end{cases} \tag{1}$$

Jigsaw [8] solved this by successfully learning to separate the fractured and the unfractured regions by predicting confidence scores for fractured points. To learn this, all fragments F_i were oriented in their known ground truth poses T_i^{GT} such that $G' = \bigcup_{i=0}^{n} T_i^{GT}(F_i)$, forming the original shape G with the sole difference that both unfractured and fractured points are included in G'. Then, a point $p_{ij} \in F_i$ is labeled as fractured if the nearest point on another fragment is within a threshold α

$$\arg \min_{q \in G' \setminus F_i} \|p_{ij} - q\|_2 < \alpha \tag{2}$$

which identifies fracture points as those close to complementary points on adjacent fragments. Instead of using a learned classifier, we directly apply this threshold-based labeling in our framework for simplicity. After segmentation, all fractured points are removed from each fragment. Resulting point clouds for all fragments F_i and global G contain exclusively unfractured surfaces and are ready for feature extraction.

3.2 Shared Global-Fragment Feature Extractor

We leverage PointNet++ [13], a hierarchical feature extractor capable of encoding rich, local neighborhood geometry around points to extract a reduced set of representative keypoints (superpoints) from each fragment F_i and global G, denoted as

S^{F_i} and S^G. Specifically, this extracts

$$S^{F_i} = \{s_1^{F_i}, s_2^{F_i}, \ldots, s_{|S^{F_i}|}^{F_i}\}, \quad |S^{F_i}| << |F_i|$$

for each fragment, and similarly

$$S^G = \{s_1^G, s_2^G, \ldots, s_{|S^G|}^G\}, \quad |S^G| << |G|$$

for the global shape. Matching at this reduced superpoint level is beneficial in capturing coarse semantic and geometric correspondences to match whole fragments with regions of G, rather than attempting precise point-to-point alignment.

PointNet++'s [13] extracts these superpoints through set abstraction layers and multi-scale grouping mechanisms. First, farthest-point-sampling selects representative centroids spanning each fragment and global surfaces. For each centroid, local neighborhoods are gathered at varying radii (scales) to capture different granularities of geometric detail. A shared MLP processes each neighborhood to create translation-invariant, permutation-invariant, and scale-aware descriptors. These layers are iteratively applied to reduce the number of centroids available at every step until a compact set of superpoints remain. With this setup, we extract superpoints S^{F_i} and S^G and their corresponding feature descriptors for each fragment $F_i \in F$ and global shape G. Each fragment superpoint $s_j^{F_i}$ has an associated feature embedding $e_j^{F_i}$, and each global superpoint s_k^G has a corresponding embedding e_k^G. By using the same backbone, we ensure fragment and global embeddings exist in a consistent, unified featured space for robust geometric comparison.

3.3 Attention-Based Feature Refinement

Each PointNet++ superpoint embedding encodes geometric details of its local neighborhood. However, these embeddings lack broader context about their global relevance. To introduce longer-range context, we refine each embedding using self and cross attention [19], allowing each superpoint to understand its geometric and semantic relationship within its fragment or global structure.

3.3.1 Intra-Fragment & Intra-Global Context

Intra-fragment context allows for each superpoint to recognize its geometric and semantic relevance relative to the fragment it belongs to, thereby capturing longer-range context from a further superpoint on the other side of, such as along an edge, of the same fragment. We incorporate point cloud self-attention using a Point-Transformer layer [22] for all fragments F_i in an object to enrich each superpoint embedding $e_j^{F_i}$ by aggregating information from neighboring superpoints. For each fragment F_i, superpoints are refined individually by treating each embedding $e_j^{F_i}$ as a query to attend to other superpoints within the same fragment. Point cloud self attention integrates both feature similarity and spatial proximity by utilizing relative positional embeddings in attention calculations. As a result, each superpoint embedding captures longer-range structural relationships within its fragment. We apply this process to the global shape G in order to enrich each global superpoint embedding e_k^G with contextual information from the entire, unbroken shape.

3.3.2 Global-Fragment Context

While intra-fragment context improves embeddings within individual fragments and the global shape, our ultimate task is to align fragments F_i to corresponding regions on global shape G. To achieve this, our framework learns fragment-global relationships with cross-attention. Specifically, fragment superpoint embeddings $e_j^{F_i}$ act as queries, while global superpoint embeddings e_k^G serve as both keys and values. Fragment superpoints selectively attend to relevant global shape superpoints based on feature similarity, enabling embeddings to capture semantic and geometric relationships between fragments and regions of the global shape. Each fragment's superpoint embedding incorporates global context, enhancing our framework's ability to accurately identify fragment-to-global correspondences.

3.4 Global-Fragment Feature Matching

With extracted fragment superpoints S^{F_i} and global shape superpoints S^G, and their attention-refined feature embeddings E^{F_i} and E^G, our goal is to establish coarse correspondences between fragment and global regions. We achieve this using second-order convolutions to robustly correlate fragment and global superpoint features.

3.4.1 High Order Feature Convolutions

High order feature convolutions are powerful mechanisms for capturing complex correlations between features. They have demonstrated effectiveness in prior works in image correspondence and 3D registration [16, 15, 9, 10, 2, 5]. In our task, we

attempt to find reliable matches between each fragment superpoint $s_j^{F_i}$ and global superpoint s_k^G to align fragments to the global shape; naturally, this can be tackled with second-order feature convolutions.

Traditional first-order convolution summarize local neighborhoods for a single point by linearly combining neighboring points' features. Given neighborhoods $N(s_j^{F_i})$ for a fragment superpoint and $N(s_k^G)$ for a global superpoint, they compute

$$\text{FirstOrderConv}(F_j^{S_i}) = \sum_{n \in N(s_j^{F_i})} K[n - s_j^{F_i}] \cdot e_n^{F_i} \tag{3}$$

$$\text{FirstOrderConv}(G_k) = \sum_{n \in N(s_k^G)} K[n - s_k^G] \cdot e_n^G \tag{4}$$

where K is a convolutional kernel mapping 3D displacement vectors $\mathbb{R}^3 \to \mathbb{R}$ to scalars. The scalar outputs from Equations 3 and 4 could be compared to give some similarity measure, but this limits the capacity to capture intricate, pair-wise relationships between points in each neighborhood, losing information such as how individual neighbors in $N(s_j^{F_i})$ pair up with individual neighbors in $N(s_k^G)$. This setup poses a few problems. For example, we cannot disambiguate whether a high overall similarity comes from one very strong neighbor-to-neighbor match or from many weak ones. Further, we lose spatial consistency cues, where we can't learn to boost or diminish scores between pairs of points across both neighborhoods that have particular relative offsets. Two very different neighborhood match patterns could even possibly average to the same scalars and appear identical.

In contrast, second-order convolutions explicitly compute pairwise interactions between neighborhoods, capturing richer geometric and semantic relationships. Given

fragment superpoints S^{F_i} and global superpoints S^G with respective embeddings $E^{F_i} = \{e_j^{F_i}\}$ and $E^G = \{e_k^G\}$, a second-order convolution computes pairwise correlations between neighborhoods around a fragment superpoint $s_j^{F_i}$ and global superpoint s_k^G, expressed as:

$$\text{SecondOrderConv}(F_i, G)_{(s_j^{F_i}, s_k^G)} = \sum_{(n,m) \in N(s_j^{F_i}) \times N(s_k^G)} C_{(n,m)} \cdot K[n - s_j^{F_i}, m - s_k^G] \quad (5)$$

where $C_{(n,m)} = e_n^{F_i} \cdot (e_m^G)^T$ represents the feature correlation between a fragment superpoint $n \in F_i$ and global superpoint $m \in G$, and K is a convolutional kernel mapping a 6 displacement vector between points between neighborhoods to scalars ($\mathbb{R}^6 \to \mathbb{R}$). The result of this convolution yields a fragment-global pairwise correlation matrix $Corr \in \mathbb{R}^{|S^{F_i}| \times |S^G|}$, where each entry represents the correlation value between the fragment superpoint $s_j^{F_i}$ and global superpoint s_k^G. We apply the Gaussian kernel as our similarity measure on $Corr$ to this correlation matrix, transforming it into an affinity (similarity) matrix $A \in [0, 1]^{S^{F_i} \times S^G}$. This affinity matrix A serves as a robust indicator of potential matching regions, where pairs of fragment-global superpoints with higher affinity indicate greater probability of matching.

$$A_{ij} = \exp\Big(-\frac{Corr_{i,j}^2}{2\sigma^2}\Big) \quad (6)$$

3.4.2 Proxy Match Transform

Though second-order feature convolutions effectively capture pairwise feature correlations, they come at quadratic complexity by explicitly computing the full neighborhood correlation matrix $C \in \mathbb{R}^{|X| \times |Y|}$ for point clouds X and Y. Each pair of

neighborhoods is compared exhaustively, making this impractical for large inputs. Recent works have introduced Proxy Match Transform (PMT) [6], which provides a sub-quadratic approximation of second-order convolutions. Instead of constructing the full correlation matrix C, PMT uses a learned low-dimensional proxy tensor P as an intermediate reference space, significantly reducing computational overhead. PMT approximates C by projecting features F_X and F_Y into the low-dimensional proxy space:

$$C_X = F_X P^\top \quad \text{and} \quad C_Y = F_Y P^\top \tag{7}$$

where $C_X \in \mathbb{R}^{|X| \times D_{proxy}}$, $C_Y \in \mathbb{R}^{|Y| \times D_{proxy}}$, and $D_{proxy} << |X|, |Y|$. Intuitively, C_X represents projecting our feature embedding into a smaller shared "reference space" defined by proxy tensor P, which effectively represents a small set of basis features / reference points. This converts our complex, high-dimensional embeddings to simpler, compressed features describing each point's similarity with these proxy basis features. This is simply done for a point, though. To introduce context from that point's neighborhood, which is integral to second order convolutions, the PMT embedding is refined and computed as

$$\text{PMT}(F_x) = \sum_{h \in [N_h]} A_x^{(h)} C_x w_x^{(h)} \tag{8}$$

where h represents the head index, $A_x^{(h)}$ represents the local attention matrix and $w_x^{(h)}$ represents a learnable weight scalar. $A_x^{(h)}$ weights local neighborhood interactions to spatially close and relevant neighbors, and the $w_x^{(h)}$ scales each embedding to ensure that each attention head contributes effectively to the overall neighborhood's

representation. If the PMT embedding is computed for two superpoints (and their features) to compare, the 2nd-order convolution is then approximated by

$$\mathrm{PMT}(F_x) \cdot \mathrm{PMT}(F_y)^\top \approx \mathrm{SecondOrderConv}(F_x, F_y) \tag{9}$$

PMT achieves sub-quadratic complexity through proxy tensor P. Essentially, P acts as a shared intermediate space, enabling information flow between fragments and global shape superpoints without explicitly computing expensive pairwise neighborhood correlations. This reduces complexity from $O(|X| * |Y| * N(\cdot)^2 * D_{embedding})$ in traditional second-order convolutions to $O((|X| + |Y|) * N(\cdot) * D_{proxy})$, where $N(\cdot)$ represents the neighborhood size and $D_{proxy} << |X|, |Y|$.

3.5 Fragment-Global Alignment

Each fragment superpoint embedding $e_j^{F_i}$ and global superpoint embedding e_k^G is converted into its respective PMT embedding $\mathrm{PMT}(e_j^{F_i})$ and $\mathrm{PMT}(e_k^G)$. We approximate the second-order convolution between these embeddings as

$$\mathrm{PMT}(e_j^{F_i}) \cdot \mathrm{PMT}(e_k^G)^\top \approx \mathrm{SecondOrderConv}(e_j^{F_i}, e_k^G) \tag{10}$$

resulting in a correlation matrix $Corr \in \mathbb{R}^{|S^{F_i}| \times |S^G|}$. Applying the Gaussian kernel (equation 6), we obtain the affinity matrix $A \in [0, 1]^{|S^{F_i}| \times |S^G|}$. For a given object, columns of A represent superpoints across all fragments in F, rows represent superpoints on the global shape G, and A_{kj} represents how similar (affine) the fragment superpoint j is to the global superpoint i.

3.5.1 Coarse Superpoint Matching

To align fragments F_i to their corresponding global shape regions in G, we establish coarse-level correspondences between their superpoints. If most superpoints from a fragment correspond to a specific subset of superpoints on the global shape, coarse pose estimation can roughly align each fragment onto its matching region on the global shape with that set of correspondences. With affinity matrix A, this correspondence-finding step naturally becomes an assignment problem. Given a set of fragment superpoints, a set of global superpoints, and the similarity A_{kj} between each fragment and global superpoint, we aim to find the best possible matching between fragment and global superpoints to maximize overall affinity. Since each fragment F_i originates from exactly one region on G, each fragment superpoint should uniquely match exactly one global superpoint. Thus, we aim to find a one-to-one (1:1) assignment between fragment superpoint $s_j^{F_i}$ and exactly one global superpoint s_k^G based on their pairwise affinity values A_{kj}.

Inference-Time Matching

The Hungarian algorithm is used to solve this 1:1 matching. Given affinity matrix A, the algorithm treats the problem as finding a maximum-weight matching in a bipartite graph by converting our similarity matrix into a cost matrix, and iteratively adjusts rows and columns to find an optimal assignment. The final result,

$$H = \text{Hungarian}(A) \tag{11}$$

is a binary permutation matrix $H \in \{0,1\}^{|S^{F_i}| \times |S^G|}$, the optimal 1:1 matching that

maximizes our overall fragment-global superpoint similarity. Formally,

$$H \in \{0,1\}^{|S^{F_i}| \times |S^G|}, \quad \text{where} \quad H_{kj} = \begin{cases} 1, & \text{if } s_j^{F_i} \text{ matches } s_k^G \\ 0, & \text{otherwise} \end{cases}$$

Training-Time Matching

The Hungarian algorithm is non-differentiable and thus unsuitable for training. Instead, the Sinkhorn algorithm provides a differentiable and efficient approximation to the 1:1 assignment matching problem. It iteratively normalizes rows and columns of the A to produce a doubly stochastic matrix S, where each row and column sums to 1, acting as probability distributions. Sinkhorn is formalized as

$$S = \text{Sinkhorn}(\exp(\frac{A}{\tau})) \tag{12}$$

where the affinity matrix A is exponentiated to ensure positive values and is divided by τ, a temperature parameter to control the sharpness of output probabilities. $S_{:,j}$ represents the probability distribution of matching fragment superpoint $j \in F$ to the global superpoint set, and $S_{k,:}$ represents the probability distribution of a global superpoint $k \in G$ to any of the fragment superpoints in F.

3.5.2 Coarse Pose Estimation

Correspondences between fragment superpoints $S^{F_i} = \{s_1^{F_i}, s_2^{F_i}, \ldots, s_{|S^{F_i}|}^{F_i}\}$ and global superpoints $S^G = \{s_1^G, s_2^G, \ldots, s_{|S^G|}^G\}$ are established by the Hungarian assignment matrix H, enabling estimation of fragment poses based on these predicted correspondences. We denote the set of global superpoints that correspond to fragment F_i's

superpoints as

$$S_{F_i}^G = \{s_k^G \text{ s.t. } H_{kj} = 1, s_j^{F_i} \in S^{F_i}\}$$

which explicitly defines $S_{F_i}^G$ as the matched subset of global superpoints from S^G, based on the predicted Hungarian correspondences from H for fragment F_i. Given these coarse correspondences, we estimate a rigid transformation to align the fragment superpoints S^{F_i} to their corresponding global superpoints $S_{F_i}^G$.

SVD Rigid Transformation

In matching scenarios where most correspondences are correct, estimating a rigid transformation between two point clouds is performed with the **Kabsch algorithm**, which computes the optimal transformation by minimizing the sum of squared distances between correspondences. Given a set of corresponding points $X \subseteq S^{F_i}$ and $Y \subseteq S_{F_i}^G$, Kabsch computes a rotation $R \in \mathbb{R}^{3\times3}$ and translation $t \in \mathbb{R}^3$ minimizing

$$\arg\min_{R,t} \sum_{i=1}^{n} ||(Rx_i + t) - y_i||^2 \tag{13}$$

where $x_i \in X$ and $y_i \in Y$. The optimal R and t is computed via singular value decomposition (SVD), by:

$$\begin{aligned} X_{\text{centered}} &= x_i - \mu_x, \quad Y_{\text{centered}} = y_i - \mu_Y \\ U, \Sigma, V^\top &= \text{SVD}(X_{\text{centered}} Y_{\text{centered}}^\top) \\ R &= VU^\top \quad \text{s.t. } \det(R) = 1 \\ t &= \mu_Y - R\mu_x \end{aligned} \tag{14}$$

where μ_X and μ_Y are the centroids of X and Y. $R_{optimal}$ and $t_{optimal}$ can then be applied to an entire fragment to coarsely align F_i to its corresponding region on G.

RANSAC Rigid Transformation

In practical scenarios, the predicted correspondences from assignment matrix H contain noisy outliers. For robust rigid transformation estimation, we use **RANSAC (Random Sample Consensus)**. RANSAC iteratively selects small random subsets of correspondences between S^{F_i} and $S^G_{F_i}$ to estimate candidate rigid transformations using the Kabsch algorithm (Equation 14) on these subsets, and evaluates the consensus between the transformed point cloud $R(S^{F_i}) + t$ and $S^G_{F_i}$ by counting the number of inliers. Inliers are defined as the number of correspondences whose transformed fragment superpoints fall within a predefined, small distance threshold to their matched global superpoints. Formally, given candidate transformations $R_{candidate}, t_{candidate}$ computed using Kabsch from a random subset of points, RANSAC maximizes the consensus set size defined by:

$$\text{inliers}(R_{candidate}, t_{candidate}) = \{(x_i, y_i) \text{ s.t. } ||(R_{candidate}x_i + t_{candidate}) - y_i||_2 < \alpha\} \tag{15}$$

where α is the specified inlier distance threshold. After a fixed number of iterations, the transformation $R_{optimal}, t_{optimal}$ with the largest number of inliers is the selected transformation. We primarily apply the RANSAC approach to ensure robust coarse rigid transformation estimations. As a result, each F_i is transformed to $F_i^{coarse} = R_{optimal}F_i + t_{optimal}$ using that fragment's optimal predicted rigid transformation.

3.5.3 Fine-Level Registration

Each fragment F_i^{coarse} aligns coarsely with its corresponding region in G, and a simple final refinement step is applied to fine-tune the coarse rigid transformations $R_{optimal}$,

$t_{optimal}$ to more closely register each fragment point F_i with G. **Point-to-plane Iterative Closest Point (Point-to-Plane ICP)** is applied to each F_i to G, which is a simple, iterative variant of the vanilla ICP algorithm used to register one point cloud to another based on point distances. It iteratively minimizes the distances from fragment points to tangent planes for the nearest points on the global shape to align their surfaces through the minimization objective

$$\arg\min_{R,t} \sum_{i=0}^{N} [n_i^\top ((R_{x_i} + t) - y_i)]^2 \tag{16}$$

where $x_i \in F_i^{coarse}$, $y_i \in G$, and n_i represents the normal for y_i. Typically, ICP requires sufficient overlapping points between two point clouds and a good initial pose for registration for accurate registration. If neither holds true, registration may result in a poor local minimum, producing an inaccurate alignment. This further explains why our coarse alignment stage is necessary to provide a reliable initialization for each fragment F_i. Equation 16 produces optimal fine rigid transformation $T_{fine} = (R_{fine}, t_{fine})$. We combine the coarse and fine transformations such that

$$T^{final} = T^{coarse} \cdot T^{fine} \tag{17}$$

$$T^{coarse} = \begin{bmatrix} R^{coarse} & t^{coarse} \\ 0 & 1 \end{bmatrix} \tag{18}$$

$$T^{fine} = \begin{bmatrix} R^{fine} & t^{fine} \\ 0 & 1 \end{bmatrix} \tag{19}$$

The final predicted aligned fragment points is then computed with the final, optimal

rigid transformation on a fragment point cloud F_i such that:

$$F_i^{predicted} = T^{final} \cdot F_i \tag{20}$$

Applying this to all initial fragments F_i, our framework completes the reassembly task by reconstructing $G = F_1^{predicted} \cup F_2^{predicted} \cup \ldots \cup F_{|F|}^{predicted}$, where each fragment F_i is aligned to their corresponding region on G.

3.6 Training Objectives

Our reassembly framework is trained with the objective of predicting robust, accurate correspondences between fragment superpoints S^F and global superpoints S^G, as covered in section 3.5.1. We believe that with good correspondences, the coarse matching step performs sufficiently well to roughly align fragments F_i to G and learn semantic and geometric relationships. Our training loss $\mathcal{L}$ is composed of two training losses, defined as:

$$\mathcal{L} = \lambda_{circle}\mathcal{L}_{circle} + \lambda_{KL}\mathcal{L}_{KL} \tag{21}$$

where $\mathcal{L}_{circle}$ represents weighted circle loss, $\mathcal{L}_{KL}$ represents KL divergence, and λ_i are tunable weights for each loss term. Alternative metrics to evaluate point cloud registration, such as chamfer distance, rotation error, or translation error, were considered but are often not as informative or discriminative to provide meaningful signals for model training.

3.6.1 Ground Truth Construction

As our objective focuses on correspondence matching between S^F and S^G, our prediction is a soft, doubly-stochastic Sinkhorn matrix S, where S_{kj} represents the similarity of matching global superpoint k to fragment superpoint j, for a given object. A ground truth matrix S^{GT} is required to compare our predicted superpoint correspondences to ground truth correct correspondences. To coarsely match each fragment F_i to its approximately correct region on G, correspondences between fragment superpoints and global superpoints that belong to the same region should be correct, while superpoints from different regions of G should be incorrect.

To learn this, the ground truth matrix S^{GT} must label global superpoints s_k^G with which fragment it belongs to. This is not provided, though, as the global G is an inferred shape with no fragment labeling. We construct S^{GT} using an approach similar to the segmentation step in section 3.1. All fragment superpoints S^{F_i} are oriented in their known ground truth orientations T_i^{GT} and superimposed on top of the ground truth G, such that $G = \bigcup_{i=1}^{n} T_i^{GT}(S^{Fi})$. Since all fragments are superimposed on top of the global shape, the fragment label for each global superpoint S_k^G is simply the label of the nearest neighbor fragment superpoint. We define the hard ground truth matrix $S^{GT} \in \{0,1\}^{|S^{F_i}| \times |S^G|}$, such that

$$S^{GT} = \begin{cases} 1 \text{ if } s_k^G \text{ and } s_j^{F_i} \text{ belong to the same fragment } F_i \\ 0 \text{ otherwise} \end{cases} \tag{22}$$

However, this approach alone lacks precision. Consider a scenario where a bottle is fractured in half with one piece containing the neck and upper body, and the other piece containing the lower body. Correspondences that match superpoints on the

neck of a fragment to superpoints on the upper body of the global shape will still be considered correct in S^{GT}, as all neck and upper body superpoints on both the fragment and global shape have the same fragment label. To address this, we refine S^{GT} by incorporating spatial distances between fragment and global superpoints. We introduce a cost matrix C^{GT}, defined as:

$$C^{GT} = \begin{cases} ||s_k^G - s_j^{F_i}||^2 \text{ if } s_k^G \text{ and } s_j^{F_i} \text{ belong to the same fragment } F_i \\ \infty \text{ otherwise} \end{cases} \tag{23}$$

This penalizes fragment-global superpoint matches proportionally to their Euclidean distances, encouraging correct correspondences to be spatially close. We convert these costs into similarity scores with a temperature parameter τ':

$$S^{GT} = \exp(\frac{-C^{GT}}{\tau'}) \tag{24}$$

Finally, each column (fragment superpoint) of S^{GT} is normalized, ensuring it represents a probability distribution over possible global superpoint matches for each fragment superpoint. S^{GT} serves as the ground truth distribution during training, guiding the model toward accurate geometric and semantic correspondences.

3.6.2 Weighted Circle Loss $\mathcal{L}_{circle}$

Weighted Circle Loss is a pairwise-similarity metric-learning loss function focusing on learning geometric features in point-matching and descriptor learning tasks, making it particularly suitable for SHARD's fragment-global superpoint matching task. It learns discriminative embeddings by explicitly distinguishing between correct

(positive) and incorrect (negative) matches, pulling positive fragment-global superpoint pairs closer together within a specified margin m_p and pushing the negatively matched pairs apart outside a specified margin m_n. First, we compute the Euclidean distance between our fragment and global superpoint embeddings $e_j^{F_i}$ and e_k^G, where

$$d_{e_j^{F_i}, e_k^G} = \|e_j^{F_i} - e_k^G\|_2 \tag{25}$$

We introduce two margins, m_p (for positives) and m_n (for negatives), to control how close / far positive and negative pairs should be from each other, which are used to define positive and negative weights w^p and w^n, where

$$w^p = \max\big(0,\ d_{e_j^{F_i}, e_k^G} - m_p\big), \qquad\qquad w^n = \max\big(0,\ m_n - d_{e_j^{F_i}, e_k^G}\big) \tag{26}$$

These weights eliminate easy fragment-global superpoint pairs (those already within or outside the desired margins m_p and m_n) and highlight hard ones. Positive and negative pairs that violate their margins are aggregated via

$$\mathcal{L}_s^p = \text{LogSumExp}(\ w^p\ (d_{e_j^{F_i}, e_k^G} - m_p)), \quad \mathcal{L}_s^n = \text{LogSumExp}(\ w^n\ (m_n - d_{e_j^{F_i}, e_k^G})) \tag{27}$$

where $\mathcal{L}_s^p$ and $\mathcal{L}_s^n$ represent the positive and negative pair losses from some superpoint s to the other set of superpoints to be matched to. These terms are computed from each global superpoint to all fragment superpoints and vice versa. As such, $\mathcal{L}_{s_k^G} \in \mathbb{R}^{|S^F|}$, and $\mathcal{L}_{s_j^F} \in \mathbb{R}^{|S^G|}$. These loss vectors are aggregated by averaging the positive and negative pair losses for the worst violating global-fragment superpoint pairs, for all possible pairs. A row and column loss term are respectively calculated to consider both the row direction (fragment-to-global matches) and similarly in the

column direction (global-to-fragment), and are then averaged via

$$\mathcal{L}_{\text{row}} = \frac{1}{N_{\text{row}}} \sum_{k} \text{softplus}((L^{p}_{s^{G}_{k}} + L^{n}_{s^{G}_{k}})), \quad \mathcal{L}_{\text{col}} = \frac{1}{N_{\text{col}}} \sum_{j} \text{softplus}(L^{p}_{s^{F_i}_{j}} + L^{n}_{s^{F_i}_{j}}) \tag{28}$$

$$\mathcal{L}_{circle} = \tfrac{1}{2} \left(\mathcal{L}_{\text{row}} + \mathcal{L}_{\text{col}}\right) \tag{29}$$

With weighted circle loss, SHARD learns to match fragment geometries to similar matching geometric regions on the global shape G.

3.6.3 Kullback-Leibler (KL) Divergence $\mathcal{L}_{KL}$

While weighted circle loss emphasizes geometric feature matching, it alone may allow ambiguous matches where fragments with similar geometry could incorrectly map onto identical global shape regions. To resolve this, we incorporate a KL divergence loss to guide predicted correspondences S towards their ground truth distributions S^{GT}. Recall that the columns of S each represent a probability distribution indicating how likely a particular fragment superpoint $s^{F_i}_{j}$ matches to the set of global superpoints s^{G}, and that the columns of S^{GT} represent the ground truth probabilities of this same distribution, as defined by the fragment-global superpoint pair's fragment label and Euclidean distance (section 3.6.1). Then, we compute a column-wise KL divergence such that

$$\mathcal{L}_{KL}(S \| S^{GT}) = \frac{1}{|S^{F}|} \sum_{j=1}^{|S^{F}|} \sum_{i=1}^{|S^{G}|} S^{GT}_{ij} \log \frac{S^{GT}_{ij}}{S_{ij}} \tag{30}$$

By explicitly focusing on each fragment superpoint's matching distribution, our model learns to accurately predict correspondences from fragments to their correct regions on the global shape, learning semantic relationships and enhancing alignment accuracy. A row-wise KL loss is not applied, as we treat each fragment superpoint as a source to match onto the reference shape G, not the other way around. However, minimizing our column-wise KL divergence also minimizes row-wise KL divergence due to the nature of the 1:1 assignment task.

4 Experiments

We evaluate our reassembly framework through experimental evaluations on a large-scale fracture assembly dataset. Our results show that SHARD shows promising results, producing accurate fragment restorations across a diverse object categories both quantitatively and qualitatively. We further focus on conducting model architecture and training objective ablation studies to analyze the contribution of various components and loss terms in SHARD's pipeline.

4.1 Protocols

Dataset

We utilize the Breaking Bad dataset [17], a comprehensive dataset designed for geometric fracture reassembly tasks. The dataset contains diverse categories of object meshes that have undergone simulated physical fractures, and is divided into three categories: `everyday` for common object meshes, `artifact` for archaeological-themed objects, and `other` for miscellaneous objects. We focus on both training and

testing a subset of category types in the `everyday` category; specifically, our dataset focuses on the following object subcategories: `BeerBottle`, `DrinkBottle`, `WineBottle`, `Bottle`, `PillBottle`, `Bowl`, `Mug`, `Plate`, `Statue`, `ToyFigure`, `Vase`. We use an $80-20$ train-test split to produce a training set of 4674 fractures, and a validation set of 1224 fractures across these object subcategories in `everyday`. Each train/test example contained the set of F fragments and the corresponding global shape G. As these were all input object meshes, each fragment and global object mesh was uniformly sampled with $N = 5000$ points from the object's surface mesh as a point cloud. A "sampling-by-object" [8] approach was taken, where the sum of points across all fragments F for an object summed up to 5000 points, the same number of points sampled from global shape G. This was critical for our 1:1 assignment task.

Evaluation Metrics

To quantitatively evaluate SHARD's reassembly results, we define and report coarse accuracy and fine accuracy, which directly translate from our coarse-matching and fine-registration objectives defined in our framework. We also consider other fragment pose estimation evaluation metrics from [8], such as Chamfer Distance, Rotation Error, and Translation Error.

Coarse accuracy evaluates whether fragment superpoints correctly corresponded to their ground truth regions on the global shape, where a correspondence was correct if a fragment superpoint $s_j^{F_i}$ and global superpoint s_k^G belong to the same fragment F_i Formally, this is given by $S_{(k,j)}^{GT} > 0$ (Equation 24). The number of correct and incorrect coarse correspondences was found, and the percentage accuracy was reported.

Fine accuracy provides a more precise measure. For each fragment superpoint $s_j^{F_i}$, the correct fine correspondence is the nearest-neighbor global superpoint s_k^G that it

SHARD	Parameter
Training Parameters	
Epochs	50
Batch Size	4
Learning Rate	0.001
Training optimizer	Adam
Learning rate scheduler	Cosine
Minimum learning rate for Cosine scheduler	1e-5
Sampling Parameters	
# Points sampled per object	5000
η: Segmentation ground truth label threshold	0.025
Model Architecture Parameters	
PointNet++ input feature dimension	6
PointNet++ output feature dimension	512
# Intra-Fragment self-attention heads	8
# Fragment-Global cross-attention heads	8
# PMT heads	8
PMT radius	0.1
τ: Temperature for affinity matrix A	1e-5
τ': Temperature for soft ground truth matrix S^{GT}	0.05
# RANSAC correspondences	3
RANSAC threshold	0.075
# RANSAC iterations	500
Loss Parameters	
λ_{circle}: Weighted circle loss term ratio	1.0
m_p: Weighted circle loss positive margin	0.1
m_n: Weighted circle loss negative margin	1.4
λ_{KL}: KL divergence loss term ratio	1.0

Table 1: Training, sampling, model architecture, and loss parameters for SHARD Reassembly Framework.

should match to if superimposed onto its corresponding region on G. Formally, the correct correspondence between fragment superpoint j and all global superpoints is given by $\arg\max S^{GT}_{:,j}$. To relax this, the predicted fragment superpoint was correct if it was within a small distance threshold of 0.01 from the actual global super-

point. The number of correct and incorrect fine correspondences was found, and the percentage accuracy was reported.

Chamfer Distance measures the similarity between our pose estimated fragments $F^{predicted}$ and our global shape G to evaluate how close our reassembly output was to the original ground truth. Ideally, $F^{predicted} = G$.

$$\text{Chamfer}(F^{predicted}, G) = \frac{1}{|F^{predicted}|} \sum_{f \in F^{predicted}} \min_{g \in G} ||f - g||^2 + \frac{1}{|G|} \sum_{g \in G} \min_{f \in F^{predicted}} ||g - f||^2 \tag{31}$$

where f is a fragment point in $F^{predicted}$ and g is a global shape point $g \in G$.

Mean Absolute Error (MAE) and **Root Mean Squared Error (RMSE)** were used to evaluate the rotation and translation accuracies of each fragment F_i when aligning them to global shape G. Specifically,

$$\text{MAE}(R^{F_i}) = \frac{1}{3}||R^{F_i}_{GT} - R^{F_i}_{pred}||_1, \quad \text{MAE}(t^{F_i}) = \frac{1}{3}||t^{F_i}_{GT} - t^{F_i}_{pred}||_1 \tag{32}$$

$$\text{RMSE}(R^{F_i}) = \frac{1}{\sqrt{3}}||R^{F_i}_{GT} - R^{F_i}_{pred}||_2, \quad \text{RMSE}(t^{F_i}) = \frac{1}{\sqrt{3}}||t^{F_i}_{GT} - t^{F_i}_{pred}||_2 \tag{33}$$

where $R^{F_i}_{pred}$ and $R^{F_i}_{GT}$ represent the predicted and ground truth rotations in Euler angles, and $t^{F_i}_{pred}$ and $t^{F_i}_{GT}$ represent the predicted and ground truth translation vectors for each fragment F_i. The mean error for each object was computed as the average error of all its fragments.

4.2 Performance

We report SHARD's performance on various object subcategories from the `everyday` mesh set as displayed in Table 1.

Evaluation Metric	**Reported Value**
Coarse Accuracy (%)	83.03 %
Fine Accuracy (%)	2.31%
Chamfer Distance ($\times 10^{-3}$)	2.076801
Rotation RMSE (degrees)	59.346165
Rotation MAE (degrees)	34.263524
Translation MAE ($\times 10^{-2}$)	15.760097
Translation RMSE ($\times 10^{-2}$)	17.790916

Table 2: Reported Evaluation Metrics for SHARD on `everyday` Object Subcategories

SHARD demonstrated promising results in coarsely aligning each fragment to the inferred global shape, achieving 83.03% in coarse accuracy. As such, we succeeded in our task to learn corresponding matching regions of the global shape for each fragment. Though the fine accuracy of 2.31% appears incredibly low, this was expected due to the strict definition of a correct fine correspondence such that a fragment and global superpoint are only considered a correct match if they are within a small distance of each other. Accurate fragment reassemblies achieved fine accuracies of $5 - 10\%$, while highly inaccurate alignments resulted in $< 1\%$ fine accuracy. In comparison to other works, Jigsaw [8] achieved fine accuracies ranging from $1 - 8\%$. SHARD produces promising fragment pose estimations with low translation RMSE of 17.79×10^{-2}, indicating that fragments were roughly aligned to their ground truth global shape region with high accuracy and proving the efficacy of our superpoint correspondence matching module. However, the reported rotation RMSE of 59.79% is higher than desired, which can be attributed to the final fine-level registration

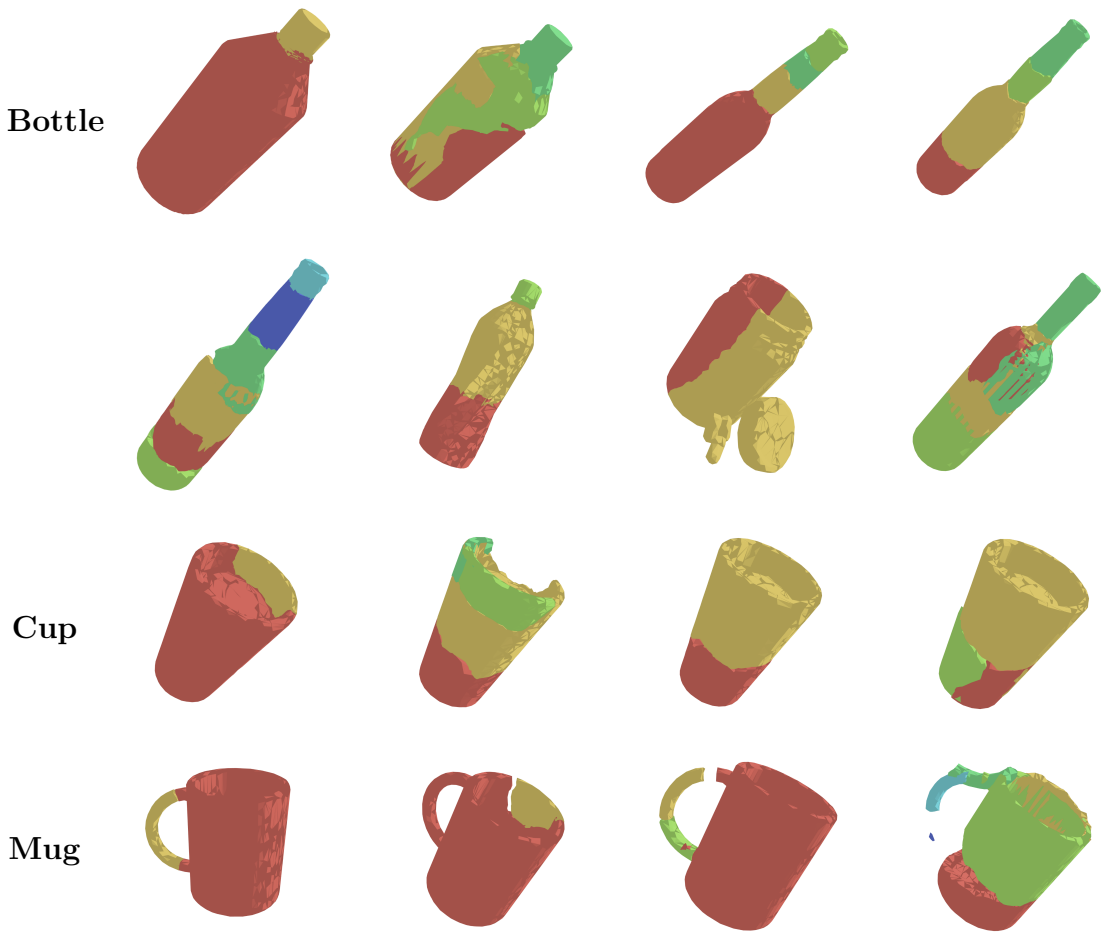

Figure 3: Qualitative results of SHARD's reassembly outputs on `Bottle, BeerBottle, DrinkBottle, WineBottle, PillBottle, Cup, Mug` object categories from the Breaking Bad `everyday` dataset. Object meshes are only used for better visualization as SHARD's outputs are point clouds. Fragments are colored for better viewing.

step of our pipeline. As we use simple point-to-plane ICP to finely register each fragment onto its predicted region on the global, this was too underpowered to escape local minima and correctly orient the fragment, even if the fragment was correctly matched to its region on the global shape (Figure 5).

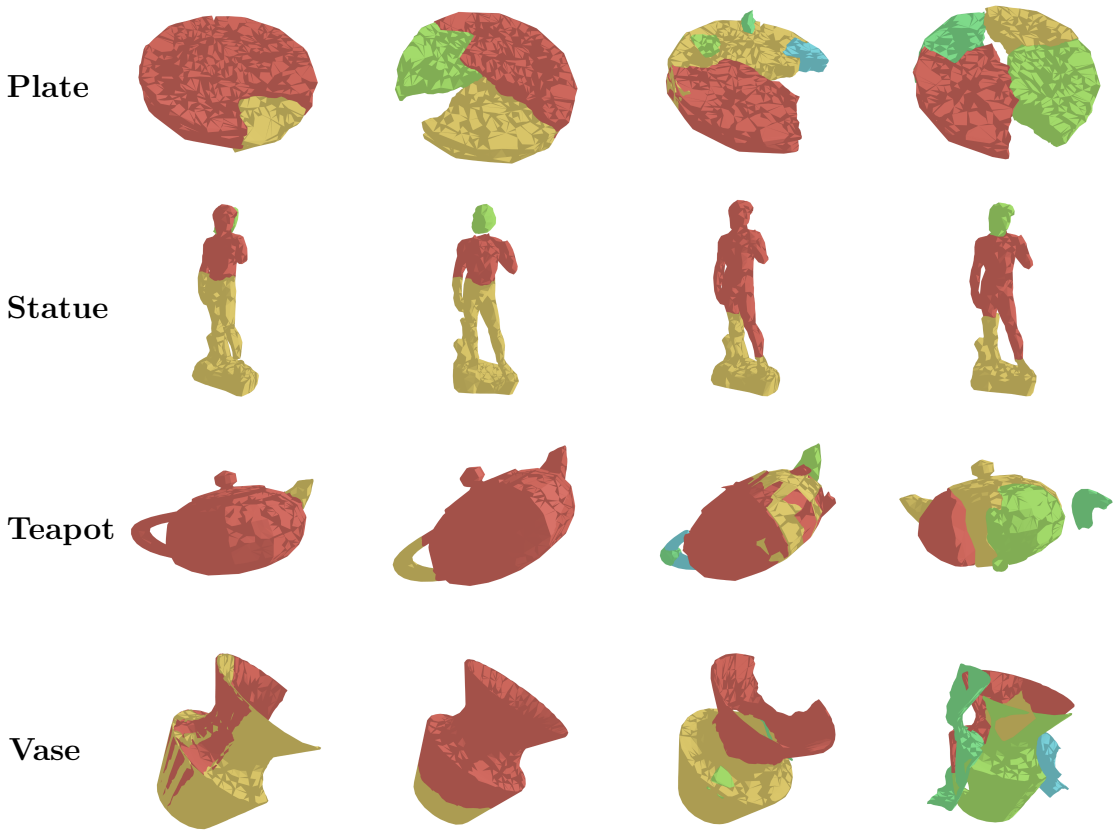

Figure 4: Qualitative results of SHARD's reassembly outputs on `Plate, Statue, Teapot, Vase` object categories from the Breaking Bad `everyday` dataset. Object meshes are only used for better visualization as SHARD's outputs are point clouds. Fragments are colored for better viewing.

Qualitative results from SHARD's reassembly output are displayed in Figures 3 and 4. Across all object categories, we observe that each fragment was matched with its corresponding region on the global shape G, resulting in largely accurate fragment positions. Several problems can be seen from these visualizations. First, there exist overlaps or gaps between fragments, which respectively suggest that some global superpoints were overcompeted for or were not robustly matched to fragment super-

Figure 5: Sample reassemblies with high coarse accuracy and low fine accuracy. In each example, a fragment was roughly matched to the correct corresponding region on the global shape, but are oriented incorrectly due to our simple fine-level registration step using point-to-plane ICP.

points. The former could be attributed to the sole column-wise KL divergence loss term that treats all global superpoints as a probability distribution for a particular fragment superpoint, and not the other way around. This would lead to fragments competing for the same global-superpoints despite our 1:1 assignment. The latter can be attributed to filtering out fracture superpoints on fragment fractured surfaces in SHARD's preprocessing step, resulting in scenarios with more global superpoints than fragment superpoints. Second, reconstructions for objects involving many fragments or symmetric fragments do not perform as well, such as `Mug`, `Plate`, and `Vase`. Both can be attributed to fragments that lack discriminative geometric detail, resulting matching fragments to many potential regions on the global shape. However, in most cases, we observe promising qualitative results across our diverse set of object categories that fracture into a reasonable number of pieces.

4.3 Ablation Studies

Model Architecture Ablations

We analyze the contributions of various modules within SHARD's pipeline by ablating these modules from our full model, training and testing them with their ablations on the same data, and evaluating them. Specifically, we ablate the attention layers (intra-piece self attention and fragment-global cross attention), the ProxyMatchTransform layer, and both. As such, we compare against the following model architecture blations: Full (Transformer + PMT), None (No Transformer, No PMT), PMT Only, and Transformer Only.

From our evaluation metrics, we observe similar coarse accuracies across all ablations, and lower fine accuracy for None. Transformer Only performed the best in Fine Accuracy, Chamfer Distance, and Rotation Error, while the full model performed the best in Translation Error. The fully ablated (None) model is expected to perform the worst in reassembly, as this involves performing fragment-global superpoint matching purely on the unrefined PointNet++ output features. As these features only describe their corresponding superpoint's local neighborhood, many ambiguous matching regions could exist for a given fragment, leading to more false correspondences. PMT Only experiences a slight improvement from this, but still struggles with ambiguity without fragment-global context. Transformer Only excels at matching fragments with their corresponding regions on the global shape; with more correct superpoint correspondences establish, it produces better rough fragment pose estimations that result in high fine accuracy, low chamfer distance, and low rotation error after ICP refinement. Though we expected the full model to outperform the ablated models for all metrics, this can be attributed to the larger

parameter space to optimize. Including both transformer and PMT layers significantly increases the number of parameters to optimize, and as such, the full model may require more data to train. Nevertheless, the full model still produces promising qualitative reconstructions across most categories as displayed in Figures 7 and 8.

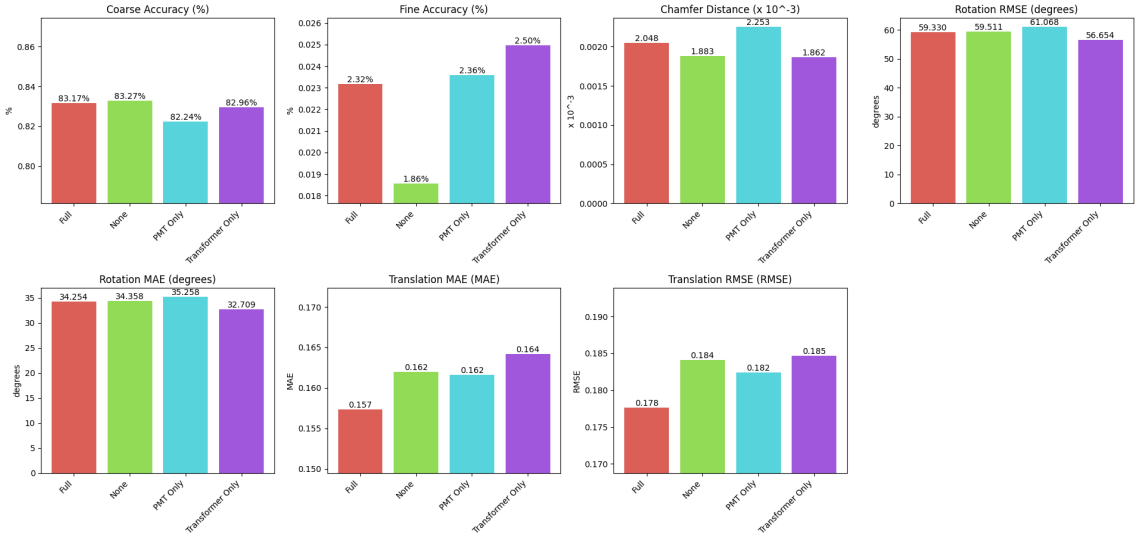

Figure 6: Evaluation metrics of SHARD's model architecture ablations. Full represents both Transformer and PMT, None represents neither Transformer nor PMT.

Loss Ablations

We also ablate each of our loss terms $\mathcal{L}_{circle}$ and $\mathcal{L}_{KL}$ to analyze their contributions in training our reassembly framework. To evaluate this, the full SHARD pipeline was trained on a small subset of `BeerBottle, DrinkBottle, WineBottle` object fracture examples, where one Full training variant contained both losses ($\mathcal{L}_{circle}$ and $\mathcal{L}_{KL}$), one contained Circle Loss Only ($\mathcal{L}_{circle}$), and one contained KL Divergence Loss Only ($\mathcal{L}_{KL}$). All training variants were tested on these object categories and evaluated against one another.

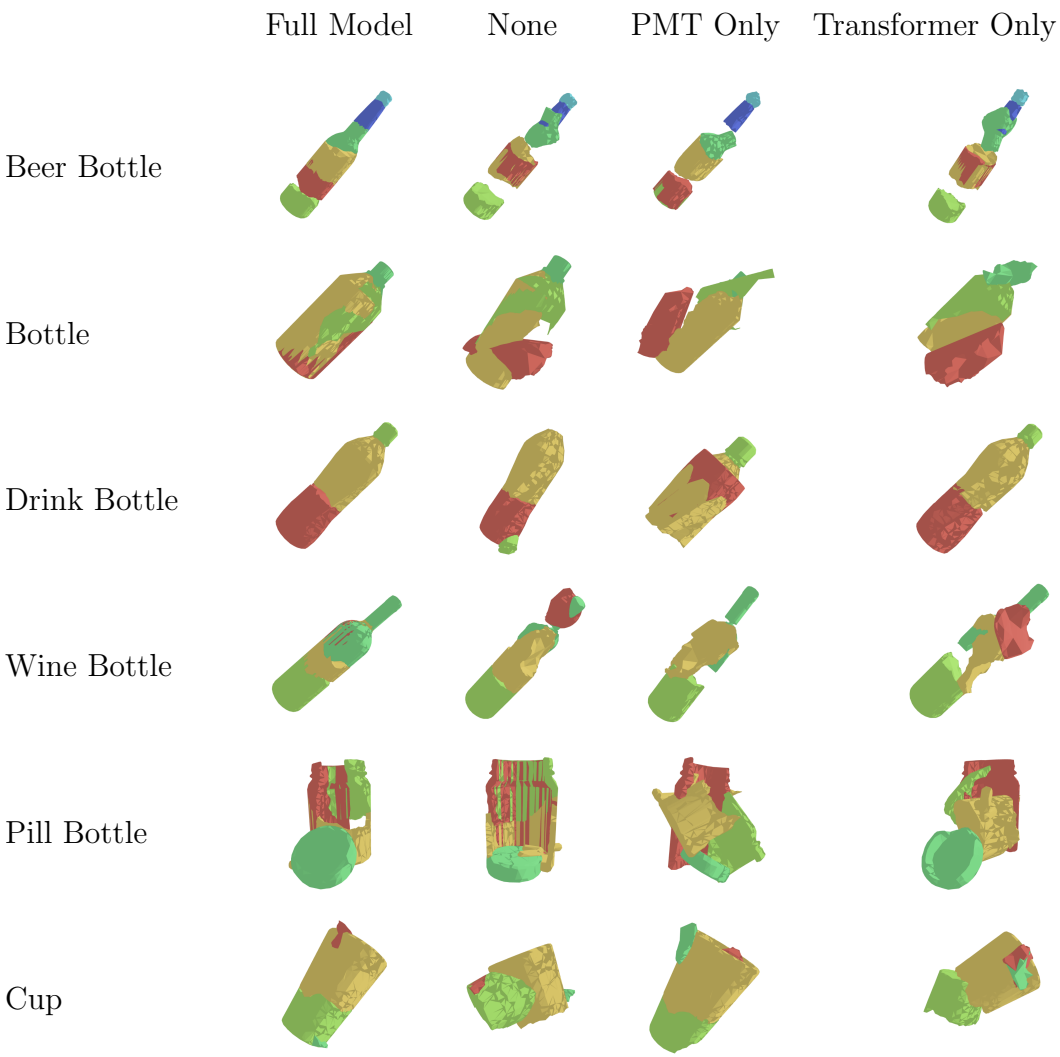

Figure 7: Model architecture ablation results of SHARD's reassembly outputs on `Beer Bottle`, `Bottle`, `Drink Bottle`, `Wine Bottle`, `Pill Bottle`, `Cup` object categories from the Breaking Bad `everyday` dataset.

Figure 9 compares three training variants across our evaluation metrics: Full (Weighted Circle Loss + KL), Circle Only, and KL Only. As shown, training SHARD with only circle loss achieves the worst performance across all metrics, while KL Only un-

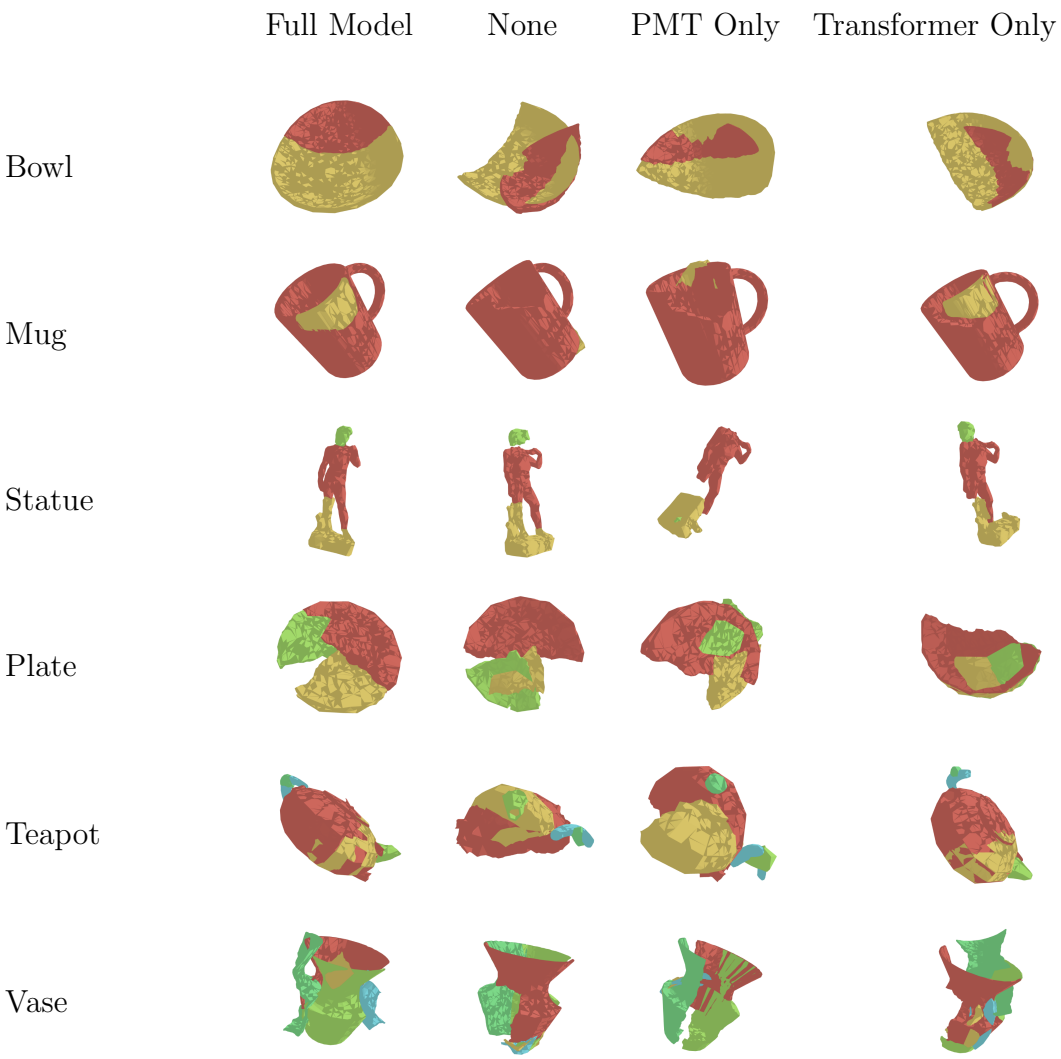

Figure 8: Model architecture ablation results of SHARD's reassembly outputs on `Bowl`, `Mug`, `Statue`, `Plate`, `Teapot`, `Vase` object categories from the Breaking Bad `everyday` dataset.

expectedly yields the best evaluations. Since the KL term directly aligns the soft correspondence distributions S to the ground truth assignment matrix S^{GT}, this provides a strong supervisory signal for correct fragment-global superpoint matching,

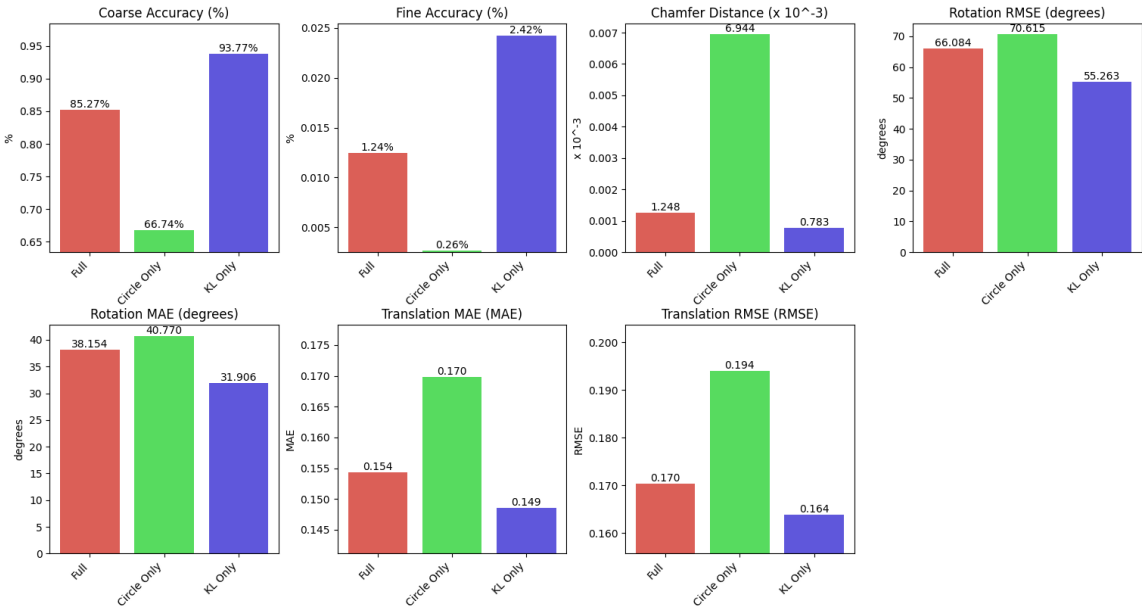

Figure 9: Evaluation metrics of SHARD's loss ablations. Full represents both $\mathcal{L}_{circle}$ and $\mathcal{L}_{KL}$, Circle Only represents $\mathcal{L}_{circle}$, and KL Only represents $\mathcal{L}_{KL}$.

explaining its effectiveness. On the other hand, weighted circle loss lacks the explicit matching distribution guidance, suggesting its poor performance when trained alone. However, when combining both losses in the Full objective, we achieve a balance between structuring our embedding space to be discriminative through circle loss and explicitly pushing our assignment distributions to be more accurate, creating a complementary effect to help our model generalize better to new fracture patterns and prevent distributional overfitting. Further, training and evaluating these ablations on the full SHARD framework using an extremely small subset of data may explain the underperformance of the Circle Loss and Full training objectives in comparison to the KL Only variant, which learns distribution alignment much faster.

5 Conclusion

In this work, we introduced SHARD, a template-guided framework for 3D fractured object reassembly that leverages an inferred global shape prior to guide local-global fragment alignment. Our pipeline combines PointNet++ superpoint extraction, self- and cross-attention for contextual feature refinement, ProxyMatchTransform for high-order feature convolution, and a two-stage coarse and fine pose estimation strategy to align fragments to their corresponding regions on the global shape, restoring the original object. Our results on the Breaking Bad dataset demonstrate that SHARD achieves strong coarse correspondence accuracy and promising fragment pose estimation predictions, and our ablation studies explore the benefits of the transformer and PMT modules in our architecture along with our training objective formulation. Nevertheless, fine alignment remains challenging for small, low-detail, or symmetric fragments due to our naive point-to-plane ICP refinement, and gaps or overlaps can appear as fragments compete for similar global regions in our framework. Future research directions may include exploring more powerful fine-registration techniques such as fracture-surface matching, explicit handling of missing fragment pieces, and training a fully end-to-end joint optimization of the template shape prior generation and assembly process to improve accuracy and robustness.

References

[1] Yun-Chun Chen et al. "Neural shape mating: Self-supervised object assembly with adversarial shape priors". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12724–12733.

[2] Christopher Choy, Wei Dong, and Vladlen Koltun. “Deep global registration”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2514–2523.

[3] Shengyu Huang et al. “Predator: Registration of 3d point clouds with low overlap”. In: *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2021, pp. 4267–4276.

[4] Qi-Xing Huang et al. “Reassembling fractured objects by geometric matching”. In: *ACM Siggraph 2006 papers*. 2006, pp. 569–578.

[5] Seungwook Kim, Juhong Min, and Minsu Cho. “Transformatcher: Match-to-match attention for semantic correspondence”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 8697–8707.

[6] Nahyuk Lee et al. “3D geometric shape assembly via efficient point cloud matching”. In: *arXiv preprint arXiv:2407.10542* (2024).

[7] Jiaxin Lu, Gang Hua, and Qixing Huang. “Jigsaw++: Imagining Complete Shape Priors for Object Reassembly”. In: *arXiv preprint arXiv:2410.11816* (2024).

[8] Jiaxin Lu, Yifan Sun, and Qixing Huang. “Jigsaw: Learning to assemble multiple fractured objects”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 14969–14986.

[9] Juhong Min and Minsu Cho. “Convolutional hough matching networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2940–2950.

[10] Juhong Min, Dahyun Kang, and Minsu Cho. "Hypercorrelation squeeze for few-shot segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 6941–6952.

[11] Georgios Papaioannou and Evaggelia-Aggeliki Karabassi. "On the automatic assemblage of arbitrary broken solid artefacts". In: *Image and Vision Computing* 21.5 (2003), pp. 401–412.

[12] Charles R Qi et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[13] Charles Ruizhongtai Qi et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space". In: *Advances in neural information processing systems* 30 (2017).

[14] Zheng Qin et al. "Geometric transformer for fast and robust point cloud registration". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11143–11152.

[15] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. "Efficient neighbourhood consensus networks via submanifold sparse convolutions". In: *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part IX 16*. Springer. 2020, pp. 605–621.

[16] Ignacio Rocco et al. "Neighbourhood consensus networks". In: *Advances in neural information processing systems* 31 (2018).

[17] Silvia Sellán et al. "Breaking bad: A dataset for geometric fracture and reassembly". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 38885–38898.

[18] Hugues Thomas et al. “Kpconv: Flexible and deformable convolution for point clouds”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 6411–6420.

[19] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[20] Yue Wang et al. “Dynamic graph cnn for learning on point clouds”. In: *ACM Transactions on Graphics (tog)* 38.5 (2019), pp. 1–12.

[21] Kang Zhang et al. “3D fragment reassembly using integrated template guidance and fracture-region matching”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2138–2146.

[22] Hengshuang Zhao et al. “Point transformer”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 16259–16268.

[23] Lifa Zhu et al. “Point cloud registration using representative overlapping points”. In: *arXiv preprint arXiv:2107.02583* (2021).