3D Geometric Shape Assembly via Efficient Point Cloud Matching

Nahyuk Lee [*1] **Juhong Min** [*2] **Junha Lee** [1] **Seungwook Kim** [2] **Kanghee Lee** [3] **Jaesik Park** [3 4] **Minsu Cho** [1 2]

Abstract

Learning to assemble geometric shapes into a larger target structure is a pivotal task in various practical applications. In this work, we tackle this problem by establishing local correspondences between point clouds of part shapes in both coarse- and fine-levels. To this end, we introduce Proxy Match Transform (PMT), an approximate high-order feature transform layer that enables reliable matching between mating surfaces of parts while incurring low costs in memory and compute. Building upon PMT, we introduce a new framework, dubbed Proxy Match TransformeR (PMTR), for the geometric assembly task. We evaluate the proposed PMTR on the large-scale 3D geometric shape assembly benchmark dataset of Breaking Bad and demonstrate its superior performance and efficiency compared to state-of-the-art methods. Project page: https://nahyuklee.github.io/pmtr.

1. Introduction

Shape assembly aims to determine a precise placement for each constituent part and construct a larger target shape as a whole. This task holds paramount significance, especially in the context of various applications encompassing robotics (Wang & Hauser, 2019; Zakka et al., 2020; Zeng et al., 2021), manufacturing (Tian et al., 2022), computer graphics (Li et al., 2012), and computer-aided design (Chen et al., 2015; Jacobson, 2017). Despite its pivotal role in industrial productivity and the plethora of applications, the field of shape assembly remains relatively underexplored in the literature due to the intricate challenge it presents: demands for comprehensive understanding of geometric structures and analyses of pairwise relationships between local surfaces of given input parts for accurate assembly.

There have been several recent attempts (Schor et al., 2019; Li et al., 2020a; Wu et al., 2020; Li et al., 2020b; Huang et al., 2020; Narayan et al., 2022; Chen et al., 2022; Wu et al., 2023b) to address the task of shape assembly, but these methods fall short of achieving accurate assembly. They typically represent each part as a global embedding and perform regression to predict a placement for each part. The global encoding strategy for each part, while simplifying the process, greatly limits local information by collapsing spatial resolutions, which is necessary to localize mating surface. Indeed, accurate shape assembly requires a detailed analysis of both fine- and coarse-level spatial information of the parts in recognizing mating surfaces and establishing correspondences between the surfaces. Therefore, a promising approach would be to retain the spatially-rich part representations during the encoding phase and analyze pairwise local correspondence relationships between them for reliable localization and matching of mating surfaces.

In the realm of correspondence analysis within image matching, prior methods (Rocco et al., 2018; Min & Cho, 2021; Kim et al., 2022; Min et al., 2021; Rocco et al., 2020) typically utilize a high-order feature transform, *i.e.*, high-dimensional convolution or attention, to achieve objectives of localizing relevant instances and establishing correspondences between them. The high-order feature transforms, which assess structural patterns of correlations in high-dimensional spaces, have been empirically validated for their efficacy in identifying accurate visual matches. However, the quadratic complexity with respect to input spatial resolution still remains as a significant drawback, *limiting their application to only low-resolution (coarse-grained) inputs*. Such a limitation becomes particularly problematic in the context of geometric assembly since meticulous alignment between parts *requires to analyze high-resolution (fine-grained)* to precisely identify 'geometric compatibility' between mating surfaces to match.

In this paper, we address this issue by introducing a new form of low-complexity high-order feature transform layer, dubbed *Proxy Match Transform (PMT)*, to tackle the challenges of geometric shape assembly. The layer is designed

[*]Equal contribution [1]Department of Computer Science and Engineering, POSTECH, Pohang, Korea [2]Graduate School of Artificial Intelligence, POSTECH, Pohang, Korea [3]Department of Computer Science and Engineering, Seoul National University, Seoul, Korea [4]Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, Korea. Correspondence to: Minsu Cho <mscho@postech.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

to align analogous local embeddings in feature space, *e.g.*, points on mating surfaces, with sub-quadratic complexity, thus offering low-complexity yet high-order approach as illustrated in Fig. 1. We theoretically prove that the proposed PMT layer can effectively approximate the conventional high-order feature transforms (Rocco et al., 2018; Choy et al., 2020; Min & Cho, 2021) under particular conditions. To demonstrate its efficacy, we incorporate the PMT layer into a coarse-to-fine matching framework Proxy Match TransformeR (PMTR), which uses PMTs for both coarse- and fine-level matching steps for establishing reliable correspondences on mating surfaces. We compare our results with recent state of the arts and provide thorough performance analysis on the standard geometric shape assembly benchmark dataset of Breaking Bad (Sellán et al., 2022). The experiments demonstrate that our method outperforms existing approaches by a significant margin while being computationally efficient compared to the baselines.

Our main contributions can be summarized as follows:

- We introduce Proxy Match Transform (PMT), a low-complexity high-order feature transform layer that effectively refines the matching of the feature pair.
- Our theoretical analysis shows that PMT effectively approximates high-order feature transform while incurring sub-quadratic memory and time complexity.
- The performance improvements in geometric shape assembly over the state-of-the-art baselines demonstrate the effectiveness and efficiency of our approach.

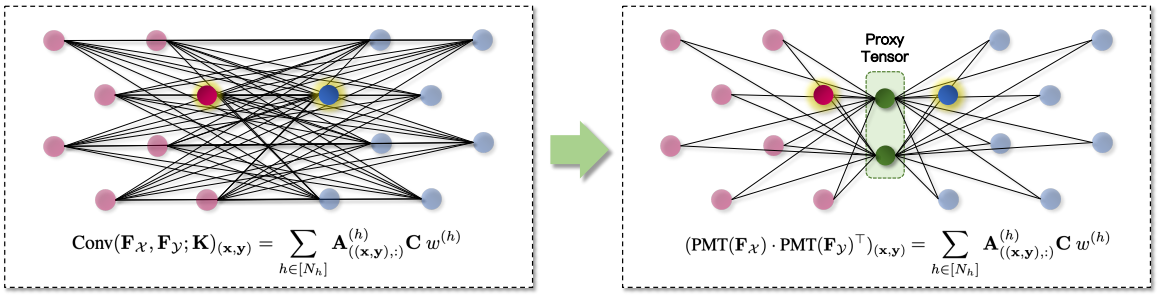
2. Related Work

3D shape assembly & registration. Previous research in generative models for 3D objects has primarily focused on building objects through the combination of basic 3D primitives. One prevalent approach trains specialized models tailored to individual object classes, enabling the assembly of objects from volumetric primitives such as cuboids (Tulsiani et al., 2017). Conversely, Khan et al. (2019) proposes a unified model that can generate cuboid primitives across various classes. Additionally, variational autoencoders (VAEs) have been employed to model objects as compositions of cuboids, offering robust abstractions that distill local geometric details and elucidate object correspondences (Kingma & Welling, 2014; Jones et al., 2020).

Parallel to these developments, research in part assembly has aimed to construct complete objects from predefined semantic parts. The method of Li et al. (2020b) predicts translations and rotations for part point clouds to assemble a target object from an image reference. Extending this, Narayan et al. (2022); Huang et al. (2020) have conceptualized part assembly as a graph learning challenge, utilizing iterative message passing techniques to integrate parts into cohesive objects. These approaches heavily rely on the PartNet dataset (Mo et al., 2019) to ensure semantic correspondence between the assembled parts and the target models, demonstrating that while geometric shapes are foundational, semantic cues can significantly guide and streamline the assembly process. Our research diverges from these methods by focusing on the assembly of parts without predefined semantics. A closely related methodology is that of Chen et al. (2022), which also tackles the problem of 3D shape assembly by integrating implicit shape reconstruction, providing a relevant benchmark.

Additionally, the concept of 3D shape assembly overlaps with the domain of 3D registration, especially in scenarios characterized by low overlap between a pair of point clouds. Techniques such as those proposed by Huang et al. (2021) and Yu et al. (2021) leverage self-attention and cross-attention mechanisms within and across point cloud features to transform 3D features, facilitating enhanced matching accuracy. Qin et al. (2022) further advances this by embedding transformation-invariant data into the positional embeddings of transformer layers, optimizing the matching process in low-overlap conditions. Despite their efficacy, the practical application of these methods in fine-grained matching scenarios is often constrained by the *quadratic complexity* associated with their matching layers, highlighting a critical area for improvement in computational efficiency and scalability. Our work addresses these challenges by proposing a novel approach that optimizes the computational demands of feature matching while maintaining high robustness.

High-order feature transform for matching. High-order feature transforms are essential in (both image and point cloud) matching tasks, helping to establish consensus among correspondences within a high-dimensional space. Initially introduced by Rocco et al. (2018), the concept of a learning-based neighborhood consensus supports the identification of accurate matches by leveraging neighboring ambiguous matches between 2D images. This approach has also been adapted for 3D registration tasks, notably by Choy et al. (2020), who utilized a 6D sparse convolutional layer to filter out outlier correspondences. Given high computational complexity associated with high-order feature transforms, several studies have proposed methods to reduce this burden. Techniques such as decomposing high-dimensional convolutional kernels (Min et al., 2021) and sparsifying the correlation map with top-k scores (Rocco et al., 2020) have been effective. Further, Shi et al. (2023) enhanced matching efficiency by creating a sparse correlation matrix through the clustering of input tokens, significantly reducing the number of tokens involved. More recent advancements have integrated the self-attention mechanism to utilize global feature consensus effectively, although these methodologies, proposed by Cho et al. (2021) and Kim et al. (2022), come at a higher computational cost.



High–order feature transform with quadratic complexity

Product of PMTs with sub–quadratic complexity

Figure 1. Given a correlation score at position $(\mathbf{x}, \mathbf{y})$ (the edge between highlighted nodes) and its neighboring scores (all other edges), vanilla high-order feature transform (shown on the left) leads to quadratic complexity due to its demand for memory-intensive pairwise correlation scores. The product of two PMTs (shown on the right) effectively approximates this high-order transform only with sub-quadratic complexity by avoiding direct construction of correlation scores, instead exchanging information through a low-dimensional proxy tensor. The red/blue nodes and black edges represent the source/target features and the correlation scores between them, respectively.

Our work introduces the Proxy Match Transform (PMT), which simplifies existing high-order feature transforms to significantly reduce computational demands. We apply PMT in a coarse-to-fine approach, identifying reliable correspondences between the mating surfaces of input parts and subsequently refining them for precise assembly. There have been several approaches relevant to ours such as leveraging local geometric cues for assembly by Lu et al. (2023), the linear approximations in convolutional networks by Denton et al. (2014), sparse attention mechanisms by Zaheer et al. (2021), low-rank approximations of self-attention by Chen et al. (2020), and Gaussian kernel approximations by Chen et al. (2021). Unlike these methods, however, which primarily enhance processing within a *single feature*, PMT uniquely addresses the challenge of efficient matching between *two distinct features*, improving both computational efficiency and the feature correspondence analysis, which are essential for diverse applications like geometric shape assembly.

3. Proposed Approach

In the task of geometric shape assembly, analyzing geometric compatibility between fractured shapes is of utmost importance; the geometric properties of the *mating surfaces* should exhibit consistency, where vertices, edges, and surfaces seamlessly fit together to form a coherent structure. To achieve reliable localization of mating surfaces between shapes, a model needs to analyze the compatibility of all possible feature correspondences and accurately identify spatially consistent matches. In the field of visual matching and its applications (Rocco et al., 2018; Choy et al., 2020; Min & Cho, 2021; Cho et al., 2021; Min et al., 2021), a trending approach for assessing match reliability is the utilization of ***high-order feature transform***, *e.g.*, convolution or self-attention. This technique effectively assesses patterns within neighborhood matches in a differentiable manner. Building upon these principles, we will now explore the theoretical formulation of high-order transform, with a specific emphasis on its application for enhancing pairwise feature correlation.

Preliminary. High-order convolution (Rocco et al., 2018; Choy et al., 2020; Min & Cho, 2021) generalizes the standard convolution by taking as input more functions, feature maps, or sets. In the context of our problem, we consider two point clouds $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^3\}_{i=1}^N$ and $\mathcal{Y} = \{\mathbf{y}_i \in \mathbb{R}^3\}_{i=1}^M$, and focus on the 2nd-order convolution with two sets of features $\mathcal{F}_\mathcal{X}$ and $\mathcal{F}_\mathcal{Y}$, associated with the two point clouds, respectively. For ease of notation, we represent these features in matrix form, *i.e.*, $\mathbf{F}_\mathcal{X} \in \mathbb{R}^{|\mathcal{X}|\times D_{\text{emb}}}$, where D_{emb} is the feature embedding dimension, and indexes each feature embedding using its associated point $\mathbf{x} \in \mathcal{X}$ such that $(\mathbf{F}_\mathcal{X})_\mathbf{x} \in \mathbb{R}^{D_{\text{emb}}}$, and same goes for $\mathcal{F}_\mathcal{Y}$. We also express the feature correlation of two points from each point cloud, $\mathbf{x}$ and $\mathbf{y}$, as $\mathbf{C}_{(\mathbf{x},\mathbf{y})} \coloneqq (\mathbf{F}_\mathcal{X})_\mathbf{x} \cdot (\mathbf{F}_\mathcal{Y})_\mathbf{y}^\top \in \mathbb{R}$. The 2nd-order convolution on $(\mathbf{F}_\mathcal{X}, \mathbf{F}_\mathcal{Y})$ with kernel K is then defined as:

$$\text{Conv}(\mathbf{F}_\mathcal{X}, \mathbf{F}_\mathcal{Y})_{(\mathbf{x},\mathbf{y})} \coloneqq \sum_{(\mathbf{n},\mathbf{m})\in\mathcal{N}(\mathbf{x})\times\mathcal{N}(\mathbf{y})} \mathbf{C}_{(\mathbf{n},\mathbf{m})} K([\mathbf{n}-\mathbf{x}, \mathbf{m}-\mathbf{y}]), \quad (1)$$

where $\mathcal{N}(\cdot)$ represents a set of neighbor points and $K : \mathbb{R}^6 \to \mathbb{R}$ is a convolutional kernel, represented as a mapping function that takes a displacement vector onto learnable weight scalar.

Building upon insights from the work of Cordonnier et al. (2020), we consider Lemma 1 which states that the conv layer in Eq. 1 can be re-formulated as a form of multi-head self-attention under sufficient conditions:

Lemma 1. *Consider a bijective mapping of natural numbers, i.e., heads, onto 6-dimensional local displacements:* $t(h) : [N_h] \to \Delta(\mathbf{x}, \mathbf{y})$. *Let* $\mathbf{A}^{(h)} \in \mathbb{R}^{|\mathcal{X}||\mathcal{Y}|\times|\mathcal{X}||\mathcal{Y}|}$ *be an*

attention matrix that holds the following:

$$\mathbf{A}^{(h)}_{(\mathbf{x},\mathbf{y}),(\mathbf{n},\mathbf{m})} = \begin{cases} 1, & \text{if } t(h) = (\mathbf{n},\mathbf{m}) - (\mathbf{x},\mathbf{y}) \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Then, for any high-dimensional convolution with a kernel $K : \mathbb{R}^6 \to \mathbb{R}$*, there exists* $\{w^{(h)} \in \mathbb{R}\}_{h\in[N_h]}$ *such that following equality holds:*

$$\text{Conv}(\mathbf{F}_\mathcal{X}, \mathbf{F}_\mathcal{Y})_{(\mathbf{x},\mathbf{y})} = \sum_{h\in[N_h]} \mathbf{A}^{(h)}_{((\mathbf{x},\mathbf{y}),:)} \mathbf{C}\, w^{(h)}. \tag{3}$$

Proof. We refer to the Appendix A for the complete proof.

As illustrated in the left of Fig. 1, the 2nd-order convolution (Eq. 1 and 3) is designed to disambiguate spatially consistent correspondences and update their correlation values by analyzing local correlation patterns around each point pair $(\mathbf{x},\mathbf{y}) \in \mathcal{X}\times\mathcal{Y}$. Despite its good empirical performance in literature (Rocco et al., 2018; Choy et al., 2020; Min & Cho, 2021; Min et al., 2021), its critical limitation lies in the quadratic complexity of correlation computation, *i.e.*, $\mathcal{O}(|\mathcal{X}|\cdot|\mathcal{Y}|)$, with respect to the input resolution, imposing significant computational burdens during both the training and inference phases. This restricts its practical applications with large spatial resolution inputs, such as the geometric shape assembly task that demands high-resolution, *i.e.*, geometric-level, input processing for geometric compatibility analysis to ensure precise correspondence alignments.

3.1. Proxy Match Transform: an efficient high-order feature transform with sub-quadratic complexity

To overcome the limitation, we introduce an efficient feature matching layer, dubbed *Proxy Match Transform*, which approximates high-order convolution with sub-quadratic complexity. Given a pair of features $(\mathbf{F}_\mathcal{X}, \mathbf{F}_\mathcal{Y})$ as inputs, PMT layers with N_h heads[1] are defined as follows:

$$\text{PMT}(\mathbf{F}_\mathcal{X}) \coloneqq \sum_{h\in[N_h]} \mathbf{A}^{(h)}_\mathcal{X} \mathbf{F}_\mathcal{X} \mathbf{P}^{(h)\top} w^{(h)}_\mathcal{X}, \tag{4}$$

$$\text{PMT}(\mathbf{F}_\mathcal{Y}) \coloneqq \sum_{h\in[N_h]} \mathbf{A}^{(h)}_\mathcal{Y} \mathbf{F}_\mathcal{Y} \mathbf{P}^{(h)\top} w^{(h)}_\mathcal{Y}, \tag{5}$$

where $w^{(h)}_\mathcal{X} \in \mathbb{R}$ is a learnable weight scalar, $\mathbf{A}^{(h)}_\mathcal{X} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ is local attention matrix[2], and $\mathbf{P}^{(h)} \in \mathbb{R}^{D_\text{proxy}\times D_\text{emb}}$ is **proxy tensor** that satisfies the following:

$$\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} = \begin{cases} \mathbf{I}_{D_\text{emb}}, & \text{if } i = j \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{6}$$

where D_proxy refers to the spatial resolution of the proxy tensor satifying $D_\text{proxy} \ll |\mathcal{X}|, |\mathcal{Y}|$. The constraint ensures orthogonality between different proxy tensors. The rationale behind this design is discussed in Sec. 3.2.

At each head, the layer initially constructs a correlation between the input feature $\mathbf{F}_\mathcal{X}$ and the proxy tensor $\mathbf{P}^{(h)}$ such that $\mathbf{C}^{(h)}_\mathcal{X} \coloneqq \mathbf{F}_\mathcal{X}\mathbf{P}^{(h)\top}$ in much smaller size of $|\mathcal{X}| \times D_\text{proxy}$, compared to the pairwise feature correlation $\mathbf{C} = \mathbf{F}_\mathcal{X}\mathbf{F}_\mathcal{Y}^\top \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{Y}|}$ as defined in Eq. 1. After applying learnable weight $w^{(h)}_\mathcal{X}$, the output at position $(\mathbf{n},\mathbf{m}) \in |\mathcal{X}| \times D_\text{proxy}$ is computed through a weighted-sum of its neighborhood matches lying on the spatial dimension of feature map $\mathbf{F}_\mathcal{X}$, *e.g.*, $\{(\mathbf{n}',\mathbf{m})\}_{\mathbf{n}'\in\mathcal{N}(\mathbf{n})}$ where $|\mathcal{N}(\mathbf{n})| = \epsilon \ll |\mathcal{X}|$. To formally put, the Proxy Match Transform output at head h given input $\mathbf{F}_\mathcal{X}$ at position $(\mathbf{n},\mathbf{m})$ is defined as

$$\begin{aligned} \text{PMT}(\mathbf{F}_\mathcal{X})^{(h)}_{(\mathbf{n},\mathbf{m})} &= \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{n},:)} \mathbf{F}_\mathcal{X} \mathbf{P}^{(h)\top}_{\ (:,\mathbf{m})} w^{(h)}_\mathcal{X} \\ &= \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{n},:)} \mathbf{C}^{(h)}_{\mathcal{X}\ (:,\mathbf{m})} w^{(h)}_\mathcal{X} \\ &= \sum_{\mathbf{n}'\in\mathcal{N}(\mathbf{n})} \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{n},\mathbf{n}')} \mathbf{C}^{(h)}_{\mathcal{X}\ (\mathbf{n}',\mathbf{m})} w^{(h)}_\mathcal{X}. \end{aligned} \tag{7}$$

$\text{PMT}(\mathbf{F}_\mathcal{Y})^{(h)}$ is similarly defined with a different set of parameters of $\mathbf{A}^{(h)}_\mathcal{Y}$ and $w^{(h)}_\mathcal{Y}$.

It is important to note that the PMT layers perform two ***independent*** transforms for ***feature matching***, one for $\mathbf{F}_\mathcal{X}$ and the other for $\mathbf{F}_\mathcal{Y}$. Despite the independence, matching between the feature pair is effectively facilitated by a shared proxy tensor $\mathbf{P}$. This proxy tensor allows for the exchange of information between the features, eliminating the need to construct and convolve memory-intensive pairwise feature correlations, which often contain sparse and limited informative match scores. We demonstrate that how the PMT effectively approximates existing high-order convolution in Sec. 3.2 and empirically prove the efficacy of the use of proxy tensor and different parameter sets in geometric shape assembly in Sec. 4.4.

3.2. Constraints for Proxy Match Transform

In order for the Proxy Match Transforms to express the high-order convolution, we assume the following constraints, (i) orthonormality constraint: $\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} = \mathbf{I}_{D_\text{emb}}$ if $i = j$, and (ii) zero-matrix constraint: $\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} = \mathbf{0} \in \mathbb{R}^{D_\text{emb}\times D_\text{emb}}$ otherwise for all $i,j \in [N_h]$. Under such conditions, a dot product between two Proxy Match Transforms can effectively approximate high-order convolution. Our main theoretical result is provided below.

Theorem 1. *If we assume* $\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} = \mathbf{I}_{D_{emb}}$ *if* $i = j$ *and* $\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} = \mathbf{0}$ *otherwise for all* $i,j \in [N_h]$*, and define* $\mathbf{A}^{(h)}_{(\mathbf{x},\mathbf{y}),(\mathbf{n},\mathbf{m})} \coloneqq \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{x},\mathbf{n})} \cdot \mathbf{A}^{(h)}_{\mathcal{Y}\ (\mathbf{y},\mathbf{m})}$ *and* $w^{(h)} \coloneqq$

[1] Similar to multi-head self-attention (Vaswani et al., 2017), each head performs distinct attentions and feature transform, allowing the layer to attend different aspects of inputs.

[2] To avoid the quadratic complexity of $|\mathcal{X}|\times|\mathcal{X}|$ in the attention matrices, we adopt an implementation strategy similar to that described in Thomas et al. (2019). We refer to Sec. 4.2 for details.

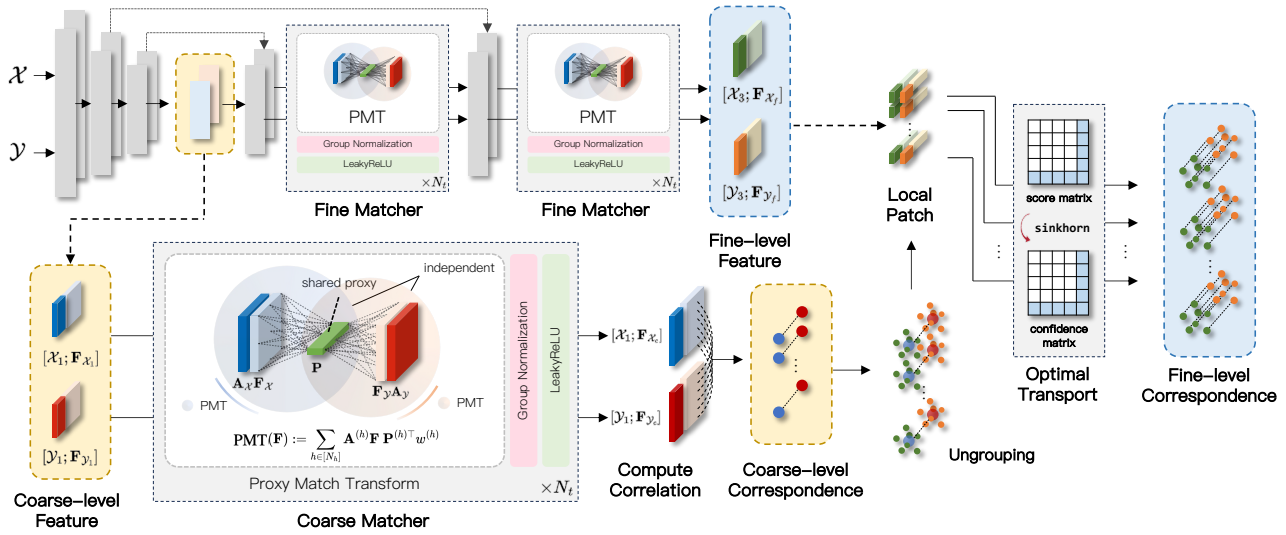


Figure 2. Overall pipeline of the Proxy Match TransformeR (PMTR) for pairwise shape assembly. The proposed architecture consists of coarse-level matching and fine-level matching. Each part of matching uses coarse-level features and fine-level features, respectively, acquired from the KPConv-FPN backbone as their input. Each matcher consists of N_t PMT layers in series. See Sec. 3.3 for details.

$w_{\mathcal{X}}^{(h)} w_{\mathcal{Y}}^{(h)}$, *then, the dot-product of Proxy Match Transform outputs with a sufficient number of heads* N_h *can express high-dimensional convolutional layer with kernel* $K : \mathbb{R}^6 \to \mathbb{R}$: $\mathit{PMT}(\mathbf{F}_{\mathcal{X}}) \cdot \mathit{PMT}(\mathbf{F}_{\mathcal{Y}})^\top = \mathit{Conv}(\mathbf{F}_{\mathcal{X}}, \mathbf{F}_{\mathcal{Y}})$.

We refer to the Appendix A for the complete proof of the theorem. For the proxy tensors to satisfy the conditions, we design two auxiliary training objectives on proxy tensors, orthonormal loss $\mathcal{L}_{\text{orth}}$ and zero loss $\mathcal{L}_{\text{zero}}$, as follows:

$$\mathcal{L}_{\text{orth}} = \sum_{(i,j)\in[N_h]^2} \delta(i,j)(\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} - \mathbf{I}_{D_{\text{emb}}}), \quad (8)$$

$$\mathcal{L}_{\text{zero}} = \sum_{(i,j)\in[N_h]^2} (1-\delta(i,j))\mathbf{P}^{(i)\top}\mathbf{P}^{(j)}, \quad (9)$$

where $\delta(i,j)$ provides 1 if $i=j$ and 0 otherwise.

3.3. Overall architecture

The proposed architecture, dubbed Proxy Match TransformeR (PMTR) comprises four main parts: (1) feature extraction, (2) coarse-level matching, (3) fine-level matching, and (4) transformation prediction & training objectives. As illustrated in Fig. 2, our pipeline begins with the point cloud pair embedding. The feature extraction network generates three pairs of features, each at distinct spatial resolutions. These feature pairs are subsequently fed to a corresponding PMT layer, which facilitates both coarse-level matching (for mating surface localization) and fine-level matching (for geometric matching). The outputs from the coarse matching phase are utilized to establish a preliminary correspondence between the mating surfaces of the input parts, which is crucial for identifying potential areas of alignment. Subsequently, the fine matching phase are designed to refine these correspondences, focusing exclusively on reliable matches identified during the coarse matching stage. This allows for precise correspondence establishment, ensuring accurate assembly as demonstrated by our experiments in Sec. 4.4.

Feature extraction. A pair of point cloud to match is fed to a feature embedding network, reducing their spatial resolution to provide coarse-level feature pair. Each of two subsequent upsampling layers connected in series provides features in more high-resolution. From this U-Net shaped architecture, similarly to KPConv-FPN (Thomas et al., 2019), the model gives three pairs of point cloud features with different spatial resolutions: $\{(\mathbf{F}_{\mathcal{X}_n}, \mathbf{F}_{\mathcal{Y}_n})\}_{n=1}^3$ where $\mathbf{F}_{\mathcal{X}_n} \in \mathbb{R}^{|\mathcal{X}_n| \times D_{n\text{-emb}}}$ with $|\mathcal{X}_1| < |\mathcal{X}_2| < |\mathcal{X}_3|$, implying $\mathbf{F}_{\mathcal{X}_1}$ is the coarse feature with the smallest number of features. $\{\mathcal{Y}_i\}_{n=1}^3$ is similarly defined. The coarse feature pair $\{(\mathbf{F}_{\mathcal{X}_1}, \mathbf{F}_{\mathcal{Y}_1})\}$ is utilized for identifying potential mating surfaces to match while the others $\{(\mathbf{F}_{\mathcal{X}_n}, \mathbf{F}_{\mathcal{Y}_n})\}_{n=2}^3$ are used precise geometric alignment between identified potential surface matches.

Coarse-level matching. At this stage, PMT processes the coarse feature pair $\{(\mathbf{F}_{\mathcal{X}_1}, \mathbf{F}_{\mathcal{Y}_1})\}$, aiming to evaluate potential local correspondence between the feature set. This is achieved without directly computing the pairwise correlation matrix $\mathbf{F}_{\mathcal{X}_1} \cdot \mathbf{F}_{\mathcal{Y}_1}{}^\top$, which would otherwise result in a quadratic dimensionality of $\mathbb{R}^{|\mathcal{X}_1| \times |\mathcal{Y}_1|}$. Instead, a pair of PMTs operates in a manner that allows them to be refined independently to provide two refined coarse-level features $(\mathbf{F}_{\mathcal{X}_c}, \mathbf{F}_{\mathcal{Y}_c})$ as follows:

$$\text{PMT}(\mathbf{F}_{\mathcal{X}_1}) = \mathbf{F}_{\mathcal{X}_c}, \quad \text{PMT}(\mathbf{F}_{\mathcal{Y}_1}) = \mathbf{F}_{\mathcal{Y}_c}. \quad (10)$$

Despite this independence, the transformations ensure that the dot product of the refined features closely approximates

the output of a high-order feature transformation. The approximation is conceptualized as if the features had undergone a high-order transform, according to Theorem 1:

$$\text{PMT}(\mathbf{F}_{\mathcal{X}_1}) \cdot \text{PMT}(\mathbf{F}_{\mathcal{Y}_1})^\top = \mathbf{F}_{\mathcal{X}_c} \cdot \mathbf{F}_{\mathcal{Y}_c}{}^\top \quad (11)$$

$$\approx \text{Conv}(\mathbf{F}_{\mathcal{X}_1}, \mathbf{F}_{\mathcal{Y}_1}). \quad (12)$$

This approach allows for the independent refinement of the features while still capturing the essence of their interaction, akin to high-order convolution, *without the direct computation of their pairwise correlation*, thereby effectively avoiding the burden of quadratic complexity.

Fine-level matching. In fine-level matching, we leverage the high-resolution feature pairs $\{(\mathbf{F}_{\mathcal{X}_n}, \mathbf{F}_{\mathcal{Y}_n})\}_{n=2}^{3}$ to achieve more precise alignment. This stage mirrors the coarse-level matching in its use of PMT layers for transforming features but in a serial configuration. This setup ensures that the output of one PMT layer serves as the input to the next, such that $\text{PMT}(\mathbf{F}_{\mathcal{X}_2}) = \mathbf{F}_{\mathcal{X}_3}$ and subsequently, $\text{PMT}(\mathbf{F}_{\mathcal{X}_3}) = \mathbf{F}_{\mathcal{X}_f}$, with an analogous sequence for providing fine-level feature $\mathbf{F}_{\mathcal{Y}_f}$. Note that PMT effectively addresses the infeasibility of employing vanilla high-order convolution for high-resolution matching, especially under conditions where $|\mathcal{X}|, |\mathcal{Y}| > 1500$, as observed in our experiments. Compared to vanilla high-order convolution with complexity of $\mathcal{O}(|\mathcal{X}| \cdot |\mathcal{Y}|)$, rendering it infeasible for large-scale applications, the proposed PMT having $\mathcal{O}(\max(|\mathcal{X}|, |\mathcal{Y}|) \cdot D_{\text{proxy}})$ complexity where $D_{\text{proxy}} \ll |\mathcal{X}|, |\mathcal{Y}|$ provides a more efficient means of analyzing feature correlations. In Sec. 4.4, we present an apples-to-apples comparisons, illustrating the practical advantages of PMT over traditional matching methods, *e.g.*, high-order convolution (Rocco et al., 2018).

Transformation prediction. After the coarse-level matching, the refined feature pair $(\mathbf{F}_{\mathcal{X}_c}, \mathbf{F}_{\mathcal{Y}_c})$ is utilized to compute correlation in size of $|\mathcal{X}_c| \times |\mathcal{Y}_c|$ where each score at position $(\mathbf{x}, \mathbf{y})$ is defined as $\exp(-||(\mathbf{F}_{\mathcal{X}_c})_{\mathbf{x}} - (\mathbf{F}_{\mathcal{Y}_c})_{\mathbf{y}}||_2^2)$ similarly to the work of Qin et al. (2022). From $|\mathcal{X}_c| \times |\mathcal{Y}_c|$ number of scores, we collect top-k matches as reliable coarse matches, laying the foundation for more granular alignment at the subsequent fine-level matching. Building on coarse-level *matches* and fine-level *features*, we employ the point-to-node grouping method (Yu et al., 2021), which clusters fine-level features that are spatially proximate to the coarse matches, effectively sharpening the broad coarse-level correspondence into more precise fine-level ones. In essence, the computation of fine-level matches is directly influenced by the groundwork laid at the coarse level, establishing a hierarchical refinement process. We then incorporate an optimal transport layer (Sarlin et al., 2020) to the fine-level matches to obtain final correspondences for the subsequent transformation prediction. Finally, similarly to Qin et al. (2022), we use the final correspondences to predict the relative transformation $\{\mathbf{R}|\mathbf{t}\}$ between the point cloud pair $(\mathcal{X}, \mathcal{Y})$.

Training objectives. Following the previous 3D matching literatures (Zhao et al., 2023; Wu et al., 2023a; Chen et al., 2023; Yu et al., 2023), we adopt overlap-aware circle loss $\mathcal{L}_{\text{oc}}$ (Qin et al., 2022), and point matching loss $\mathcal{L}_{\text{p}}$ (Sarlin et al., 2020), as our main training objectives for coarse- and fine-level correspondence matching respectively. We direct readers to the work of Qin et al. (2022) for further details of $\mathcal{L}_{\text{oc}}$ and $\mathcal{L}_{\text{p}}$. With two auxiliary losses in Eq. 9, our main training objective is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{oc}} + \mathcal{L}_{\text{p}} + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}} + \lambda_{\text{zero}}\mathcal{L}_{\text{zero}}, \quad (13)$$

where λ_{orth} and λ_{zero} are hyperparameters which are set to 1.0 in our experiments.

4. Experiments

In this section, we discuss the dataset and evaluation metrics used (Sec. 4.1), implementation details (Sec. 4.2), the results of pairwise shape assembly with comprehensive analysis (Sec. 4.3), an in-depth ablation study to inspect the efficacy of the proposed techniques (Sec. 4.4), and extension of our evaluation to the task of multi-part assembly (Sec. 4.5).

4.1. Dataset and Evaluation Metrics

Dataset. In our experiments, we utilize the Breaking Bad dataset (Sellán et al., 2022) which is a large-scale dataset of fractured objects for the task of geometric shape assembly, which consists of over 1 million fractured objects simulated from 10K meshes of PartNet (Mo et al., 2019) and Thingi10k (Zhou & Jacobson, 2016). For *pairwise* assembly training and evaluation, we exclusively select a subset of the Breaking Bad dataset containing two-part objects (Sec. 4.3). For *multi-part* assembly, we expand our evaluation to include all samples in the dataset, encompassing objects with 2 to 20 parts (Sec. 4.5).

Evaluation metrics. Following the evaluation protocol of Sellán et al. (2022), we measure the root mean square error (RMSE) between the ground-truth and predicted rotation (R) and translation (T) parameters, and the Chamfer distance (CD) between the assembly results and ground-truth. In addition, we introduce and report a new metric, called CoRrespondence Distance (CRD), which is defined as Frobenius norm between the input pair of the assembled point cloud; unlike CD, CRD offers a more comprehensive measure of correspondence, capturing both proximity and structural alignment between the assembled objects. We compute the evaluation metrics of RMSE (R) and RMSE (T) based on *relative transformation*, *e.g.*, rotation and translation, between the input fracture pair, instead of absolute pose as in previous literature (Chen et al., 2022; Wu et al., 2023b) by setting the largest fracture as an anchor and compute

Table 1. Pairwise shape assembly results. Numbers in bold indicate the best performance and underlined ones are the second best.

Method	Estimator Type	Target $\{\mathbf{R}\|\mathbf{t}\}$	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})
			everyday				artifact			
Global (2019; 2020a)	MLP	absolute pose	27.77	15.26	110.74	30.61	19.26	7.16	86.30	21.02
LSTM (2020)			20.04	7.77	84.60	22.07	19.52	6.45	84.42	21.33
DGL (2020)			20.32	6.40	86.23	22.38	19.82	6.19	85.46	21.65
NSM (2022)			21.71	11.09	83.38	23.71	19.44	6.33	83.22	21.41
Wu et al. (2023b)			20.65	11.66	84.58	22.90	19.17	7.97	85.04	20.90
GeoTransformer (2022)	correspondence alignment	relative transformation	<u>0.61</u>	<u>0.51</u>	<u>22.81</u>	7.28	<u>0.89</u>	<u>0.70</u>	<u>33.23</u>	10.30
Jigsaw (2023)			5.48	1.34	38.73	**2.73**	6.36	1.45	39.71	**3.02**
PMTR (Ours)			**0.39**	**0.25**	**17.14**	<u>5.53</u>	**0.60**	**0.42**	**23.28**	<u>7.27</u>

the relative transformation. The formal definitions of the evaluation metrics can be found in Appendix D.

4.2. Implementation details

We implement our PMTR using PyTorch Lightning (Falcon & team, 2019). Experiments were conducted on a machine with Intel(R) Xeon(R) Gold 6342 CPU @ 2.80GHz and NVIDIA GeForce RTX 3090 GPU. For all experiments, except the ones include GeoTransformer, we use ADAM (Kingma & Ba, 2015) optimizer with a learning rate of 1×10^{-3} for 150 epochs. For GeoTransformer, we use the identical settings but only reduce the learning rate to 1×10^{-4} to prevent model divergence. To ensure uniform point density among fractures, we uniform-sample approximately 5,000 points on the surface of holistic objects and allocate the number of sample points for each fracture proportional to the surface area of each fracture. Each of both coarse-level and fine-level matchers consists of 2 PMT($\cdot$) layers ($N_t = 2$) with nonlinearity and group norm (Wu & He, 2018). See Appendix C for further details.

Avoiding quadratic complexity of attention in PMT. In our actual implementation, we use local, *i.e.*, sparse, attention for $\mathbf{A}_{\mathcal{X}}^{(h)}$ by collecting attention scores of 'neighborhood' of each position, thus reducing attention size to $|\mathcal{X}| \times \epsilon$ instead of $|\mathcal{X}| \times |\mathcal{X}|$ where $\epsilon \in \mathbb{N}^+$ is the number of neighbors: $\epsilon \ll |\mathcal{X}|$. Specifically, attention at position $\mathbf{x} \in \mathbb{R}^3$ denoted as $\mathbf{A}_{\mathcal{X}(\mathbf{x},:)}^{(h)} \in \mathbb{R}^{1\times\epsilon}$ are limited to the neighborhood of $\mathbf{x}$, represented by $\mathcal{N}(\mathbf{x})$. Then, the output of PMT at $\mathbf{x}$ is formulated as $\text{PMT}(\mathbf{F}_{\mathcal{X}})_{(\mathbf{x},:)} = \sum_{h\in[N_h]} \mathbf{A}_{\mathcal{X}(\mathbf{x},:)}^{(h)} \mathbf{F}_{\mathcal{X}(\mathcal{N}(\mathbf{x}),:)} \mathbf{P}^{(h)\top} w_{\mathcal{X}}^{(h)}$ where $\mathbf{F}_{\mathcal{X}(\mathcal{N}(\mathbf{x}),:)} \in \mathbb{R}^{\epsilon\times D_{\text{emb}}}$ is neighborhood features of position $\mathbf{x}$. This method significantly reduces the computational complexity typically associated with full pairwise attention, which would otherwise be quadratic, *i.e.*, $|\mathcal{X}| \times |\mathcal{X}|$. This reduction in complexity mirrors strategies found in existing literature, such as those described by Thomas et al. (2019). For simplicity in presentation, however, this paper narrates with a conventional square attention matrix $\mathbf{A}_{\mathcal{X}}^{(h)} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ to illustrate our methodology. A similar approach applies to the other matrix $\mathbf{A}_{\mathcal{Y}}^{(h)} \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$.

Assessment with relative transformations. Note that the previous methods for pairwise geometric assembly (Li et al., 2020a; Wu et al., 2020; Huang et al., 2020; Chen et al., 2022) predict *two different transformation parameters* for the input pair of parts to assemble them in 3D space. However, this approach has a limitation in accurate evaluation: even if a model perfectly assembles the pair of parts, the assessment may be inaccurate if the assembled object does not match the specific *absolute* pose of the ground truth. To address this issue, we suggest to predict the *relative* transformation between input parts, allowing us to focus solely on the assembly rather than the predefined absolute poses.

4.3. Pairwise Shape Assembly

To evaluate our method, we categorize previous baseline methods into two groups based on their approach to transformation parameters $\{\mathbf{R}|\mathbf{t}\}$ prediction. The first group includes 'regression methods' that encode each part into a global embedding and directly regress their absolute transformations using MLP: Global (Li et al., 2020a), LSTM (Wu et al., 2020), DGL (Huang et al., 2020), NSM (Chen et al., 2022), and Wu et al. (2023b). The second group consists of 'matching-based methods' that estimate relative transformations by aligning their predicted correspondences between each pair of parts: GeoTransformer (Qin et al., 2022) and Jigsaw (Lu et al., 2023).

Experimental results and analysis. We evaluate our method and compare it against baseline methods on the `everyday` and `artifact` subsets of the Breaking Bad dataset. Tab. 1 presents the results, demonstrating that our method consistently outperforms all baseline methods on both subsets. In Fig. 4, we provide qualitative comparisons between ours and the baselines, using mesh representation for better visualization.

To provide deeper insights to the learned shared proxy $\mathbf{P}^{(h)}$, we visualize how the proxy and the refined coarse-level features ($\mathbf{F}_{\mathcal{X}_c}$ and $\mathbf{F}_{\mathcal{Y}_c}$) are distributed in the feature space via t-SNE. As shown in Fig. 3, The visualization reveals that

Table 2. **Ablation study on the proxy sharing.** By sharing proxy tensor in each Proxy Match Transform layer, two independent feature transforms share information, yielding the highest score.

Ref.	proxy	shared proxy	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})
(a)	✗	✗	0.53	0.47	21.04	6.93
(b)	✓	✗	0.44	0.31	18.66	5.97
Ours	✓	✓	**0.39**	**0.25**	**17.14**	**5.53**

Table 3. **Ablation study on the contribution of $\mathcal{L}_{\text{orth}}$ and $\mathcal{L}_{\text{zero}}$.** They constrains Proxy Match Transform in approximating the high-dimensional convolution layers, yielding the highest score.

Ref.	$\mathcal{L}_{\text{orth}}$	$\mathcal{L}_{\text{zero}}$	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})
(a)	✗	✗	0.43	0.31	18.82	6.23
(b)	✓	✗	0.43	0.32	17.87	5.73
(c)	✗	✓	0.43	0.27	18.77	6.02
Ours	✓	✓	**0.39**	**0.25**	**17.14**	**5.53**

subsets of the source and target features (orange and lightblue) and the proxy (purple) form distinct clusters (Fig. 3 (a)) with closer proximity, implying higher correlation with the proxy. In Fig. 3. (b), we visually mark those points in 3D space. Notably, the points with the highest correlation (red and blue) with the proxy are predominantly located on the "mating surfaces" of the parts, revealing that the proxy effectively facilitates the information exchange between given feature pair without the burden of quadratic complexity.

4.4. Ablation studies

Effect of proxy tensor in assembly. To verify the effect of proxy tensor in PMT, we conducted a series of ablation studies on the `everyday` subset of the Breaking Bad dataset. Specifically, we examine the impact of the shared proxy tensor by either removing it or using two different proxies instead of a shared one. The results, summarized in Tab. 2, clearly indicate that both removing the proxy and not sharing it lead to a significant decline in assembly performance. This underscores the efficacy of the shared proxy in facilitating information exchange in PMT.

Next, we explore the impact of $\mathcal{L}_{\text{orth}}$ and $\mathcal{L}_{\text{zero}}$, which serve as the sufficient conditions to constrain the PMT layer to represent the high-dimensional convolutional layers, as detailed in Sec. 3.2. The results are presented in Tab. 3; as evident from the table, the best performance is achieved when both losses are incorporated. This highlights that the significance of these constraining conditions for PMT, as they are crucial in enabling PMT to effectively approximate the high-dimensional convolution.

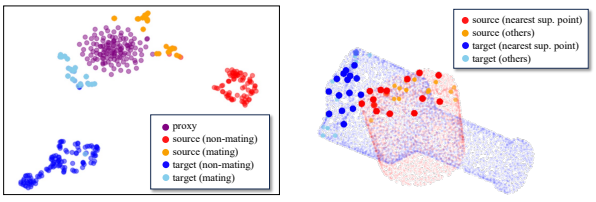
Comparison between different matchers. To demonstrate the efficacy and efficiency of the proposed matching layer, PMT, we conduct ablations by either removing it (None) or replacing it with different layers: a single linear transformation (Linear), multi-layer perceptron (MLP), high-dimensional convolution (HDC by Min et al. (2021)), and GeoTransformer (GeoTr by Qin et al. (2022)). In Tab. 4, we compare ours with other layers at fine-level; Undoubtedly, the layers without any information exchange between source and target features, *e.g.*, None, Linear, and MLP, show dramatic drops in performance. While the matching layers of HDC and GeoTr cause out-of-memory-error due to their quadratic complexity, being unable to be incorporated at fine-level with large input spatial resolutions, the proposed PMT not only efficiently processes source and target features without memory burden but also effectively

Table 4. **Ablation study on the choice of fine-level matcher.** Proxy Match Transform layer at fine-level yields the best assembly accuracy while incurring low-compute complexity than baselines.

Ref.	Coarse-level Matcher	Fine-level Matcher	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})
(a)	PMT	None	0.53	0.43	20.70	6.63
(b)		Linear	0.47	0.37	17.55	5.68
(c)		MLP	0.49	0.38	17.35	5.69
(d)		HDC	Out of memory error			
(e)		GeoTr	Out of memory error			
Ours		PMT	**0.39**	**0.25**	**17.14**	**5.53**

Table 5. **Ablation study on the impact of Proxy Match Transform as a fine-level matcher.** Proxy Match Transform layer consistently boosts performance with various coarse-level matchers.

Ref.	Coarse-level Matcher	Fine-level Matcher	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})
(a)	None	None	0.69	0.57	27.71	8.78
(b)		PMT	**0.60**	**0.52**	**24.66**	**7.42**
(c)	Linear	None	0.64	0.53	26.14	7.42
(d)		PMT	**0.55**	**0.50**	**22.44**	**6.69**
(e)	MLP	None	0.66	0.50	26.93	7.35
(f)		PMT	**0.57**	**0.44**	**23.74**	**7.03**
(g)	HDC	None	0.76	0.63	27.75	8.68
(h)		PMT	**0.63**	**0.48**	**23.43**	**7.08**
(i)	GeoTr	None	0.61	0.51	**22.81**	**7.28**
(j)		PMT	**0.48**	**0.33**	23.91	7.32
(k)	PMT	None	0.53	0.43	20.70	6.63
Ours		PMT	**0.39**	**0.25**	**17.14**	**5.53**



(a) (b)

Figure 3. (a) t-SNE visualization of proxy tensor (colored in purple), source features $\mathbf{F}_{\mathcal{X}_c}$ and target features $\mathbf{F}_{\mathcal{Y}_c}$. The source and target features are colored in warm (red) and cool (blue) tones, respectively, and those on mating surfaces are colored in orange and lightblue. (b) Feature visualization in 3D space. Source $\mathcal{X}_1$ and target features $\mathcal{Y}_1$ with closer proximity to the proxy tensor are highlighted in red and blue, respectively, and features on mating surfaces are highlighted in orange and lightblue. For this visualization, we use proxy tensor at a head index of $h = 0$: $\mathbf{P}^{(0)}$.

Table 6. Multi-part assembly results on the Breaking Bad dataset.

Method	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})	PA_{CRD} ↑ (%)	PA_{CD} ↑ (%)	CRD ↓ (10^{-2})	CD ↓ (10^{-3})	RMSE (R) ↓ (°)	RMSE (T) ↓ (10^{-2})	PA_{CRD} ↑ (%)	PA_{CD} ↑ (%)
	everyday						artifact					
Global (2019; 2020a)	27.79	15.30	55.42	15.31	36.42	37.90	26.42	14.92	54.41	14.48	36.67	36.97
LSTM (2020)	27.69	15.23	54.78	15.24	36.74	38.97	28.15	14.61	53.59	15.49	36.67	37.25
DGL (2020)	27.90	13.23	55.76	15.33	36.99	39.70	27.48	13.91	54.66	15.10	36.66	37.40
Wu et al. (2023b)	28.18	19.70	54.98	15.59	35.66	36.28	26.02	15.81	54.35	14.27	36.63	37.02
Jigsaw (2023)	14.13	11.82	41.12	11.74	52.48	60.26	16.10	9.53	42.01	17.47	56.93	65.58
PMTR (Ours)	**6.51**	**5.56**	**31.57**	**9.95**	**66.95**	**70.56**	**5.67**	**4.33**	**31.58**	**10.08**	**66.96**	**71.61**

Figure 4. Qualitative results of pairwise shape assembly (Upper row) and multipart shape assembly (Bottom row) on Breaking Bad dataset.

exchanges information between them via proxy tensor. In Tab. 5, similar experiments are conducted at coarse-level. As evident from the tables, incorporating the PMT layer as both fine and coarse matcher consistently leads to superior performance, affirming its superiority over the state-of-the-art matching layers (Min et al., 2021; Qin et al., 2022).

4.5. Multi-part Assembly

To assess the generalizibility of our method, we extend our evaluation to include multiple input parts, *i.e.*, multi-part assembly, which requires the model to understand the pairwise correspondence relationships among all input parts. Utilizing the two-part assembly framework (Fig. 2), it begins with computing relative transformations between each pair of the P parts. We then construct a *pose graph* wherein each node and factor respectively represent an individual part and the predicted relative transformation, *i.e.*, pose, between two parts. To optimize this pose graph for assembly, we employ a recent transformation averaging method detailed in the work of Dellaert et al. (2020). After the optimization, we evaluate our method using the metrics from pairwise assembly, supplemented by Part Accuracy (PA_{CD}) (Huang et al., 2020) – the percentage of parts with Chamfer Distance less than the predefined threshold of 0.01 – as well as CRD-based Part Accuracy (PA_{CRD}) with 0.1 threshold. As seen from Tab. 6 and Fig. 4, our method significantly surpasses all baselines on all metrics on the multi-part assembly, demonstrating robust generalization to multiple input scenarios. For details on the evaluation metrics, refer to Appendix D.

5. Scope and Limitations

Despite the advances in efficient point cloud matching and shape assembly, our method still faces several limitations. First, the accuracy of our method can be compromised in scenarios with extremely low overlap between point clouds, which can hinder the identification of reliable correspondences. Second, our method, like many others in the field, requires extensive training on domain-specific datasets to achieve optimal performance. Third, while our experiments demonstrate the efficacy of PMT in shape assembly tasks, it have not been extensively tested across other potential applications such as robotics, manufacturing, digital artistry, or even restoration of ancient artifacts via more accurate and detailed part assembly. Thus, the applicability of our approach beyond geometric shape assembly remains to be fully validated. We leave this to future work.

6. Conclusion

We have introduced a low-complexity, high-order feature transform layer, Proxy Match Transform, designed for efficient approximation of traditional compute-intensive high-order feature transforms. The significant performance improvements over the recent state of the arts with lower computational load indicate that its effective real-world applicability from artifact reconstruction to manufacturing. Although the proposed method has been applied exclusively to geometric shape assembly, its remarkable improvements across various evaluation metrics indicate its profound potential for broad applications.

Acknowledgements

This work was supported by IITP grants (RS-2022-II220290: Visual Intelligence for Space-Time Understanding and Generation (30%), RS-2021-II212068: AI Innovation Hub (60%), RS-2019-II191906: AI Graduate School Program at POSTECH (5%), RS-2021-II211343: AI Graduate School Program at SNU: 2021-0-01343 (5%)) funded by the Korea government.

Impact Statement

The advancements in geometric shape assembly hold paramount potentials across numerous fields, from archaeological artifact reconstruction to industrial manufactures. This research can also advance the field manufacturing, robotics, digital artistry, and even restoration of ancient artifacts via more accurate and robust shape assembly.

References

Chen, S., Xu, H., Li, R., Liu, G., Fu, C.-W., and Liu, S. Sirapcr: Sim-to-real adaptation for 3d point cloud registration. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2023.

Chen, X., Zhang, H., Lin, J., Hu, R., Lu, L., Huang, Q.-X., Benes, B., Cohen-Or, D., and Chen, B. Dapper: decompose-and-pack for 3d printing. *ACM Trans. Graph.*, 34(6):213–1, 2015.

Chen, Y., Zeng, Q., Ji, H., and Yang, Y. Skyformer: Remodel self-attention with gaussian kernel and nyström method. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Chen, Y.-C., Li, H., Turpin, D., Jacobson, A., and Garg, A. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chen, Z., Gong, M., Ge, L., and Du, B. Compressed self-attention for deep metric learning with low-rank approximation. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

Cho, S., Hong, S., Jeon, S., Lee, Y., Sohn, K., and Kim, S. Cats: Cost aggregation transformers for visual correspondence. *Advances in Neural Information Processing Systems*, 34:9011–9023, 2021.

Choy, C., Dong, W., and Koltun, V. Deep global registration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations (ICLR)*, 2020.

Dellaert, F., Rosen, D. M., Wu, J., Mahony, R., and Carlone, L. Shonan rotation averaging: Global optimality by surfing so (p)^ n so (p) n. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

Denton, R., Zaremba, W., Bruna, J., LeCun, Y., and Fergus, R. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Falcon, W. and team, T. P. L. Pytorch lightning, 2019. URL https://github.com/PyTorchLightning/pytorch-lightning.

Huang, J., Zhan, G., Fan, Q., Mo, K., Shao, L., Chen, B., Guibas, L., and Dong, H. Generative 3d part assembly via dynamic graph learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., and Schindler, K. Predator: Registration of 3d point clouds with low overlap. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Jacobson, A. Generalized matryoshka: Computational design of nesting objects. In *Computer Graphics Forum*, volume 36, pp. 27–35. Wiley Online Library, 2017.

Jones, R. K., Barton, T., Xu, X., Wang, K., Jiang, E., Guerrero, P., Mitra, N. J., and Ritchie, D. Shapeassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG)*, 39(6): 1–20, 2020.

Khan, S. H., Guo, Y., Hayat, M., and Barnes, N. Unsupervised primitive discovery for improved 3d generative modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9739–9748, 2019.

Kim, S., Min, J., and Cho, M. Transformatcher:match-to-match attention for semantic correspondence. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Li, H., Alhashim, I., Zhang, H., Shamir, A., and Cohen-Or, D. Stackabilization. *ACM Transactions on Graphics, (Proc. of SIGGRAPH Asia 2012)*, 31(6), 2012.

Li, J., Niu, C., and Xu, K. Learning part generation and assembly for structure-aware shape synthesis. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2020a.

Li, Y., Mo, K., Shao, L., Sung, M., and Guibas, L. Learning 3d part assembly from a single image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pp. 664–682. Springer, 2020b.

Lu, J., Sun, Y., and Huang, Q. Jigsaw: Learning to assemble multiple fractured objects. *arXiv preprint arXiv:2305.17975*, 2023.

Min, J. and Cho, M. Convolutional hough matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2940–2950, June 2021.

Min, J., Kang, D., and Cho, M. Hypercorrelation squeeze for few-shot segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021.

Min, J., Zhao, Y., Luo, C., and Cho, M. Peripheral vision transformer. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Mo, K., Zhu, S., Chang, A. X., Yi, L., Tripathi, S., Guibas, L. J., and Su, H. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 909–918, 2019.

Narayan, A., Nagar, R., and Raman, S. Rgl-net: A recurrent graph learning framework for progressive part assembly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 78–87, 2022.

Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., and Xu, K. Geometric transformer for fast and robust point cloud registration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., and Sivic, J. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Rocco, I., Arandjelović, R., and Sivic, J. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. Superglue: Learning feature matching with graph neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Schor, N., Katzir, O., Zhang, H., and Cohen-Or, D. Componet: Learning to generate the unseen by part synthesis and composition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

Sellán, S., Chen, Y.-C., Wu, Z., Garg, A., and Jacobson, A. Breaking bad: A dataset for geometric fracture and reassembly. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*, 2022.

Shi, Y., Cai, J.-X., Shavit, Y., Mu, T.-J., Feng, W., and Zhang, K. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J. Kpconv: Flexible and deformable convolution for point clouds. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

Tian, Y., Xu, J., Li, Y., Luo, J., Sueda, S., Li, H., Willis, K. D., and Matusik, W. Assemble them all: Physics-based planning for generalizable assembly by disassembly. *ACM Trans. Graph.*, 41(6), 2022.

Tulsiani, S., Su, H., Guibas, L. J., Efros, A. A., and Malik, J. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2635–2643, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wang, F. and Hauser, K. Stable bin packing of non-convex 3d objects with a robot manipulator. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8698–8704. IEEE, 2019.

Wu, Q., Shen, Y., Jiang, H., Mei, G., Ding, Y., Luo, L., Xie, J., and Yang, J. Graph matching optimization network for point cloud registration. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023a.

Wu, R., Zhuang, Y., Xu, K., Zhang, H., and Chen, B. Pq-net: A generative part seq2seq network for 3d shapes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Wu, R., Tie, C., Du, Y., Zhao, Y., and Dong, H. Leveraging se (3) equivariance for learning 3d geometric shape assembly. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2023b.

Wu, Y. and He, K. Group normalization. In *Proc. European Conference on Computer Vision (ECCV)*, 2018.

Yu, H., Li, F., Saleh, M., Busam, B., and Ilic, S. Cofinet: Reliable coarse-to-fine correspondences for robust point-cloud registration. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yu, J., Ren, L., Zhang, Y., Zhou, W., Lin, L., and Dai, G. Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zakka, K., Zeng, A., Lee, J., and Song, S. Form2fit: Learning shape priors for generalizable assembly from disassembly. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9404–9410. IEEE, 2020.

Zeng, A., Florence, P., Tompson, J., Welker, S., Chien, J., Attarian, M., Armstrong, T., Krasin, I., Duong, D., Sindhwani, V., et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pp. 726–747. PMLR, 2021.

Zhao, H., Wei, S., Shi, D., Tan, W., Li, Z., Ren, Y., Wei, X., Yang, Y., and Pu, S. Learning symmetry-aware geometry correspondences for 6d object pose estimation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2023.

Zhou, Q. and Jacobson, A. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

A. Theoretical Analysis of Proxy Match Transform

We now derive sufficient conditions such that Proxy Match Transform can express high-dimensional convolution. Our main theoretical result is given below.

Theorem 1. *If we assume* $\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} = \mathbf{I}_{D_{emb}}$ *if* $i = j$ *and* $\mathbf{P}^{(i)\top}\mathbf{P}^{(j)} = \mathbf{0}$ *otherwise for all* $i, j \in [N_h]$*, and define* $\mathbf{A}^{(h)}_{(\mathbf{x},\mathbf{y}),(\mathbf{n},\mathbf{m})} \coloneqq \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{x},\mathbf{n})} \cdot \mathbf{A}^{(h)}_{\mathcal{Y}\ (\mathbf{y},\mathbf{m})}$ *and* $w^{(h)} \coloneqq w^{(h)}_{\mathcal{X}} w^{(h)}_{\mathcal{Y}}$*, then, the dot-product of Proxy Match Transform outputs with a sufficient number of heads* N_h *can express high-dimensional convolutional layer with kernel* $K : \mathbb{R}^6 \to \mathbb{R}$*:* $\mathit{PMT}(\mathbf{F}_{\mathcal{X}}) \cdot \mathit{PMT}(\mathbf{F}_{\mathcal{Y}})^\top = \mathit{Conv}(\mathbf{F}_{\mathcal{X}}, \mathbf{F}_{\mathcal{Y}})$.

Proof. We first take the dot-product of Proxy Match Transform outputs and simplify:

$$\mathrm{PMT}(\mathbf{F}_{\mathcal{X}}) \cdot \mathrm{PMT}(\mathbf{F}_{\mathcal{Y}})^\top = \left(\sum_{h\in[N_h]} \mathbf{A}^{(h)}_{\mathcal{X}} \mathbf{F}_{\mathcal{X}} \mathbf{P}^{(h)\top} w^{(h)}_{\mathcal{X}}\right) \left(\sum_{h\in[N_h]} \mathbf{A}^{(h)}_{\mathcal{Y}} \mathbf{F}_{\mathcal{Y}} \mathbf{P}^{(h)\top} w^{(h)}_{\mathcal{Y}}\right)^\top \tag{14}$$

$$= \sum_{(i,j)\in[N_h]^2} w^{(i)}_{\mathcal{X}} \mathbf{A}^{(i)}_{\mathcal{X}} \mathbf{F}_{\mathcal{X}} \mathbf{P}^{(i)\top} \mathbf{P}^{(j)} \mathbf{F}_{\mathcal{Y}}{}^\top \mathbf{A}^{(j)\top}_{\mathcal{Y}} w^{(j)}_{\mathcal{Y}} \tag{15}$$

$$= \sum_{(i,j)\in[N_h]^2} \delta(i,j) \left(w^{(i)}_{\mathcal{X}} \mathbf{A}^{(i)}_{\mathcal{X}} \mathbf{F}_{\mathcal{X}} \mathbf{F}_{\mathcal{Y}}{}^\top \mathbf{A}^{(j)\top}_{\mathcal{Y}} w^{(j)}_{\mathcal{Y}}\right) \tag{16}$$

$$= \sum_{h\in[N_h]} w^{(h)}_{\mathcal{X}} \mathbf{A}^{(h)}_{\mathcal{X}} \mathbf{F}_{\mathcal{X}} \mathbf{F}_{\mathcal{Y}}{}^\top \mathbf{A}^{(h)\top}_{\mathcal{Y}} w^{(h)}_{\mathcal{Y}}, \tag{17}$$

where $\delta(i,j)$ provides 1 if $i = j$ and 0 otherwise. Using definitions of $\mathbf{A}^{(h)} \in \mathbb{R}^{|\mathcal{X}||\mathcal{Y}|\times|\mathcal{X}||\mathcal{Y}|}$ and $w^{(h)} \in \mathbb{R}$, the output at a specific position $(\mathbf{x},\mathbf{y}) \in \mathbb{R}^6$ is as follows:

$$(\mathrm{PMT}(\mathbf{F}_{\mathcal{X}}) \cdot \mathrm{PMT}(\mathbf{F}_{\mathcal{Y}})^\top)_{(\mathbf{x},\mathbf{y})} = \sum_{h\in[N_h]} \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{x},:)} \mathbf{F}_{\mathcal{X}} \mathbf{F}_{\mathcal{Y}}{}^\top \mathbf{A}^{(h)\top}_{\mathcal{Y}\ (:,\mathbf{y})} w^{(h)} \tag{18}$$

$$= \sum_{h\in[N_h]} \sum_{(\mathbf{n},\mathbf{m})\in\mathcal{X}\times\mathcal{Y}} \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{x},\mathbf{n})} \mathbf{F}_{\mathcal{X}\ (\mathbf{n},:)} \mathbf{F}_{\mathcal{Y}}{}^\top{}_{(:,\mathbf{m})} \mathbf{A}^{(h)\top}_{\mathcal{Y}\ (\mathbf{m},\mathbf{y})} w^{(h)} \tag{19}$$

$$= \sum_{h\in[N_h]} \left(\sum_{(\mathbf{n},\mathbf{m})\in\mathcal{X}\times\mathcal{Y}} \mathbf{A}^{(h)}_{\mathcal{X}\ (\mathbf{x},\mathbf{n})} \cdot \mathbf{A}^{(h)}_{\mathcal{Y}\ (\mathbf{y},\mathbf{m})}\right) \mathbf{C}_{(\mathbf{n},\mathbf{m})} w^{(h)} \tag{20}$$

$$= \sum_{h\in[N_h]} \mathbf{A}^{(h)}_{((\mathbf{x},\mathbf{y}),:)} \mathbf{C}\, w^{(h)}. \tag{21}$$

Now consider the following Lemma:

Lemma 1. *Consider a bijective mapping of natural numbers, i.e., heads, onto 6-dimensional local displacements:* $t(h) : [N_h] \to \Delta(\mathbf{x},\mathbf{y})$*. Let* $\mathbf{A}^{(h)} \in \mathbb{R}^{|\mathcal{X}||\mathcal{Y}|\times|\mathcal{X}||\mathcal{Y}|}$ *be an attention matrix that holds the following:*

$$\mathbf{A}^{(h)}_{(\mathbf{x},\mathbf{y}),(\mathbf{n},\mathbf{m})} = \begin{cases} 1, & \text{if } t(h) = (\mathbf{n},\mathbf{m}) - (\mathbf{x},\mathbf{y}) \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

Then, for any high-dimensional convolution with a kernel $K : \mathbb{R}^6 \to \mathbb{R}$*, there exists* $\{w^{(h)} \in \mathbb{R}\}_{h\in[N_h]}$ *such that following equality holds:*

$$\mathrm{Conv}(\mathbf{F}_{\mathcal{X}}, \mathbf{F}_{\mathcal{Y}})_{(\mathbf{x},\mathbf{y})} = \sum_{h\in[N_h]} \mathbf{A}^{(h)}_{((\mathbf{x},\mathbf{y}),:)} \mathbf{C}\, w^{(h)}. \tag{23}$$

Proof. Consider high-dimensional convolution at position $(\mathbf{x}, \mathbf{y})$:

$$\begin{aligned}
\text{Conv}(\mathbf{F}_{\mathcal{X}}, \mathbf{F}_{\mathcal{Y}})_{(\mathbf{x},\mathbf{y})} &\coloneqq \sum_{(\mathbf{n},\mathbf{m})\in\mathcal{N}(\mathbf{x})\times\mathcal{N}(\mathbf{y})} \mathbf{C}_{(\mathbf{n},\mathbf{m})} K([\mathbf{n}-\mathbf{x}, \mathbf{m}-\mathbf{y}]) \\
&= \sum_{(\boldsymbol{\nu},\boldsymbol{\mu})\in\Delta(\mathbf{x},\mathbf{y})} \mathbf{C}_{(\mathbf{x},\mathbf{y})+(\boldsymbol{\nu},\boldsymbol{\mu})} K((\boldsymbol{\nu},\boldsymbol{\mu})) \\
&= \sum_{h\in[N_h]} \mathbf{C}_{(\mathbf{x},\mathbf{y})+t(h)} K(t(h)) && (\, t(h) : [N_h] \to \Delta(\mathbf{x},\mathbf{y}) \,) \\
&= \sum_{h\in[N_h]} \mathbf{C}_{(\mathbf{x},\mathbf{y})+t(h)} w^{(h)} && (\, w^{(h)} \coloneqq K(t(h)) \in \mathbb{R} \,) \\
&= \sum_{h\in[N_h]} \left(\sum_{(\mathbf{n},\mathbf{m})\in\mathcal{X}\times\mathcal{Y}} \mathbb{1}[t(h) = (\mathbf{n},\mathbf{m}) - (\mathbf{x},\mathbf{y})] \mathbf{C}_{(\mathbf{n},\mathbf{m})} \right) w^{(h)} \\
&= \sum_{h\in[N_h]} \mathbf{A}^{(h)}_{((\mathbf{x},\mathbf{y}),:)} \mathbf{C} w^{(h)}. && (24)
\end{aligned}$$

By applying Lemma 1, we conclude that the dot-product of Proxy Match Transform outputs is equivalent to the high-order convolution. ■

B. Efficiency of Proxy Match Transform

To demonstrate the superiority of the proposed PMT, we provide the efficiency comparison between different matchers, *e.g.*, Geometric Transformer (GeoTr) by Qin et al. (2022) and Proxy Match Transform (PMT), both during training and inference phases in Tab. 7. "Coarse-only" and "Coarse + Fine" refer to two different Proxy Match TransformeR (PMTR) models with PMT integrated only at the coarse-level and both levels, respectively. Specifically, we measure the computational efficiency by employing Floating Point Operations Per Second (FLOPS), and to assess the memory overhead and footprint, we record the peak memory usage for each method during both the training and inference phases, as well as the number of parameters. We also provide the training/inference times required for each matcher. For clarity in our comparison, when measuring the FLOPS, number of parameters, and train/inference times, we exclude those associated with the backbone and focus solely on the matchers: the coarse- or fine-level matcher.

Table 7. Efficiency comparison results between GeoTr (Qin et al., 2022) and PMT. Lower is better.

Method	Coarse-level Matcher	Fine-level Matcher	FLOPS ↓ (G)	# Param. ↓ (K)	Mem. train ↓ (GB)	Mem. test ↓ (GB)	Train time ↓ (ms)	Inference time ↓ (ms)
GeoTransformer (2022)	GeoTr	None	9.67	926.85	6.96	3.10	8.93	8.04
PMTR (Coarse-only)	PMT	None	**0.45**	**273.85**	**2.12**	**0.28**	**4.06**	**3.23**
PMTR (Coarse + Fine)	PMT	PMT	0.78	296.15	3.78	0.88	5.35	3.75

The results clearly indicates that PMT delivers substantial reductions not only in training/inference time but also in memory requirements. Notably, PMT is approximately $\times$**21.5** more efficient in FLOPS, needs $\times$**3.4** more compact number of parameters and $\times$**3.28** / $\times$**11.07** less required memory for training/inference phases compared to GeoTr. Such efficiency is crucial, as it facilitates the practical deployment of our fine-level matcher for intricate matching tasks.

C. Additional implementation Details

Attention Calculation. We adopt the relative-position encoding strategy of PerViT (Min et al., 2022) to compute the attention $\mathbf{A}_{\mathcal{X}}^{(h)}$. Specifically, we compute pairwise Euclidean distances $\mathbf{R}_{\mathcal{X}} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ each of which entry at position $\mathbf{q}, \mathbf{k} \in \mathbb{R}^3$ is defeind as $(\mathbf{R}_{\mathcal{X}})_{\mathbf{q},\mathbf{k}} = ||\mathbf{q}-\mathbf{k}||_2$. An MLP processes this to provide an attention score $\mathbf{A}_{\mathcal{X}}^{(h)}$. $\mathbf{A}_{\mathcal{Y}}^{(h)}$ is similarly defined. We refer the readers to the work of Min et al. (2022) for additional details.

Model hyperparameters. For the backbone network, we utilize KPConv-FPN (Thomas et al., 2019) with subsampling radius of 0.01. We leverage global attention matrix for coarse-level matcher, and local attention matrix (See Sec. 4.2) for fine-level matchers. The number of attention heads N_h is set to 4. Refer to Tab. 8 the rest of hyperparameters. Each matcher takes a specific input and output feature pair, applies a type of attention mechanism, and uses various hyperparameters crucial for its operation.

Table 8. Detail configurations and hyperparameters of different type of matchers. $\mathbf{A}_{\mathcal{Y}}$ similarly defined.

Matcher Type	Input Feature Pair	Output Feature Pair	Attention Type	D_{emb}	i-th PMT	D_{proxy}
Coarse-level matcher	$\{\mathbf{F}_{\mathcal{X}_1}, \mathbf{F}_{\mathcal{Y}_1}\}$	$\{\mathbf{F}_{\mathcal{X}_c}, \mathbf{F}_{\mathcal{Y}_c}\}$	global attention $\mathbf{A}_{\mathcal{X}_1}^{(h)} \in \mathbb{R}^{\lvert\mathcal{X}_1\rvert\times\lvert\mathcal{X}_1\rvert}$	512	1 2	32 128
Fine-level matcher	$\{\mathbf{F}_{\mathcal{X}_2}, \mathbf{F}_{\mathcal{Y}_2}\}$	$\{\mathbf{F}_{\mathcal{X}_3}, \mathbf{F}_{\mathcal{Y}_3}\}$	local attention $\mathbf{A}_{\mathcal{X}_2}^{(h)} \in \mathbb{R}^{\lvert\mathcal{X}_2\rvert\times\epsilon}$	256	1 2	16 64
Fine-level matcher	$\{\mathbf{F}_{\mathcal{X}_3}, \mathbf{F}_{\mathcal{Y}_3}\}$	$\{\mathbf{F}_{\mathcal{X}_f}, \mathbf{F}_{\mathcal{Y}_f}\}$	local attention $\mathbf{A}_{\mathcal{X}_3}^{(h)} \in \mathbb{R}^{\lvert\mathcal{X}_3\rvert\times\epsilon}$	128	1 2	8 32

D. Evaluation Metrics

We employ four different metrics to assess the results. Consider a pair of input point clouds $\{\mathcal{X}, \mathcal{Y}\}$. The ground truth SE(3) relative pose between the point clouds is represented by $\{\mathbf{R}^{\text{GT}}, \mathbf{t}^{\text{GT}}\}$, while the prediction is denoted as $\{\mathbf{R}, \mathbf{t}\}$. We define $\mathbf{T}(\cdot)$ as a function that transform input pose with corresponding rotation $\mathbf{R}$ and translation $\mathbf{t}$.

Chamfer Distance (CD). The chamfer distance between two point clouds S_1, S_2 is defined as

$$d_{\text{CD}}(S_1, S_2) = \frac{1}{S_1}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2 + \frac{1}{S_2}\sum_{y\in S_2}\min_{x\in S_1}\|x-y\|_2^2, \tag{25}$$

which measures the sum of the distance between nearest neighbor correspondences between point clouds. To assess the quality of shape assembly, we measure the chamfer distance between ground truth assembly and the prediction as:

$$\text{CD} = d_{\text{CD}}(\mathbf{T}(\mathcal{X})\cup\mathcal{Y}, \mathbf{T}^{\text{GT}}(\mathcal{X})\cup\mathcal{Y}). \tag{26}$$

CoRrespondence Distance (CRD). While the Chamfer distance calculates the distance between two point clouds, its ability to capture more complex features of the object's geometry, such as symmetry and rotation, is limited. To overcome this limitation, we define a new metric, CoRrespondence Distance (CRD). CRD is simply defined as the Frobenius norm between two point clouds:

$$\text{CRD} = \frac{1}{L}\sum_{i=1}^{L}\|(\mathbf{T}(\mathcal{X})\cup\mathcal{Y})_i - (\mathbf{T}^{\text{GT}}(\mathcal{X})\cup\mathcal{Y})_i\|_F, \tag{27}$$

where $L = |\mathcal{X}| + |\mathcal{Y}|$ is the size of assembled object. By considering all pairwise distances between point clouds, it offers a more comprehensive measure of similarity, capturing both proximity and structural alignment:

Rotational-, Translational-RMSE (RMSE(R), RMSE(T)). Finally, to directly measure the prediction accuracy of transformation parameters, we compute the root mean square error (RMSE) between predicted and ground-truth rotation and translation, respectively. Following the protocols of Sellán et al. (2022), we use Euler angle representation for rotation:

$$\text{RMSE(R)} = \frac{1}{\sqrt{3}}\|\mathbf{R}-\mathbf{R}^{\text{GT}}\|_F, \qquad \text{RMSE(T)} = \frac{1}{\sqrt{3}}\|\mathbf{t}-\mathbf{t}^{\text{GT}}\|_F. \tag{28}$$

Additional metrics for multi-part assembly. Also, we employ two additional metrics to evaluate multi-part assembly performance. Consider a set of input point clouds $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^{P}$ with P parts. The ground truth SE(3) relative poses between the point clouds are represented by $\{\mathbf{T}_i^{\mathrm{GT}}\}_{i=1}^{P}$, while the prediction is denoted as $\{\mathbf{T}_i\}_{i=1}^{P}$. Similar to pairwise assembly, the assembled object can be represented with $\bigcup_{i=1}^{P} \mathbf{T}_i(\mathcal{P}_i)$. Note that in our context, the direction of pose is defined as the transformation that aligns each part $\mathcal{P}_i$ with the coordinate frame of largest fracture as anchor.

Part Accuracy (Chamfer Distance-based). Part accuracy (PA) (Li et al., 2020b) is defined as the percentage of fractures with Chamfer Distance (CD) less than the predefined threshold $\tau_{\mathrm{CD}} = 0.01$:

$$\mathrm{PA}_{\mathrm{CD}} = \frac{1}{P} \sum_{i=1}^{P} \mathbb{1} \left(d_{\mathrm{CD}}(\mathbf{T}_i(\mathcal{P}_i), \mathbf{T}_i^{\mathrm{GT}}(\mathcal{P}_i)) < \tau_{\mathrm{CD}} \right). \tag{29}$$

Part Accuracy (Correspondence Distance-based). Our proposed CoRrespondence Distance (CRD) can be seamlessly adapted for Part Accuracy (PA) evaluation. This adaptation involves substituting the Chamfer Distance (CD) with the CoRrespondence Distance (CRD), and setting the threshold $\tau_{\mathrm{CRD}} = 0.1$:

$$\mathrm{PA}_{\mathrm{CRD}} = \frac{1}{P} \sum_{i=1}^{P} \mathbb{1} \left(\frac{1}{|\mathcal{P}_i|} \sum_{j=1}^{|\mathcal{P}_i|} \| \mathbf{T}_i(\mathcal{P}_i)_j - \mathbf{T}_i^{\mathrm{GT}}(\mathcal{P}_i)_j \|_F < \tau_{\mathrm{CRD}} \right). \tag{30}$$

E. Additional Qualitative Results

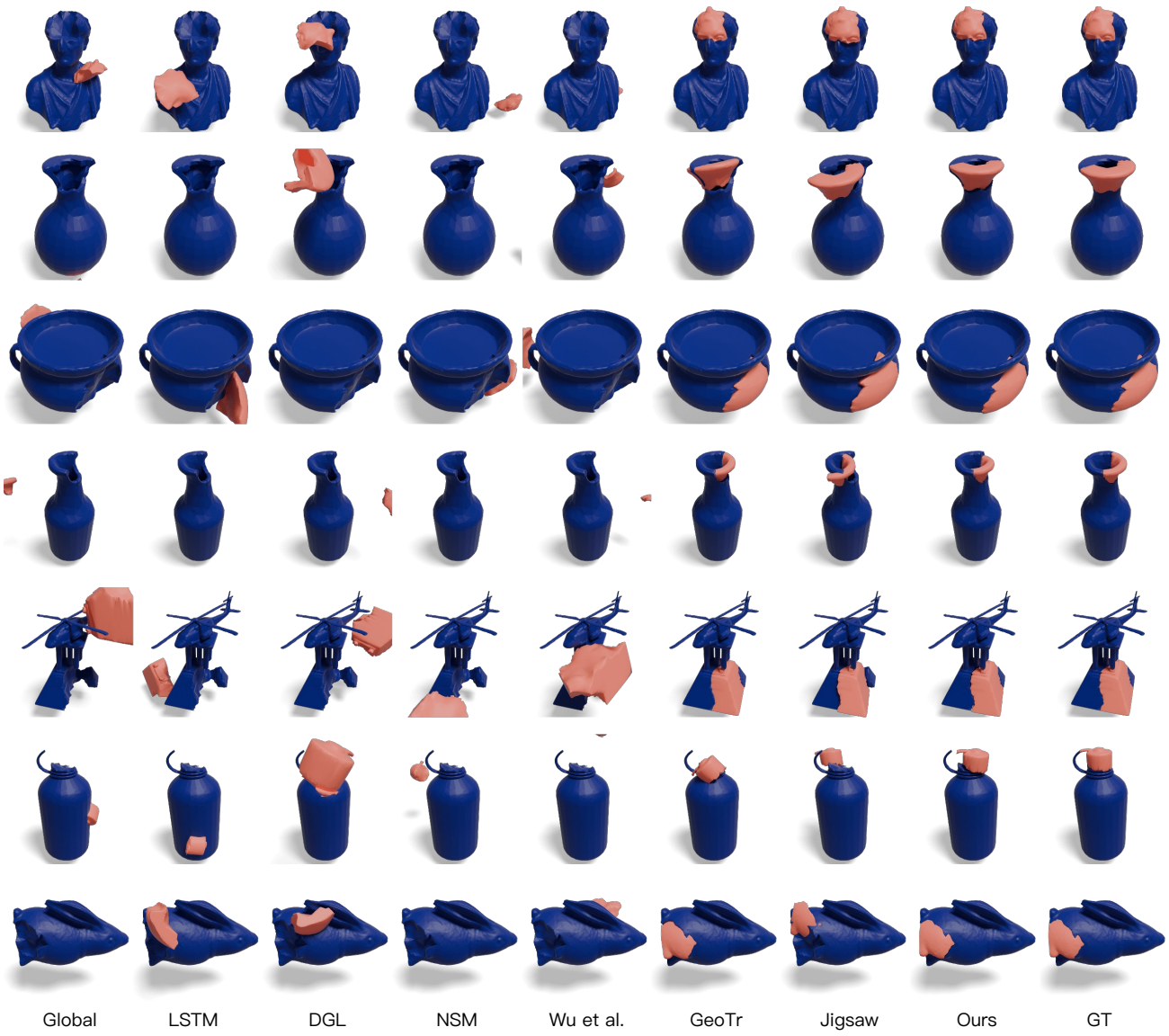


Figure 5. Additional qualitative results of pairwise shape assembly on Breaking Bad dataset.

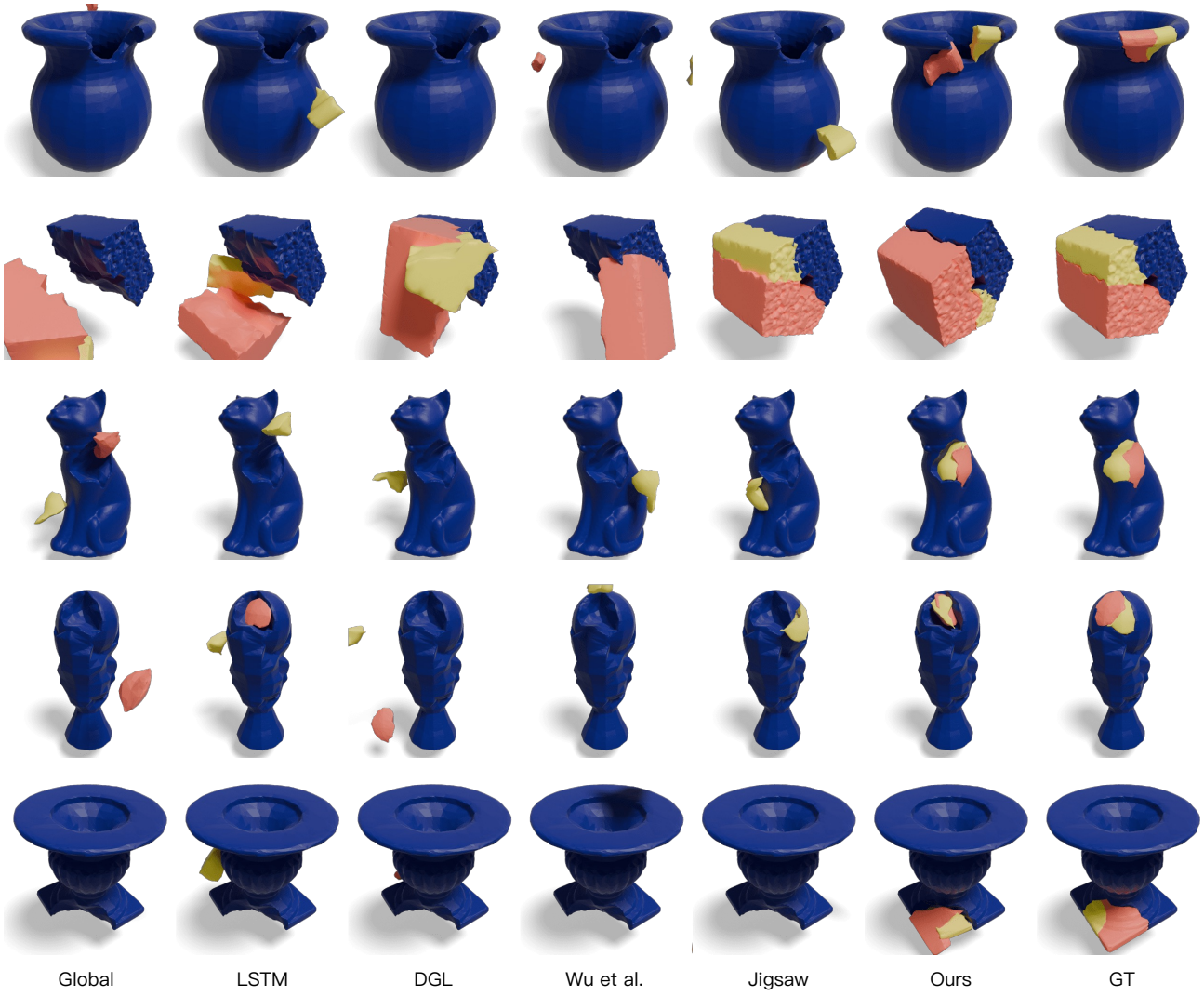


Figure 6. Additional qualitative results of multipart shape assembly on Breaking Bad dataset.