3DPGS: 3D Probabilistic Graph Search for Archaeological Piece Grouping

Junfeng Cheng, Yingkai Yang, Tania Stathaki

Department of Electrical and Electronic Engineering
Imperial College London
London, UK, SW7 2AZ
{junfeng.cheng20, yingkai.yang19, t.stathaki}@imperial.ac.uk

Abstract

In this paper, we propose a new benchmark called "Archaeological Piece Grouping." In the field of archaeology, it is common for broken archaeological pieces, such as artifact fragments, to be mixed. Archaeologists often spend significant time distinguishing these pieces and categorizing them into different groups. Our benchmark introduces a novel, comprehensive dataset named ArcPie, along with new evaluation metrics for this task. Additionally, we propose a new framework called "3D Probabilistic Graph Search" (3DPGS) to address the problem of grouping mixed archaeological pieces. This framework includes a relation network designed to learn the relationships among all the input 3D pieces. Utilizing the relationships learned, our framework generates a probabilistic matching graph that describes the affinity of any two pieces. We also introduce a novel search algorithm to identify groups according to this matrix. Our framework significantly outperforms other baselines.

Code — https://github.com/J-F-Cheng/3dpgs-grouping

1 Introduction

In the field of archaeology, an important task is the restoration of cultural relics, such as ancient handicrafts, steles, sculptures, etc. However, this task is challenging because the objects are often broken into pieces due to their age, spanning hundreds or thousands of years. The pieces from different objects are usually mixed together, adding significant complexity to the process of fixing these objects. To successfully restore these objects, archaeologists typically need a lot of time to compare and separate these mixed pieces. This procedure is extremely time-consuming, further complicating the restoration process for researchers.

The scenario described above involves a thought-provoking problem: given mixed pieces with complex shapes, how do we separate them into different groups based on their original affiliations? For convenience, we name this problem "Archaeological Piece Grouping." We use Fig. 2 in Sec. 3.1 to better illustrate this challenge. This is a complicated task for three main reasons: First, we need to use algorithms to understand the relationships among all the pieces. Second, it is hard to formulate the output of the algorithms because the number of potential groups is unknown. In addition to these issues, the lack of appropriate archaeological data is also a critical problem for researching this topic.

Figure 1: We propose a new dataset, ArcPie (**Arc**haeological **Pie**ce Dataset), for the newly introduced benchmark of the Archaeological Piece Grouping task, which aims to group all the mixed and shuffled pieces. In the figure, the colored 3D meshes represent the fractured pieces, and the white 3D objects represent their original shapes.

Previous works (Cheng et al. 2023; Huang et al. 2020; Zhang et al. 2022; Xu, Zheng, and Jin 2024; Narayan, Nagar, and Raman 2022; Cheng and Stathaki 2024; Du et al. 2024) have proposed algorithms to process separate furniture parts or other types of 3D objects. Huang et al. (2020) propose a novel GNN-based architecture that can automatically assemble parts into a complete shape. Cheng and Stathaki (2024) use an auto-regressive algorithm to group mixed furniture parts. Sellán et al. (2022) propose a new dataset for the assembly task, using simulation methods to break the 3D meshes to construct the dataset.

Although these works have made some achievements in 3D part assembly/grouping tasks, processing pieces from archaeological objects still has three unaddressed problems. First, this type of task requires a large broken archaeological piece dataset for training and testing. However, the current state-of-the-art broken archaeological piece data established by Sellán et al. (2022) is small and contains only around 200 shapes. Additionally, their dataset does not contain category

labels for objects, which brings difficulty in the performance analysis. Secondly, for the grouping task, the performance of Cheng and Stathaki (2024)'s algorithm is still limited, making it difficult to achieve high-quality grouping. Thirdly, the evaluation methods proposed by Cheng and Stathaki (2024) are hardly able to assess the algorithms' performance on the entire predicted groups, making it difficult to reflect the true performance of the algorithms.

To overcome the challenges discussed above, we propose a new dataset called ArcPie (Archaeological Piece Dataset) for the Archaeological Piece Grouping task to address the issue of lacking datasets. ArcPie contains 980 shapes across 11 classes, providing a new platform for us and other researchers to investigate and research archaeological objects. More details about ArcPie arediscussed in Sec. 3. We also propose new evaluation methods for the Archaeological Piece Grouping task. Given the limitations of current grouping algorithms, we propose a new framework named 3D Probabilistic Graph Search (3DPGS) for high-quality piece grouping. In this framework, we introduce a new mechanism to produce a probabilistic matching graph to describe the relationships among all the pieces. Additionally, we propose a novel method to search for groups based on the probabilistic matching graph. We demonstrate our framework in Sec. 4. We summarize our contributions in this paper as follows:

Contributions

- We propose a new benchmark called "Archaeological Piece Grouping", which focuses on solving the problem of separating mixed archaeological pieces into their affiliated groups.
- To address the issue of lacking broken archaeological pieces data, we establish a new dataset called ArcPie for this task, collecting 11 classes of data, totaling 980 objects for the research of the Archaeological Piece Grouping task. Additionally, our data includes pose information for the broken pieces, allowing the dataset to be applied to the 3D Part Assembly task, a task which aims to reassemble fractured pieces into a complete shape. (This paper only considers its usage for the Archaeological Piece Grouping task.)
- We present a novel framework called 3DPGS for high-quality "Archaeological Piece Grouping". Our framework surpasses the previous state-of-the-art algorithms by a large margin.
- We propose a new method called "Group-Level F1" for a more comprehensive evaluation of the grouping task. This method considers the completeness of the predicted groups, which can better reflect the performance of the algorithms.

2 Related Works

2.1 3D Part Grouping

3D Part Grouping (Cheng and Stathaki 2024) is a recently proposed task that aims to identify all possible combinations of given 3D parts. For example, if the parts from two IKEA chairs are carelessly mixed together, how can we use an algorithm to automatically separate these parts? Cheng and Stathaki (2024) propose three algorithms to address the 3D Part Grouping task.

Gradient-Field-based Auto-Regressive Sampling (G-FARS) G-FARS (Cheng and Stathaki 2024) is the current state-of-the-art algorithm in the area of 3D Part Grouping. The core idea of this algorithm is to use a gradient-field-based graph neural network to learn the probability distribution of the part groups. With a well-trained model, the G-FARS framework can auto-regressively sample the part groups from the mixed part set.

Comp-Net Comp-Net (Cheng and Stathaki 2024) is another algorithm for the 3D Part Grouping task. This algorithm can determine whether two input parts can be grouped together. Based on this idea, it can identify all the possible groups in the mixed part set.

GRU-Mask GRU-Mask (Cheng and Stathaki 2024) uses a GRU-based network to encode the input part set and an MLP to output all selection masks, representing the groups in the given part set. However, a drawback of this algorithm is that it requires a predefined output size for the MLP, which may limit the maximum number of groups that GRU-Mask can predict.

2.2 Breaking Bad Dataset

Breaking Bad is a dataset of fractured objects proposed by Sellán et al. (2022). This dataset generates various fracture patterns from a single object based on a realistic physical algorithm, which precomputes multiple fracture modes by solving a generalised eigenvalue problem, as mentioned in Eqn. 1. Breaking Bad is categorised into three main classes, namely, Everyday, Artifact, and Other. Given our research objective on archaeological piece grouping, we solely use Artifact objects to ensure a suitable nature of the training data. However, the Artifact category of this dataset is small, which only contains 200 objects. Besides, the objects are not detailed labelled, which brings difficulty in the experimental analysis.

3 Archaeological Piece Grouping Task and ArcPie Dataset

This section first discusses the problem definition of the "Archaeological Piece Grouping" task. Then, we will introduce how we collect ArcPie , how we create fractures, and how we create a mixed dataset for our benchmark.

3.1 Problem Formulation

We intuitively demonstrate the definition of the Archaeological Piece Grouping task in Fig 2. In our task, our goal is to separate the mixed input pieces, and assign them into different groups. Mathematically, assume we have a mixed and shuffled piece set $\mathcal{P} = \{p_m\}_{m=1}^{M}$ which contains M pieces p_m. Each piece p_m is represented by a point cloud. These pieces are originally belongs to N groups $\mathcal{G} = \{G_n\}_{n=1}^{N}$, but they are mixed together. **The potential number of groups N is NOT known.** Our goal is to use algorithms to find out all the groups $\mathcal{G} = \{G_n\}_{n=1}^{N}$ based on the given

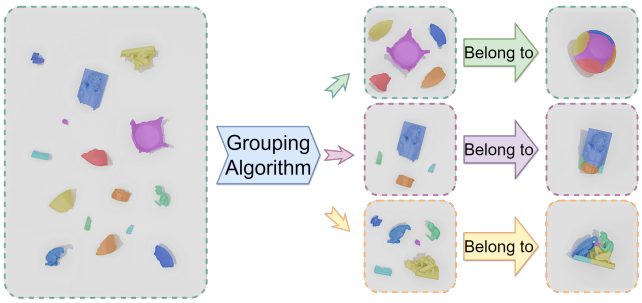

Figure 2: An intuitive demonstration of the problem formulation for the "Archaeological Piece Grouping" task. Our goal is to use an algorithm to automatically identify all possible groups from a completely mixed and shuffled 3D set.

piece set $\mathcal{P}$. In the following content, we use $\hat{\mathcal{G}} = \{\hat{G}_k\}_{k=1}^{K}$ to represent the groups predicted by the algorithms. **It is important to note that K may differ from N if the algorithm wrongly produces extra or fewer groups.**

3.2 Data Collection

For our ArcPie dataset, the original mesh files were collected from ScanTheWorld (2024), the world's largest repository of free, 3D printable cultural objects, created from global 3D scan data. The objects in our dataset were selected based on specific criteria: they must be commonly found in archaeological fields and be fragile, likely to be discovered in fractured states. For instance, coins, despite their frequent presence at archaeological sites, are excluded due to their sturdy nature. This strict selection standard ensures that our dataset accurately reflects real-world scenarios.

All mesh files were manually examined and labelled into 11 distinct categories, spanning Human Sculpture, Musical Instrument, Vase, etc. Details of the collected data are presented in Table 1.

To evaluate the generalization ability of algorithms across various types of archaeological objects, we further divide the collected data into "Seen" and "Unseen" groups based on their categories. The "Seen" objects are used in both training and testing, while the "Unseen" objects are used solely in testing. Our purpose is to verify whether the algorithms can generalize to categories that are not seen during the training procedure.

3.3 Breaking the Objects

Multiple fracture simulation approaches has been proposed in the fields of computer graphics, film making, and physical simulation. Some simulation methods, such as finite elements (Kaufmann et al. 2009; Koschier, Lipponer, and Bender 2015; Pfaff et al. 2014; Wicke et al. 2010), boundary elements (Hahn and Wojtan 2015, 2016), and mass-spring methods (Hirota, Tanoue, and Kaneko 1998; Norton et al. 1991), offer very high spatial resolution yet require considerate time for a single simulation (Sellán et al. 2022), making it unsuitable for generating our large-scale datasets for machine learning purpose. Although some prefracture-based methods are fast enough for scalable application (Chen et al. 2022; Oh, Shin, and Jun 2012; Müller, Chentanez, and Kim 2013; Raghavachary 2002; Su, Schroeder, and Fedkiw 2009), they have limitations such as lack of diversity or being not physically based, which makes generalization to real-world applications more challenging (Sellán et al. 2022).

[*] HS: Human Sculpture, [+] MI: Musical Instrument, [§] AS: Animal Sculpture

Seen (Total: 725)				
Class Name	Bowl	HS[*]	MI[+]	Plate
Number	39	73	17	25
Class Name	Stele	Vase	Weapon	
Number	416	116	39	
Unseen (Total: 255)				
Class Name	AS[§]	Jade	Jar	Tool
Number	130	13	76	36

Table 1: The statistics of our ArcPie dataset. We separate our shapes into "Seen" and "Unseen" categories. "Seen" categories are shapes used in both training and testing, while "Unseen" are the shapes that are only used for testing.

Consequently, similar to the Breaking Bad Dataset, the collected meshes were broken into pieces in various fracture modes using Breaking Good, a real-time destruction breaking algorithm proposed by Sellán et al. (2022). Initially, it computes 20 orthogonal fracture modes naturally in a scalable manner (Sellán et al. 2022) and subsequently simulates 80 different impact-dependent fracture patterns accordingly.

Generating 20 Fracture Modes Breaking Good algorithm assigns values to each corner of a tetrahedron rather than to each vertex, allowing for discontinuities within the mesh. The fracture modes are represented as columns of a matrix $\tilde{\mathbf{U}} \in \mathbb{R}^{12m \times k}$, which solve the generalised eigenvalue problem:

$$\arg \min_{\tilde{\mathbf{U}}^T \tilde{\mathbf{M}} \tilde{\mathbf{U}} = \mathbf{I}} \frac{1}{2} \sum_{r=1}^{k} \text{trace} \left(\tilde{\mathbf{U}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{U}} \right) + \omega \sum_{r=1}^{k} E_D(\tilde{\mathbf{u}}_r) \quad (1)$$

Here, $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{M}}$ are expanded versions of $\mathbf{Q}$, the Hessian matrix of elastic strain energy, and $\mathbf{M}$, the FEM mass matrix. $E_D(\tilde{\mathbf{u}}_r)$ is a convex objective function that measures discontinuity.

Generating 80 Impact-Dependent Fracture Patterns Let $\mathbf{w} \in \mathbb{R}^{12m}$ denote the impact on the shape, it can be projected onto aforementioned precomputed fracture modes to obtain fractured displacement:

$$w^* = \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T \tilde{\mathbf{M}} w \quad (2)$$

This allows the computation of various efficient, impact-dependent fracture patterns by selecting different discontinuity thresholds τ.

In total, 100 distinct fracture patterns can be generated for each mesh file in our ArcPie Dataset.

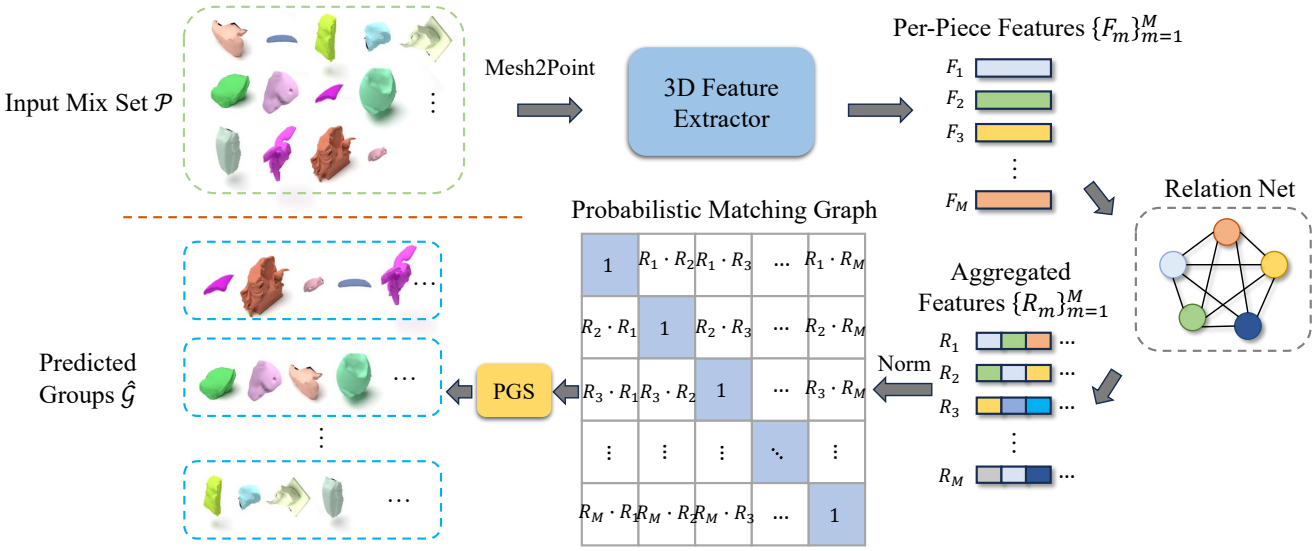

Figure 3: The overview of our proposed 3DPGS framework. Our framework can produce a Probabilistic Matching Graph that indicates the affinity between any two input pieces. Based on the predicted matching graph, we propose a Probabilistic Graph Search algorithm to find out the groups.

3.4 Random Mixing

After obtaining all the fractures, we can create our mixed dataset. The creation pipeline is as follows: first, we randomly select N objects and collect their fractures. Next, we shuffle these pieces, and finally store the preprocessed data.

Cross-Class Mixing Our ArcPie dataset contains "Seen" and "Unseen" categories. For each category, we not only mix data from the same class but also cross-class data (*i.e.* we mix data from different classes), as this setting more closely resembles a real-world scenario.

Mixed Breaking-Bad Artifact Besides training and testing on ArcPie , we also use the same mixing method discussed here to create the mixed Breaking-Bad Artifact dataset Sellán et al. (2022). In the following content, we use the abbreviation "BBArtifact" to represent this dataset.

4 Method

In this section, we introduce our proposed framework, 3D Probabilistic Graph Search (3DPGS). We demonstrate the core idea of 3DPGS in Fig 3. Essentially, our framework can be abstracted into two parts: Probabilistic Matching Graph Learning (PMGL) and Probabilistic Graph Search (PGS). The basic idea of PMGL is to produce a probabilistic matching graph that describes the affinity between any two pieces. In other words, this graph shows how likely any two pieces can be grouped together. PGS is responsible for identifying all possible groups based on the probabilistic matching graph. We propose a general idea to implement 3D probabilistic graph search, thus the networks in our framework, such as the 3D feature extractor and relation network, can be implemented using any reasonable architecture. For example, the 3D feature extractor can be PointCNN (Li et al. 2018), PointNet (Qi et al. 2017a), PointNet++ (Qi et al. 2017b), etc.; the relation network can be a Graph Convolutional Network (Kipf and Welling 2016), EdgeConv (Wang et al. 2019), Transformer (Vaswani et al. 2017), etc.

In the following content, we introduce the details of 3DPGS. We first discuss the implementation of Probabilistic Matching Graph Learning, which includes 3D feature extraction, feature aggregation, training labels, and loss function. The second part introduces the working principle of our Probabilistic Graph Search algorithm.

4.1 Probabilistic Matching Graph Learning (PMGL)

3D Feature Extraction The first step of implementing PMGL is to use 3D feature extraction algorithms to obtain per-piece features $\{F_m\}_{m=1}^M$. We first transform the input meshes into point clouds, and then use a 3D point cloud processing algorithm to extract the features. As mentioned, this feature extractor can be implemented using any reasonable technique. For fair comparisons with the current state-of-the-art (SoTA) algorithms (Cheng and Stathaki 2024), we apply PointNet (Qi et al. 2017a), as they do.

Relation Net and Feature Aggregation In the archaeological piece grouping problem, as suggested by Cheng and Stathaki (2024), it is important to use an efficient algorithm to determine the relationships among all the pieces. In our design, we employ a Relation Net to achieve this goal. Basically, the Relation Net aggregate and combine the previous per-part features $\{F_m\}_{m=1}^M$ to obtain our aggregated features $\{R_m\}_{m=1}^M$, and we intuitively demonstrate this procedure in Fig. 3. As stated before, the relation net can be implemented with any reasonable architecture. For fair comparison purposes, we use a fully-connected EdgeConv-based GNN (Wang et al. 2019) to implement our Relation Net, the

Algorithm 1: Probabilistic Graph Search (PGS)

Input: Probabilistic Matching Graph $\mathcal{MAT}_{\mathbf{G}}$, Probability Threshold T_P

Output: Predicted Groups $\hat{\mathcal{G}} = \{\hat{G}_k\}_{k=1}^{K}$

$M \leftarrow |\mathcal{MAT}_{\mathbf{G}}|$ {Number of vertices in the graph, which equals to the number of pieces M}
$visited \leftarrow [false, \ldots, false]$ {Visited vertices indicator, filled with M "false"s}
$\hat{\mathcal{G}} \leftarrow \emptyset$ {Initialize predicted groups set}
for $v \leftarrow 1$ to M **do**
  if $\neg visited[v]$ **then**
    $\hat{G} \leftarrow \emptyset$
    $S \leftarrow \{v\}$ {Initialize stack with vertex v}
    while $S \neq \emptyset$ **do**
      $u \leftarrow S$.pop()
      if $\neg visited[u]$ **then**
        $visited[u] \leftarrow$ true
        append u to $\hat{G}$
        for all $\{w : \mathcal{MAT}_{\mathbf{G}}[u][w] \geq T_P\}$ **do**
          if $\neg visited[w]$ **then**
            S.push(w) {Select neighbors by threshold}
          end if
        end for
      end if
    end while
    $\hat{\mathcal{G}} \leftarrow \hat{\mathcal{G}} \cup \{\hat{G}\}$ {Add new group to the set}
  end if
end for
return $\hat{\mathcal{G}}$ {Return the set of predicted groups}

same as the state-of-the-art (SoTA) algorithms (Cheng and Stathaki 2024) use. Our experiments in Sec. 5.3 prove that our framework, 3DPGS, can achieve a performance boost through relation learning. The aggregated features are normalized for the next step.

Probabilistic Matching Graph (PMG) Prediction Based on the two steps described above, we can produce our Probabilistic Matching Graph. Our method involves pairwise computation of the dot product among all the normalized aggregated features. Assume that the predicted PMG is represented by the mathematical symbol $\mathcal{MAT}_{\mathbf{G}}$, and each element can be obtained using the following formula:

$$\mathcal{MAT}_{\mathbf{G}}[i][j] = R_i \cdot R_j. \tag{3}$$

We also demonstrate this procedure in Fig. 3. With our prediction algorithm, we can obtain a matrix graph to describe the affinity between any two pieces (i.e., the probability that the pair can be placed in the same group).

PMG Label Generation and Loss Function Our learning objective is to create a Probabilistic Matching Graph that describes the affinity between any two pieces. Finding a suitable method for creating training labels is crucial in this case. In our training procedure, we directly use information from the ground truth groups to define our labels. We follow this rule: If two pieces belong to the same group, we set their affinity (matching probability) to 1.0; otherwise, it is set to 0.0. With this rule, we can create an accurate ground truth Probabilistic Matching Graph. In the training procedure, we use Mean Squared Error (MSE) as our loss function.

It should be mentioned that for negative pairs (*i.e.*, those with 0.0 affinity), we transform this value to -1.0 in the training procedure, as the range for the dot product between two normalized vectors is $[-1.0, 1.0]$. In the PGS process (*i.e.*, our inference procedure), we adjust the range of the Matching Graph from $[-1.0, 1.0]$ back to $[0.0, 1.0]$.

4.2 Probabilistic Graph Search (PGS)

After obtaining our probabilistic matching graph, the remaining challenge is how to retrieve groups from this matching information. To address this, we propose the Probabilistic Graph Search (PGS) algorithm, as detailed in Algorithm 1. Recall that our Probabilistic Matching Graph illustrates the likelihood that any two pieces can be grouped together. In our algorithm, we start from one node in the graph and attempt to visit nodes as distant as possible until there are no more nodes left to visit. Then, we backtrack to previous nodes to see if there are other nodes that can be visited. Utilizing this procedure, the PGS algorithm enables us to identify all the groups.

Probability Threshold To determine whether pieces have sufficient affinity to be considered in one group, we use a parameter called the "Probability Threshold" (T_P). Only when the connection probability between two nodes exceeds this threshold do we consider them as potentially grouped together. This approach allows us to disregard pairs of nodes with very weak connections and focus only on those with stronger connections. The advantage of this threshold is significant: consider a scenario where, if the probability graph wrongly associates two pieces that should be in different groups, the PGS algorithm might incorrectly merge their affiliated groups into one. This issue can greatly impair the performance of 3DPGS, but the Probability Threshold can address this problem. We demonstrate the importance of the Probability Threshold through an ablation study in Sec. 5.3.

5 Experiments

5.1 Metrics

We apply both Part-level (Piece-level) and Group-level Evaluation in our experiments. Recall the problem definition, we define $\{G_n\}_{n=1}^{N}$ as the ground truth groups and $\{\hat{G}_k\}_{k=1}^{K}$ as the predicted groups. We use these mathematical symbols in the following content.

Part-level (Piece-level) Evaluation In the piece grouping task, evaluating the methods' performance is not easy. The most critical issue is the inconsistency between the predicted groups and the ground truth groups. Compared with the ground truth groups, the predicted groups may even have different numbers of groups and different orders among these groups. Cheng and Stathaki (2024) propose a way to evaluate the grouping algorithms. Their core idea is to use Jaccard Similarity (Eqn. 4) to match the predicted groups with the

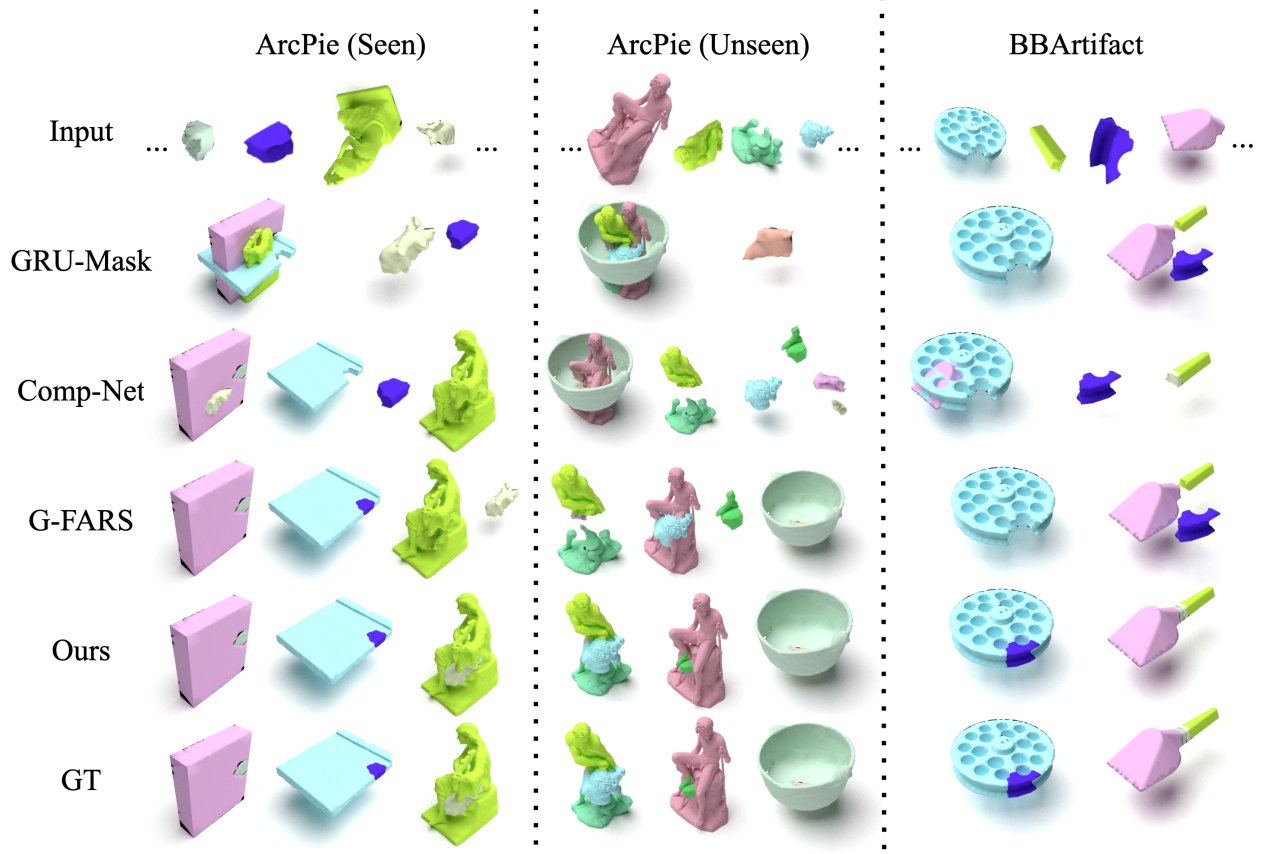

Figure 4: The comparison results among all the algorithms. We can observe that only our algorithm is able to correctly group all the input mixed sets. In this figure, for intuitive demonstration, the pieces are translated and rotated to the ground truth poses. **Please note that the information on ground truth poses is NOT used in the grouping procedure.**

ground truth groups. Each predicted group is matched with the most "similar" ground truth group (*i.e.* with the highest Jaccard Similarity score). Based on the matching, they calculate how many parts are correctly assigned to the affiliated group. With the above procedure, the TP, FP, and FN can be obtained, and naturally, the Piece-level F1 score (PL-F1, including two averaging method: ssa / oa (Cheng and Stathaki 2024)) can be calculated for the algorithms.

$$J(\hat{G}_k, G_n) = \frac{|\hat{G}_k \cap G_n|}{|\hat{G}_k \cup G_n|} \tag{4}$$

However, a great drawback of this evaluation way is that it does not consider the completeness of the groups. Even if a group only has one part correctly assigned, it is considered as a true positive part. To address this issue, we propose a new metric which considers the completeness of all the predicted groups. We introduce our new metric in the following content.

Group-level Evaluation To successfully evaluate the completeness of each group, we propose a new metric called Group Level F1 Score (GL-F1) which draws inspiration from the evaluation methods from object detection tasks.

When we calculate Average Precision in object detection tasks, we normally use a method like IoU to compare the predicted detection boxes and the ground truth boxes, and we set a threshold to judge whether the predicted boxes are True Positives. Similarly, in our task, we use Jaccard Similarity to compare our predicted groups with the ground truth groups, and see whether the score can exceed a certain threshold. However, due to the invalidity of the confidence scores produced by the models, we can calculate only the F1 score rather than the Average Precision in our task. The GL-F1 is a more comprehensive way to test the performance of the grouping algorithms, which overcomes the issue in the Part-level Evaluation. In our evaluation, we calculate GL-F1 scores under three Jaccard Similarity thresholds, which are 0.5 (GL-F1 0.5), 0.75 (GL-F1 0.75), and 0.5: 0.95: 0.05 (GL-F1 [0.5, 0.95]).

5.2 Results and Discussions

We demonstrate the quantitative results in Table 2. The results show that our algorithm outperforms other baselines by a large margin, particularly on the metric of GL-F1. These results underscore the strong grouping capability of our proposed 3DPGS. Besides, our algorithm still achieves a very high score on Unseen category data, which further proves the generalization ability of 3DPGS.

Dataset	Metric	GRU-Mask	Comp-Net	G-FARS	3DPGS
ArcPie Seen	PL-F1 ssa / oa	0.720 / 0.705	0.786 / 0.767	0.826 / 0.804	**0.854 / 0.830**
	GL-F1 0.5	0.615	0.653	0.700	**0.806**
	GL-F1 0.75	0.236	0.292	0.442	**0.547**
	GL-F1 [.5:.95]	0.301	0.347	0.486	**0.587**
ArcPie Unseen	PL-F1 ssa / oa	0.706 / 0.687	0.759 / 0.738	0.798 / 0.773	**0.829 / 0.789**
	GL-F1 0.5	0.575	0.603	0.646	**0.769**
	GL-F1 0.75	0.208	0.236	0.375	**0.491**
	GL-F1 [.5:.95]	0.269	0.294	0.422	**0.533**
BBArtifact	PL-F1 ssa / oa	0.698 / 0.667	0.718 / 0.689	0.771 / 0.747	**0.800 / 0.756**
	GL-F1 0.5	0.565	0.560	0.598	**0.727**
	GL-F1 0.75	0.185	0.159	0.306	**0.419**
	GL-F1 [.5:.95]	0.254	0.221	0.353	**0.466**

Table 2: The comparison results on the two datasets. Expect from our dataset ArcPie, we also apply the Artifact data from Breaking Bad dataset (Sellán et al. 2022) to further test the performance of all the algorithms.

We show the qualitative in Fig. 4. **We rotate and translate them to the ground truth poses ONLY for demonstration purposes. The ground truth poses are NOT used in the grouping procedure.** The results show that only our algorithm can correctly group all the input mixed sets. It is difficult for other algorithms to produce the correct groups.

5.3 Ablation Study

	metrics	w/o rel	w/o dot	3DPGS
Seen	PL-F1 ssa	0.693	0.845	**0.854**
	PL-F1 oa	0.662	0.811	**0.830**
	GL-F1 0.5	0.627	0.787	**0.806**
	GL-F1 0.75	0.163	0.538	**0.547**
	GL-F1 [.5:.95]	0.229	0.575	**0.587**
Unseen	PL-F1 ssa	0.681	0.818	**0.829**
	PL-F1 oa	0.648	0.772	**0.789**
	GL-F1 0.5	0.594	0.744	**0.769**
	GL-F1 0.75	0.142	0.471	**0.491**
	GL-F1 [.5:.95]	0.208	0.514	**0.533**

Table 3: Ablation study on architecture: "w/o rel" means 3DPGS without the relation net; "w/o dot" is another variant of 3DPGS, where the Probabilistic Matching Graph is produced by an MLP rather than by dot product operations.

Our ablation study contains two parts. The first part (Table 3) is the architecture ablation, which tests the influences of different important modules. The second part (Fig. 5) is the research on the proposed PGS algorithm. In this ablation study, we focus on testing how the threshold affects the performance of our framework. The dataset used in the ablation study is ArcPie.

Architecture Ablation (w/o rel) In this ablation, we remove the relation net from our 3DPGS, and the results in Table 3 show that we have a great performance decreasing.

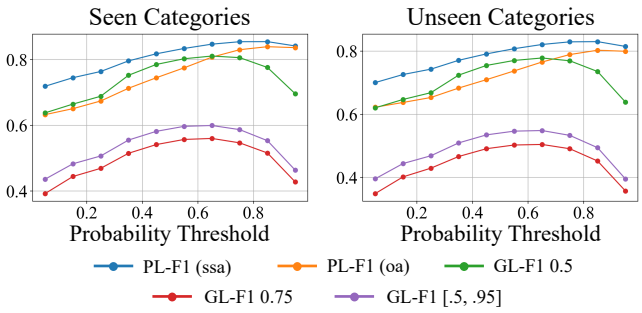

Figure 5: Ablation study results on Probability Threshold.

This proves the effectiveness and importance of the relation net.

Arachitecture Ablation (w/o dot) We present a variation of which use an MLP to produce our Pobalistic Maching Graph. We can also observe a score-decreasing. This phenomenon proves the importance of the dot product operation when producing our matching graph.

Probability Threshold The ablation study in Fig. 5 shows that T_P has a significant impact on the performance of 3DPGS. When T_P is set to 0.65 or 0.75, 3DPGS achieves relatively high performance across different metrics. We apply 0.75 in our experiments.

6 Conclusion

In this paper, we present a novel benchmark called "Archaeological Piece Grouping." This benchmark includes a new dataset called ArcPie, which can not only be used for the grouping task but also has the potential to be applied to other 3D tasks, such as 3D part assembly. Additionally, we propose a new evaluation metric for our benchmark. We also introduce a new algorithm called 3DPGS for the grouping task and demonstrate its performance and strong capabilities. In the future, we will consider transferring our algorithm to the physical world, such as in robotic tasks.

Acknowledgments

The authors would like to express their sincere gratitude to Ms. Xinyi Zhu for her invaluable help and support in collecting and creating ArcPie, which greatly contributed to the success of this work.

References

Chen, Y.-C.; Li, H.; Turpin, D.; Jacobson, A.; and Garg, A. 2022. Neural Shape Mating: Self-Supervised Object Assembly with Adversarial Shape Priors. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cheng, J.; and Stathaki, T. 2024. G-FARS: Gradient-Field-based Auto-Regressive Sampling for 3D Part Grouping. In *Computer Vision and Pattern Recognition*.

Cheng, J.; Wu, M.; Zhang, R.; Zhan, G.; Wu, C.; and Dong, H. 2023. Score-pa: Score-based 3d part assembly. *arXiv preprint arXiv:2309.04220*.

Du, B.; Gao, X.; Hu, W.; and Liao, R. 2024. Generative 3D Part Assembly via Part-Whole-Hierarchy Message Passing. *arXiv preprint arXiv:2402.17464*.

Hahn, D.; and Wojtan, C. 2015. High-resolution brittle fracture simulation with boundary elements. *ACM Transactions on Graphics*, 34(4): 1–12.

Hahn, D.; and Wojtan, C. 2016. Fast approximations for boundary element based brittle fracture simulation. *ACM Transactions on Graphics*, 1–11.

Hirota, K.; Tanoue, Y.; and Kaneko, T. 1998. Generation of crack patterns with a physical model. *The Visual Computer*, 14(3): 126–137.

Huang, J.; Zhan, G.; Fan, Q.; Mo, K.; Shao, L.; Chen, B.; Guibas, L. J.; Dong, H.; et al. 2020. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems*, 33: 6315–6326.

Kaufmann, P.; Martin, S.; Botsch, M.; Grinspun, E.; and Gross, M. 2009. Enrichment textures for detailed cutting of shells. *ACM Transactions on Graphics*, 28(3): 1–10.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Koschier, D.; Lipponer, S.; and Bender, J. 2015. Adaptive tetrahedral meshes for brittle fracture simulation. *Symposium on Computer Animation,Symposium on Computer Animation*.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31.

Müller, M.; Chentanez, N.; and Kim, T.-Y. 2013. Real time dynamic fracture with volumetric approximate convex decompositions. *ACM Transactions on Graphics*, 1–10.

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

Norton, A.; Turk, G.; Bacon, B.; Gerth, J.; and Sweeney, P. 1991. Animation of fracture by physical modeling. *The Visual Computer*, 7(4): 210–219.

Oh, S.; Shin, S.; and Jun, H. 2012. Practical simulation of hierarchical brittle fracture. *Computer Animation and Virtual Worlds*, 23(3–4): 291–300.

Pfaff, T.; Narain, R.; de Joya, J. M.; and O'Brien, J. F. 2014. Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics*, 33(4): 1–9.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Raghavachary, S. 2002. Fracture generation on polygonal meshes using Voronoi polygons. In *ACM SIGGRAPH 2002 conference abstracts and applications*.

ScanTheWorld. 2024. ScanTheWorld. www.myminifactory.com/scantheworld/. Accessed: May 03, 2024.

Sellán, S.; Chen, Y.-C.; Wu, Z.; Garg, A.; and Jacobson, A. 2022. Breaking Bad: A Dataset for Geometric Fracture and Reassembly. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Sellán, S.; Luong, J.; Silva, L. M. D.; Ramakrishnan, A.; Yang, Y.; and Jacobson, A. 2022. Breaking Good: Fracture Modes for Realtime Destruction. arXiv:2111.05249.

Su, J.; Schroeder, C.; and Fedkiw, R. 2009. Energy stability and fracture for frame rate rigid body simulations. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5): 1–12.

Wicke, M.; Ritchie, D.; Klingner, B. M.; Burke, S.; Shewchuk, J. R.; and O'Brien, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Transactions on Graphics*, 1–11.

Xu, B.; Zheng, S.; and Jin, Q. 2024. SPAFormer: Sequential 3D Part Assembly with Transformers. *arXiv preprint arXiv:2403.05874*.

Zhang, R.; Kong, T.; Wang, W.; Han, X.; and You, M. 2022. 3d part assembly generation with instance encoded transformer. *IEEE Robotics and Automation Letters*, 7(4): 9051–9058.