FragmentDiff: A Diffusion Model for Fractured Object Assembly

QUN-CE XU, BNRist, Department of Computer Science and Technology, Tsinghua University, China
HAO-XIANG CHEN, BNRist, Department of Computer Science and Technology, Tsinghua University, China
JIACHENG HUA, Department of Computer Science and Technology, Tsinghua University, China
XIAOHUA ZHAN, Department of Foreign Languages and Literatures, Tsinghua University, China
YONGLIANG YANG, Department of Computer Science, University of Bath, United Kingdom
TAI-JIANG MU*, BNRist, Department of Computer Science and Technology, Tsinghua University, China

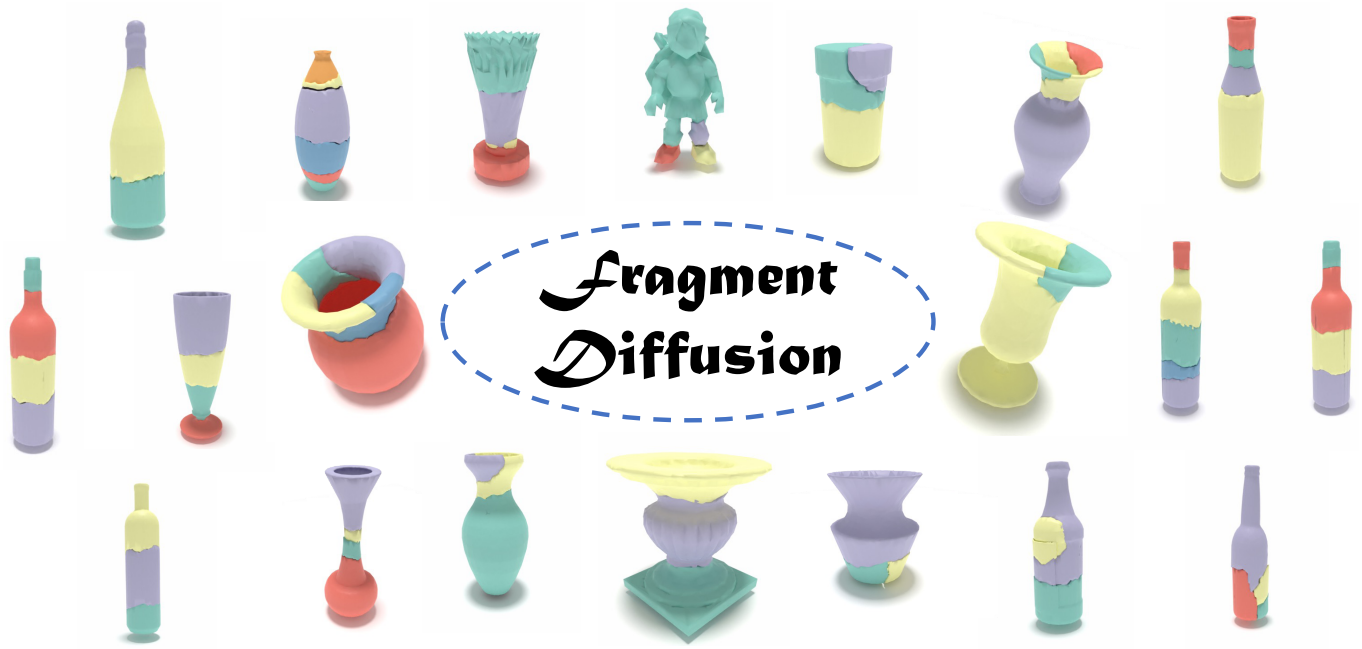

Fig. 1. Our proposed FragmentDiff accurately assembles objects from randomly posed fractured pieces by diffusing in the pose parameter space.

Fractured object reassembly is a challenging problem in computer vision and graphics with applications in industrial manufacturing and archaeology. Traditional methods based on shape descriptors and geometric registration often struggle with ambiguous features, resulting in lower accuracy. Recent data-driven methods are inherently affected by the representation and learning ability of the trained models. To address this, we propose a novel approach inspired by diffusion models and transformers. Our method applies diffusion denoising via a transformer to predict the pose parameter of each fragment, taking advantage of their global feature correlation and pose prior learning abilities. We evaluate our approach on a fractured object dataset and demonstrate superior performance compared to state-of-the-art methods. Our method offers a promising solution for accurate and robust fractured object reassembly, advancing the field in complex shape analysis and assembly tasks.

CCS Concepts: • **Computing methodologies** → **Shape analysis**.

Additional Key Words and Phrases: fractured object, fragment assembly, diffusion model

ACM Reference Format:
Qun-Ce Xu, Hao-Xiang Chen, Jiacheng Hua, Xiaohua Zhan, Yongliang Yang, and Tai-Jiang Mu. 2024. FragmentDiff: A Diffusion Model for Fractured Object Assembly. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 03–06, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3680528.3687673

*Corresponding author: Tai-Jiang Mu (taijiang@tsinghua.edu.cn).

Authors' Contact Information: Qun-Ce Xu, BNRist, Department of Computer Science and Technology, Tsinghua University, China, quncexu@tsinghua.edu.cn; Hao-Xiang Chen, BNRist, Department of Computer Science and Technology, Tsinghua University, China, chx20@mails.tsinghua.edu.cn; Jiacheng Hua, Department of Computer Science and Technology, Tsinghua University, China, hjc21@mails.tsinghua.edu.cn; Xiaohua Zhan, Department of Foreign Languages and Literatures, Tsinghua University, China, jasonzhanthu@gmail.com; Yongliang Yang, Department of Computer Science, University of Bath, United Kingdom, strongyang@gmail.com; Tai-Jiang Mu, BNRist, Department of Computer Science and Technology, Tsinghua University, China, taijiang@tsinghua.edu.cn.

SA Conference Papers '24, December 03–06, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687673

1 Introduction

Fractured object assembly is a fundamental task in computer vision and graphics. It plays a crucial role in various applications such as industrial manufacturing, archaeology, object reconstruction, and cultural heritage preservation. The ability to accurately and efficiently assemble fragmented objects is essential for restoring damaged artifacts, optimizing production processes, and understanding complex geometries. By assembling fragmented parts, one can extract valuable information from the recovered objects, leading to advancements in quality control, reverse engineering, and historical artifact analysis. Since the target shape is often unknown, and the fragmented parts typically have irregular shapes, it is rather challenging to identify the correct poses of individual parts and align them accordingly. Therefore, manually solving the problem is often time-consuming, skill-demanding, and sometimes even impractical if the number of parts is significant.

With the development of 3D acquisition and shape modeling techniques, large 3D model datasets such as [Chang et al. 2015; Wu et al. 2015] emerged and attracted research attention on 3D shape learning. Other datasets [Mo et al. 2019; Sellán et al. 2022] were also introduced to benchmark the object reassembly task, which aims to facilitate data-driven object assembly from their constituent parts. Early unsupervised assembly methods, such as [Brown et al. 2008; Huang et al. 2006; Papaioannou and Karabassi 2003], focused on defining and matching hand-crafted features based on the fragment shapes. These methods analyzed the geometric properties of fragments (e.g., sharp features, curvatures) to solve the relative positions and orientations between fragments. With the explosion of data scale and the advancements in machine learning, supervised methods [Chen et al. 2022; Jones et al. 2021; Li et al. 2020; Narayan et al. 2022; Zhan et al. 2020] were proposed to address the fractured object assembly problem. These methods leveraged shape assembly priors to estimate fragment poses, where features are automatically learned and correlated from labeled data.

Recently, diffusion-based models [Ho et al. 2020; Song et al. 2021c] have proved their capability in high-fidelity 2D and 3D content generation [Poole et al. 2023; Rombach et al. 2022], as well as discriminative tasks such as image and volume segmentations [Baranchuk et al. 2022; Xing et al. 2023]. By regarding the fragmented parts as a noisy object and the reassembled complete object as the denoised one, we model the fractured object assembly as a denoising process, which has not been well explored. To this end, we propose *FragmentDiff*, a framework that predicts accurate fragment poses for shape assembly, which contains a complete diffusion denoising pipeline based on a transformer architecture. In our setting, the signal for noising and denoising in the diffusion process is the fragment poses rather than their shapes. Specifically, given a set of fragmented parts with arbitrary poses broken from a complete object, we first transfer them to feature embeddings and then feed them as the condition of the diffusion process, similar to those methods for text-conditioned image generation. We leverage the learning ability of transformers to effectively correlate part features for diffusion noise prediction and part adjacency matrix inferences. To facilitate model training of the proposed method, we have constructed a dataset composed of fractured objects based on existing benchmarks [Sellán et al. 2022]. Both quantitative and qualitative evaluations demonstrate that our model is highly effective and outperforms existing baselines.

In summary, our work makes the following major contributions:

- We present a novel approach that addresses the fractured assembly problem based on a diffusion model and its sampling and denoising strategy.
- We devise a comprehensive pipeline that utilizes a transformer architecture to predict diffusion noises in the parameter space as well as the fragment adjacency matrix.
- We conduct quantitative and qualitative evaluations to validate the effectiveness and superiority of our model.

2 Related Works

Fractured Object Reassembly. Automatic reassembly of fractured objects has garnered significant research interest in the last decades. Early research attempts addressed this problem using unsupervised methods that involved analyzing object fragments with traditional geometry processing techniques such as shape segmentation, feature extraction, and shape matching. For 3D solid shape assembly, [Papaioannou and Karabassi 2003; Papaioannou et al. 2001] made early contributions of automatically reconstructing objects from fragmented parts. Subsequent studies addressed more challenging cases, including assembling parts with intricate geometries and working on low-quality data. [Brown et al. 2008; Huang et al. 2006; Koller and Levoy 2006; Shen et al. 2012] were the representatives that tackled these challenges. Recently, a pairwise part reassembly method was proposed based on extracting and matching breaking curves [Alagrami et al. 2023]. Albeit greatly facilitating the reassembly process, traditional methods still rely on defining and matching hand-crafted feature descriptors, which can be easily affected by shape ambiguity (e.g., symmetry, plainness) due to the lack of consideration of more global shape information.

As an early attempt of learning-based approach, Funkhouser et al. [2011] utilized regression trees as the classifier to match hand-crafted features (e.g., thickness, color, convexity) for fragment assembly. More recently, the field of shape assembly has witnessed significant progress due to the availability of 3D shape benchmarks with semantic labels, such as [Chang et al. 2015; Mo et al. 2019]. These benchmarks have paved the way for learning-based methods for reassembling shapes from their semantic parts. Several learning-based approaches [Jones et al. 2021; Narayan et al. 2022; Zhan et al. 2020] have been proposed to address this problem involving semantic information, leveraging machine-learning techniques to reassemble shapes based on their semantic parts. These methods utilize the semantic labels of the parts to guide the reassembly process. Besides, other methods have focused on specific domains, such as [Guo et al. 2022; Willis et al. 2022; Yu et al. 2022], which have presented approaches explicitly tailored for reassembling CAD models. These methods take into account the unique characteristics and constraints of CAD models during the reassembly process.

In contrast to semantic part assembly, [Chen et al. 2022] proposed a learning-based approach for fractured object reassembly without relying on part semantics through a transformer-based network combined with adversarial loss. [Lu et al. 2023] achieved fractured object assembly using a transformer-based learning method with

the proposed primal-dual descriptor. [Wu et al. 2023] introduced the concept of SE(3) equivariance, which helps to improve assembly accuracy and efficiency, especially for complex assemblies involving multiple parts. A very recent work [Scarpellini et al. 2024] proposed a uniform diffusion-based method to assemble puzzles in 2D and 3D domains following a graph structure. However, this work does not perform as well in 3D as in 2D, and there is still room for improvement due to the limitation of graph neural networks. Moreover, to facilitate research in the field, [Sellán et al. 2022] introduced a large-scale benchmark dataset specifically designed for the fractured object reassembly task. This dataset also provides a standardized evaluation framework for assessing the performance of different reassembly methods.

Point Cloud Registration. This problem aims to align overlapping point clouds which are often raw data captured by 3D scanners from different views. Point cloud registration has been extensively studied. While point registration techniques like RANSAC [Fischler and Bolles 1981] and Iterative Closest Point (ICP) [Besl and McKay 1992; Chen and Medioni 1991] are widely used for rigid point cloud registration, the emergence of deep learning drives the research on differentiable RANSAC [Brachmann et al. 2017] and ICP [Wang and Solomon 2019] suitable for learning-based tasks. Readers may refer to existing surveys [Deng et al. 2022; Tam et al. 2013] for a comprehensive understanding.

Here, we mainly review works that concern low-overlap registration. These types of works focus on aligning point clouds with only shallow overlap regions in between, similar to the fractured object assembly task. Recent works [Huang et al. 2012, 2021; Yan et al. 2021] addressed the specific challenges posed by point clouds with small overlapping areas. The key is how to extract feature points from only the shared interface area for robust alignment. Our task also targets the alignment of fragmented parts but with no requirement that there always exist overlaps between parts.

Diffusion Models for Discriminative Tasks. Classic diffusion models, such as those proposed in [Ho et al. 2020; Sohl-Dickstein et al. 2015], have been successfully applied to generative tasks, producing high-fidelity results. This impact extends to both 2D and 3D generation tasks, as demonstrated by works such as [Dhariwal and Nichol 2021; Rombach et al. 2022; Song et al. 2021c; Vahdat et al. 2021; Zeng et al. 2022; Zhou et al. 2021], where diffusion models are used for denoising the input representation or in the latent space.

In contrast to the works above, many researchers have adopted diffusion models for deterministic tasks rather than generative ones. For instance, [Chen et al. 2020; Tritrong et al. 2021; Voynov and Babenko 2020; Voynov et al. 2021; Zhang et al. 2021] utilized GAN models to perform image segmentation. This highlights the potential that generative models can be employed for discriminative tasks. [Baranchuk et al. 2022] proposed semantic segmentation methods based on denoising diffusion models, which effectively capture high-level semantic information. Subsequent works by [Amit et al. 2021; Wu et al. 2024; Xing et al. 2023] tackled segmentation and depth estimation problems using diffusion models.

Similar to the usage of generative models above, we formulate pose prediction using a diffusion-based pipeline, which adapts the powerful generative model for our challenging fractured object assembly task.

3 Fractured Object Assembly Task

We formulate the fractured object assembly task as a diffusion problem. Specifically, given K fragmented parts denoted as $\mathcal{P} = \{P_i | i = 1, 2, ..., K\}$, each of which is represented as a point cloud as in [Sellán et al. 2022]. Our objective is to estimate the canonical SE(3) pose $\widehat{T}_i = \{\widehat{R}_i, \widehat{t}_i | \widehat{R}_i \in \mathrm{SO}(3), \widehat{t}_i \in \mathbb{R}^3\}$ for each part P_i. The recovered object O can be obtained by combining the transformed parts: $O = \bigcup_{i=1}^{K} P_i \otimes \widehat{T}_i$, where $\otimes$ denotes the pose transformation operation $P_i \otimes \widehat{T}_i = \widehat{R}_i P_i + \widehat{t}_i$. By regarding the fragmented parts as a noisy object in total, the assembly problem can be formulated as a denoising process to recover poses for the complete object. Different from previous works which regress the pose $\widehat{T}$ directly by neural networks, we adopt a diffusion model to learn the pose distribution conditioned on the fragmented parts.

4 Methodology

In this section, we present the proposed diffusion-based pose estimation model FragmentDiff, specifically devised for the fractured object assembly task. We first give an overview of our model, then detail the network structure as well as the training procedure and losses.

4.1 Overview

As shown in Fig. 2, the fragments are first transformed into feature vectors in the latent space through a point cloud encoder. During the training phase, we introduce noises to the input poses and train a transformer as a noise predictor. In addition, each timestep is encoded into an embedding and combined with the feature vectors of the fragments as a condition for the transformer, which can effectively process and correlate these feature embeddings. Meanwhile, we employ the learned fragment features to infer the explicit relationships among fragments by referring to their adjacency matrix using a transformer. We recognize its critical role in guiding the prediction of arbitrary fragment poses by establishing their spatial relations, which is in line with traditional methods. During the reverse diffusion process, the model learns to reconstruct the original signal (poses) from a noisy sample. It generates new samples by starting from a random noise input and iteratively removing the noise until the poses are generated for fragment reassembly.

4.2 Fragment Diffusion

4.2.1 Generic Diffusion Model. The generic Denoising Diffusion Probabilistic Models (DDPM) [Ho et al. 2020] employ a Markov chain trained via variational inference to generate samples that conform to the data distribution within a finite timeframe. The model encompasses a forward process that incrementally introduces Gaussian noise to the data, transforming the original data x_0 into a noised state x_T. This forward process is defined by:

$$\begin{aligned} q(x_{1:T}|x_0) &= \prod_{t=1}^{T} q(x_t|x_{t-1}), \\ q(x_t|x_{t-1}) &= \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \end{aligned} \tag{1}$$

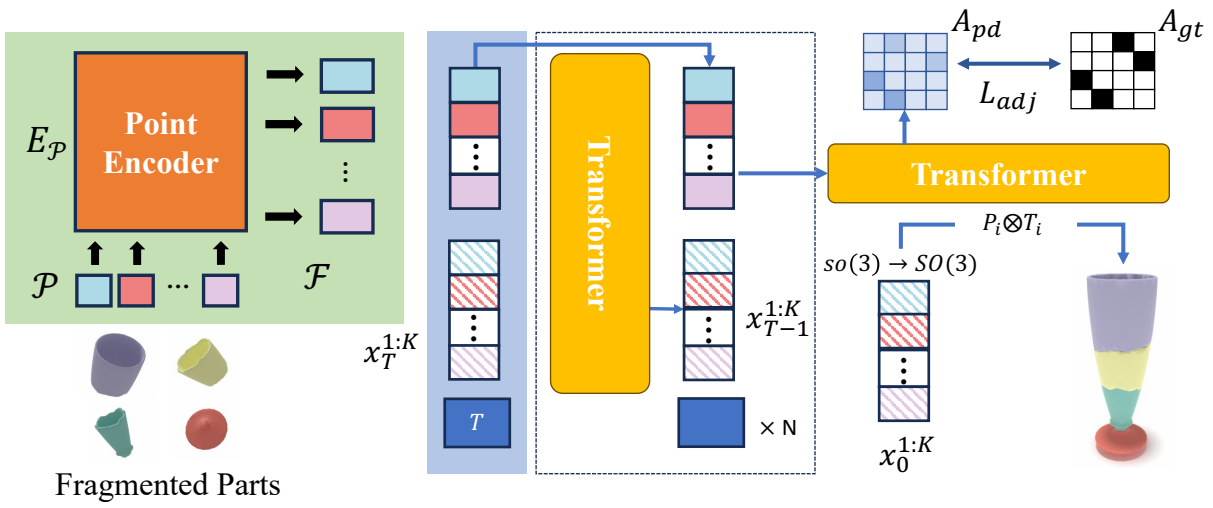

Fig. 2. The diffusion-based pipeline of FragmentDiff. Given fragments $\mathcal{P}$ of an object O, they are first embedded into latent features $\mathcal{F}$. FragmentDiff then denoises the poses $x_T^{1:K}$ conditioned on $\mathcal{F}$ while inferring the spatial relationships between fragments by referring to their adjacency matrix.

Here, β_t dictates the noise level at each timestep, with the forward process being meticulously crafted to be reversible by design.

The reverse process is the counterpart to the forward diffusion, aiming to reconstruct the original data from x_T. It is a learned Markov chain that iteratively refines the estimate x_{t-1} from the noised data x_t at each timestep t, with the neural network predicting the mean $\mu_\theta(x_t, t)$ and variance $\Sigma_\theta(x_t, t)$ of the conditional distribution $p_\theta(x_{t-1}|x_t)$. The reverse process is mathematically described as:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t),$$
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2}$$

During training, the neural network's parameters θ are optimized to effectively denoise the data by reversing the diffusion process, culminating in the generation of high-quality samples that align with the data distribution after a number of iterations T. The training is often guided by a weighted variational bound that is tailored to emphasize the reconstruction of more significant noise levels, which is crucial for the generation of high-fidelity samples.

4.2.2 Our Fragment Diffusion. We model the fractured object assembly task as a generation problem, where we aim to sample poses from pose distribution $q(x_0^{1:K})$ under the condition of fragmented parts $P_{1:K}$ using a neural network approximation $p_\theta(x_0^{1:K}|P_{1:K})$. Following DDPM (Denoising Diffusion Probabilistic Models) [Ho et al. 2020], a noising process can be defined as:

$$q(x_t^{1:K}|x_{t-1}^{1:K}) := \mathcal{N}(x_t^{1:K}; \sqrt{1-\beta_t}x_{t-1}^{1:K}, \beta_t \mathbf{I}) \tag{3}$$

where $t \in [0, T]$ is the timestep and β_t is the schedule at t .

By gradually adding Gaussian noise to $x_0^{1:K}$, the final sample $x_T^{1:K}$ looks like Gaussian noise. Note that given a timestep t, the sample $x_t^{1:K}$ could be directly computed without running the whole chain:

$$x_t^{1:K} := \sqrt{\bar{\alpha}_t}x_0^{1:K} + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{4}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $\bar{\alpha}_t := \prod_{i=0}^{t}(1-\beta_i)$.

To train the diffusion model, we approximate $q(x_{t-1}^{1:K}|x_t^{1:K})$ as network $p_\theta(x_{t-1}^{1:K}|x_t^{1:K})$. We reverse the noising process to get the noiseless sample x_0:

$$p(x_0^{1:K}|P_{1:K}) := p(x_T^{1:K}) \prod_{t=0}^{T} p_\theta(x_{t-1}^{1:K}|x_t^{1:K}, P_{1:K}) \tag{5}$$

where $p_\theta(x_{t-1}^{1:K}|x_t^{1:K}, P_{1:K})$ can be parameterized as a diagonal Gaussian distribution when the step size is small enough. Following [Ho et al. 2020; Nichol and Dhariwal 2021], we parameterize this distribution by predicting the noise ϵ, which is added to a sample $x_t^{1:K}$ and the variance $\sum$ of the distribution. We choose DDPM sampler to sample the poses of fragments. Please note that there are other choices for diffusion sampling strategy [Karras et al. 2022; Song et al. 2021a]. We choose the original DDPM sampler only to demonstrate the effectiveness of using generative models for pose estimation in the fractured object assembly task.

4.2.3 Network Architecture.

Point Encoder. We begin by individually processing the fragmented parts $\mathcal{P}$ through a shared point cloud encoder $\mathbf{E}_p$, which produces their initial latent features $\mathcal{F} = \{F_i \mid i = 1, 2, ..., K\}$. To strike a balance between performance and efficiency, we adopt the Dynamic Graph CNN (DGCNN) [Wang et al. 2019] as the backbone for extracting point cloud features. The latent vectors extracted by DGCNN have a dimension of $n \times W$, where n corresponds to the number of points per fragment and W represents the feature dimension per point. For computation efficiency, we further pool the initial features $\mathcal{F}$ into a $1 \times W$ feature $\widetilde{\mathcal{F}}$ for each fragment.

Multi-head Transformer. Encouraged by Point-E [Nichol et al. 2022], we adopt Multi-head Transformer as our diffusion network $p_\theta(x_{t-1}^{1:K}|x_t^{1:K}, P_{1:K})$. In particular, we convert the rotation matrix $\widehat{R} \in \mathrm{SO}(3)$ of each fragment into $\widehat{r} \in \mathrm{so}(3)$, which represents 3D

ALGORITHM 1: Training Procedure of FragmentDiff

input: fracture $P_{1:K}$, poses $x_0^{1:K}$, diffusion model $\mathcal{G}$, and Point encoder $\mathcal{E}$.
output: L_{hybrid}.
$x_t^{1:K}$ shape: (6,K)
sample $\epsilon \sim \mathcal{N}(0, I)$.
$x_t^{1:K} = \sqrt{\bar{\alpha}_t} x_0^{1:K} + \sqrt{1-\bar{\alpha}_t}\epsilon$
$\epsilon_\theta, \Sigma_\theta \leftarrow \mathcal{G}(x_t, \mathcal{E}(F_i), t)$
compute L_{hybrid} according to Equation 6 .
return L_{hybrid}

rotations based on Lie algebra and can be directly outputted by a neural network. Together with translation $\widehat{t} \in \mathbb{R}^3$, we obtain a tensor of shape $K \times 6$ for the fragment poses, where K is the number of fragments. All translations and rotations are normalized by the mean and variance of the whole dateset. The transformer takes the timestep t, individual fragment features $\widetilde{\mathcal{F}}_{1:K}$, and noised poses $x_t^{1:K}$ as inputs to predict ϵ_θ and Σ_θ. More specifically, our diffusion transformer is designed based on the standard self-cross attention mechanism, with 16 multi-heads of eight layers, and a width of 1024.

Adjacency Matrix. We have designed an adjacency matrix module that takes the $K \times W$ point cloud features $\widetilde{\mathcal{F}}$ outputted by the point encoder and feeds them into a simple transformer module to predict a $K \times K$ adjacency matrix. Subsequently, we train the adjacency matrix module by minimizing the differences between the predicted adjacency matrices and those ground truth ones (see next subsection). With the help of this module, our model can better learn the spatial relations between fragments, thus benefiting pose estimation.

4.2.4 Optimization Objectives. Following [Nichol and Dhariwal 2021], we utilize a hybrid loss consisting of a denoising loss and a variation lower bound to train the transformer-based diffusion model:

$$L_{hybrid} = E_{t, x_0^{1:K}, \epsilon}[\|\epsilon - \epsilon_\theta(x_t^{1:K}, t, \widetilde{\mathcal{F}}_{1:K})\|] + \lambda L_{vlb} \tag{6}$$

where the variation lower bound L_{vlb} is computed as follows:

$$L_{vlb} = \sum_{i=0}^{T} L_i \tag{7}$$

$$L_0 = -\log p_\theta(x_0^{1:K} | x_1^{1:K}) \tag{8}$$

$$L_{t-1} = D_{KL}(q(x_{t-1}^{1:K} | x_t^{1:K}, x_0^{1:K}) || p_\theta(x_{t-1}^{1:K} | x_t^{1:K})) \tag{9}$$

$$L_T = D_{KL}(q(x_T^{1:K} | x_0^{1:K}) || p(x_T^{1:K})) \tag{10}$$

Note that L_T does not depend on θ. It is close to zero if the forward noising process fully destroys the data distribution. The above denoising process is briefly depicted in Algorithm 1

Adjacency Loss. This loss function aims to quantify the difference between the predicted adjacency matrix A_{pd} and the ground truth adjacency matrix A_{gt}, which is measured based on the binary cross entropy suitable for binary distributions:

$$L_{adj}(A_{pd}, A_{gt}) = -\sum_{i=1}^{K}\sum_{j=1}^{K}[A_{gt}^{ij} \log(A_{pd}^{ij}) + (1 - A_{gt}^{ij}) \log(1 - A_{pd}^{ij})] \tag{11}$$

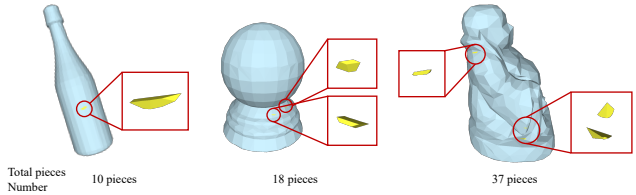

Fig. 3. Example of tiny fragments generated using the Breaking Good methodology [Sellán et al. 2023]. It is observed that the frequency of such diminutive pieces increases with the number of fragments.

where $A_{pd} \in \{0,1\}^{K \times K}$ is the predicted adjacency matrix, K is the number of fragments, $A_{gt} \in \{0,1\}^{N \times N}$ is the ground truth adjacency matrix, A_{pd}^{ij} and A_{gt}^{ij} are elements from the predicted and ground truth adjacency matrices, respectively.

5 Experiments

5.1 Data Preparation

To facilitate model training and evaluation, we introduce a new dataset tailored to the proposed fractured object assembly task based on the Breaking Bad benchmark for fractured object reassembly [Sellán et al. 2022]. Following other baselines, we primarily utilize the "everyday" subset of the Breaking Bad dataset for training, as the object categories represented in this subset are everyday objects commonly encountered in fractured object assembly scenarios. Each object category in our dataset contains multiple instances that vary in scale and pose.

To obtain the fractured objects, we utilized the scripts provided by [Sellán et al. 2022] to extract the fractured objects as original mesh files for further processing. The Breaking Bad dataset offers various shape categories, featuring 543 model instances and over 54,000 fractured patterns with different numbers in the 'everyday' subset. To ensure the dataset's suitability and avoid extreme cases, such as tiny pieces that are challenging to sample or highly sensitive to noise or point perturbation, we pass fractured objects through a filter based on the fragment volume as follows.

If one fragment is smaller than 0.1% of the volume of the complete object, it will not be considered for the reassembly. Such tiny fragments are commonly seen in the Breaking Bad dataset (see Fig. 3), and their poses are hard to predict due to the following reasons. First, tiny pieces usually lack useful information for alignment given very limited and often featureless overlapping areas with other parts. Second, due to the extremely small volume, the point cloud sampled from such a piece would be unexpectedly dense (highly different from those sampled from a regular fragment), which makes the full model hard to converge and even suffer from gradient explosion. Third, it follows the common practices where large fragments are prioritized for joint reassembly followed by embedding tiny pieces separately.

To convert the data from surface meshes to more general point clouds, we initially performed dense sampling to generate 10,000 points from each fragment. Subsequently, we downsampled the point cloud to $n = 1,024$ points. For each group of fragments, we

Table 1. Quantitative results on the *everyday* subset and the unseen *artifact* subset. **Bold** numbers indicate the best and underlined numbers indicate the second best. M(R) and M(T) denote rotation and translation errors measured by M, respectively. Ours-XM denotes our model trained with different model size. ↓ /↑: Lower / higher is better.

Method	Test on 'Everyday'					Test on 'Artifact'		
	RMSE(R) degree↓	MAE(R)↓	RMSE(T) $\times 10^{-2}$↓	MAE(T)↓	PA ↑	RMSE(R) degree↓	RMSE(T) $\times 10^{-2}$↓	PA ↑
LSTM [Wu et al. 2020]	87.420	118.077	16.71	12.781	0.201	88.284	18.928	0.182
DGL [Zhan et al. 2020]	82.708	90.812	15.630	11.642	0.261	85.970	17.812	0.209
Global [Li et al. 2020]	77.632	91.150	14.263	11.334	0.272	80.644	16.554	0.221
NSM [Chen et al. 2022]	86.758	95.30	15.890	12.464	0.160	88.812	17.031	0.148
Jigsaw [Lu et al. 2023]	38.174	32.329	10.688	8.269	0.640	42.315	12.981	0.547
SE(3)-Equiv [Wu et al. 2023]	75.920	82.230	15.088	10.734	0.268	78.151	17.882	0.232
DiffAssemble [Scarpellini et al. 2024]	73.201	80.529	14.722	10.064	0.301	77.181	18.103	0.253
Ours-2M	25.066	22.824	8.692	7.294	0.752	30.219	10.933	0.704
Ours-40M	**13.682**	**11.510**	**7.411**	**5.837**	**0.902**	**18.182**	**8.124**	**0.823**

normalized their scales by dividing them by the length of the longest bounding box diagonal. During the training phase, we randomly sampled rotation matrices and translation vectors to generate diverse poses for each fragment. The dataset was split at the object instance level into training, validation, and test sets with a 60/20/20 scheme. All the baselines and our methods are trained and tested in the same setting.

5.2 Baseline Methods

Traditional unsupervised methods are challenging to compare with due to the complexity of the problem and the lack of public implementations. On the other hand, given the emerging nature of the learning-based fractured object assembly task, a limited number of baseline approaches are available for comparison. Based on an extensive review, we have carefully selected the following baseline methods for comparative evaluation: DGL [Zhan et al. 2020], LSTM [Wu et al. 2020], Global [Li et al. 2020], NSM [Chen et al. 2022], Jigsaw [Lu et al. 2023], SE(3)-Equiv [Wu et al. 2023], and DiffAssemble [Scarpellini et al. 2024]. The selected baseline methods well represent the state-of-the-art research in fractured object assembly, enabling us to effectively evaluate the proposed framework's performance. Regarding the implementation of baseline methods: DGL [Zhan et al. 2020], LSTM [Wu et al. 2020] and Global [Li et al. 2020] are following the benchmark [Sellán et al. 2022]. Jigsaw [Lu et al. 2023], SE(3)-Equiv [Wu et al. 2023], and DiffAssemble [Scarpellini et al. 2024] are based on their official open-source codes. Note that the official implementation of NSM [Chen et al. 2022] only applies to pairwise assembly. We adopt the implementation from [Wu et al. 2023] which supports multiple-part NSM.

5.3 Evaluation Metric

In line with previous research efforts in the field, we adopt a standardized evaluation metric for the 3D shape assembly task, as employed in prior works such as [Chen et al. 2022; Li et al. 2020; Lu et al. 2023; Sellán et al. 2022]. Our evaluation metric encompasses rotation difference, translation difference, and part accuracy. Rotation and translation differences are measured using mean absolute error (MAE) and root mean square error (RMSE) metrics. Note that the rotation difference is calculated under degree format. As proposed in [Zhan et al. 2020], part accuracy (PA) evaluates the ratio of perfectly assembled pieces in the obtained result compared to the ground truth. It is calculated based on the average Chamfer distance between the source and target point clouds, with a threshold of 0.01.

5.4 Implementation Details

Our model was implemented using Pytorch and trained using a batch size of 64 with 150K iterations in about 54 hours on a Linux server, which is equipped with an Intel Xeon Sliver CPU, eight 4090 RTX GPU and 24GB memory. In our standard 40M network configuration, the width W of the point cloud encoder $\mathbf{E}_p$ is set to 128. The Multi-head Transformer within the diffusion network consists of 8 multi-heads, each with a width of 512 and spanning 12 layers. We used the Adam optimizer [Kingma and Ba 2015] with a learning rate of 5×10^{-5}. We applied the warmup and cosine annealing strategy to adjust the learning rate. In terms of inference time, our model typically takes 15 seconds for sampling (depending on sample times) during the reverse diffusion process. Due to the limitation of computational resources, we keep the same 150K iterations but use different numbers and types of GPUs for other baseline methods.

5.5 Comparison

Overall Performance. Table 1 presents a comprehensive quantitative comparison of our proposed method with other baseline approaches on the *everyday* subset. Please refer to Figs. 7 and 8 for detailed visual comparisons. As we can see, our method demonstrated superior performance compared to other techniques across all evaluation metrics. Ours obtained an average rotation error of 11.51°, which signifies a substantial improvement over the current state-of-the-art Jigsaw [Lu et al. 2023], which had an error of 32.33°. This translates to a remarkable decrease of approximately 64%. Moreover, our translation error is superior with an average of 5.84×10^{-2}, outperforming the best baseline approach which reported an error of 8.27×10^{-2}. Regarding part accuracy, our methods have achieved an

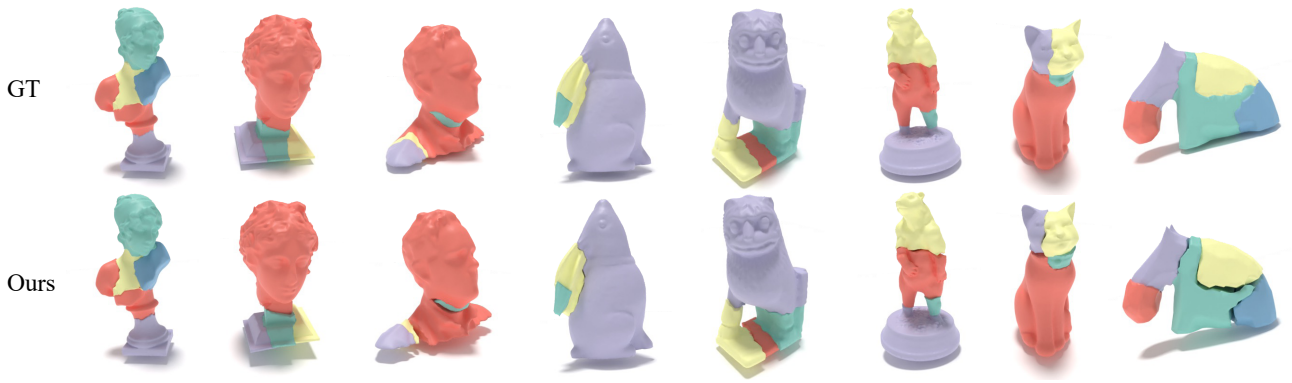

Fig. 4. Qualitative results on the Artifact subset. Our method is generalizable to unseen categories.

impressive 90.2% success rate in recovering the shape to its canonical pose. Additionally, our model exhibits generalization capabilities to unseen categories in the Artifact subset (see Fig. 4 for visual results and Table 1 for quantitative comparisons).

Our outstanding results are mainly attributed to two reasons: the strong ability of the transformer model to learn the global relationship between poses and point cloud features, and the formulation of the fractured object assembly task as a diffusion-based generation problem. The transformer model outperforms others in many learning-based tasks, as well as shape assembly which has been shown in [Chen et al. 2022]. Compared with regression-based methods, formulating the task as a pose generation problem conditioned on fragments can not only establish the mapping from fragments to poses, but also allow an effective learning of the pose prior.

On the other hand, although SE(3)-Equiv [Wu et al. 2023] demonstrated strong performance in the two-part assembly task in the original paper, we observed that the transformer-based SE(3)-Equiv model struggles with the multi-part assembly task in our experiments, exhibiting similar difficulties to NSM [Chen et al. 2022], despite the two sharing a comparable network architecture. Conversely, while DiffAssemble [Scarpellini et al. 2024] faces challenges in accurately handling rotations, it compensates for this with a trade-off in translation accuracy, thereby maintaining its competitiveness against Jigsaw [Lu et al. 2023]. Although Jigsaw [Lu et al. 2023] utilizes a neural network, it also incorporates a global fracture alignment phase as a post-processing step, which contributes to more accurate results compared to other purely learning-based baselines.

5.6 Ablation Study

We performed ablation studies to thoroughly examine the impact of various parameters and different training and testing strategies on the overall performance of our pipeline.

Model Parameter Size. We trained another transformer model with a different amount of parameters (2M), denoted "Ours-2M" in Table 1. In the same training environment, we observed that the larger parameter size ("Ours-40M") performs better, which aligns with our intuition. Despite a slight decrease in performance, the smaller model remains competitive with the state-of-the-art. The smaller model exhibits approximately twice the error in rotation and translation compared to the larger model. However, the part accuracy decreases by approximately 15%, indicating that the model can still assemble shapes in almost all situations.

Table 2. The influence of the number of fragments in our 40M model setting.

#PN	RMSE(R) degree↓	MAE(R)↓	RMSE(T) $\times 10^{-2}$↓	MAE(T)↓	PA ↑
2	**6.588**	**5.022**	**0.689**	**0.844**	**0.979**
3	9.103	7.150	2.971	2.323	0.952
4+	25.121	20.277	10.238	7.881	0.601

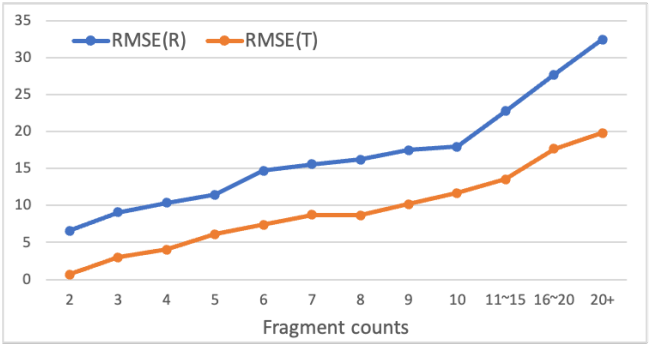

Fig. 5. Model(40M) performance with different fragment numbers.

Fragment Number. In our dataset, the average fragment number per object is 4.5818, with the maximum reaching 32. We tested our method to assemble objects with different fragment numbers. The results are shown in Table 2. As expected, the larger the number is, the more difficult the assembly behaves. This is also demonstrated in Fig. 5 where we show the rotation error RMSE(R) and translation error RMSE(T) while changing the fragment number. It can be seen that larger fragment numbers pose challenges for learning-based approaches as individual parts become 'featureless'. Although the pose errors increase (bottom row of Table 2), our performance is still much better even compared with the overall performance of

Table 3. The influence of adjacency matrix, w/o means 'without'.

#Model	RMSE(R) degree↓	MAE(R)↓	RMSE(T) $\times 10^{-2}$↓	MAE(T)↓	PA ↑
with Adj(2M)	25.066	22.824	8.692	7.294	0.752
w/o Adj(2M)	28.330	26.291	12.699	10.504	0.714
with Adj(40M)	**13.682**	**11.510**	**7.411**	**5.837**	**0.902**
w/o Adj(40M)	18.452	16.566	10.938	9.411	0.875

Table 4. The influence with missing part (MP) as input, w/o means 'without'.

#Condition	RMSE(R) degree↓	MAE(R)↓	RMSE(T) $\times 10^{-2}$↓	MAE(T)↓	PA ↑
with MP(40M)	18.920	17.112	10.599	8.378	0.764
w/o MP(40M)	**13.682**	**11.510**	**7.411**	**5.837**	**0.902**

Table 5. The influence of including the tiny parts (TP), w/o means 'without'.

#Condition	RMSE(R) degree↓	MAE(R)↓	RMSE(T) $\times 10^{-2}$↓	MAE(T)↓	PA ↑
with TP(40M)	30.891	27.702	13.544	10.934	0.727
w/o TP(40M)	**13.682**	**11.510**	**7.411**	**5.837**	**0.902**

prior methods (upper part of Table 1), demonstrating the learning ability of our diffusion and transformer based model.

Adjacency Matrix Module. We also conducted ablation studies to demonstrate that the adjacency matrix prediction module is beneficial for optimizing the pose estimation between fragments from our diffusion model as shown in Table 3.

With Missing Parts. We also evaluate the robustness of our model by introducing missing fragments. Specifically, we removed 20–30% random parts from the input within the 'everyday' subset. The performance was not affected much (see Table 4). This indicates that our model learns not only the spatial relationships between adjacent fragments but also the canonical poses of isolated fragments.

With Tiny Parts. In our dataset preparation phase, we removed the tiny parts to enhance data quality for stable training and faster convergence. We also trained and tested our model on the 'everyday' subset, which includes these small pieces, to validate our assumption. As discussed in Section 5.1, the presence of these tiny pieces significantly affected the training phase. Uniform sampling with 1024 points in these low-volume pieces results in an extremely high point density compared to other, more regular pieces. This characteristic made the training process challenging, as it led to gradient explosions that required manual adjustments. Even after the complete training, the model's performance substantially deteriorated under this condition (see Table 5).

Fig. 6. Failure cases. Our method may fail for complex fragment shapes (right) and complex fragment patterns (middle and left).

6 Conclusion and Limitation

In this work, we propose FragmentDiff, a novel approach to tackle the fractured assembly problem by incorporating a diffusion model. Our comprehensive pipeline leverages a transformer architecture acting on point clouds to effectively correlate features and accurately predict diffusion noises, resulting in plausible fragment poses. The quantitative and qualitative evaluations have demonstrated the superior performance of our approach compared to existing methods, highlighting its potential for practical applications.

In our experiments, we also tested on the DDIM (Denoising Diffusion Implicit Models) sampler [Song et al. 2021b], but the performance decreased rapidly. Therefore, we choose DDPM as a more stable sampler. It indicates that the pose distribution is sensitive to the sampling strategy which is different from the distribution of 2D images. In the future, we would like to find a more stable parameter space for poses. We also observed some failure cases with complex fragment shapes or patterns, as shown in Fig. 6. These cases may be solved by reducing the complexity of fragmented parts using the divide and conquer algorithm, which divides tough cases into smaller subsets, each consisting of a reduced number of fragments. Or we may involve some more prior knowledge (e.g., overlaps) as guidance for the model to learn the relationships between fragments to help pose estimation. Other practical considerations such as fragment collisions could also be taken into account as training losses to enhance performance. Also, there are remaining challenges for real applications such as the lack of realistic data with erosion/abrasion effects, the assembly of multiple objects with missing and spurious parts, etc., which are worth exploring in the future.

Acknowledgments

We thank all the reviewers for their useful suggestions. This work was supported by the National Natural Science Foundation of China (62220106003), the Research Grant of Beijing Higher Institution Engineering Research Center, Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, Tsinghua University Initiative Scientific Research Program, and UKRI grant CAMERA (No. EP/T022523/1).

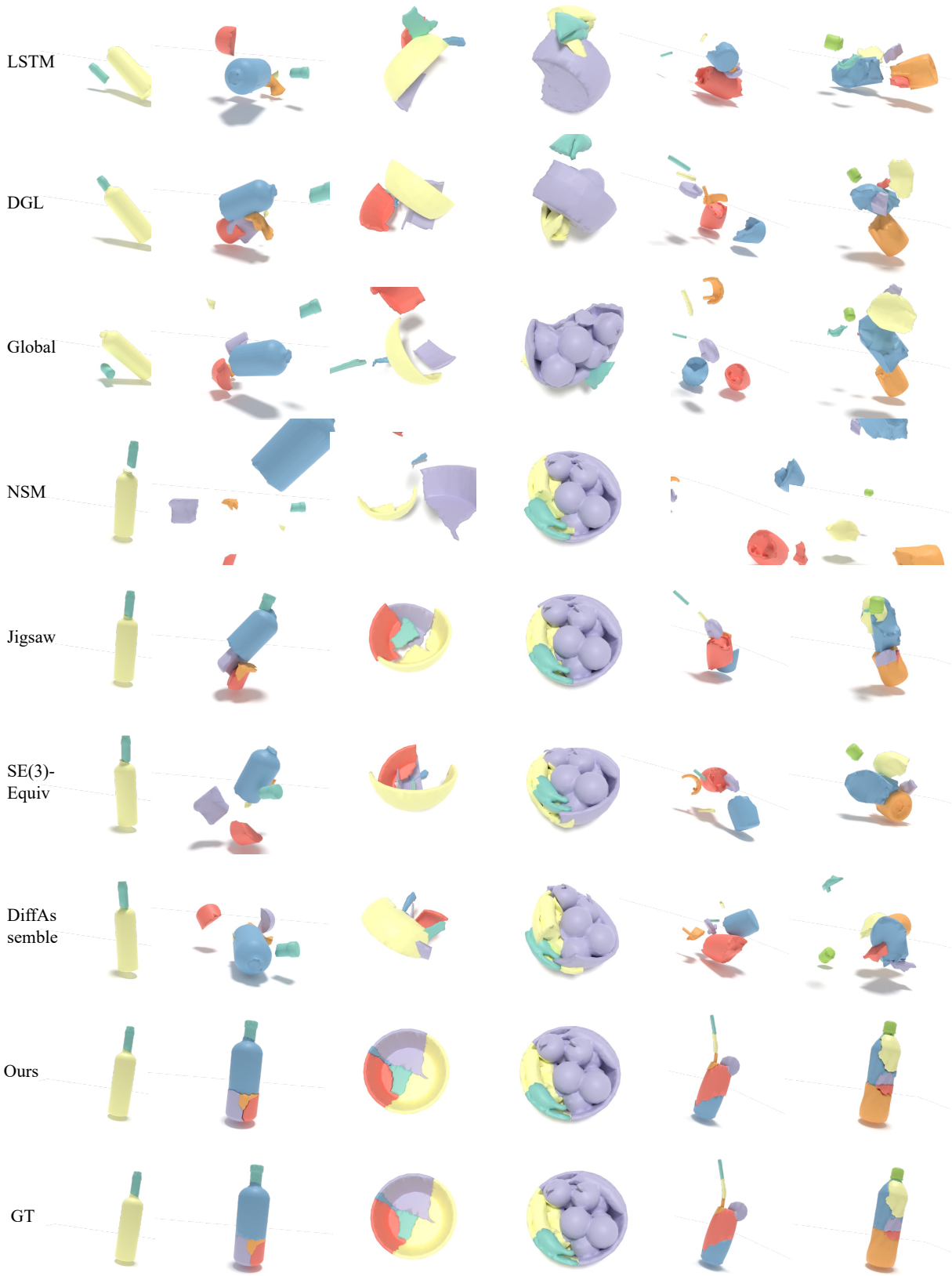

Fig. 7. Visual comparisons between LSTM, DGL, Global, NSM, Jigsaw, SE(3)-Equiv, DiffAssemble, ours, and the ground truth (GT).

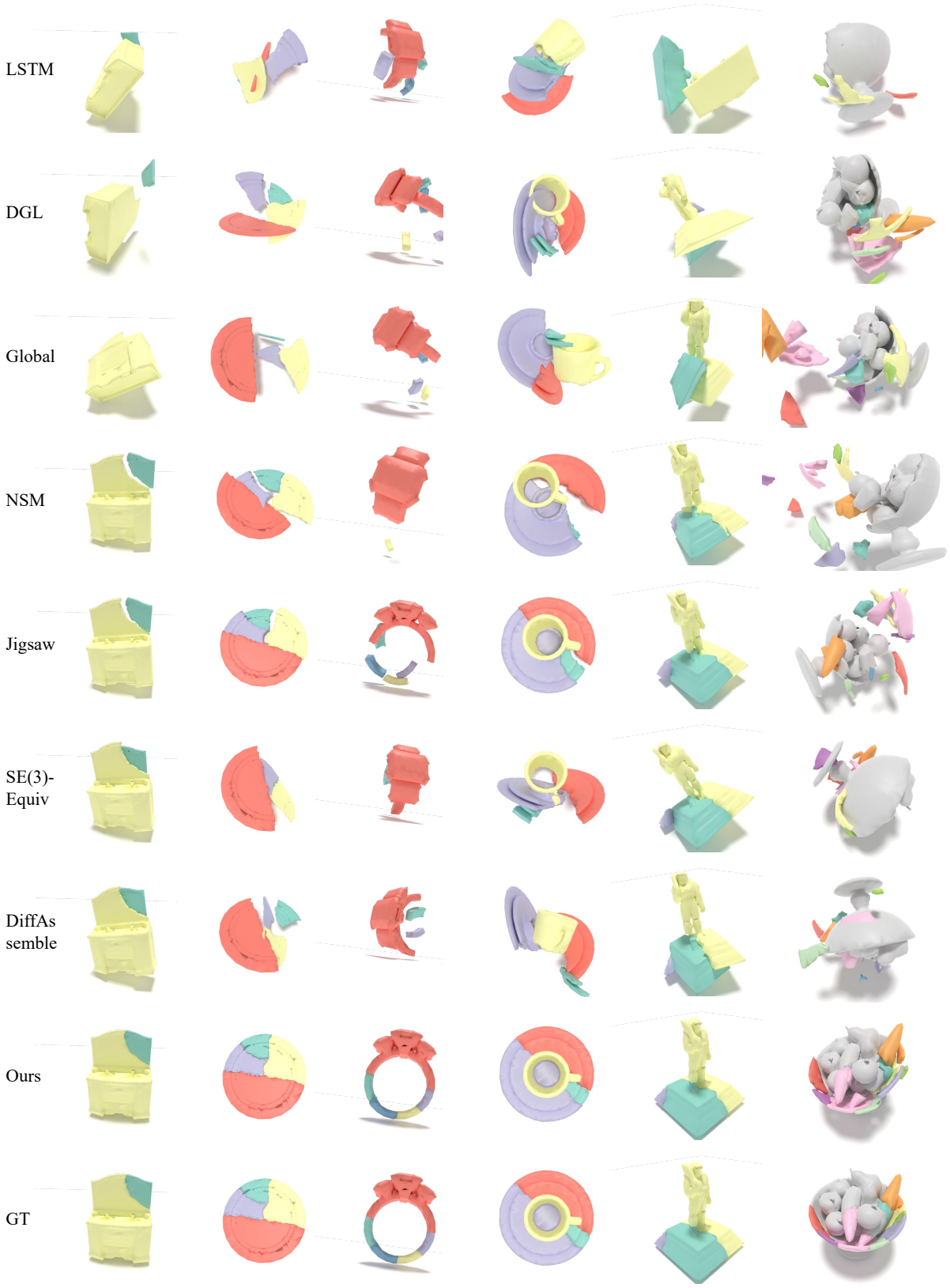

Fig. 8. Additional visual comparisons between baseline results and ours.

References

Ali Alagrami, Luca Palmieri, Sinem Aslan, Marcello Pelillo, and Sebastiano Vascon. 2023. Reassembling Broken Objects using Breaking Curves. *arXiv preprint arXiv:2306.02782* (2023).

Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. 2021. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390* (2021).

Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2022. Label-Efficient Semantic Segmentation with Diffusion Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

P.J. Besl and Neil D. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (1992), 239–256.

Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. 2017. Dsac-differentiable ransac for camera localization. In *Proc. IEEE Conf. CVPR*. 6684–6692.

Benedict J. Brown, Corey Toler-Franklin, Diego Nehab, Michael Burns, David P. Dobkin, Andreas Vlachopoulos, Christos Doumas, Szymon Rusinkiewicz, and Tim Weyrich. 2008. A system for high-volume acquisition and matching of fresco fragments: reassembling Theran wall paintings. *ACM Trans. Graph.* 27, 3 (2008), 84.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *International Conference on Machine Learning*. 1691–1703.

Y. Chen and G. Medioni. 1991. Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. 2724–2729 vol.3.

Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. 2022. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *Proc. IEEE Conf. CVPR*. 12724–12733.

Bailin Deng, Yuxin Yao, Roberto M. Dyke, and Juyong Zhang. 2022. A Survey of Non-Rigid 3D Registration. *Computer Graphics Forum* 41, 2 (2022), 559–589.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.

Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 6 (1981), 381–395.

Thomas Funkhouser, Hijung Shin, Corey Toler-Franklin, Antonio García Castañeda, Benedict Brown, David Dobkin, Szymon Rusinkiewicz, and Tim Weyrich. 2011. Learning how to match fresco fragments. *Journal on Computing and Cultural Heritage (JOCCH)* 4, 2 (2011), 1–13.

Haoxiang Guo, Shilin Liu, Hao Pan, Yang Liu, Xin Tong, and Baining Guo. 2022. ComplexGen: CAD reconstruction by B-rep chain complex generation. *ACM Trans. Graph.* 41, 4 (2022), 129:1–129:18.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

Hui Huang, Minglun Gong, Daniel Cohen-Or, Yaobin Ouyang, Fuwen Tan, and Hao Zhang. 2012. Field-guided registration for feature-conforming shape composition. *ACM Trans. Graph.* 31, 6 (2012), 179:1–179:11.

Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. 2006. Reassembling fractured objects by geometric matching. *ACM Trans. Graph.* 25, 3 (2006), 569–578.

Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. 2021. Predator: Registration of 3d point clouds with low overlap. In *Proc. IEEE Conf. CVPR*. 4267–4276.

Benjamin T. Jones, Dalton Hildreth, Duowen Chen, Ilya Baran, Vladimir G. Kim, and Adriana Schulz. 2021. AutoMate: a dataset and learning approach for automatic mating of CAD assemblies. *ACM Trans. Graph.* 40, 6 (2021), 227:1–227:18.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems* 35 (2022), 26565–26577.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

David Koller and Marc Levoy. 2006. Computer-aided reconstruction and new matches in the forma urbis romae. *Computer-aided Reconstruction and new Matches in The Forma Urbis Romae* (2006), 103–125.

Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. 2020. Learning 3d part assembly from a single image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020*. 664–682.

Jiaxin Lu, Yifan Sun, and Qixing Huang. 2023. Jigsaw: Learning to Assemble Multiple Fractured Objects. *Advances in Neural Information Processing Systems* 36 (2023).

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. 2019. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proc. IEEE Conf. CVPR*. 909–918.

Abhinav Narayan, Rajendra Nagar, and Shanmuganathan Raman. 2022. RGL-NET: A recurrent graph learning framework for progressive part assembly. In *Proc. IEEE Conf. CVPR*. 78–87.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *arXiv* abs/2212.08751 (2022).

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*. 8162–8171.

Georgios Papaioannou and Evaggelia-Aggeliki Karabassi. 2003. On the automatic assemblage of arbitrary broken solid artefacts. *Image and Vision Computing* 21, 5 (2003), 401–412.

Georgios Papaioannou, E-A Karabassi, and Theoharis Theoharis. 2001. Virtual archaeologist: Assembling the past. *IEEE Computer Graphics and Applications* 21, 2 (2001), 53–59.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proc. IEEE Conf. CVPR*. 10674–10685.

Gianluca Scarpellini, Stefano Fiorini, Francesco Giuliari, Pietro Morerio, and Alessio Del Bue. 2024. DiffAssemble: A Unified Graph-Diffusion Model for 2D and 3D Reassembly. In *Proc. IEEE Conf. CVPR*.

Silvia Sellán, Yun-Chun Chen, Ziyi Wu, Animesh Garg, and Alec Jacobson. 2022. Breaking Bad: A Dataset for Geometric Fracture and Reassembly. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Silvia Sellán, Jack Luong, Leticia Mattos Da Silva, Aravind Ramakrishnan, Yuchuan Yang, and Alec Jacobson. 2023. Breaking good: Fracture modes for realtime destruction. *ACM Transactions on Graphics* 42, 1 (2023), 1–12.

Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. 2012. Structure recovery by part assembly. *ACM Trans. Graph.* 31, 6 (2012), 180:1–180:11.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. 2256–2265.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021a. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021b. Denoising Diffusion Implicit Models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021c. Score-Based Generative Modeling through Stochastic Differential Equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Gary K.L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. 2013. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics* 19, 7 (2013), 1199–1217.

Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. 2021. Repurposing gans for one-shot semantic part segmentation. In *Proc. IEEE Conf. CVPR*. 4475–4485.

Arash Vahdat, Karsten Kreis, and Jan Kautz. 2021. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems* 34 (2021), 11287–11302.

Andrey Voynov and Artem Babenko. 2020. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*. 9786–9796.

Andrey Voynov, Stanislav Morozov, and Artem Babenko. 2021. Object segmentation without labels with large-scale generative models. In *International Conference on Machine Learning*. 10596–10606.

Yue Wang and Justin M Solomon. 2019. Deep closest point: Learning representations for point cloud registration. In *Proc. IEEE Int. Conf. Computer Vision*. 3523–3532.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* 38, 5 (2019), 146:1–146:12.

Karl DD Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G Lambourne, Armando Solar-Lezama, et al. 2022. Joinable: Learning bottom-up assembly of parametric cad joints. In *Proc. IEEE Conf. CVPR*. 15849–15860.

Junde Wu, Wei Ji, Huazhu Fu, Min Xu, Yueming Jin, and Yanwu Xu. 2024. Medsegdiff-v2: Diffusion-based medical image segmentation with transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 6030–6038.

Ruihai Wu, Chenrui Tie, Yushi Du, Yan Zhao, and Hao Dong. 2023. Leveraging SE (3) Equivariance for Learning 3D Geometric Shape Assembly. In *Proc. IEEE Int. Conf. Computer Vision*. 14311–14320.

Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. 2020. Pq-net: A generative part seq2seq network for 3d shapes. In *Proc. IEEE Conf. CVPR*. 829–838.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proc. IEEE Conf. CVPR*. 1912–1920.

Zhaohu Xing, Liang Wan, Huazhu Fu, Guang Yang, and Lei Zhu. 2023. Diff-UNet: A Diffusion Embedded Network for Volumetric Segmentation. *arXiv preprint arXiv:2303.10326* (2023).

Zihao Yan, Zimu Yi, Ruizhen Hu, Niloy J Mitra, Daniel Cohen-Or, and Hui Huang. 2021. Consistent two-flow network for tele-registration of point clouds. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2021), 4304–4318.

Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. 2022. CAPRI-Net: Learning Compact CAD Shapes with Adaptive Primitive Assembly. In *Proc. IEEE Conf. CVPR*. 11758–11768.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. 2022. LION: Latent point diffusion models for 3D shape generation. *arXiv preprint arXiv:2210.06978* (2022).

Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, and Hao Dong. 2020. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems* 33 (2020), 6315–6326.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. 2021. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proc. IEEE Conf. CVPR*. 10145–10155.

Linqi Zhou, Yilun Du, and Jiajun Wu. 2021. 3d shape generation and completion through point-voxel diffusion. In *Proc. IEEE Int. Conf. Computer Vision*. 5826–5835.