Score-PA: Score-based 3D Part Assembly

Junfeng Cheng[1]
junfeng.cheng20@imperial.ac.uk
Mingdong Wu[2]
wmingd@pku.edu.cn
Ruiyuan Zhang[3]
zhangruiyuan@zju.edu.cn
Guanqi Zhan[4]
guanqi@robots.ox.ac.uk
Chao Wu[5]
chao.wu@zju.edu.cn
Hao Dong[†2]
hao.dong@pku.edu.cn

[1] Department of Electrical and Electronic Engineering,
Imperial College London,
London, UK
[2] School of CS,
Peking University,
Beijing, China
[3] College of Computer Science and Technology,
Zhejiang University,
Hangzhou, China
[4] Department of Engineering Science,
University of Oxford,
Oxford, UK
[5] School of Public Affairs,
Zhejiang University,
Hangzhou, China

Abstract

Autonomous 3D part assembly is a challenging task in the areas of robotics and 3D computer vision. This task aims to assemble individual components into a complete shape without relying on predefined instructions. In this paper, we formulate this task from a novel generative perspective, introducing the Score-based 3D Part Assembly framework (Score-PA) for 3D part assembly. Score-based methods are typically time-consuming during the inference stage. To address this issue, we introduce a novel algorithm called the Fast Predictor-Corrector Sampler (FPC) that accelerates the sampling process within the framework. We employ various metrics to assess assembly quality and diversity, and our evaluation results demonstrate that our algorithm outperforms existing state-of-the-art approaches. We release our code at `https://github.com/J-F-Cheng/Score-PA_Score-based-3D-Part-Assembly`.

1 Introduction

Assuming you purchase a piece of IKEA furniture, assembling the separate parts into a complete structure can be challenging without proper guidance (*e.g.*, the instructions in the manual). If we were to use a robot to assist us with furniture assembly, a key issue would be enabling the robot to understand the relationships among all the parts and autonomously

† Corresponding Author

assemble them. Tackling autonomous 3D part assembly is a complex and demanding task in the fields of robotics and 3D computer vision. This challenge arises due to the requirement for algorithms to explore and navigate through a large pose space for each input part, identifying the correct orientation and position for assembly while also accounting for various constraints and dependencies between the components.

Recent years have witnessed significant explorations in the field of 3D part assembly. Dynamic Graph Learning (DGL) [11] is a representative algorithm in this area, featuring an iterative graph neural network designed for the task. This network can predict a 6-DoF part pose for each input part and transform input part point clouds to assemble the shape. RGL-NET [10], another notable algorithm for 3D part assembly, leverages the order information of input point clouds to enhance assembly capabilities.

According to Huang et al., the 3D part assembly task faces two primary challenges: assembly quality and assembly diversity. The former necessitates that the designed algorithm accurately assemble parts into a complete shape, while the latter requires the algorithm to produce a range of reasonable assembly outcomes. Existing methods, which focus on minimizing the distance between the network-predicted pose and the ground truth pose, struggle to generate diverse results. To address this, we recast the problem as a generative task. Mathematically, we learn a conditional probability $p(\mathbf{Q}_\mathbf{P} \mid \mathbf{P})$ to generate new poses, where $\mathbf{Q}_\mathbf{P}$ represents the pose set for the input part set $\mathbf{P}$. We propose the Score-based 3D Part Assembly framework (Score-PA), which can learn the proposed conditional distribution. Our experiments demonstrate that our framework attains superior quality and greater diversity in results compared to other baselines. However, the inference stage of a score-based model is typically time-consuming. We thus propose the Fast Predictor-Corrector Sampler (FPC) to accelerate the inference stage of our framework.

Following Huang et al. [11], we employ Shape Chamfer Distance (SCD) [11], Part Accuracy (PA) [15], and Connectivity Accuracy (CA) [11] to assess the quality of the generated results. However, Huang et al. only provide a qualitative evaluation of algorithm diversity and do not use any quantitative metrics for this aspect. To address this gap, we introduce Quality-Diversity Score (QDS) and Weighted Quality-Diversity Score (WQDS) to compare our algorithm's diversity with that of other baselines. Both qualitative and quantitative results demonstrate that our proposed method significantly surpasses previous algorithms in terms of diversity while achieving state-of-the-art performance for quality.

We summarise our contribution as follows:

- We approach the autonomous 3D part assembly task from a novel generative perspective, proposing the Score-based 3D Part Assembly framework (Score-PA). This framework learns a conditional probability for the pose set of input parts, resulting in high-quality and diverse results.
- We propose a new sampler called FPC to speed up the convergence of our Score-based 3D Part Assembly framework.
- We also propose two new metrics to quantitatively evaluate the assembly diversity.

2 Related Work

2.1 Assembly-based 3D modeling

Although our work does not focus on 3D modelling, it is necessary to review assembly-based 3D modelling techniques, as they have already proposed assembly methods for creating new

3D models. Assembly-based 3D modelling is a promising approach to expanding the accessibility of 3D modelling. In assembly-based modelling, new models are constructed from shape components extracted from a database [5]. One approach, proposed by Funkhouser et al. [8], creates new 3D shapes by assembling parts from a repository. Other works [5, 13, 14] facilitate shape modelling using probabilistic models to encode semantic and geometric relationships among shape components. More recently, some studies [15, 20, 31] generate certain parts and then predict a per-part transformation for the generated parts to obtain a new shape.

2.2 Autonomous 3D Part Assembly

The autonomous 3D part assembly task, as introduced by Huang et al. [11], focuses on predicting a 6-DoF pose (comprising rotation and translation) for the composition of individual parts. To accomplish this, Huang et al. [11] proposed an assembly-oriented dynamic graph learning framework that demonstrated impressive performance using their specially designed algorithm. Following this development, a recurrent graph learning framework was introduced [10], which takes advantage of the order information of parts during the assembly process. This approach highlights the significant improvements that can be achieved by incorporating order information. However, in practical applications, such order information is often not readily available for 3D part assembly problems. As a result, our study continues to adhere to the setting proposed by Huang et al. [11], focusing on assembling parts in random order. This ensures that our approach remains applicable in real-world scenarios where order information might not be accessible.

2.3 Score-Based Generative Models

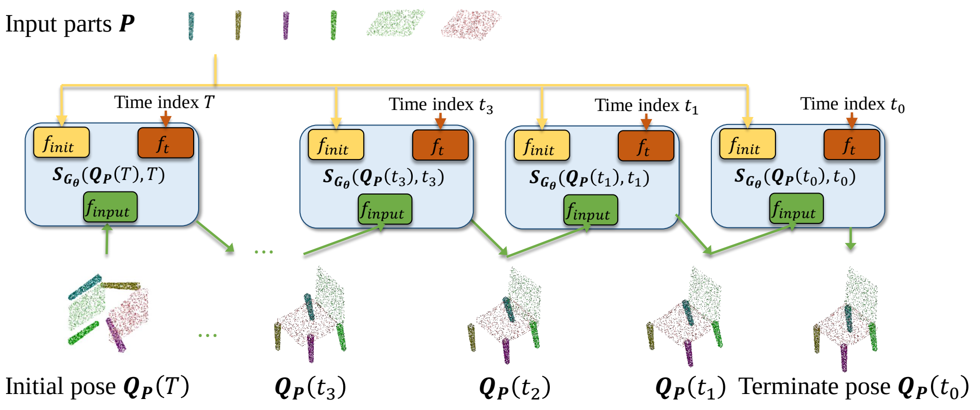
Score-based generative models aim to estimate specific distributions [12, 24, 25, 26, 27]. The primary objective is to minimize the squared distance between the estimated gradients and the gradients of the log-density of the data distribution [12, 23].

A recent variation of score-based models, known as score-based generative modelling with stochastic differential equations (SDEs) [26], employs SDEs for data perturbation, achieving remarkable success in generation tasks. This approach diffuses the data distribution during training using SDEs and generates data by reversing the diffusion process, *i.e.*, through the inverse SDE.

Score-based modelling has achieved significant success in various tasks, such as point cloud generation [3], molecular conformation generation [21], scene graph generation [28], point cloud denoising [16], human pose estimation [6], object rearrangement [30], *etc.*

3 Method

This section focuses on introducing our proposed approach to tackle the challenges discussed above. We discuss the problem definition in Section 3.1. We then overview our Score-based 3D Part Assembly framework in Section 3.2, and the training algorithm in Section 3.3. To speed up the inference stage of our framework, we further propose a new sampler, FPC, for fast sampling purpose in Section 3.4.



Input parts $\boldsymbol{P}$
Time index T
Time index t_3
Time index t_1
Time index t_0
f_{init}
f_t
f_{input}
$S_{G_\theta}(Q_P(T),T)$
$S_{G_\theta}(Q_P(t_3),t_3)$
$S_{G_\theta}(Q_P(t_1),t_1)$
$S_{G_\theta}(Q_P(t_0),t_0)$
Initial pose $\boldsymbol{Q_P}(T)$
$\boldsymbol{Q_P}(t_3)$
$\boldsymbol{Q_P}(t_2)$
$\boldsymbol{Q_P}(t_1)$
Terminate pose $\boldsymbol{Q_P}(t_0)$

Figure 1: The sampling procedure of our Score-PA. We here show a process of N steps sampling. t_n represents the time value at the step n (the algorithm starts from step $N-1$ to step 0). We can observe that the chair is assembled from coarse to fine.

3.1 Problem Definition

Assuming we have a set of separate parts, the core concept of autonomous 3D part assembly involves designing an algorithm that learns the assembly rules from a dataset and moves the separate parts to appropriate locations accordingly. Specifically, the goal of the 3D part assembly task is to predict a 6-DoF pose set $\mathbf{Q_P} = \{\mathbf{q}_i\}_{i=1}^N$ for parts transformation, corresponding to the given 3D part point clouds $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N$ (the order of the input parts is random). Each $\mathbf{p}_i \in \mathbb{R}^{1000\times 3}$ is a point cloud that conveys the geometric information of a part, and each $\mathbf{q}_i \in \mathbb{R}^6$ is a combination of a translation vector $\mathbf{t}_i \in \mathbb{R}^3$ and an Euler angle vector $\mathbf{e}_i \in \mathbb{R}^3$. By using the predicted pose set $\mathbf{Q_P} = \{\mathbf{q}_i\}_{i=1}^N$, the given point clouds $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N$ can be transformed into an assembled shape $\mathbf{P}^*$ through rotation and translation.

3.2 Overview of Score-PA

As stated in Section 1, our goal is to learn a conditional probability distribution $p(\mathbf{Q_P} \mid \mathbf{P})$. Instead of directly predicting the final part pose, we learn how to transform the input part by predicting the gradient field that can guide each part to the conditional distribution. Score-based generative model, as an emerging generative technique, provides a direct way to learn the gradient field. Our goal is to build score function to approach the conditional data distribution $\nabla_{\mathbf{Q_P}} log(p_t(\mathbf{Q_P} \mid \mathbf{P}))$. To achieve this, we design a graph neural network to model our score function $\mathbf{S}_{\mathbf{G}_\theta}(\mathbf{Q_P}, t) = \nabla_{\mathbf{Q_P}} log(p_t(\mathbf{Q_P} \mid \mathbf{P}))$ [26], where $\mathbf{G}_\theta$ is our designed graph neural network, and time t is used to index the diffusion process of 6-DoF pose $\{\mathbf{Q_P}(t)\}_{t=0}^T$. The overview of our framework is shown in Fig. 1, our objective is to train the formulated score-based model to generate the 6-DoF pose set $\mathbf{Q_P}$ based on the given 3D part point clouds information of the parts $\mathbf{P}$, and $\mathbf{P}$ can be easily transformed to the assembled shape $\mathbf{P}^*$. In the proposed algorithm, the graph $\mathbf{G}_\theta = (\mathbf{V}, \mathbf{E})$ represents the 3D geometric information of the given parts' point clouds. The nodes in this graph neural network are fully connected for message passing. The set of nodes $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^N$ is obtained via the parts' point clouds $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^N$ through a parametric function f_{init} which is designed as a vanilla PointNet [19].

It can extract the geometric information from the parts' point clouds:

$$\mathbf{V} = f_{init}(\mathbf{P}), \tag{1}$$

The *Input Encoder* f_{input} shown in Fig. 1 is parameterized as an MLP, which is used to encode the input of the designed network. The embedding layer f_t includes a *Gaussian Fourier Projection* layer and a linear layer, which enable the score-based model to be conditioned on the input time t [26, 29].

Subsequently, for training the model, as indicated by Song et al. [26], training with perturbed data can help the model better estimate the score. In our designed framework, we train our model with the data perturbed by SDEs, and under the certain condition $\mathbf{P}$, a diffusion process of 6-DoF pose $\{\mathbf{Q_P}(t)\}_{t=0}^{T}$ is constructed for this purpose, where $t \in [0, T]$. $\mathbf{Q_P}(0) \sim p_0$ represents the i.i.d. samples in the given dataset, and $\mathbf{Q_P}(T) \sim p_T$ is a prior distribution that we have already known. The diffusion process can be expressed by the following mathematical model:

$$d\mathbf{Q_P} = \mathbf{f_d}(\mathbf{Q_P}, t)dt + g(t)d\mathbf{w}, \tag{2}$$

where $\mathbf{f_d}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ represents the drift coefficient of $\mathbf{Q_P}$, and $g(t) \in \mathbb{R}$ is diffusion coefficient [26]. $\mathbf{w}$ is the standard Brownian motion [26]. More details about the training algorithm are discussed in Section 3.3.

After training, we can generate poses for the separate parts through sampling. Assume a well-trained model is obtained, and we can sample the 6-DoF pose set $\mathbf{Q_P}(0)$ by using reverse-time SDE shown as follows:

$$d\mathbf{Q_P} = [\mathbf{f_d}(\mathbf{Q_P}, t) - g^2(t)\nabla_{\mathbf{Q_P}} \log(p_t(\mathbf{Q_P} \mid \mathbf{P}))]dt + g(t)d\bar{\mathbf{w}}, \tag{3}$$

where $\bar{\mathbf{w}}$ represents a reverse time Brownian motion, and dt is an infinitesimal negative time step. In the real scenario, we normally use an iterative algorithm (*e.g.*, Predictor-Corrector sampling algorithm) to solve the inverse SDEs. The simplified sampling process is described in Fig. 1. Assuming we conduct N steps sampling, the iterative algorithm starts with the initial pose $\mathbf{Q_P}(T)$. Then the initialized value is processed by the trained score-based model to obtain $\mathbf{Q_P}(t_{N-2})$. The iterative algorithm continues until the terminate pose $\mathbf{Q_P}(t_0)$ is obtained ($t_0 = 0$), and the final value is used for translating and rotating the input parts. In our framework, we design a new algorithm, Fast Predictor-Corrector sampler (FPC), to speed up the sampling procedure. More details about our sampling algorithm are discussed in Section 3.4.

3.3 Loss Function and Training Algorithm

Our proposed algorithm aims to estimate the conditional distribution of training data $p(\mathbf{Q_P}(0) \mid \mathbf{P})$, and the original score-matching method is not suitable for this scenario since it is designed for single random variable estimation. We propose a new objective function to solve this problem. Different from the original score-matching objective function [26], our objective estimates the gradient fields of log-conditional-density $\nabla_{\mathbf{Q_P}} log(p_t(\mathbf{Q_P}(t) \mid \mathbf{P}))$. The formula is shown as follows:

$$\min_{\theta} \mathbb{E}_t \mathbb{E}_{\mathbf{Q_P}(0)|\mathbf{P}} \mathbb{E}_{\mathbf{Q_P}(t)|\mathbf{Q_P}(0),\mathbf{P}} \left[\lambda(t) \|\mathbf{S}_{\mathbf{G}_\theta}(\mathbf{Q_P}(t), t) - \nabla_{\mathbf{Q_P}(t)} \log(p_{0t}(\mathbf{Q_P}(t) \mid \mathbf{Q_P}(0), \mathbf{P}))\|_2^2\right] \tag{4}$$

We show the training procedure in Algorithm 1. In the algorithm, we set the perturbation SDE as $d\mathbf{Q_P} = \sigma^t d\mathbf{w}$, where $t \in [0,T]$. We select $\lambda(t) = \frac{1}{2\log\sigma}(\sigma^{2t}-1)$ in our experiment. Intuitively, for each iteration, we first select the training data from the training dataset and sample t from the uniform distribution. Then the ground truth pose $\mathbf{Q_P}(0)$ is perturbed by a Gaussian noise with variance $\frac{1}{2\log\sigma}(\sigma^{2t}-1)\mathbf{I}$. Finally, we calculate the objective function shown in Equation 4 and use the gradient-descent algorithm to optimize the parameter of the designed graph neural network.

Algorithm 1: Training algorithm of our Score-PA

```
Input: Training dataset $\mathcal{D}_{train}$
Parameters: $T$, $\sigma$
for $N$ epochs do
    for each $\mathbf{P}, \mathbf{I_P}, \mathbf{Q_P}(0)$ from the training dataset $\mathcal{D}_{train}$ do
        Sample $t$ from uniform distribution $\mathcal{U}(0,T)$
        $\mathbf{Q_P}(t) \leftarrow \mathbf{Q_P}(0) + \mathcal{N}(\mathbf{0}, \frac{1}{2\log\sigma}(\sigma^{2t}-1)\mathbf{I})$
        Optimize Equation 4.
    end
end
```

3.4 Fast Predictor-Corrector Sampler for Inference

Assume a well-trained model is obtained by using the training method discussed above, and we can then use this trained model to sample the poses of the separate parts. We first apply the Predictor-Corrector sampler (PC) proposed by Song et al. [26] in our framework to sample poses. However, we find that the PC sampler requires a large number of sampling steps to achieve high performance. Fig. 3 in Section 4.4 shows that the PC sampler requires 400 steps to achieve optimal sampling results. This means it brings a large latency in the inference stage. Motivated by accelerating the sampling speed, we propose Fast Predictor-Corrector sampler (FPC). Algorithm 2 shows the sampling process of our proposed algorithm. Assume we conduct N steps sampling with C_F steps final correction. From step $N-1$ to step 1, the algorithm executes the same program as the Predictor-Corrector sampler algorithm proposed by [26]. After that, the algorithm conducts C_F steps Langevin MCMC [1, 2, 4, 9] with noise decay for correction. Finally, a one-step prediction step without noise is used to obtain the results. d is a parameter which controls the decay rate of the noise. The experimental results discussed

Algorithm 2: Fast Predictor-Corrector Sampling Algorithm (FPC)

```
Input: Testing dataset $\mathcal{D}_{test}$, Graph Neural Network $\mathbf{G}_\theta$
Require: $T$, $\sigma$, $N$, $d$, $C$, $C_F$, $r$
Select $\mathbf{P}, \mathbf{I_P}$ from the testing dataset $\mathcal{D}_{test}$.
Sample $\mathbf{Q_P}(T) \sim \mathcal{N}(\mathbf{0}, \frac{1}{2\log\sigma}(\sigma^{2T}-1)\mathbf{I})$.
for $n \leftarrow N-1$ to $1$ do
    // Original Predictor-Corrector sampling algorithm
    $t_p \leftarrow \frac{(n+1)T}{N}$ $t \leftarrow \frac{nT}{N}$
    for $i \leftarrow 1$ to $C$ do
        $\mathbf{Q_P}(t_p) \leftarrow Corrector(\mathbf{Q_P}(t_p))$
    end
    $\mathbf{Q_P}(t) \leftarrow Predictor(\mathbf{Q_P}(t_p))$
end
$t_p \leftarrow \frac{T}{N}$
for $i \leftarrow 0$ to $C_F - 1$ do
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{g} \leftarrow \mathbf{S}_{\mathbf{G}_\theta}(\mathbf{Q_P}(t_p), t_p)$
    $\varepsilon \leftarrow 2(r\|\mathbf{z}\|_2/\|\mathbf{g}\|_2)$
    $\mathbf{Q_P}(t_p) \leftarrow \mathbf{Q_P}(t_p) + \varepsilon\mathbf{g} + \sqrt{2\varepsilon}(1-\frac{i}{C_F})^d\mathbf{z}$
        // Corrector with noise decay
end
$\mathbf{Q_P}(0) \leftarrow \mathbf{Q_P}(t_p) + \frac{T}{N}\sigma^{2t}\mathbf{S}_{\mathbf{G}_\theta}(\mathbf{Q_P}(t_p), t_p)$
    // Predictor without noise
return $\mathbf{Q_P}(0)$
```

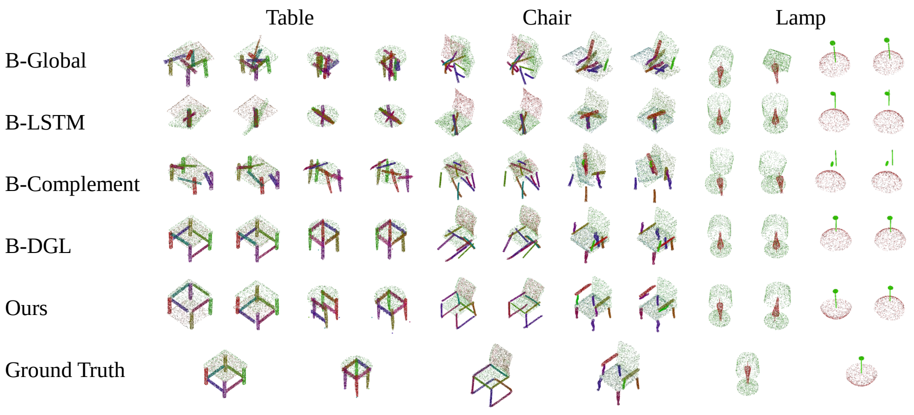


Figure 2: The qualitative comparisons between our algorithm and other baselines. For each algorithm, we show two generated assemblies. The results show that only our framework is able to generate diverse results with high quality. It is hardly possible for B-Global, B-LSTM and B-Complement to generate reasonable results. B-DGL can generate some reasonable assemblies, but the generated assemblies lack diversity. We show more results in the supplementary.

in Section 4.4 show that our algorithm generates better and more diverse results compared with that of the original Predictor-Corrector sampler.

The design rationale of the FPC We initially apply $N-1$ steps of normal PC sampling. The objective of this stage is to find the **approximate value** of a reasonable pose. At this stage, we refrain from implementing noise decay techniques as they could potentially impede both the **convergence** of the algorithm and the **diversity** of the produced results. Then, once the algorithm finds the approximate value of a reasonable pose, we utilize C_F steps of Langevin MCMC to obtain the **exact value** of the pose. In the part assembly task, even slight noise can corrupt the pose. In this case, we implement the noise decay technique at this stage to avoid corrupting the sampled pose. At this point, the noise decay no longer hinders convergence and diversity (because the algorithm has already found the approximate value), but instead enhances the quality of sampling, which brings a more accurate estimation of the pose. As a result, the algorithm can also achieve better connectivity.

4 Experiment

In this section, we discuss our datasets, baselines, evaluation metrics, experimental results and ablation study. For the experimental details, please refer to the supplementary.

4.1 Datasets and Baselines

The datasets used for experiments follow [11], which consist of three categories: chair, table and lamp (these datasets are subsets of PartNet [17], which contain a large number of fine-grained shapes and hierarchical part segmentations). Chair dataset, Table dataset and Lamp dataset contain 6,323, 8,218 and 2,207 shapes, respectively.

We compare our proposed algorithm with B-Global [11], B-LSTM [11], B-Complement [11] and Dynamic Graph Learning (B-DGL) [11]. The results show that our algorithm achieves state-of-the-art performance over other baselines.

4.2 Evaluation Metrics

In our experiments, we follow Huang et al. [11] to use Shape Chamfer Distance (SCD) [11], Part Accuracy (PA) [15] and Connectivity Accuracy (CA) [11] to evaluate the assembly quality. SCD and PA are used to evaluate the assembly quality of the whole shape and each individual part, respectively, and CA can show the quality of the connections between each pair of parts. We generate ten shapes for each set of separate parts in the evaluation procedure. Quality evaluations are based on Minimum Matching Distance [7], which means the minimum distance between the assembled shapes and the ground truth is measured for the quality evaluation.

According to Huang et al. [11], apart from assembly quality, assembly diversity is another crucial aspect of the 3D part assembly task. However, Huang et al. only provide a qualitative evaluation of the diversity of the assembly algorithm in their paper without presenting a quantitative method to assess the diversity. In the following, we introduce our proposed metrics, the Quality-Diversity Score (QDS) and the Weighted Quality-Diversity Score (WQDS).

Diversity Score (DS) [18, 22] evaluates the diversity of the results: The formula of DS is $\mathrm{DS} = \frac{1}{N^2}\sum_{i,j=1}^{N}(\mathrm{Dist}(\mathbf{P}_i^*, \mathbf{P}_j^*))$, where $\mathbf{P}_i^*$ and $\mathbf{P}_j^*$ represent any two assembled shapes. However DS only evaluates the average distances between pairs of transformed shapes but does not consider whether the shapes are reasonably assembled. Therefore, this metric is not suitable for the task of 3D part assembly. To solve this problem, we propose QDS and WQDS that can not only test the diversity among all transformed shapes but also consider the quality of these transformed shapes. The formula is shown in Equation 5 and 6. We apply SCD as the distance metric Dist in QDS and WQDS.

$$\mathrm{QDS} = \frac{1}{N^2}\sum_{i,j=1}^{N}[\mathrm{Dist}(\mathbf{P}_i^*, \mathbf{P}_j^*) \cdot \mathbb{1}(\mathrm{CA}(\mathbf{P}_i^*) > \tau_q) \cdot \mathbb{1}(\mathrm{CA}(\mathbf{P}_j^*) > \tau_q)], \tag{5}$$

$$\mathrm{WQDS} = \frac{1}{N^2}\sum_{i,j=1}^{N}[\mathrm{Dist}(\mathbf{P}_i^*, \mathbf{P}_j^*) \cdot \mathrm{CA}(\mathbf{P}_i^*) \cdot \mathrm{CA}(\mathbf{P}_j^*)], \tag{6}$$

The only difference between QDS/WQDS and DS is that we add constraints to the comparison pair. Specifically, for QDS, the constraint is given by $\mathbb{1}(\mathrm{CA}(\mathbf{P}_i^*) > \tau_q) \cdot \mathbb{1}(\mathrm{CA}(\mathbf{P}_j^*) > \tau_q)$. In the case of WQDS, the constraint takes the form $\mathrm{CA}(\mathbf{P}_i^*) \cdot \mathrm{CA}(\mathbf{P}_j^*)$. As discussed above, CA evaluates the connectivity accuracy of the algorithms. The constraints for the two metrics mean that the pair $\mathbf{P}_i^*$ and $\mathbf{P}_j^*$ contribute to the diversity value if and only if both assembled shapes have sufficiently high connectivity accuracy. In other words, both assembled shapes should have high-quality of connections between each pair of parts. These two new metrics, QDS and WQDS, can meet the requirements of our tasks, which involve evaluating the diversity between reasonably assembled pairs. In our experiments, we set $\tau_q = 0.5$ for QDS.

4.3 Compare with the Baselines

Table 1 presents the quantitative evaluation results of our algorithm in comparison to other baselines. It is clear to see that our algorithm attains the highest score in most metrics. These results indicate that our algorithm can generate diverse and high-quality assembly outcomes.

Metrics	Category	B-Global	B-LSTM	B-Complement	B-DGL	Ours
SCD $\downarrow$	Chair	0.0178	0.0230	0.0197	0.0089	**0.0071**
	Table	0.0077	0.0159	0.0116	0.0051	**0.0042**
	Lamp	0.0111	**0.0104**	0.0157	0.0105	0.0111
PA $\uparrow$	Chair	13.35	8.92	10.99	38.51	**44.51**
	Table	20.01	8.39	15.84	46.57	**52.78**
	Lamp	13.87	28.24	11.57	33.43	**34.32**
CA $\uparrow$	Chair	10.01	11.2	10.59	23.51	**30.32**
	Table	18.08	18.78	15.65	39.63	**40.59**
	Lamp	27.73	28.67	32.2	40.19	**49.07**
QDS (10^{-5}) $\uparrow$	Chair	0.152	0.036	0.086	1.688	**3.355**
	Table	0.2	0.246	0.057	3.048	**9.172**
	Lamp	0.758	0.629	2.814	1.835	**6.836**
WQDS (10^{-4}) $\uparrow$	Chair	0.188	0.074	0.207	0.553	**1.71**
	Table	0.169	0.163	0.180	0.342	**1.8**
	Lamp	0.175	0.211	1.0	0.31	**1.02**

Table 1: The quantitative comparison between our proposed algorithm and other baselines. In the testing stage, the sequence of the input parts is randomly shuffled. The results show our framework outperforms all the baselines for most metrics (only the SCD score of our framework is slightly below B-LSTM's SCD in Lamp dataset testing). In particular, the QDS of our framework outperforms other baselines by a large margin.

Qualitative results are shown in Fig. 2, which compares both the quality and diversity of our algorithm and other baselines. The results show that only our algorithm can assemble parts in a diverse way while keeping high quality. It is hardly possible for B-Global, B-LSTM, and B-Complement to generate reasonable results. B-DGL can generate some reasonable assembly results, but the generated results lack diversity. The substantial diversity offered by our method implies that Score-PA is also suitable for 3D shape design tasks.

4.4 Ablation Study

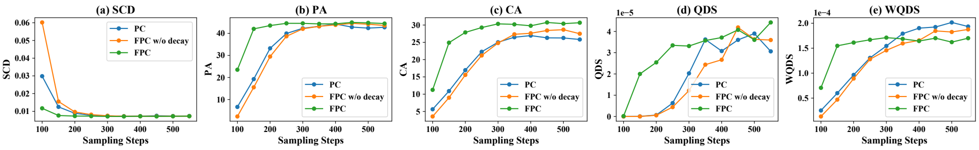


Figure 3: The quantitative results among PC, FPC without noise decay and FPC. The three samplers are tested with the sampling steps from 100 to 550. The results show that our algorithm has the fastest convergence speed.

Fig. 3 shows the ablation results among the vanilla PC sampler, the FPC sampler without noise decay and the full algorithm of the FPC sampler. The ablation experiments are conducted on the Chair dataset. We test the three samplers with different sampling steps. The results of the five metrics consistently show that the convergence speed of the FPC sampler is much faster than that of the other two algorithms. FPC sampler only requires 200 steps to achieve a relatively high score and 300 steps to achieve optimal performance. In comparison, the vanilla PC sampler and the FPC sampler without noise decay need 400 steps and 500 steps, respectively, to achieve similar results to FPC (200 steps). The detailed data can be found in Table 2. Compared with vanilla PC sampler, FPC-200, FPC-250 and

Samplers	SCD ↓	PA ↑	CA ↑	QDS (10^{-5}) ↑	WQDS (10^{-4}) ↑	Avg. Time (s) ↓
PC	0.0073	44.22	26.97	3.078	**1.902**	0.84 (×1)
FPC w/o decay	0.0072	44.13	28.65	**3.629**	1.825	0.99 (×0.85)
FPC-200	0.0073	43.52	27.88	2.545	1.612	**0.37** (×2.27)
FPC-250	0.0073	**44.53**	29.27	3.345	1.668	0.47 (×1.79)
FPC-300	**0.0071**	44.51	**30.32**	3.355	1.71	0.57 (×1.47)

Table 2: Compare FPC (200, 250 and 300 sampling steps) with the optimal PC and FPC without noise decay sampler. All the samplers are tested in the same hardware environment (R9 3900x with RTX 3090).

FPC-300 achieve ×2.27, ×1.79 and ×1.47 acceleration respectively. Besides, the comparison between FPC and FPC without noise decay proves that the technique of noise decay can indeed help accelerate the sampling algorithm.

Connectivity accuracy enhancement. Our proposed FPC algorithm also has a good performance on connectivity accuracy, which can be proved by both the results shown in Fig. 3 and the qualitative results shown in Fig. 4. Fig. 4 shows that the results sampled by PC or FPC without noise decay are easier to be disconnected (see the red circles in Fig. 4), while this does not appear in the results sampled by FPC sampler.

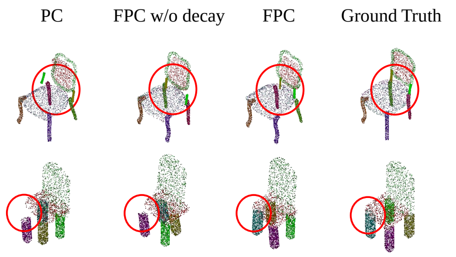


Figure 4: The qualitative ablation experiments among PC, FPC without noise decay and FPC (all the samplers are tested with their optimal sampling steps). The results sampled by FPC have the best connectivity.

5 Conclusion

In this work, we propose a novel framework, Score-PA, for the 3D part assembly task, viewing the 3D part assembly as a problem of conditional probability distribution estimation. We modify the original score-matching objective function [26] for log-conditional-density estimation purposes, and develop a graph neural network for score function modelling. Besides, we propose a new sampling method, FPC, to speed up the inference of our framework. The experiments demonstrate that our designed framework achieves the current state-of-the-art performance over other baselines for both assembly quality and assembly diversity.

Acknowledgement

This project was supported by the National Natural Science Foundation of China - General Program (62376006) and The National Youth Talent Support Program (8200800081).

References

[1] Stephen Brooks. Markov chain monte carlo method and its application. *Journal of the royal statistical society: series D (the Statistician)*, 47(1):69–100, 1998.

[2] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[3] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *European Conference on Computer Vision*, pages 364–381. Springer, 2020.

[4] Chain Monte Carlo. Markov chain monte carlo and gibbs sampling. *Lecture notes for EEB*, 581:540, 2004.

[5] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *ACM SIGGRAPH 2011 papers*, pages 1–10. 2011.

[6] Hai Ci, Mingdong Wu, Wentao Zhu, Xiaoxuan Ma, Hao Dong, Fangwei Zhong, and Yizhou Wang. Gfpose: Learning 3d human pose prior with gradient fields. *arXiv preprint arXiv:2212.08641*, 2022.

[7] Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shalah, Raphaël Groscot, and Leonidas J Guibas. Composite shape modeling via latent space factorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8140–8149, 2019.

[8] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM transactions on graphics (TOG)*, 23(3):652–663, 2004.

[9] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992.

[10] Abhinav Narayan Harish, Rajendra Nagar, and Shanmuganathan Raman. Rgl-net: A recurrent graph learning framework for progressive part assembly. *arXiv preprint arXiv:2107.12859*, 2021.

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

[12] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[13] Prakhar Jaiswal, Jinmiao Huang, and Rahul Rai. Assembly-based conceptual 3d modeling with unlabeled components using probabilistic factor graph. *Computer-Aided Design*, 74:45–54, 2016.

[14] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *Acm Transactions on Graphics (TOG)*, 31(4):1–11, 2012.

[15] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3d part assembly from a single image. In *European Conference on Computer Vision*, pages 664–682. Springer, 2020.

[16] Shitong Luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4583–4592, 2021.

[17] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019.

[18] Kaichun Mo, He Wang, Xinchen Yan, and Leonidas Guibas. Pt2pc: Learning to generate 3d point cloud shapes from part tree conditions. In *European Conference on Computer Vision*, pages 683–701. Springer, 2020.

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[20] Nadav Schor, Oren Katzir, Hao Zhang, and Daniel Cohen-Or. Componet: Learning to generate the unseen by part synthesis and composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8759–8768, 2019.

[21] Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International Conference on Machine Learning*, pages 9558–9568. PMLR, 2021.

[22] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3859–3868, 2019.

[23] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[25] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020.

[26] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[27] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021.

[28] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13936–13945, 2021.

[29] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[30] Mingdong Wu, Fangwei Zhong, Yulong Xia, and Hao Dong. TarGF: Learning target gradient field for object rearrangement. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=Euv1nXN98P3`.

[31] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838, 2020.