Generative 3D Part Assembly via Part-Whole-Hierarchy Message Passing

Bi'an Du[1], Xiang Gao[1], Wei Hu[1]*, Renjie Liao[2,3,4]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]University of British Columbia, [3]Vector Institute for AI, [4]Canada CIFAR AI Chair
pkudba@stu.pku.edu.cn, gyshgx868@pku.edu.cn, forhuwei@pku.edu.cn, rjliao@ece.ubc.ca

Abstract

Generative 3D part assembly involves understanding part relationships and predicting their 6-DoF poses for assembling a realistic 3D shape. Prior work often focus on the geometry of individual parts, neglecting part-whole hierarchies of objects. Leveraging two key observations: 1) super-part poses provide strong hints about part poses, and 2) predicting super-part poses is easier due to fewer super-parts, we propose a part-whole-hierarchy message passing network for efficient 3D part assembly. We first introduce super-parts by grouping geometrically similar parts without any semantic labels. Then we employ a part-whole hierarchical encoder, wherein a super-part encoder predicts latent super-part poses based on input parts. Subsequently, we transform the point cloud using the latent poses, feeding it to the part encoder for aggregating super-part information and reasoning about part relationships to predict all part poses. In training, only ground-truth part poses are required. During inference, the predicted latent poses of super-parts enhance interpretability. Experimental results on the PartNet dataset show that our method achieves state-of-the-art performance in part and connectivity accuracy and enables an interpretable hierarchical part assembly. Code is available at https://github.com/pkudba/3DHPA.

1. Introduction

Generative 3D Part Assembly [7, 15, 16, 25, 32] is an emerging research area that aims to generate complex 3D shapes via assembling simple 3D parts without relying on prior semantic knowledge. Different from traditional 3D shape generation, it focuses on generating diverse, plausible configurations of given parts. It facilitates the generation of complex objects and scenes with compositionality, flexibility, and efficiency. With the rapid evolution of 3D printing technology and the increasing demand for diverse 3D shapes, generative 3D part assembly finds applications in various scenarios, attracting attention from experts in computer vision, graphics, robotics, and machine learning.

However, achieving generative 3D part assembly presents a significant challenge due to the vast number of potential part arrangements and orientations, coupled with the intricate dependencies among the parts. Adding to the complexity, part geometry exhibits notable variation even within the same object category, making it exceedingly difficult to generalize learned assembly patterns across different objects.

Previous approaches primarily focus on designing architectures capable of learning powerful representations for individual 3D parts. The hope is that these learned representations could facilitate accurate part assembly, either in a one-shot manner or sequentially [14, 21, 33, 34]. However, these methods often ignore the inherent part-whole hierarchies in 3D shapes in representation learning. For instance, as illustrated in Figure 1, a chair consists of *super-parts* such as seats, backs, and legs, with each super-part further divisible into *parts* like seat surfaces and seat frames. Understanding the pose of a super-part provides insights into the poses of constituent parts within the same super-part, as they often share similar orientations or exhibit symmetry (*e.g.*, left-right symmetry in chair legs and arms). Moreover, predicting super-part poses is typically easier due to the fewer number of super-parts compared to parts. Incorporating these hierarchies into the modeling would make the learning process and potentially improve the overall performance.

In this paper, we introduce a part-whole-hierarchy message passing network for 3D shape assembly, predicting 6 degrees of freedom (6-DoF) poses for super-parts and parts in a hierarchical manner. We establish the correspondence between parts and super-parts (*i.e.*, subsets of parts) by grouping parts based on their geometric similarities, following the approach from [33, 34]. Importantly, we treat super-part poses as latent variables to be learned, thereby eliminating the requirement of ground-truth super-part poses in our work. Our model comprises two sequential modules: the super-part encoder and the part encoder.

The first super-part encoder takes the 3D point clouds of all parts as input and employs the attention-based message

*Corresponding Author: Wei Hu (forhuwei@pku.edu.cn).

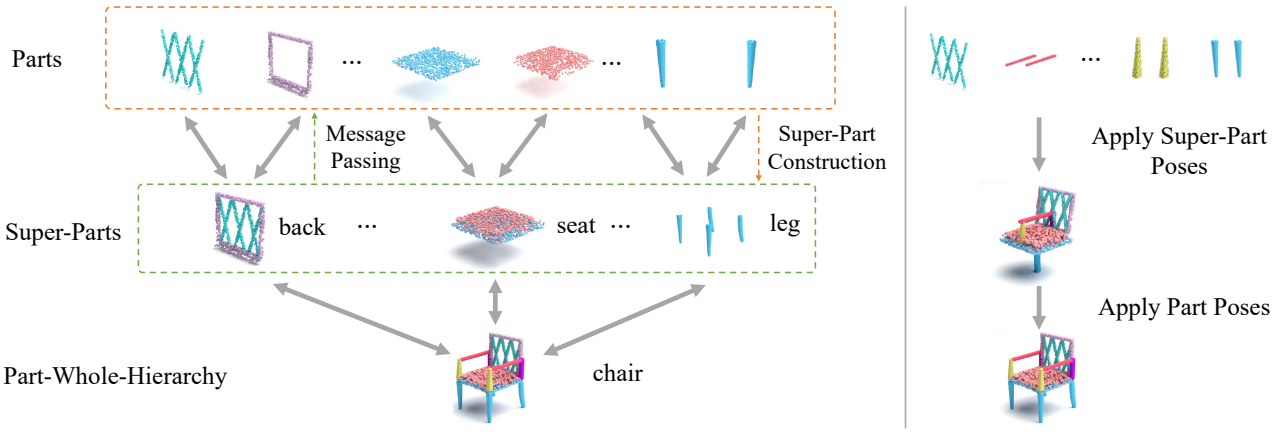

Figure 1. Left: an illustration of the part-whole-hierarchy for 3D shapes; Right: the part assembly process via the proposed part-whole-hierarchy message passing network.

passing to predict super-parts poses. The representation of each super-part is aggregated from representations of parts within it. These super-part poses provide initial estimation for the poses of individual parts.

Subsequently, the whole point clouds are transformed based on the predicted super-part poses and fed to the part encoder. This module, using a cross-attention mechanism, extracts features from the transformed point cloud, transferring super-part level information to the part level. It then leverages the attention-based message passing again to capture relationships among individual parts.

Following previous works [14, 21, 33, 34], we train and evaluate our model on the PartNet dataset [18]. Overall, our model achieves the state-of-the-art performances, outperforming the strongest competitor [34] by a significant margin, *i.e.*, with an almost 2% improvement in mean part accuracy and a 3% improvement in mean connectivity accuracy. Moreover, through visual analysis of the assembly process for both super-parts and parts, we not only showcase accurate generation of part poses but also demonstrate interpretability via the predicted super-part poses. This interpretability feature further improves the utility of our model.

2. Related Work

Assembly3D modeling Numerous prior studies have approached the challenging 3D part assembly through the joint estimation of part poses. The influential work by [7] introduces an intelligent scissoring technique to tackle this problem for part components. Subsequent research by [2, 10, 12] employs graphical models to capture the semantic and geometric relationships among shape components, enabling exploration in assembly-based shape modeling. PAGENet [14] presents a network that is aware of individual parts and generates semantic parts along with their poses. [31] proposes to decompose shapes using a library of 3D parts provided by the user. However, these studies either assumed prior knowledge of part semantics or relied on existing shape databases. In a more practical setting, the authors in [3, 21, 33, 34] focus on the pose estimation of individual parts without relying on shape databases or known semantic information. Specifically, DGL [33] employs a dynamic part graph to iteratively refine the poses of individual parts. A progressive strategy using the recurrent graph learning framework has been investigated in [21]. Additionally, the authors in [3, 34] utilizes the Transformer [28] to model the structural relationships and performs the simultaneous assembly of all parts. We follow this line of work and propose novel improvement to generate structurally-coherent part assemblies.

Structural Shape Generation In recent years, deep generative models, such as generative adversarial networks (GANs) [9] and variational autoencoders (VAEs) [5], have garnered significant attention for shape generation tasks. Notably, the work by [8] introduces a two-level variational autoencoder that simultaneously learns the overall shape structure and detailed part geometries. Both GRASS [13] and StructureNet [17] employ techniques to compress the shape structure into a latent space while considering the relationships between different parts. Additionally, [19] uses a part-tree decomposition to conditionally generate 3D shapes, and [11] adopts a procedural programmatic representation to establish connections between part cuboids. Inspired by the Seq2Seq networks in machine translation, [29] introduces a sequential encoding and decoding approach for the regression of shape parameters. While many of these approaches focus on directly generating new part shapes given random latent codes, our main focus revolves around the rigid transformation of existing parts to facilitate their assembly.

3. Part-Whole-Hierarchy Message Passing

This section provides a detailed explanation of our proposed part-whole-hierarchy message passing network. We begin by

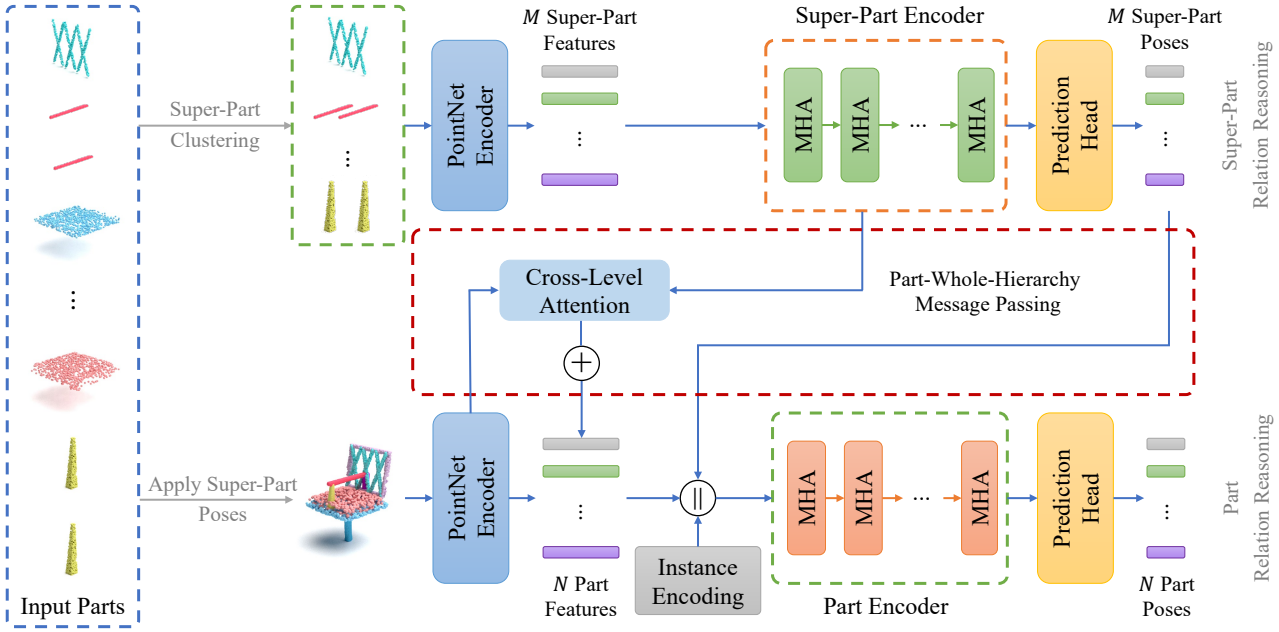

Figure 2. The overall architecture of our model consists of two modules: *super-part encoder* and *part encoder*. We first obtain super-parts via grouping parts based on their geometric similarities in an unsupervised fashion. The super-part encoder takes point cloud as input and predicts latent super-part poses (**no ground truth is needed**). The point cloud is then transformed based on super-part poses and fed to the part encoder. We incorporate both cross-level and within-level attention in the part encoder to predict part poses.

presenting how we construct super-parts in an unsupervised manner. Then we introduce our super-part encoder that predicts the latent poses of super-parts. Subsequently, we describe how our part encoder leverages the latent poses and the part-whole hierarchy to predict part poses. The overall model is illustrated in Figure 2. Finally, we introduce the loss functions and the training process for diverse generation.

3.1. Super-Part Construction

We denote the input point clouds as $\mathcal{P} = \{\mathbf{P}_i | i = 1 \dots N\}$, where $\mathbf{P}_i \in \mathbb{R}^{d \times 3}$ corresponds to the point cloud of the i-th given part of the 3D shape. The values of N and d represent the number of parts and the number of points per part respectively. Note that their values may differ from object to object. For notation convenience, we assume objects are padded to the maximum number of parts, and each part is padded to the maximum number of points per part. The goal of our task is to predict a set of 6-DoF part poses $\{(t_i, r_i)\}_{i=1}^{N}$, where $t_i \in \mathbb{R}^3$ and $r_i \in \mathbb{R}^4$ represent the translation and rigid rotation for each part, respectively. The complete part assembly for a 3D shape is $\mathcal{S} = \cup_{i=1}^{N} \mathbf{T}_i(\mathbf{P}_i)$, and $\mathbf{T}_i$ represents a transformation in $SE(3)$ that consists of a 3D rotation in the rotation group $SO(3)$ and a 3D translation in the translation group, induced by (t_i, r_i).

A super-part is a subset of parts that is ideally semantically meaningful. However, since we do not have ground-truth super-parts, we need to construct them in an unsupervised manner. In particular, we compute axis-aligned bounding boxes for each part and evaluate the similarity between these 3D boxes. Parts are grouped into the same super-part if the difference between their respective enclosing boxes is below a specified threshold. Following previous works [33, 34], we set the threshold to 0.2. Although the super-parts obtained through this method may lack semantic meaning, they provide a coarse abstraction grounded in geometry similarities without any supervised labels. Then we group the input point cloud parts $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^{N}$ into a set of M super-parts $\mathcal{P}' = \{\mathbf{P}'_i\}_{i=1}^{M}$ based on the part-whole hierarchy of a given object. Here $\mathbf{P}'_i$ represents the i-th super-part. To maintain consistent notation, we again pad the number of parts per super-part to the maximum M. Note that padding is used here for clarity and understanding. In practice, our model can handle sets of parts with varying sizes without padding.

3.2. Super-Part Relation Reasoning

To reason about the relationships among super-parts, we utilize the super-part encoder. The computational process is illustrated in the top half of Figure 2.

Given a set of point clouds of a super-part $\mathbf{P}'_i$, we first compute the set-level permutation-invariant representation $\mathbf{F}'_i$ via a shared PointNet [22]. We use the same architecture and hyperparameters as in previous works [33, 34] for a fair comparison. This feature representation captures the global characteristics of each super-part while being invariant to the order of the parts within the set. Based on the features

$\mathbf{F}'_i$ extracted for each super-part from the PointNet, we use a two-layer multi-head attention (MHA) module to learn the relationships between super-parts. Specifically, following [28], we calculate an attention matrix $\mathbf{A}'$, where $\mathbf{A}'_{i,j}$ represents the attention weight of the i-th super-part to the j-th super-part. Then we multiply each super-part feature $\mathbf{F}'_j$ by $\mathbf{A}'_{i,j}$ and sum them up to obtain the attention-weighted feature $\mathbf{G}'_i$. In our model, we set the embedding dimension to 256 and the number of heads for the MHA module to 8.

For super-part pose prediction, we feed the attention-weighted feature $\mathbf{G}'_i$ into a prediction head containing 4 fully-connected layers to obtain the pose $\{(t'_i, r'_i)\}_{i=1}^{M}$. We apply *tanh* operation to the translation vector, which restricts the range of the part center offset as $(-1, 1)$. Additionally, for simplicity, we predict the quaternion vector $r'_i = (r'_{i0}, r'_{i1}, r'_{i2}, r'_{i3})$ instead of the rotation matrix. We then use the *Rodrigues formula* [23] to obtain the rotation matrix $\mathbf{R}'_i$ corresponding to r'_i one by one as follows:

$$\mathbf{R}'_i = \begin{bmatrix} 1 - 2{r'_{i2}}^2 - 2{r'_{i3}}^2, 2r'_{i1}r'_{i2} - 2r'_{i0}r'_{i3}, 2r'_{i1}r'_{i3} + 2r'_{i0}r'_{i2} \\ 2r'_{i1}r'_{i2} + 2r'_{i0}r'_{i3}, 1 - 2{r'_{i1}}^2 - 2{r'_{i3}}^2, 2r'_{i2}r'_{i3} - 2r'_{i0}r'_{i1} \\ 2r'_{i1}r'_{i3} - 2r'_{i0}r'_{i2}, 2r'_{i2}r'_{i3} + 2r'_{i0}r'_{i1}, 1 - 2{r'_{i1}}^2 - 2{r'_{i2}}^2. \end{bmatrix}$$

The transformed point cloud of the super-part is expressed as $\mathbf{T}_i(\mathbf{P}'_i) = \mathbf{R}'_i\mathbf{P}'_i + t'_i$. To ensure that it is a unit quaternion, we normalize the rigid rotation vector such that $\|r'_i\| = 1$.

Since we do not have ground-truth super-part poses, the supervision for this module solely comes from the back-propagation signal from the subsequent module. As a result, the super-part poses are considered as latent variables.

3.3. Part Relation Reasoning

As previously mentioned, the poses of super-parts offer valuable clues about the poses of their corresponding parts. To leverage this information, we design a part encoder to incorporate the latent super-part poses to predict the part poses.

Transformation via Latent Super-Part Poses Firstly, we apply the latent super-part poses to transform all point cloud parts, resulting in transformed parts denoted as $\hat{\mathcal{P}} = \{\hat{\mathbf{P}}_i\}_{i=1}^{N}$. These transformed parts are then fed into the part encoder for further processing. Similarly to the super-part encoder, we employ another PointNet to extract part-level feature denoted as $\hat{\mathbf{F}}_i$. In order to better utilize the part-whole hierarchy of objects, we design two ways to integrate the super-part level information to the part level, *i.e.*, cross-level and within-level attention modules.

Cross-Level Attention Part representations $\hat{\mathbf{F}}_i$ from the PointNet encoder does not leverage the part-whole hierarchy. To better fuse the information from the super-parts, we propose a cross-level attention module. Specifically, we treat the part representation $\hat{\mathbf{F}}_i$ as query and super-part representations $\mathbf{F}'_i$ as keys and values. We then employ a MHA module to perform an attention-weighted aggregation of super-part representations $\mathbf{F}'_i$ to obtain the updated part representations $\mathbf{F}_i$. By doing so, the information from part-whole hierarchy is explicitly integrated into part representations. Here we specify the embedding dimension as 256 and the MHA module is set to utilize 8 heads.

Within-Level Attention To model the part relationships, we leverage another within-level attention module, which is depicted in the bottom half of Figure 2. We first concatenate the representation from the previous module F_i with an additional instance encoding vector that is unique for each part following [34]. This allows us to distinguish geometrically-equivalent parts. Then we perform message passing among parts using another MHA module where the attention computation is aware of the part-whole hierarchy. Specifically, we concatenate the latent super-part poses to the part-level feature before computing the attention scores at each layer. Consequently, in each message passing layer, a part receives pose hints from its parent super-part and updates its representation. The resulting part feature is denoted as $\mathbf{G}_i = \sum_{j=1}^{N} \mathbf{A}_{i,j}\mathbf{F}_i$, where $\mathbf{A}_{i,j}$ represents the attention weight of the i-th part to the j-th part. This design enables the part-to-part message passing to effectively leverage the part-whole hierarchy information. We set the number of heads for the MHA module to 8, the dimension of the part-level feature to 256, and concatenate it with the 40-dimensional instance encoding vector.

Prediction Head Finally, we feed the attention-weighted part-level feature $\mathbf{G}_i$ to a part pose prediction head, which consists of four fully-connected layers. The channel sizes of the first three fully-connected layers are set to 256, 256 and 1024, respectively, followed by ReLU activation functions. We use a linear projection layer to predict the part pose (t_i, r_i), encompassing a 4-dimensional rigid rotation and 3-dimensional translation. This prediction head outputs the part poses $\{(t_i, r_i)\}_{i=1}^{N}$, where t_i represents the translation vector and r_i represents the rigid rotation vector for each part. We adopt the same design as in the super-part encoder to restrict the translation vectors to the range of $(-1, 1)$ and normalize the rigid rotation vector r_i to have a unit norm.

3.4. Training Objective for Diverse Generation

Considering the same set of input point cloud parts, geometrically equivalent parts (such as legs of a chair) can be interchanged and decorated parts may have multiple placement options, resulting in multiple possible shapes. For example, a semi-cylindrical part can be placed on top of a chair backrest as a headrest, or placed under the backrest as a lumbar pillow. To account for such diversified structural variations and configurations, we introduce random noise into the system following [21, 33, 34] and employ the Min-of-N

(MoN) loss [6] to simultaneously consider assembly accuracy and assembly diversity. By considering the minimum of a set of N possible assemblies, we can encourage the model to generate diverse and valid shape configurations while maintaining overall accuracy in the assembly process. This helps address the challenge of capturing multiple plausible solutions that arise from the interchangeable and decoratable nature of the parts.

Let $\mathcal{F}(\mathcal{P}, \delta_j)$ denote our network outputs and $\mathcal{F}^*(\mathcal{P})$ denote the ground-truth point clouds, then the MoN loss is:

$$\mathcal{L}_{\text{MoN}} = \min_{\delta_j \sim \mathcal{N}(0,1)} \mathcal{L}\left(\mathcal{F}(\mathcal{P}, \delta_j), \mathcal{F}^*(\mathcal{P})\right) \tag{1}$$

where δ_j is a random noise vector drawn from standard Normal distribution in an IID fashion. Following [34], we sample 5 random vectors δ_j during training. The loss function $\mathcal{L}$ consists of the following components for both local part and global shape losses.

Firstly, we supervise the translation via an l_2 loss between our prediction t_i and the ground-truth translation vector t_i^* for each part:

$$\mathcal{L}_t = \sum_{i=1}^{N} \|t_i - t_i^*\|_2^2 \tag{2}$$

Then we compute the Chamfer distance [6] between the predicted and the ground-truth rigid rotation for each part:

$$\mathcal{L}_r = \sum_{i=1}^{N} d_c\left(r_i(\mathbf{P}_i), r_i^*(\mathbf{P}_i)\right) \tag{3}$$

The Chamfer distance $d_c(\mathcal{X}, \mathcal{Y})$ is a metric commonly used in point cloud comparison, which measures the dissimilarity between two point sets $\mathcal{X}$ and $\mathcal{Y}$ by calculating the average distance between each point in one set and its nearest neighbor in the other set as follows:

$$d_c(\mathcal{X}, \mathcal{Y}) = \sum_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \|x - y\|_2^2 + \sum_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \|x - y\|_2^2 \tag{4}$$

In this case, we use the Chamfer distance to assess the discrepancy between the predicted rotation of each part and its corresponding ground-truth rotation.

Similarly, the full shape $\mathcal{S}$, *i.e.*, the set of point clouds belonging to all parts, is supervised via the Chamfer distance from the ground-truth shape $\mathcal{S}^*$:

$$\mathcal{L}_s = d_c(\mathcal{S}, \mathcal{S}^*) \tag{5}$$

In summary, the overall loss function is defined as:

$$\mathcal{L} = \lambda_t \mathcal{L}_t + \lambda_r \mathcal{L}_r + \lambda_s \mathcal{L}_s \tag{6}$$

where λ_t, λ_r and λ_s represent the weights assigned to the three loss terms. Based on the cross validation, we empirically set $\lambda_t = 1$, $\lambda_r = 10$, and $\lambda_s = 1$ in our experiments. By minimizing this loss function, we aim to improve the alignment of the predicted poses with the ground-truth poses, thereby enhancing the overall accuracy of the assembly.

4. Experiments

In this section, we demonstrate the effectiveness of the proposed model by comparing our results with state-of-the-art methods. We also provide visual analysis of the assembly process for super-parts and parts, which helps to illustrate the hierarchical assembly process from parts to the whole.

4.1. Dataset

Following [4, 14, 21, 24, 27, 29, 33, 34], we use the PartNet [18] dataset for both training and evaluation. This dataset consists of 26,671 shapes across 24 different 3D object categories. To effectively validate and compare different methods, we select the three largest categories, *i.e.*, chairs, tables and lamps, following [4, 14, 21, 33, 34]. In total, we have 6,323 chairs, 8,218 tables, 2,207 lamps in the finest-grained level, and we adopt the official train/validation/test splits (70%/10%/20%) to conduct the experiments. For each part point cloud, 1000 points are sampled from the original part meshes using Farthest Point Sampling (FPS) [20]. All parts are transformed into its canonical space using PCA.

4.2. Evaluation Metrics

We generate a variety of shapes by adding different Gaussian noises to a given set of input parts, and find the closest shape to the ground-truth using minimum matching distance (MMD) [1]. To measure part assembly quality, we follow [21, 33, 34] to use *shape Chamfer distance* (SCD), *part accuracy* (PA) and *connectivity accuracy* (CA). *Shape Chamfer distance* is defined in Equation (4) and Equation (5). Meanwhile, to compare the diversity of assembled parts, we propose two quantitative evaluation metrics including quality-diversity score (QDS) and the weighted quality-diversity score (WQDS) following [4]. We also visualize the generation results to qualitatively evaluate the diversity. More details of these metrics are provided in the appendix.

Mean Part/Connectivity Accuracy (mPA/mCA). PA and CA depend on the Chamfer distance threshold τ_p and τ_c for judging whether the assembly is accurate. In order to provide a more comprehensive evaluation of the performance of the model, we average the results under multiple thresholds to get *mean part accuracy* (mPA) and *mean connectivity accuracy* (mCA), formally,

$$\text{mPA} = \frac{1}{T_p} \sum_{\tau_p \in T_p} \text{PA}(\tau_p), \quad \text{mCA} = \frac{1}{T_c} \sum_{\tau_c \in T_c} \text{CA}(\tau_c),$$

where T_p and T_c are set to $\{0.01, 0.02, 0.03, 0.04, 0.05\}$.

QDS. Diversity score (DS) [19, 26] evaluates the diversity of the results as $\text{DS} = \frac{1}{N^2} \sum_{i,j=1}^{N} (d_c(\mathbf{P}_i^*, \mathbf{P}_j^*))$, where $\mathbf{P}_i^*$

Methods	SCD(10^{-2}) ↓			PA(%) ↑			CA(%) ↑			QDS(10^{-5}) ↑			WQDS(10^{-5}) ↑		
	Chair	Table	Lamp	Chair	Table	Lamp	Chair	Table	Lamp	Chair	Table	Lamp	Chair	Table	Lamp
B-Global [14, 24]	1.46	1.12	0.79	15.70	15.37	22.61	9.90	33.84	18.60	0.15	0.20	0.76	1.25	1.40	0.58
B-LSTM [29]	1.31	1.25	0.77	21.77	28.64	20.78	6.80	22.56	14.05	0.04	0.27	0.63	1.07	1.43	1.54
B-Complement [27]	2.41	2.98	1.50	8.78	2.32	12.67	9.19	15.57	26.56	0.09	0.06	2.81	1.28	1.75	2.08
DGL [33]	0.91	0.50	0.93	39.00	49.51	33.33	23.87	39.96	41.70	1.69	3.05	1.84	1.35	2.97	1.73
Score [4]	0.71	0.42	1.11	44.51	52.78	34.32	30.32	40.59	49.07	3.36	9.17	6.83	1.70	3.81	2.82
RGL [21]	0.87	0.48	0.72	49.06	54.16	37.56	32.26	42.15	57.34	3.55	7.63	6.82	2.12	4.07	2.96
IET [34]	0.54	0.35	1.03	62.80	61.67	38.68	48.45	56.18	62.62	4.15	9.09	6.98	2.74	4.56	3.29
Ours	**0.51**	**0.28**	**0.70**	**64.13**	**64.83**	**38.80**	**49.28**	**58.45**	**64.16**	**5.62**	**9.58**	**7.12**	**3.06**	**4.81**	**3.90**

Table 1. Comparison between our approach and other methods under the Chamfer distance threshold 0.01.

Methods	mPA ↑			mCA ↑		
	Chair	Table	Lamp	Chair	Table	Lamp
DGL [33]	45.84	57.86	48.32	29.17	43.88	51.75
IET [34]	74.92	73.20	52.80	62.37	68.49	75.94
Ours	**76.79**	**76.19**	**53.31**	**65.32**	**70.26**	**78.15**

Table 2. Comparison between our approach and other methods under multiple Chamfer distance thresholds from 0.01 to 0.05.

and $\mathbf{P}_j^*$ represent any two assembled shapes. Based on DS, we propose the quality-diversity score (QDS) as below,

$$\text{QDS} = \frac{1}{N^2}\sum_{i,j=1}^{N} d_c(\mathbf{P}_i^*, \mathbf{P}_j^*)\mathbb{1}(\text{CA}(\mathbf{P}_i^* > \tau_q))\mathbb{1}(\text{CA}(\mathbf{P}_j^* > \tau_q)).$$

QDS imposes constraints to remove pairs that are of low assembly quality, thus assessing not only diversity but also the quality of generated shapes. Following [4], we adopt SCD as the distance metric. The value of τ_q is set to 0.5 in both QDS and WQDS.

WQDS. Based on QDS, we further propose the weighted quality-diversity score (WQDS) as below,

$$\text{WQDS} = \frac{1}{N^2}\sum_{i,j=1}^{N} d_c(\mathbf{P}_i^*, \mathbf{P}_j^*)\text{CA}(\mathbf{P}_i)\text{CA}(\mathbf{P}_j) \mathbb{1}(\text{CA}(\mathbf{P}_i^* > \tau_q))\mathbb{1}(\text{CA}(\mathbf{P}_j^* > \tau_q)).$$

WQDS weights the QDS of a pair by their connectivity accuracy, thus favoring assembled shapes that demonstrate a high-quality connection between each pair of parts.

4.3. Comparisons with State-of-the-Art

We compare our method with the state-of-the-art methods on PartNet [18] dataset. Following previous works, we first present the quantitative results under a certain Chamfer distance threshold of 0.01 in Table 1. As we can see, our method consistently outperforms all competitors. We also show the visualization of assembly results in Figure 3. It is clear that our method is better than others in terms of both the coherence of the assembled structure and the connectivity between parts, *e.g.*, the positioning of crossbeams between chair legs, and the connection between table legs and the tabletop.

To eliminate the sensitivity of performances with respect to threshold values and provide a more comprehensive evaluation, we compare the mPA and mCA of our method with two competitive baselines in Table 2. For the **Chair** category, our method demonstrates a significant improvement over IET [34], with a 1.87% increase in mPA and a 2.95% increase in mCA, while the improvement in the **Lamp** category is relatively minor. This may be attributed to the limited variation in the geometry of lamp parts, making it difficult to clearly differentiate between various levels of the part-whole hierarchy. In addition, Figure 5 illustrates the performance of various methods on the **Chair** and **Table** categories using five different Chamfer distance thresholds ranging from 0.01 to 0.05. As the Chamfer distance threshold increases, the advantage of our method becomes more pronounced, which shows the superiority of our approach. We also evaluate the diversity of generative part assembly. As shown in Table 1, our method outperforms all competing methods in both QDS and WQDS. In Figure 4, we use the model to generate multiple assembly shapes for the same set of parts (with different noises) to show the variation of generated shapes.

4.4. Human Study

Since many inaccurate results are still valid visually, we also conduct a human study to evaluate the quality of generated assemblies. Specifically, we perform an A/B test with a group of students, utilizing the identical evaluation system as in [30, 35]. Participants were presented with pairs of randomly chosen assemblies of the same object from two different methods. In total, 28 participants labeled 93 assembly pairs. We compare the proposed method with three main baselines and present the outcomes in Table 5. In over 95% of the cases, our method is favored over other three competing methods, unequivocally demonstrating the superior visual quality of our method's output.

4.5. Ablation Study

We now investigate the importance of different loss components and verify the effectiveness of the proposed super-part

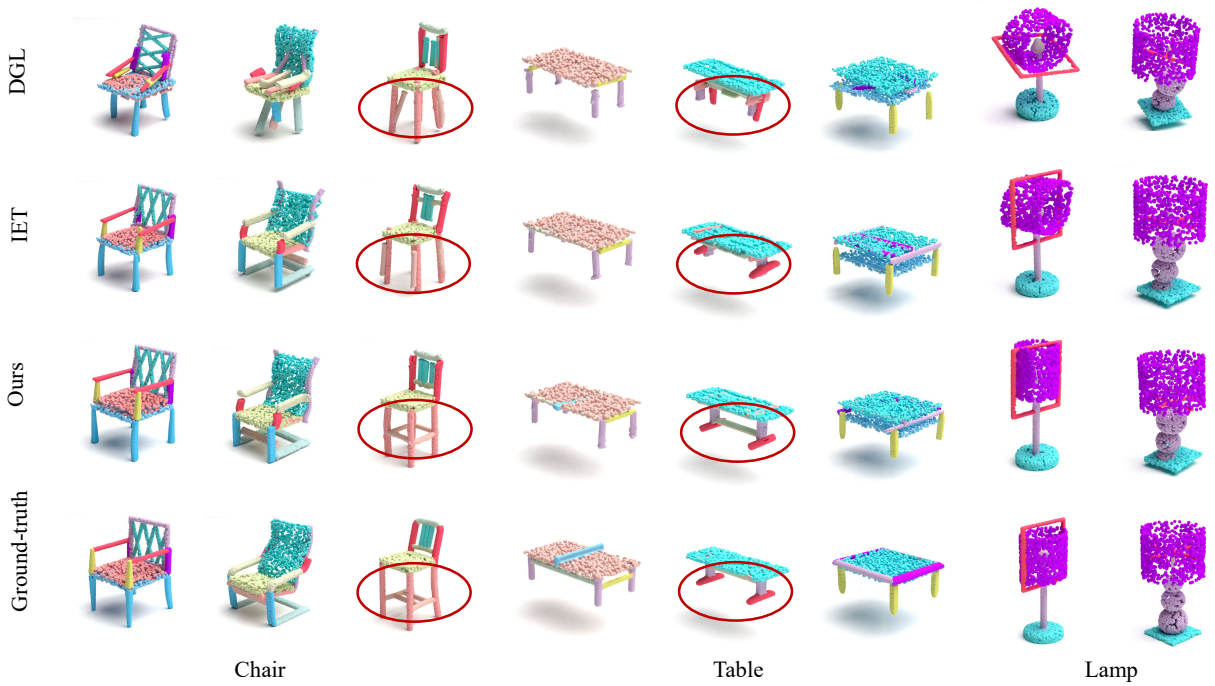

Figure 3. The qualitative comparisons between our method and two most competitive baselines on PartNet [18]. We highlight some areas where the assembly quality of ours is clearly better.

Figure 4. Diverse results on the unseen PartNet[18] test dataset generated by our network to demonstrate the structural variation in part assembly, providing different artistic results while maintaining reasonable object structures.

encoder on the Chair category.

Super-Part Encoder. Since the super-part encoder is at the core of our part-whole hierarchy message passing network, we conduct an experiment where we remove this module (denoted as **w/o Super-Part Enc.**). To mitigate the impact of the number of model parameters, we also create another baseline by increasing the number of parameters of the **w/o Super-Part Enc.** setting (denoted as **w/o Super-Part Enc. + Augmented Param.**). This involves increasing the number of parameters in the part encoder and the multi-head attention module, ensuring that the overall number of parameters is on the same order of magnitude as the full model setting. As shown in Table 3, the super-part encoder contributes significantly to the final performance. We also observe that augmenting the model parameters without incorporating the

Settings	SCD↓	PA↑	CA↑
w/o Super-Part Enc. + Augmented Param.	0.0038	60.70	53.46
w/o Super-Part Enc.	0.0036	61.04	55.39
Full Setting	**0.0028**	**64.83**	**58.45**

Table 3. Ablation studies on part-whole hierarchy on the **Table** category under a certain Chamfer distance threshold 0.01.

$\mathcal{L}_t$	$\mathcal{L}_r$	$\mathcal{L}_s$	SCD↓	PA↑	CA↑
	✓	✓	0.0075	35.36	32.73
✓		✓	0.0039	60.02	54.21
✓	✓		0.0033	62.74	56.98
✓	✓	✓	**0.0028**	**64.83**	**58.45**

Table 4. Ablation studies on loss components on the **Table** category under a certain Chamfer distance threshold 0.01.

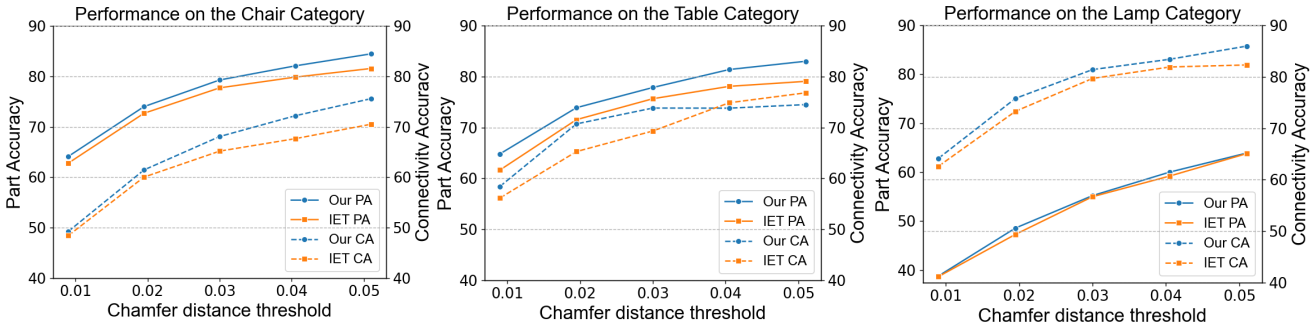

Figure 5. Performance on the **Chair**, **Table** and **Lamp** categories under multiple Chamfer distance thresholds.

Method	Percent Prefer Ours
Ours vs IET[34]	95.2%
Ours vs RGL [21]	97.6%
Ours vs DGL [33]	99.5%

Table 5. Human study results on PartNet[18].

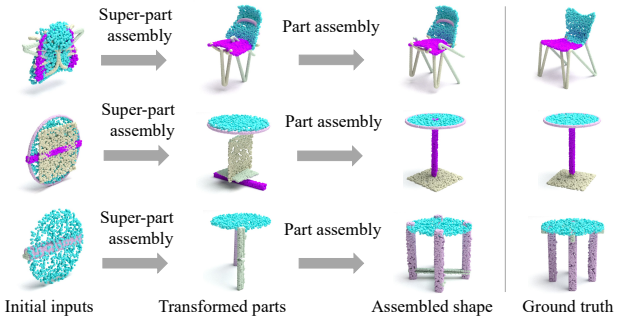

Figure 6. Visualization of our predicted hierarchical assembly process from parts to the whole.

super-part encoders actually leads to a performance drop. This finding further highlights the effectiveness of the proposed part-whole hierarchy network for part assembly.

Loss Functions. We now evaluate the importance of individual loss terms in Equation (6). As shown in Table 4, removing the part translation loss $\mathcal{L}_t$ leads to a significant performance degradation. Similarly, the remaining loss terms, including the rigid rotation loss $\mathcal{L}_r$ and the shape assembly loss $\mathcal{L}_s$, demonstrate their significance in facilitating precise part rotations and ensuring the overall coherence of the assembled parts, respectively. This shows that each loss term does play an indispensable role in achieving high-quality assemblies.

4.6. Hierarchical Part Assembly Analysis

We further provide a visual analysis of the assembly process for super-parts and parts in Figure 6. It is obvious that during the super-part assembly phase, important components such as chair seats and chair backs are often assembled correctly first. This aligns with our intuition that super-part assembly is relative easier and provides strong hints for predicting poses of parts like chair legs and arms, thus showing further evidence for the effectiveness of the part-whole-hierarchy message passing network.

5. Conclusion

In this paper, we propose the part-whole-hierarchy message passing network to address the challenging generative 3D part assembly task. We first group the point cloud of individual parts to form super-parts in an unsupervised way. Taking the point cloud as input, our super-part encoder predicts latent poses of super-parts which are used to transform the point cloud. We then feed the transformed point cloud to the part encoder. Relying on the cross-level and within-level attention based message passing, the part encoder takes the transformed point cloud as input and leverages the information from super-parts and predicts poses of parts. Experiments on the PartNet dataset demonstrate that our method achieves state-of-the-art performances as well as provides interpretable assembly process. In the future, we are interested in combining 3D part generation with our 3D part assembly model to generate 3D shapes from scratch.

Acknowledgments

This work was funded, in part, by NSERC DG Grants (No. RGPIN-2022-04636), the Vector Institute for AI, and Canada CIFAR AI Chair. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through the Digital Research Alliance of Canada alliance.can.ca, and companies sponsoring the Vector Institute www.vectorinstitute.ai/#partners, and Advanced Research Computing at the University of British Columbia. Additional hardware support was provided by John R. Evans Leaders Fund CFI grant.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 5

[2] 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. 2

[3] Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12724–12733, 2022. 2

[4] Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. Score-pa: Score-based 3d part assembly. *arXiv preprint arXiv:2309.04220*, 2023. 5, 6

[5] P Kingma Diederik, Max Welling, et al. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 2

[6] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 5

[7] 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. 1, 2

[8] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. Sdm-net: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 2

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[10] 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. 2

[11] R Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J Mitra, and Daniel Ritchie. Shapeassembly: Learning to generate programs for 3d shape structure synthesis. *ACM Transactions on Graphics (TOG)*, 39(6):1–20, 2020. 2

[12] 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. 2

[13] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 2

[14] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3d part assembly from a single image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 664–682. Springer, 2020. 1, 2, 5, 6

[15] Yuval Litvak, Armin Biess, and Aharon Bar-Hillel. Learning pose estimation for high-precision robotic assembly using simulated depth images. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3521–3527. IEEE, 2019. 1

[16] Jianlan Luo, Eugen Solowjow, Chengtao Wen, Juan Aparicio Ojea, Alice M Agogino, Aviv Tamar, and Pieter Abbeel. Reinforcement learning on variable impedance controller for high-precision robotic assembly. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3080–3087. IEEE, 2019. 1

[17] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 2

[18] 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. 2, 5, 6, 7, 8

[19] Kaichun Mo, He Wang, Xinchen Yan, and Leonidas Guibas. Pt2pc: Learning to generate 3d point cloud shapes from part tree conditions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 683–701. Springer, 2020. 2, 5

[20] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003. 5

[21] Abhinav Narayan, Rajendra Nagar, and Shanmuganathan Raman. Rgl-net: A recurrent graph learning framework for progressive part assembly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 78–87, 2022. 1, 2, 4, 5, 6, 8

[22] 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. 3

[23] Olinde Rodrigues. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de mathématiques pures et appliquées*, 5:380–440, 1840. 4

[24] 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. 5, 6

[25] Lin Shao, Toki Migimatsu, and Jeannette Bohg. Learning to scaffold the development of robotic manipulation skills. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5671–5677. IEEE, 2020. 1

[26] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree struc-

tured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3859–3868, 2019. 5

[27] Minhyuk Sung, Hao Su, Vladimir G Kim, Siddhartha Chaudhuri, and Leonidas Guibas. Complementme: Weakly-supervised component suggestions for 3d modeling. *ACM Transactions on Graphics (TOG)*, 36(6):1–12, 2017. 5, 6

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 4

[29] 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. 2, 5, 6

[30] Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Learning compact representations for lidar completion and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1074–1083, 2023. 6

[31] Xianghao Xu, Paul Guerrero, Matthew Fisher, Siddhartha Chaudhuri, and Daniel Ritchie. Unsupervised 3d shape reconstruction by part retrieval and assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8559–8567, 2023. 2

[32] Kevin Zakka, Andy Zeng, Johnny Lee, and Shuran Song. Form2fit: Learning shape priors for generalizable assembly from disassembly. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9404–9410. IEEE, 2020. 1

[33] Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, Hao Dong, et al. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems*, 33:6315–6326, 2020. 1, 2, 3, 4, 5, 6, 8

[34] Rufeng Zhang, Tao Kong, Weihao Wang, Xuan Han, and Mingyu You. 3d part assembly generation with instance encoded transformer. *IEEE Robotics and Automation Letters*, 7(4):9051–9058, 2022. 1, 2, 3, 4, 5, 6, 8

[35] Vlas Zyrianov, Xiyue Zhu, and Shenlong Wang. Learning to generate realistic lidar point clouds. In *European Conference on Computer Vision*, pages 17–35. Springer, 2022. 6