SPAFormer: Sequential 3D Part Assembly with Transformers

Boshen Xu
Renmin University of China
boshenx@ruc.edu.cn

Sipeng Zheng
BAAI
spzheng@baai.ac.cn

Qin Jin*
Renmin University of China
qjin@ruc.edu.cn

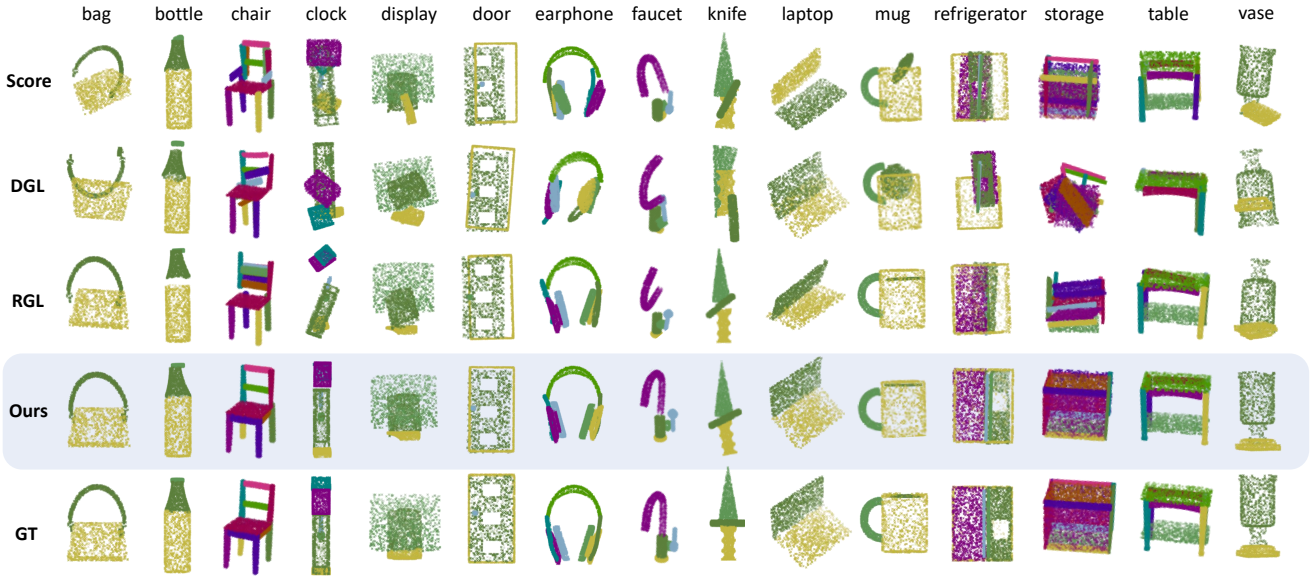


Figure 1. We propose SPAFormer, a novel sequence-conditioned transformer-based method to assemble objects from given parts with 3D point clouds. SPAFormer efficiently leverages the part geometry and sequence information, achieving significantly more plausible assemblies on our constructed benchmark PartNet-Assembly than other baseline methods including ScorePA [3], DGL [14], and RGL [35].

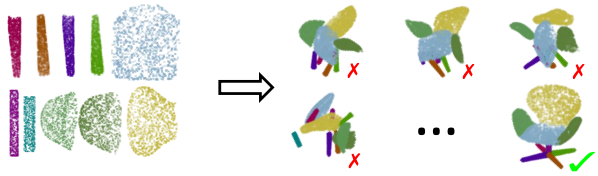
Abstract

We introduce SPAFormer, an innovative model designed to overcome the combinatorial explosion challenge in the 3D Part Assembly (3D-PA) task. This task requires accurate prediction of each part's poses in sequential steps. As the number of parts increases, the possible assembly combinations increase exponentially, leading to a combinatorial explosion that severely hinders the efficacy of 3D-PA. SPAFormer addresses this problem by leveraging weak constraints from assembly sequences, effectively reducing the solution space's complexity. Since the sequence of parts conveys construction rules similar to sentences structured through words, our model explores both parallel and autoregressive generation. We further strengthen SPAFormer through knowledge enhancement strategies that utilize the attributes of parts and their sequence information, enabling it to capture the inherent assembly pattern and relationships among sequentially ordered parts. We also construct a more challenging benchmark named PartNet-Assembly covering 21 varied categories to more comprehensively validate the effectiveness of SPAFormer. Extensive experiments demonstrate the superior generalization capabilities of SPAFormer, particularly with multi-tasking and in scenarios requiring long-horizon assembly. Code is available at https://github.com/xuboshen/SPAFormer.

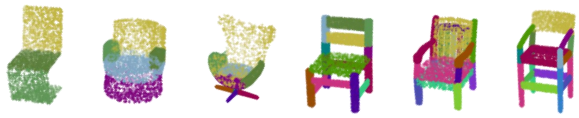
1. Introduction

There has been strong anticipation for agents to autonomously assemble parts into an intricate structural object [4, 13, 14, 19, 49]. To achieve this goal, 3D part assembly (3D-PA) is essential [14], which requires the ability of an agent to identify, reorient, and connect geometric parts in the 3D space, ultimately forming a desired object. Such capability empowers an agent to undertake a variety of ap-

*Qin Jin is the corresponding author.



(a) 3D part assembly task



(b) Assembled objects with increasing parts

Figure 2. Illustration of the combinatorial explosion challenge inherent in the assembly process. Specifically: (a) For an object composed of n parts, where we assume each part can occupy one of m discrete positions, the potential combinations of these parts grow at an extraordinary rate, exceeding $O(m^n)$ in complexity. (b) the number of constituent parts increases when the target object for assembly becomes more complex.

plications in real-world scenarios, ranging from assembling IKEA furniture [49] to crafting designs in Computer-Aided Design (CAD) environments [4].

Due to its critical importance, the 3D-PA task has attracted increasing attention from researchers. Traditionally, many approaches [3, 14, 57] achieve target object assembly by relying solely on part-level information or basic structural knowledge (e.g., object category label) without additional constraints. We consider these approaches as condition-free approaches. However, the 3D-PA task encounters a substantial obstacle, known as "combinatorial explosion" (depicted in Figure 2), where the number of possible combinations rises beyond exponentially with increasing parts. This phenomenon poses a significant challenge and makes condition-free approaches less effective in achieving accurate part assembly, particularly as the assembly sequence lengthens.

To address the combinatorial explosion challenge, several studies [3, 25–27, 35, 58] have leveraged visual priors of the object, such as point clouds [26] or RGB images [25, 48], to inform the assembly process. These visual-conditioned methods enable the foresight of an object's final form and overall appearance to guide the assembly process. However, the acquisition of such detailed visual priors incurs potential costs, reducing the practicality of these approaches in real-world applications. As an alternative, recent initiatives explore a more balanced way by utilizing structural prior knowledge. This includes the adoption of a hierarchical structure graph [12, 17, 33, 42, 47] or employing peg-hole joints on parts [27] for guidance. While this strategy decreases the need for detailed visual priors and is generally less burdensome, the reliance on structural priors still implies an additional expense compared to condition-free approaches. Given the limitations of previous works, we propose that leveraging the assembly sequences represents a more viable solution,which is advantageous due to its weak constraints: assembly sequences are readily obtainable from diverse sources, including instructional manuals [28, 48, 49, 56], the analysis of physical disassembly process [30, 40, 43], or through the prediction of assembly sequences themselves [31, 44]. Similar to how the sequence of words in a sentence determines the meaning, the assembly order of parts conveys the rules for object construction. Consequently, recognizing patterns within these sequences can effectively benefit 3D-PA, offering an efficient method to address combinatorial explosion.

In this work, we propose Sequential 3D Part Assembly with Transformers (SPAFormer), a novel transformer-based model to address the combinatorial explosion challenge in the 3D-PA task. SPAFormer conditioned on the assembly sequence enhances the 3D geometric understanding through reasoning relations among object parts. Specifically, we explore the effectiveness of potential transformer paradigms for 3D-PA, including parallel generator and autoregressive generator. Moreover, we employ knowledge enhancement strategies to enrich the model with part attributes and assembly sequence insights, by incorporating critical features such as part symmetry (e.g., the legs of a chair), assembly order, and relative position of the parts. To comprehensively evaluate our model, we expand the scope of the traditional benchmark [14], enriching it from the limited 3 categories (chair, table, lamp) to a wider variety of 21 categories that range from structured furniture (e.g., chairs, tables) to everyday items (e.g., earphones, clocks). Our findings reveal that SPAFormer not only significantly outperforms existing methods that are condition-free or sequence-conditioned (Figure 1), but also achieves competitive performance compared to approaches employing visual priors.

Our contributions can be summarized in three-fold:

- **Innovative Framework:** We propose SPAFormer, a transformer-based model for 3D object assembly with sequential parts. Our model particularly leverages sequential part information and incorporates knowledge enhancement strategies to significantly improve the assembly performance.
- **Generalization of SPAFormer:** SPAFormer shows superior generalization in object assembly from three crucial perspectives: a) category-specific, enabling it to handle various objects in the same category; b) multi-task, showcasing its versatility across diverse object categories using the same model for the first time; and c) long-horizon, proving its ability in managing complex assembly tasks with numerous parts.
- **More Comprehensive Benchmark:** To facilitate a thorough evaluation of different models, we introduce an ex-

tensive benchmark named PartNet-Assembly, covering up to 21 object categories and providing a broad spectrum of object assembly tasks.

2. Related Works

Assembly-based 3D Modeling. The assembly-based 3D modeling involves creating structural 3D shapes by integrating information from individual parts. Prior works [1, 10, 16, 18] have explored constructing diverse 3D shapes from a given set of part candidates with precise language semantics. With the development of generative methods, some works [11, 12, 22, 33, 51] focus on leveraging generative models, such as variational autoencoders [23, 53], to generate 3D shapes. Although generative models produce aesthetic 3D shapes, they are not able to deal with assembling true objects in the real world. Many works focus on fracture object assemblies [2, 15, 21, 29, 37, 52], where the model aggregates broken pieces into an object. However, the task of restoring broken objects is not applicable to complex furniture or toy assemblies. Different from previous tasks, the autonomous 3D object assembly [14] is more beneficial for robotic assembly [4, 13, 19, 20, 55] where the parts maintain semantics like IKEA assembly.

Autonomous 3D Object Assembly. In the context of 3D object assembly, the objective is to accurately predict the translation and rotation parameters of individual parts to reconstruct the object's shape. Many studies [3, 14, 50, 57] try to tackle this task by simply utilizing the part information and category-level priors. Notably, Huang *et al*. [14] employ graph neural networks to facilitate the understanding of interrelations between parts. Additionally, by expanding data through the inclusion of fractured objects, Zhang *et al*. [58] demonstrate significant improvements in assembling chairs. Despite these advancements, the problem of combinatorial explosion persists. To mitigate this issue, various prior knowledge on objects has been provided as extra information, including ground-truth object image [25], peg-hole joints on parts [27] and object point clouds [26]. However, these constraints may be costly and less feasible in real-world applications. Instead, advancements in sequence-conditioned part assembly, such as RGL [35], have demonstrated the potential of using weakly constrained assembly sequences for exploring free-form assembly. Building upon these insights, our work introduces a transformer-based framework designed to leverage the advantages offered by assembly sequences in the 3D-PA task.

Positional Encoding in Transformers. Positional encoding facilitates transformers to interpret position information, as evidenced across a diverse range of research domains [6, 7, 9, 24, 32, 45, 54]. Traditionally, the literature consists of two primary encodings: absolute and relative. Absolute positional encoding [46] enables transformers to differentiate words through position-dependent embeddings. Nonetheless, it exhibits limitations in representing the order of words relative to one another [38]. To address this issue, relative positional encoding aims to better characterize the relationships between adjacent words. A notable implementation is the RoPE [39], which rotates the input embedding to effectively extract relative positioning. In the 3D-PA task, Li *et al*. [25] and Narayan *et al*. [35] attempt to differentiate identical parts using positional encodings, but the unexpected similar features lead to their suboptimal results. Considering these works, we propose to combine variants of positional encodings to identify assembly patterns among sequential parts.

3. Methodology

Given a set of N object parts $\mathcal{P} = \{p_i\}_{i=1}^N$, the part assembly task aims to predict their poses $\{y_i\}_{i=1}^N \subset \mathrm{SE}(3)$, where p_i denotes the 3D point cloud of the i-th part and $\mathrm{SE}(3)$ includes all translation and rotation transformations. The resulting assembled 3D shape $S = \bigcup_i^N y_i(p_i)$ is formulated by transforming each part-level point cloud p_i using its corresponding pose y_i. During the assembly process, the assembly chain $\mathcal{C} = \{(p_i, p_j) \mid p_i \preceq p_j,\ \forall i \leq j\}$ is imposed on the set $\mathcal{P}$, where $p_i \preceq p_j$ denotes that part p_i is assembled before p_j. The goal of our work is sequential part assembly, which predicts the poses $\{y_i\}_{i=1}^N$ of $\mathcal{P}$ guided by the predefined assembly chain $\mathcal{C}$.

The overall framework of our proposed SPAFormer is illustrated in Figure 3. In the following sections, we first introduce our knowledge enhancement strategies to effectively incorporate part attributes and relation information in Section 3.1. Then we discuss generator variants for better modeling part relationship in Section 3.2. These strategies include order encoding (OEnc) and relation encoding (REnc) which aim to capture assembly sequence patterns, and symmetry encoding (SEnc) which focuses on enriching part features with symmetry information. In Section 3.3, we introduce our training process.

3.1. Knowledge Enhancement for Parts

Given the point clouds of object parts constrained by assembly sequence $\mathcal{C}$, a shared 3D encoder [36] is firstly used to extract their geometric representation $\mathcal{V}_0 = \{v_i\}_{i=1}^N$, which contains the shape knowledge of each part.

Due to the data deficiency, which can lead to inferior results when using the data-hungry transformers, we employ several strategies to enhance the model's understanding of parts. These strategies include order encoding (OEnc) $h(\cdot)$, relation encoding (REnc) $f_{\{q,k\}}$ to enhance the model's understanding of neighbor parts, and symmetry encoding (SEnc) $g(\cdot)$ for modeling parts attributes.

Order Encoding. To leverage the assembly pattern in part sequence, order encoding reveals the order information to

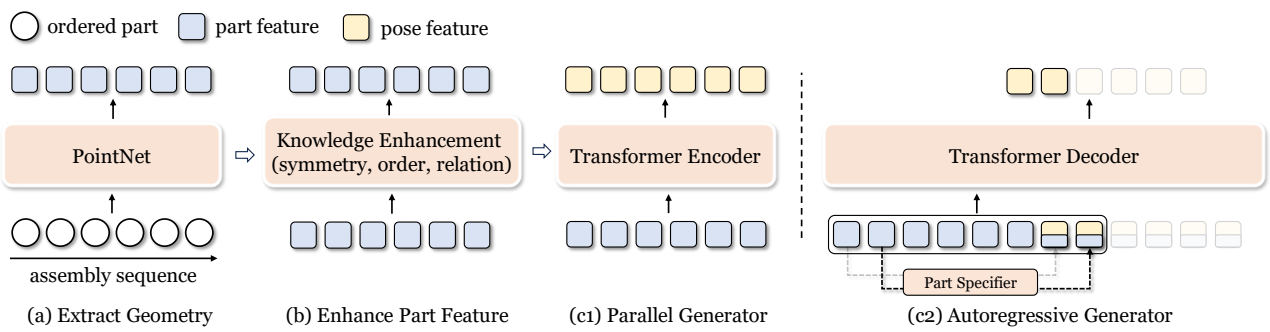


Figure 3. Illustration of overall end-to-end framework of SPAFormer. (a) The shared 3D backbone extracts the geometry feature of individual parts, followed by (b) knowledge enhancement of part features, which incorporates symmetry, order, and relation information into part features through positional encodings, then generates poses by either (c1) parallel generator, which generates poses of all parts at once, or (c2) autoregressive generator, which decodes poses of parts according to assembly sequences step by step.

transformers: $h(v_i, i) = v_i \oplus e_N(i)$, where $\oplus$ denotes feature concatenation and $e_N(i)$ is a one-hot embedding with the i-th position being 1 in a vector of length N. This encoding integrates the i-th order information into the part feature v_i before passing the part feature to the transformers.

Relation Encoding. We introduce relation encoding to capture the relative positions of assembling parts such as the bottom-to-top assembly order in IKEA furniture. Relation encoding is implemented using rotary embedding [39] to enhance the attention mechanism, which rotates the representations of query-key pairs in self-attention. We leverage its long-term decay property on attention weights to encode relative relations between parts:

$$f_{\{q,k\}}(v_i, i) = R^d_{\Theta,i} v_i \tag{1}$$

where $R^d_{\Theta,i} = \mathrm{diag}(A_1, \ldots, A_{d/2})$, $A_i = \begin{bmatrix} \cos(m\theta_i) & -\sin(m\theta_i) \\ \sin(m\theta_i) & \cos(m\theta_i) \end{bmatrix}$ is a rotation matrix, and $\mathrm{diag}(\cdot)$ denotes diagonalizing block matrices. The relation encoding is applied to queries and keys to capture the relative order information:

$$q_i \cdot k_j = (R^d_{\Theta,i} v_i) \cdot (R^d_{\Theta,j} v_j) = v_i \cdot R^d_{\Theta,i-j} v_j \tag{2}$$

where $\cdot$ denotes dot product, and the attention weights decay as the relative position between query and key increase. We refer to Su *et al*. [39] for details.

Symmetry Encoding. Symmetrical relations, such as translational (e.g., identical and parallel arms of a chair), reflective (e.g., mirror-symmetrical arms of a chair), and rotational symmetry (e.g., identical but not parallel legs of a chair), are proven beneficial for creating structured objects in generative methods [33]. To incorporate symmetrical relations into part features, we use principal component axes via PCA to acquire 3D bounding boxes and calculate bounding box sizes, which are invariant to input poses. The parts are organized based on identical bounding box sizes, resulting in the partition $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_M\}$, where each part p_i is associated with a unique group $s_i \in \{1, \ldots, M\}$. Symmetry encoding explicitly models symmetrical information into part features: $g(v_i) = v_i \oplus e_N(s_i)$.

Consequently, symmetrical parts within the groups share similar features, producing distinguishable predictions between different groups. To prevent collapsed prediction within a group, bipartite matching is conducted inside each group $\mathcal{P}_i = \{p_i^j\}_{j=1}^{n_i}$ between the transformed parts $\{y_i^j(p_i^j)\}_{j=1}^{n_i}$ and the ground truths $\{p_i^{j*}\}_{j=1}^{n_i}$, where the asterisk symbols $(*)$ represents the respective ground truths in all equations. While previous works [14, 25, 35] assign distinct encodings to symmetrical parts inside a group $\mathcal{P}_i$ to handle the ambiguity prediction of symmetrical parts, we suggest distinguishing different groups and assemble symmetrical parts by bipartite matching.

3.2. Generator Variants for Part Assembly

Then we discuss different generator variants to predict the 3D poses of object parts.

Parallel Generator. As illustrated in Figure 3 (c1), this is the most straightforward way adopted by 3D-PA approaches [14, 58]. The parallel generator simultaneously outputs the 3D poses of the sequential parts in parallel, which can be denoted as:

$$\{y_i\}_{i=1}^N = \mathcal{G}(\mathcal{F}(\{v_i\}_{i=1}^N)) \tag{3}$$

where $\mathcal{F}$ denotes the transformer encoder [46] with L layers, which aims to build relationships among parts using self-attention, and $\mathcal{G}$ is a shared MLP that maps part representations to poses.

Autoregressive Generator. Instead of parallel generation, another alternative is to autoregressively decode the 3D poses of object parts following the assembly sequence.

To provide a global context of the potential assembled object, all part features are prepended as prompts before de-

coding the poses. At each step i, the model generates a pose feature z_i, concatenates it with the next part feature v_{i+1}, and feeds it to the transformer model as input recursively. To decide the next part and maintain consistency in dimension, a part specifier indexes the upcoming part following the assembly sequence and projects the part feature for concatenation with the last pose feature, as depicted in Fig. 3 (c2). We use two linear projectors ϕ_1 and ϕ_2 to downsample the part v and pose z features from $\mathbb{R}^d$ to $\mathbb{R}^{\frac{d}{2}}$, respectively, where d is the hidden dimension of the transformers. As a result, the generation progress is formulated as:

$$y_i = \mathcal{G}(\mathcal{F}(\{v_k\}_{k=1}^{N}; \{(\phi_1(z_j) \oplus \phi_2(v_{j+1})\}_{j=0}^{i-1})) \tag{4}$$

3.3. Training Process

Training Loss. Supervision of both individual parts and the entire shape is applied during training. The part pose y_i is represented by translation $t_i \in \mathbb{R}^3$ and rotation $r_i \in \mathbb{R}^4$ in the form of quaternion. The translation is supervised via a MSE loss:

$$\mathcal{L}_t = \sum_{i=1}^{N} \|t_i - t_i^*\|_2^2 \tag{5}$$

The distance between two sets of point clouds $\mathcal{X}_1, \mathcal{X}_2$ can be measured by Chamfer distance [8]:

$$\mathcal{CD}(\mathcal{X}_1, \mathcal{X}_2) = \sum_{i,j \in \{1,2\}} \sum_{x_i \in \mathcal{X}_i} \min_{x_j \in \mathcal{X}_j} \|x_i - x_j\|_2^2 \tag{6}$$

Since rotation is not unique, the rotation is supervised via Chamfer Distance on the rotated part point cloud instead of directly applying an MSE loss on r_i:

$$\mathcal{L}_r = \sum_{i=1}^{N} (\mathcal{CD}(y_i(p_i), y_i^*(p_i))) \tag{7}$$

In order to better optimize global shapes, we also apply Chamfer Distance on the complete shape:

$$\mathcal{L}_s = \mathcal{CD}(S, S^*) \tag{8}$$

The final objective for 3D part assembly is calculated as:

$$\mathcal{L} = \lambda_t \mathcal{L}_t + \lambda_r \mathcal{L}_r + \lambda_s \mathcal{L}_s \tag{9}$$

where λ_t, λ_r and λ_s are hyper-parameters.

4. Experiments

4.1. Experiment Setup

Dataset Setup. We notice that prior studies only consider assembling merely three object categories (chair, table, lamp), which is a far cry from diverse objects in reality. To this end, we expand the available objects to include 21 categories from PartNet [34], thereby introducing a more diversified set of objects to verify the SPAFormer's generalization. We refer to this enhanced benchmark as PartNet-Assembly, and it is split into two subsets: (1) *furniture object*, containing three structurally complex categories (chair, table, and storage) that offer sufficient and diverse samples, and (2) *daily object*, containing the remaining 18 categories that represent commonly used everyday items (e.g. earphone). In total, PartNet-Assembly contains 22,873 shapes from 21 object categories with 173,765 distinct parts. The dataset is partitioned into training (70%), validation (10%), and testing (20%) sets, following the distribution protocol proposed by Mo *et al*. [34].

Implementation Details. We utilize the Adam optimizer initialized with a learning rate of 1.5e-4, and employ an exponential decay strategy, reducing the learning rate by a factor of 0.8 every 80 epochs. The training process is conducted on 2 GPUs, employing a batch size of 128, over a total of 800 epochs. The optimal epoch is selected through performance assessment using Part Accuracy (PA) metrics on the unseen validation split. The transformer architecture is configured with a hidden dimension size of 512 across 6 layers. Following Huang *et al*. [14], the hyperparameters are set as follows: $\lambda_t = 1$, $\lambda_r = 10$, $\lambda_s = 1$. Furthermore, all target shapes and input parts within PartNet-Assembly are normalized to the canonical space using PCA, which eliminates the impact of initial poses.

Evaluation Metrics. In this study, we evaluate the quality of the assembled shapes using four key metrics:

- Shape Chamfer Distance (SCD) [25], measures the overall distance between the predicted and ground-truth shapes using the equation below, with all values scaled by 10^3 for simplicity:

$$\text{SCD} = \mathcal{CD}(\cup_{i=1}^{N} y_i(x_i), \cup_{i=1}^{N} y_i^*(x_i)) \tag{10}$$

- Part Accuracy (PA) [25], assesses the correctness of part placement. A part is considered accurately placed if its error distance is below a specified threshold $\epsilon = 0.01$:

$$\text{PA} = \mathbb{I}\left[\mathcal{CD}(y_i(x_i), y_i^*(x_i)) < \epsilon\right] \tag{11}$$

- Connectivity Accuracy (CA) [14], measures the quality of connections between adjacent part pairs in the assembled shape. With $\tau = 0.01$ as the threshold, and $C = \{(c_{ij}, c_{ji})\}$ representing the set of all connected point pairs between adjacent parts (p_i, p_j):

$$\text{CA} = \frac{1}{|C|} \sum_{(c_{ij}, c_{ji}) \in C} \mathbb{I}\left[(y_i(c_{ij}) - y_j(c_{ji}))^2 < \tau\right] \tag{12}$$

- Success Rate (SR) [26], is defined as 1 for an object if the part accuracies of all components are equal to 1.

4.2. Ablation Study

1. Effect of different generator variants for assembly.

Table 1. Comparison of different generator variants on table assembly, where "para" and "auto" denote the parallel and autoregressive generator, respectively.

Generator	SCD↓	PA↑	CA↑	SR↑
para	3.80	64.38	57.60	33.50
auto	4.14	62.18	59.41	31.92

Table 2. Comparison of different assembly sequence patterns on chair assembly.

Pattern	SCD↓	PA↑	CA↑	SR↑
diagonal	6.74	55.88	36.39	16.40
top-to-bottom	9.36	45.29	25.45	10.30
bottom-to-top	9.07	45.72	27.21	11.24
descending size	10.88	38.09	21.91	9.65
random	14.38	29.40	16.69	5.99

In Table 1, we compare the performance of different variants outlined in Section 3.2 on the table assembly task. Both variants implement identical knowledge enhancement strategies. The results indicate that the parallel generator (denoted as "para") outperforms the autoregressive generator (denoted as "auto") across a majority of the 3D-PA metrics, with the exception of CA. Specifically, the "para" generator excels in SCD, PA, and SR, highlighting its proficiency in assembling objects that are globally coherent. Conversely, the "auto" generator brings gains in CA, indicating its ability to ensure better connectivity among adjacent parts of the objects. Due to the ease of training, we adopt the "para" generator in our following experiments.

2. Effect of different assembly sequence patterns.

We carry out ablations in Table 2 to evaluate the effectiveness of different assembly sequences: a) diagonal, which involves assembling an object from one corner diagonally to the opposite corner (e.g., assembly a chair from a front leg to the back); b) top-to-bottom, where objects are assembled from the top down; c) bottom-to-top, which is the reverse process of top-to-bottom, where assembly starts from the bottom; d) descending size, which is determined by the sizes of object components, calculated via principal component analysis (PCA), starting with the largest. The outcomes reveal that the diagonal pattern outperforms the others. This superior performance can be attributed to its ability to sequentially assemble symmetric components (such as the four legs of a chair), a feat that is more efficiently accomplished than other patterns like top-to-bottom. Based on these findings, we adopt the diagonal assembly pattern as our default strategy in subsequent experiments.

3. Effect of knowledge enhancement strategies.

Table 3 reveals the impact of various encoding strategies on 3D part assembly. As can be seen, each strategy can bring improvements. For example, employing OEnc and SEnc augment the assembly outcomes by $+8.08\%$ and $+17.95\%$ on PA, by comparing Row 3-4 with Row 1. This demonstrates the value of incorporating order and symmetry information into the model, which significantly benefit the assembly process. However, while REnc alone does not notably advance assembly capabilities in Row 2, its combination with other encodings acts as a pivotal enhancer, akin to a catalyst. For instance, the comparison between Row 6 and Row 7 reveals that the collaborative effect of REnc with other encodings improves SR by $+2.35\%$ and PA by $+1.71\%$. Qualitative examples are provided in Figure 4 to present the gains of integrating each encoding strategy.

Table 3. Comparison of different knowledge enhancement strategies on chair assembly, where OEnc, REnc, and SEnc denote order, relation, and symmetry encoding, respectively.

	OEnc	REnc	SEnc	SCD↓	PA↑	CA↑	SR↑
1	✗	✗	✗	14.38	29.40	16.69	5.99
2	✗	✓	✗	14.71	28.45	16.38	5.34
3	✗	✗	✓	13.35	37.48	21.41	7.68
4	✓	✗	✗	8.30	47.35	29.57	12.08
5	✓	✓	✗	8.04	48.51	30.61	12.18
6	✓	✗	✓	7.91	54.17	35.31	14.05
7	✓	✓	✓	**6.74**	**55.88**	**36.39**	**16.40**

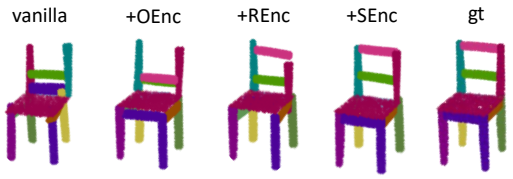


Figure 4. Visualizations of assembly results when enhancing knowledge by adding new encoding patterns in a stepwise way.

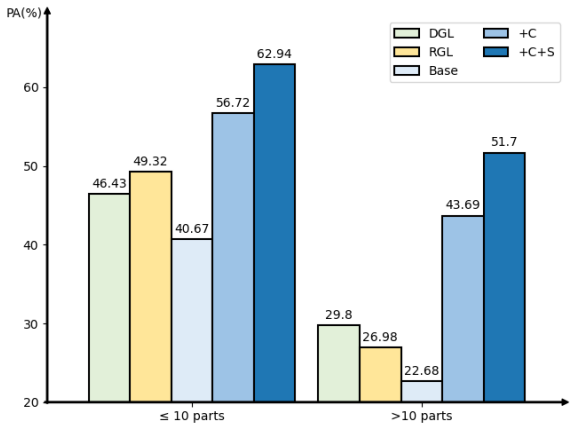


Figure 5. Comparison of varied assembly length. Our model presents notable improvements particularly in long-horizon assembly (>10 parts) when the model is enhanced by incorporating OEnc and REnc (+C), as well as SEnc (+S).

4. Effect of varied assembly sequence length.

We conduct an analysis of varied assembly lengths for

Table 4. Comparison with previous SoTA works on the furniture assembly task.

		SCD↓			PA↑			CA↑			SR↑		
Method	Seq	Chair	Table	Storage	Chair	Table	Storage	Chair	Table	Storage	Chair	Table	Storage
DGL [14]	✗	9.1	5.0	12.1	39.00	49.51	12.39	23.87	39.96	17.23	8.44	20.35	0.00
IET [57]	✗	12.3	5.6	6.8	39.30	49.21	29.31	17.83	35.82	22.51	6.18	21.34	1.07
Score-PA [3]	✗	7.4	4.5	6.9	42.11	51.55	27.68	34.27	32.79	21.82	8.32	11.23	0.35
CCS [58]	✗	7.0	-	-	53.59	-	-	**38.97**	-	-	-	-	-
Complement [41]	✓	43.4	16.5	29.3	9.85	12.90	8.61	19.39	22.75	31.17	0.00	5.94	3.57
LSTM [51]	✓	18.7	8.0	11.9	11.21	25.63	8.50	12.30	29.66	27.14	0.46	8.79	0.35
RGL [35]	✓	10.9	5.5	6.4	42.16	59.99	48.02	24.85	49.89	38.45	7.12	26.43	3.92
Ours-specific	✓	**6.7**	**3.8**	4.5	**55.88**	64.38	56.11	36.39	57.60	49.98	**16.40**	33.50	7.85
Ours-multitask	✓	7.0	4.4	**3.8**	55.84	**64.44**	**67.14**	38.75	**61.04**	**65.20**	13.96	**33.57**	**15.71**

Table 5. Comparison of daily object assembly on 18 categories.

Method	SCD↓	PA↑	CA↑	SR↑
Score-PA [3]	**6.38**	32.39	32.16	15.05
DGL [14]	6.57	36.55	40.01	18.73
RGL [35]	10.93	51.33	50.52	32.00
Ours	8.61	**54.32**	**51.60**	**37.12**

Table 6. Comparison between category-specific and universal implementations of SPAFormer, evaluated on 21 object categories.

	SCD↓	PA↑	CA↑	SR↑
ours-specific	6.54	57.79	49.95	28.80
ours-multitask	6.37	59.78	53.74	29.40

the identification of assembly patterns and symmetry information in Figure 5. It is generally accepted that assembling objects composed of more parts presents a higher degree of difficulty. To this end, we categorize the testing set into two sets based on the number of parts: short-horizon assembly, which consists of fewer than or equal to 10 parts, and long-horizon assembly, encompassing those with more than 10 parts. For example, a chair usually comprises fewer than 10 parts including four legs, a back, and a seat. Compared with the vanilla attention baseline, our model that captures sequence information (denoted as "+C") not only exhibits a significant performance improvement of $+16.05\%$ in short-horizon assembly but also shows an impressive gain of $+21.01\%$ in the more complex long-horizon assembly. Further analysis with SEnc (marked as "+S") demonstrates the use of SEnc as a versatile technique that uniformly enhances PA performance across different assembly lengths. Such results indicate that our model is able to effectively capture complex structural information, marking a promising avenue for tackling challenging long-horizon tasks. In contrast, existing approaches [14, 35] struggle with long-horizon assembly, with a decrease in PA exceeding -21.9%.

5. Effect of multi-task joint training.

An ideal assembly agent should be proficient in assembling a wide array of objects through a unified model. However, the exploration of such a universal assembly methodology remains underrepresented in existing works, and the category-specific approach is still widely adopted. To address this gap, our work delves deeper into the feasibility of a general-purpose assembly approach. In addition to vanilla category-specific SPAFormer (denoted as "ours-specific") where each model is tailored to a specific category, we develop a multi-task model (denoted as "ours-multitask") that is trained across all accessible object categories. We compare the two implementations in Table 6.

The results reveal that the multi-task model excels in assembling objects from categories with both abundant (e.g., tables, storage furniture) and scarce (e.g., mugs with 120 samples, bowls with 130 samples) training data (Details can be found in supplementary). To be specific, the multi-task model presents a modest enhancement of $+1.99\%$ on PA, +3.79% on CA, and +0.6% on SR, highlighting the viability of a general-purpose approach to part assembly. Given these positive outcomes, we intend to further explore towards this direction, focusing on identifying which categories most benefit from shared assembly knowledge and controlling the generated categories in future works.

4.3. Comparison with State of the Arts

Comparison of Furniture Assembly. Assembling furniture is challenging due to the complex and varied shapes of structured objects, which troubles many assembly methods. As shown in Table 4, our model significantly outperforms both previous score-based and graph-based models in the assembly of structured furniture objects across almost all evaluated metrics. For instance, when assembling unseen tables, our category-specific approach (denoted as "Ours-specific") remarkably exceeds the performance of the lead-

Table 7. Comparison of conditional assembly on chair assembly.

Method	Constraints	PA↑	SR↑
Score-PA [3]	condition-free	42.11	8.32
Img-PA [25]	image gt	49.10	-
GPAT [26]	point cloud gt	**57.7**	19.30
Ours	sequence	57.29	**21.97**

ing prior model, RGL [35], achieving improvements of $+4.39\%$ on PA, $+7.71\%$ on CA, and $+7.07\%$ on SR. These results demonstrate the efficacy of our proposed SPAFormer in handling furniture assembly tasks.

Additionally, we delve into a comparison of various conditional assembly techniques with a focus on the chair assembly task due to its structural property, as presented in Table 7. To ensure fair comparisons, we conduct experiments on the dataset adopted by Li *et al*. [25] and evaluate these models on PA and SR, which are fairly compared metrics across different constraints. Despite being compared against methods that utilize ground truth inputs such as images [25] or 3D point clouds [26], our model demonstrates commendable performance. This achievement highlights the distinct advantage of our sequence-conditioned assembly approach, which effectively generalizes to novel structures solely based on the relationships between parts, without relying on external visual information.

Comparison of Daily Object Assembly. To assess our model's performance in broader scenarios, we conduct experiments on the collection of 18 everyday objects (e.g. earphone, bottle), showcasing its capability to generalize across a wider range of items. As illustrated in Table 5, our SPAFormer demonstrates a substantial advantage over condition-free methods, achieving an improvement of at least $+18.39\%$ on SR, and surpassing another sequence-conditioned RGL method by 5.12% on SR. Despite condition-free approaches [3, 14] performs better on SCD, they significantly underperform across other metrics. We attribute this to the optimization collapse, where the value of SCD loss is low but the assembled shapes are less reasonable, as shown in Row 1-2 in Figure 1.

4.4. Qualitative Analysis

Failure cases. In exploring the limitations of our model, Figure 6 highlights two primary factors leading to assembly failures: out-of-distribution (OOD) shapes and our model's inherent weakness. The left two columns in Figure 6 depict OOD shapes that are rarely encountered in reality, such as recliner chair and sofa without chair legs. The examples on the right side of Figure 6 shed light on inherent weaknesses in our model. Despite making significant strides in addressing long-horizon assembly challenges as evidenced in Figure 5, our model exhibits limitations in assembling complex objects, such as chairs composed of up to 17 parts. Furthermore, our model encounters difficulties in accurately recognizing part orientations, primarily due to its limited ability to detect the junctions between parts. To address these issues, labeling parts with peg-hole joints [27] and reframing the problem as bipartite matching is a costly alternative.

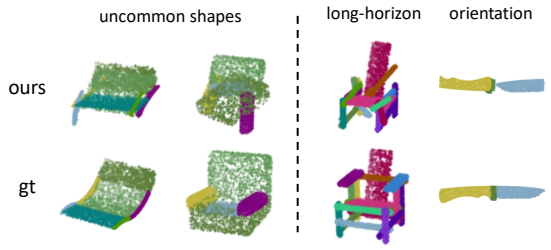


Figure 6. Visualizations of failed object assemblies, including rare shapes, long-horizon assembly, and wrong orientation.

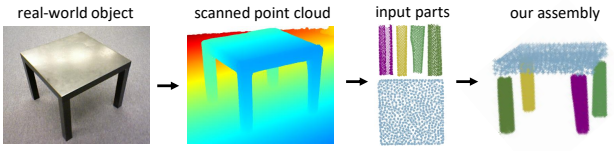


Figure 7. Real-world experiment on table assembly.

Real-world experiment. Additionally, we conduct a real-world experiment, where the object scan is sourced from the Redwood dataset [5], to successfully assemble a table by our model, suggesting potential generalization to real-world scenarios where the part point clouds are imperfect.

5. Conclusion

In this study, we introduce a novel model SPAFormer, which is designed to tackle the issue of combinatorial explosion in 3D part assembly tasks. Building upon insights from prior research, SPAFormer employs assembly sequences to effectively narrow down the extensive solution space. We explore different assembly generator variants and incorporate multiple knowledge enhancement strategies for capturing the inherent assembly pattern. To facilitate comprehensive evaluation, we extend the existing benchmark to PartNet-Assembly, which spans 21 diverse categories including both furniture and daily objects. Experimental analyses are conducted to validate components of SPAFormer and the enhanced performance over existing methods.

Acknowledgements

This work was partially supported by the Beijing Natural Science Foundation (No. L233008).

References

[1] 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. 3

[2] 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 (CVPR)*, pages 12724–12733, 2022. 3

[3] Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. Score-pa: Score-based 3d part assembly. *British Machine Vision Conference (BMVC)*, 2023. 1, 2, 3, 7, 8

[4] Fedor Chervinskii, Sergei Zobov, Aleksandr Rybnikov, Danil Petrov, and Komal Vendidandi. Auto-assembly: a framework for automated robotic assembly directly from cad. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11294–11300, 2023. 1, 2, 3

[5] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 8

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 3

[7] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022. 3

[8] 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

[9] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 3

[10] 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. 3

[11] Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomir Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. Learning generative models of shape handles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 402–411, 2020. 3

[12] 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, 3

[13] Seyed Kamyar Seyed Ghasemipour, Satoshi Kataoka, Byron David, Daniel Freeman, Shixiang Shane Gu, and Igor Mordatch. Blocks assemble! Learning to assemble with large-scale structured reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 7435–7469. PMLR, 2022. 1, 3

[14] Jialei Huang, Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas Guibas, and Hao Dong. Generative 3d part assembly via dynamic graph learning. In *The IEEE Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 3, 4, 5, 7, 8

[15] Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling fractured objects by geometric matching. In *ACM siggraph 2006 papers*, pages 569–578. 2006. 3

[16] 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. 3

[17] 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

[18] 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. 3

[19] Satoshi Kataoka, Youngseog Chung, Seyed Kamyar Seyed Ghasemipour, Pannag Sanketi, Shixiang Shane Gu, and Igor Mordatch. Bi-manual block assembly via sim-to-real reinforcement learning. *arXiv preprint arXiv:2303.14870*, 2023. 1, 3

[20] Sung-Kyun Kim and Maxim Likhachev. Parts assembly planning under uncertainty with simulation-aided physical reasoning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4074–4081. IEEE, 2017. 3

[21] Nikolas Lamb, Cameron Palmer, Benjamin Molloy, Sean Banerjee, and Natasha Kholgade Banerjee. Fantastic breaks: A dataset of paired 3d scans of real-world broken objects and their complete counterparts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4681–4691, 2023. 3

[22] 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. 3

[23] Jun Li, Chengjie Niu, and Kai Xu. Learning part generation and assembly for structure-aware shape synthesis. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11362–11369, 2020. 3

[24] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5554–5564, 2022. 3

[25] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3d part assembly from a sin-

gle image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 664–682. Springer, 2020. 2, 3, 4, 5, 8

[26] Yulong Li, Andy Zeng, and Shuran Song. Rearrangement planning for general part assembly. In *Conference on Robot Learning*. PMLR, 2023. 2, 3, 5, 8

[27] Yichen Li, Kaichun Mo, Yueqi Duan, He Wang, Jiequan Zhang, Lin Shao, Wojciech Matusik, and Leonidas Guibas. Category-level multi-part multi-joint 3d shape assembly. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 8

[28] Yunong Liu, Cristobal Eyzaguirre, Manling Li, Shubh Khanna, Juan Carlos Niebles, Vineeth Ravi, Saumitra Mishra, Weiyu Liu, and Jiajun Wu. IKEA manuals at work: 4d grounding of assembly instructions on internet videos. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 2

[29] Jiaxin Lu, Yifan Sun, and Qixing Huang. Jigsaw: Learning to assemble multiple fractured objects, 2023. 3

[30] Lin Ma, Jiangtao Gong, Hao Xu, Hao Chen, Hao Zhao, Wenbing Huang, and Guyue Zhou. Planning assembly sequence with graph transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12395–12401. IEEE, 2023. 2

[31] Lin Ma, Jiangtao Gong, Hao Xu, Hao Chen, Hao Zhao, Wenbing Huang, and Guyue Zhou. Planning assembly sequence with graph transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12395–12401. IEEE, 2023. 2

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[33] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38(6):Article 242, 2019. 2, 3, 4

[34] 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. 5

[35] 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, 3, 4, 7, 8

[36] 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

[37] Silvia Sellán, Yun-Chun Chen, Ziyi Wu, Animesh Garg, and Alec Jacobson. Breaking bad: A dataset for geometric fracture and reassembly. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 3

[38] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, 2018. Association for Computational Linguistics. 3

[39] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 3, 4

[40] Sujay Sundaram, Ian Remmler, and Nancy M Amato. Disassembly sequencing using a motion planning approach. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, pages 1475–1480. IEEE, 2001. 2

[41] 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. 7

[42] Xi Tian, Yong-Liang Yang, and Qi Wu. Shapescaffolder: Structure-aware 3d shape generation from text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2715–2724, 2023. 2

[43] Yunsheng Tian, Jie Xu, Yichen Li, Jieliang Luo, Shinjiro Sueda, Hui Li, Karl DD Willis, and Wojciech Matusik. Assemble them all: Physics-based planning for generalizable assembly by disassembly. *ACM Transactions on Graphics (TOG)*, 41(6):1–11, 2022. 2

[44] Yunsheng Tian et al. Asap: Automated sequence planning for complex robotic assembly with physical feasibility. In *ICRA*, 2024. 2

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[46] 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. 3, 4

[47] Kai Wang, Paul Guerrero, Vladimir G Kim, Siddhartha Chaudhuri, Minhyuk Sung, and Daniel Ritchie. The shape part slot machine: Contact-based reasoning for generating 3d shapes from parts. In *European Conference on Computer Vision*, pages 610–626. Springer, 2022. 2

[48] Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Chin-Yi Cheng, and Jiajun Wu. Translating a visual lego manual to a machine-executable plan. In *European Conference on Computer Vision*, pages 677–694. Springer, 2022. 2

[49] Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Ran Zhang, Chin-Yi Cheng, and Jiajun Wu. IKEA-manual: Seeing shape assembly step by step. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1, 2

[50] Weihao Wang, Rufeng Zhang, Mingyu You, Hongjun Zhou, and Bin He. 3d assembly completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2663–2671, 2023. 3

[51] 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. 3, 7

[52] Ruihai Wu, Chenrui Tie, Yushi Du, Yan Zhao, and Hao Dong. Leveraging se(3) equivariance for learning 3d geometric shape assembly. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14311–14320, 2023. 3

[53] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3

[54] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13569–13578, 2021. 3

[55] 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. 3

[56] Jiahao Zhang, Anoop Cherian, Yanbin Liu, Yizhak Ben-Shabat, Cristian Rodriguez, and Stephen Gould. Aligning step-by-step instructional diagrams to video demonstrations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[57] 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. 2, 3, 7

[58] Ruiyuan Zhang, Jiaxiang Liu, Zexi Li, Hao Dong, Jie Fu, and Chao Wu. Scalable geometric fracture assembly via co-creation space among assemblers. *arXiv preprint arXiv:2312.12340*, 2023. 2, 3, 4, 7