This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Category-Level Multi-Part Multi-Joint 3D Shape Assembly

Yichen Li[1,2] Kaichun Mo[3] Yueqi Duan[4] He Wang[5] Jiequan Zhang[1] Lin Shao[6]
Wojciech Matusik[2] Leonidas Guibas[1]
[1]Stanford University [2]MIT CSAIL [3]NVIDIA [4]Tsinghua University [5]Peking University
[6]National University of Singapore

Abstract

Shape assembly composes complex shapes geometries by arranging simple part geometries and has wide applications in autonomous robotic assembly and CAD modeling. Existing works focus on geometry reasoning and neglect the actual physical assembly process of matching and fitting joints, which are the contact surfaces connecting different parts. In this paper, we consider contacting joints for the task of multi-part assembly. A successful joint-optimized assembly needs to satisfy the bilateral objectives of shape structure and joint alignment. We propose a hierarchical graph learning approach composed of two levels of graph representation learning. The part graph takes part geometries as input to build the desired shape structure. The joint-level graph uses part joints information and focuses on matching and aligning joints. The two kinds of information are combined to achieve the bilateral objectives. Extensive experiments demonstrate that our method outperforms previous methods, achieving better shape structure and higher joint alignment accuracy.

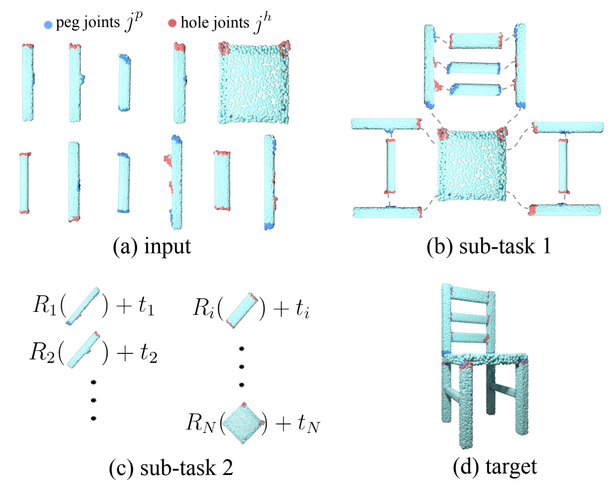

Figure 1. (a) Joint-annotated part point cloud input where blue points indicate peg joints and red points denote hole joints. (b) joint pairing process to produce a bipartite matching between pegs and holes (c) joint-matching-aware SE(3) pose prediction for each part to achieve the target (d) of assembling a shape of a valid structure and with all pegs and holes aligned.

1. Introduction

Shape assembly composes complex shape geometries by arranging a set of simple or primitive part geometries. Many important tasks and applications rely on shape assembly algorithms. For example, assembling Ikea furniture requires one to identify, reorient, and connect the relevant parts. Computer-Aided Design (CAD) modeling requires designers to reposition and align a set of part geometries to create complex designs. An accurate and robust shape assembly algorithm is critical to the development of autonomous systems for furniture assembly or CAD modeling [34, 41, 48].

In this paper, we aim to tackle the task of *multi-part multi-joint* shape assembly. This task simulates the real-world furniture assembly setting, where multiple shape parts are connected in different ways through contacting joints to make a complex shape geometry [23, 25]. As Fig. 1 shows, we are given (a) *multiple shape parts*, where each part contains *multiple joints*. For our setting, we use peg-hole joint pairs to represent the allowed connections, similar to bolts and nuts where matching is only allowed between male and female pieces of the same contacting geometry [49]. Our goal is to (b) correctly connect all peg joints with hole joints and (c) piece these parts together to (d) make a desired shape.

Many research efforts have been made towards devising shape assembly algorithms [3, 5, 7, 12, 15, 18, 26, 30, 39, 42–44, 49, 52, 53, 58, 60]. These prior works take the holistic geometric perspective of modeling shapes from parts. These works produce shapes with great aesthetic value. However, this pure geometric perspective is agnostic of the rotational and reflective symmetry of parts, and thus results in upside-down, flipped, and rotated part pose predictions. These noisy predictions can lead to unmatched joints or mismatches between joints, making it difficult to directly employ them in the context of autonomous assembly [17, 41, 43, 50, 57] or modeling of functional shapes [32, 62].

Many challenges exist in the multi-part multi-joint assembly setting: 1) large matching search space, 2) non-continuous optimization, and 3) error compounding. Previously, Willis et al. [49] have considered joints for shape assembly, but they focus on assembling shapes from only *two parts* with one pair of joints. In the two-part assembly setting, the pairing of joints is explicit and thus the desired assembly can be directly achieved through continuous pose optimization. However, our multi-part multi-joint task requires solving for a bipartite joint pairing in a very large matching space. Additionally, our task requires interleaved *discrete* and *continuous* optimization. Joint pairing is a combinatorial problem in a discrete solution space, whereas pose estimation is in a continuous solution space. In the multi-part multi-joint setting, part poses need to simultaneously satisfy both the combinatorial and the continuous constraints. Finally, optimization of this task is sensitive to error compounding. When one pair of joints are mismatched, the poses for the two parts need to be falsely adjusted in order to align these wrongly matched joints. These erroneous pose predictions reciprocally affect other joints on the two parts. These local errors propagate and eventually lead to the deterioration of the entire shape structure.

To tackle these challenges, we propose an end-to-end graph learning approach in a divide-and-conquer manner. We decouple the complex task objective into combinatorial and continuous subgoals modeled by two levels of graph representation learning. The joint-level graph uses joint information and focuses on matching part joints, and the part graph takes part geometries as input to build the desired shape structure. The two levels of graphs are then combined to achieve both two objectives through hierarchical feature aggregation. Since joints are special locations on parts, we aggregate all joint-level features for each part to form a set of joint-centric part features. These joint-centric part features are combined with the learned part graph to predict part poses to meet both the shape structure and joint matching objectives. To alleviate the error compounding issue, we use several graph iterations to assemble shapes in a coarse-to-fine manner. Each graph iteration learns to correct and refine part pose predictions from the previous iteration to eventually achieve the multi-level objectives. Extensive experiments demonstrate that we are able to achieve higher joint matching accuracy and more reliable shape structure over prior works. Our contributions are summarized as follows:

- We consider the concept of joint for the problem of category-level multi-part 3D shape assembly. We introduce a joint-annotated part dataset as well as a set of evaluation metrics to examine the performance.
- We propose a novel hierarchical graph network that simultaneously optimizes for both holistic shape structure and the joint alignment accuracy.
- We conduct extensive experiments to demonstrate the advantages of our approach over prior works on both task objectives of holistic shape structure and joint alignment accuracy. We also show ablation experiments to validate our design choices.

2. Related Work

Assembly-based 3D Modeling. Part assembly plays an important role in many tasks [13, 24, 33, 47, 61, 62]. As a pioneering work, [8] proposes a data-driven synthesis approach for 3D geometric surface model reconstruction. Since then, various methods have been proposed to generate shapes from parts [9–11, 14, 18, 20]. Many works [2, 16, 19] focus on using probabilistic graphical models to encode semantic and geometric relations among shape parts. Other works [4, 40, 42, 54] build 3D shapes conditioned on partial shapes. While most of these methods require a third-party shape repository, some generative methods have been presented in recent years [9, 10, 15, 18, 26, 35, 54]. For example, [54] first learn to generate shape parts and then estimate the transformation of parts to compose shapes. [12, 15] design a dynamic graph learning approach by reasoning about part poses and relations iteratively. Our task is different from these prior assembly-based shape synthesis works in that our goal is not to generate a variety of shapes, but to solve for the set of part poses that make *one* desired shape and match all the joints at the correct locations. Previously, [30] explores the problem of single-image-guided 3D part assembly, using an image as guidance to predict 6D part poses to assemble the desired shape, but they neglect the information of part joints and contact surfaces. Recently, Willis et al. [49] also considers joints for shape assembly but with only two parts and one pair of joints, whereas our task considers multiple shape parts and each comes with multiple contact joints. [49] also assumes watertight part geometry, whereas we relax this assumption and use simple point cloud representation that can be easily obtained using commercial scanners [32].

Graph Learning for Part Relationship Graph neural network has been proposed to study the relationship between entities to better understand objects and scenes. Recently, a line of research [6, 31, 56, 59] learns the scene graph information from the labeled object relationship with a graph neural network, which benefits object detection on large-scale image dataset such as Visual Genome [21]. Other works [28, 29, 38, 46, 63] explore the physical relationships between objects with geometrical and statistical heuristics, which are encoded in the iterative neural encoding and achieve decent performance on the 3D scene generation task. Inspired by the success of graph learning in various tasks, other research [10, 12, 15, 27, 55] apply the part relation reasoning to learn shape structure and geometry for 3D shape modeling. These prior works deals with more apparent object-level or part-level relationships and leverage explicit relationship supervision, which is calculated from shape topology, adjacency, and support. We are different from these works in that our task deals with a hierarchy of relationships, the relationships among joints and the relationship among parts. We use a hierarchical graph learning technique to simultaneously achieve bilateral objectives.

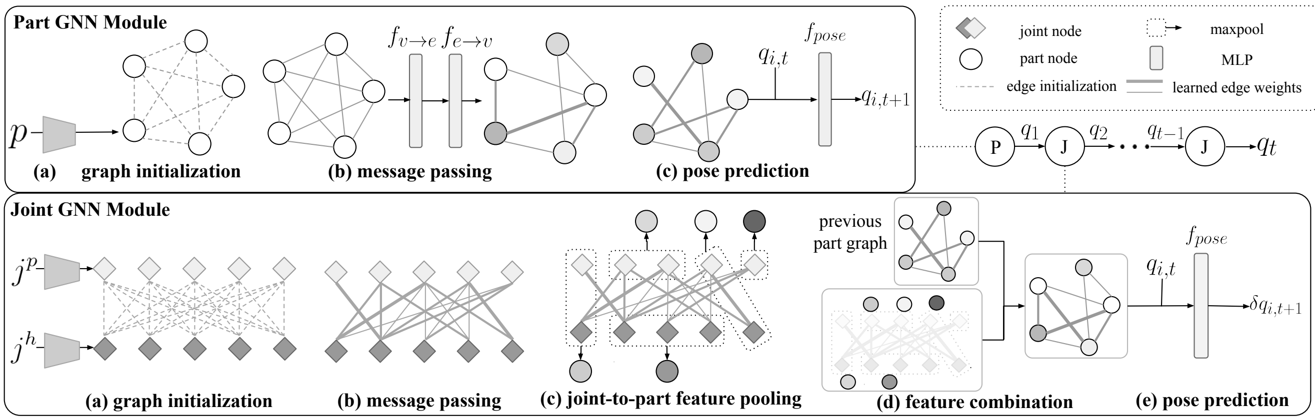

Figure 2. Our multistage graph network is composed of two main GNN modules: the part graph module and the joint graph module. The part graph module is responsible for predicting poses for each part to construct the desired shape structure. The joint graph module helps to correct part poses to connect the matched joints. Joint graph message passing (b) contains four message-passing layers: $f_{v\to e}, f_{e\to v}, f_{v\to r}, f_{r\to v}$.

3. Method

Problem Setup. Our multi-part multi-joint shape assembly task is defined as follows: given 1) a set of 3D part point clouds $\mathcal{P} = \{p_i\}_{i=1}^N$ and 2) each part should contain a number of peg and/or hole joints $\mathcal{J} = \{j_i^p, j_i^h\}_{i=1}^M$, we aim to predict a set of 6-DoF part pose $q_i = (R_i, t_i), q_i \in SE(3)$ for *all* input parts $\mathcal{P}$ to satisfy the *bilateral* objectives: 1) the union of the transformed parts $S = \cup_i q_i(p_i)$ that forms a desired 3D shape, 2) all joints are matched, and the matched pegs J^p and holes J^h are close to each other.

Overview. Our multi-part multi-joint shape assembly task has several challenges 1) find the set of one-to-one peg-hole matching from a very *large matching search space* ($O(M^2)$), 2) predict poses for all parts such that they simultaneously achieve two objectives of *connecting all matched joints* and *forming desired shape structure*, 3) local joint matching or pose prediction errors can easily propergate to the entire shape and leads to degeneration.

To deal with the first challenge, we introduce a shape prior heuristic to reduce the matching search space. Inspired by previous works [15, 30], we use the part geometry information to propose an initial rough shape structure via a part graph. Then, our joint graph works with the rough shape structure to find an initial peg-hole matching. We address the second challenge by having the two levels of graph representation learning focusing on each of the two objectives. Joint graph module matches joints. Part graph constructs shape. We then combine the joint-level and part-level information using hierarchical feature aggregation to predict part poses subject to both objectives. Finally, we alleviate error compounding by iterative graph convolution to gradually refine part poses to meet the two objectives.

Part Graph Pose Proposal. The part graph aims to propose desired shape structure from a given set of part geometries. Inspired by [15, 30], we directly regress part poses from part geometries. Therefore, we initialize our part graph $\mathcal{G}^p = (\mathcal{V}^p, \mathcal{E}^p)$ by encoding part geometry $\mathcal{P}$ features on each graph node v_i^p and edges $e_{i,j}^p$ running between all part nodes. The part geometric features are extracted using PointNet [37]. In order to explicitly model the relationship between parts to form the desired shape, we use graph message passing, a mechanism for nodes to exchange information with its neighbors through edge connections. Part-level message passing is achieved through iteratively updating the edge features $e_{ij}^p = f_{v\to e}^p(v_i^p, v_j^p)$ and node features $v_i^{p\prime} = f_{e\to v}^p(v_i, \frac{1}{N}\sum_{j=1}^N e_{ij})$. We use the updated graph for pose prediction, as shown in the top section in Fig. 2. In the first iteration of part graph convolution, part pose vectors are decoded from the part node features, $q_i = f_{pose}(v_i^{p\prime})$. For any subsequent iterations, the pose vector $q_{i,t+1}$ is predicted given the previous step pose prediction and the updated node features $q_{i,t+1} = f_{pose}(v_i^{p\prime}, q_{i,t})$.

Joint Graph Relationship Reasoning. We use a joint graph to infer and refine joint connectivity relationships. As shown in the bottom section of Fig. 2, we first initialize joint node features v_i^j using PointNet to extract the joint-geometry feature vectors. The joint edges e_{ij} are initialized to be a set of bipartite edges running between all pegs nodes and all holes nodes $\mathcal{E}^j = \{e_{p,h}^j\}$ to reflect all possible allowed connections. We then use message passing to update the edge and node features iteratively. Specifically, we first update features of each edge e_{ij} with the neural messages calculated from its connected node features, $e_{ij} = f_{v\to e}(v_i, v_j)$. For the subsequent step, we update the node features v_i by aggregating information from all connected joint edges $v_i' = f_{e\to v}(v_i, \frac{1}{M}\sum_{j=1}^M e_{ij})$. We further update the joint node features v_i by explicitly modeling the joint connectivity relationships. Joint connectivity depends on two critical information, contact surface geometry and relative part positions. Therefore, we model the joint matching relationship from joint geometry p_i^{joint} and part position $\{q_i\}$. We learn a joint connectivity matrix $r_{ij} \in [0, 1]$ to reflect how joints

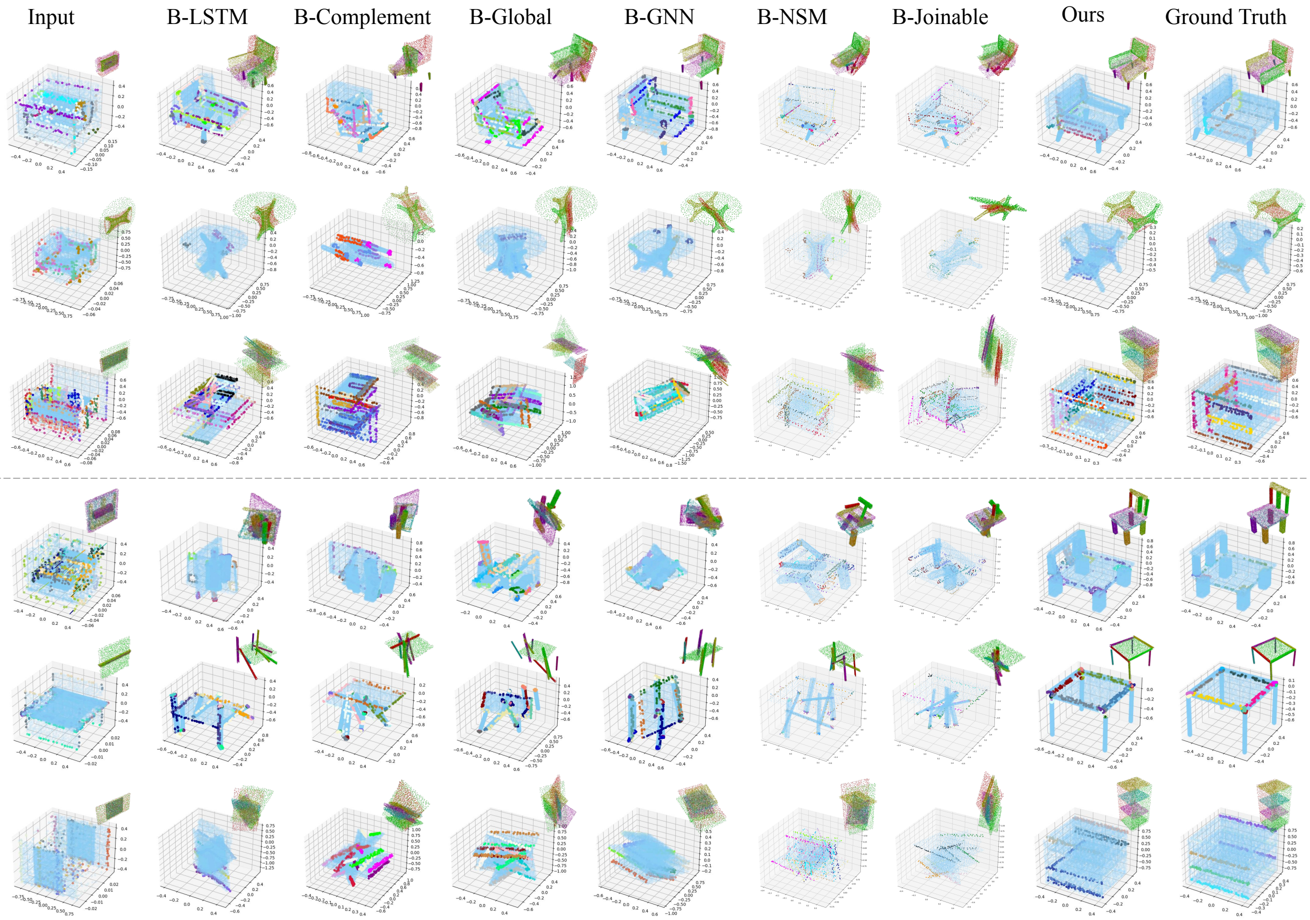

Figure 3. Qualitative comparison of our method and baselines (best viewed in color). We show the predictions in both the shape view (top-right corner) and the joint view (blue shapes), where the paired joint point sets are in the same color. In top three rows, the baselines are trained with their original setup. In bottom three rows, the baseline methods are trained with our joint input and loss. Directly imposing our proposed input and loss setting on baseline methods leads to collapsed shape prediction, whereas our proposed method produces the most structurally sound and joint optimized predictions.

are connected. The connectivity matrix is then used as edge weights applied to edge features e_{ij}, and we further update the joint nodes by aggregating the weighted edge features.

$$r_{ij}^{(t)} = f_{v\rightarrow r}\left(f_r\left(q_i; p_i^{joint}\right), f_r\left(q_j; p_j^{joint}\right)\right) \tag{1}$$

$$e'_{ij} = e_{ij} r_{ij}, \quad v''_i = f_{r\rightarrow v}\left(v'_i, \frac{\sum_j e'_{ij}}{\sum_j r_{ij}}\right). \tag{2}$$

Joint-Aware Pose Prediction. In order to generate part poses that simultaneously achieve both joint matching and shape structure objectives, we need to combine information from both the part graph and the joint graph. The joint-part relationship is hierarchical, since joints are the contacting locations on parts. We propose to model this relationship using hierarchical feature aggregation. Specifically, we use pooling operations on all relevant joint nodes for a part $\{v_t^j\}_{k=1}^{n_i}$ to form a new joint-centric part feature v_i^{p*}, as shown in section (c) in bottom of Figure 2.

$$v_i^{p*} = MaxPool(\{v_k^j\}_{k=1}^{n_i}), \tag{3}$$

These joint-aggregated part node features are then combined with the original part graph through part-wise feature concatenation for joint-aware pose prediction $v_i^{p'} = \{v_i^p; v_i^{p*}\}$, as shown in section (d) in Fig. 2. Now with the new part features $v_i^{p'}$ containing both joint and part information, we conduct the joint-aware pose proposal with the new updated part graph. Conditioning on part poses $\{q_{i,t}\}$ generated by previous graph iteration, we predict a refinement part pose operator $\delta q_{i,t+1} = f_{joint-pose}(v_i^{p'}|q_{i,t})$, as shown in section (e) in Fig. 2. The new pose prediction is compose of the predicted pose operator and previous stage part pose,

$$q_{i,t+1} = \delta q_{i,t+1}(q_{i,t}) = [\delta R_{i,t+1}(R_{i,t}), \delta t_{i,t+1} + t_{i,t}] \tag{4}$$

where the new rotation is calculated by applying the new rotation difference on the previous rotation prediction, and trans-

Setting	Method	Shape Chamfer Distance ↓				Part Pose Accuracy ↑			
		Chair	Table	Cabinet	Average	Chair	Table	Cabinet	Average
Original Setting	B-Global	0.015	0.013	0.008	0.013	32.8	30.1	33.6	31.4
	B-LSTM	0.017	0.026	0.007	0.021	39.4	22.5	44.4	30.8
	B-Complement	0.028	0.034	0.222	0.046	11.0	5.33	0.0	7.2
	B-GNN	0.007	0.008	0.006	0.007	65.3	61.4	45.0	61.7
	B-NSM	0.013	0.022	0.012	0.018	25.3	48.2	18.9	37.0
Our Full Setting (joint input and loss)	B-Global	0.029	0.022	0.013	0.024	5.4	12.0	15.0	9.6
	B-LSTM	0.037	0.029	0.017	0.031	4.4	4.1	15.3	5.1
	B-Complement	0.048	0.044	0.029	0.044	4.5	8.0	11.6	6.9
	B-GNN	0.034	0.039	0.021	0.036	11.5	3.2	10.4	7.0
	B-NSM	0.014	0.032	0.020	0.024	19.0	12.1	14.7	15.0
	B-Joinable	0.026	0.037	0.025	0.032	12.6	7.3	12.1	9.7
	Ours	**0.006**	**0.007**	**0.005**	**0.006**	**72.8**	**67.4**	**63.3**	**69.2**

Table 1. Quantitative comparison for the **Shape Structure Metrics** between our approach and the baseline methods under two settings. Original Setting on the top rows shows the performance of the baselines method with inputs and losses as originally proposed. Our full setting on the bottom rows shows the baseline performance with our joint-annotated part inputs and our joint-aware losses, same to our method. Down arrow indicates that lower numerical values corresponds to better performances. Up arrow means higher number is better.

lation is updated by adding the translation difference and previous translation prediction, more details in Appendix 7.2.

Loss Functions. We leverage two sets of loss functions: shape loss $\mathcal{L}_{shape}$ and joint loss $\mathcal{L}_{joint}$ to optimize our multistage graph network. $\mathcal{L}_{shape}$ aims to help the part graph network to generate valid shape structure, and $\mathcal{L}_{joint}$ helps the joint graph to match and connect all joints.

Shape loss We focus on the aspects of translation, rotation, and holistic shape structure in devising our shape loss $\mathcal{L}_{shape}$, where $\mathcal{L}_{shape} = \lambda_1\mathcal{L}_t + \lambda_2\mathcal{L}_r + \lambda_3\mathcal{L}_a$. We use $\mathcal{L}_2$ loss to supervise translation, and CD to supervise rotation and holistic shape structure.

$$\begin{aligned}
\mathcal{L}_t &= \sum_{i=1}^{N}||t_i - t_i^{gt}||_2^2, \\
\mathcal{L}_r &= \sum_{i=1}^{N} d_{chamfer}(R_i(p_i), R_i^{gt}(p_i)) \\
\mathcal{L}_a &= d_{chamfer}\left(\sum_{i=1}^{N}(q_i(p_i)), \sum_{i=1}^{N}\left(q_i^{gt}(p_i)\right)\right)
\end{aligned} \tag{5}$$

where Chamfer Distance (CD) is defined as [1]:

$$d_{chamfer}(a,b) = \sum_{x\in a}\min_{y\in b}||x-y||_2^2 + \sum_{x\in b}\min_{y\in a}||x-y||_2^2. \tag{6}$$

Additionally, as inspired by [30], we ensure our shape loss to be an order invariant loss metric to address the geometrically congruent parts, e.g. legs of a chair. Specifically, we perform Hungarian matching [22] within each congruent part class to supervise with the closest ground truth part pose.

Joint loss The joint matching task is very sensitive to prediction errors; one small matching error can lead to the deterioration of the entire shape. Therefore, we supervise for the joint matching objective in a coarse-to-fine manner with three loss components: $\mathcal{L}_{joint} = \lambda_4\mathcal{L}_{flip} + \lambda_5\mathcal{L}_{coarse} + \lambda_6\mathcal{L}_{fine}$. The first loss term $\mathcal{L}_{flip}$ directly corrects the flipped pose predictions. Inspired by [30], we use rotation L2 loss to correct upside-down predictions for parts with reflective symmetry:

$$\mathcal{L}_{flip} = \sum_{i=1}^{N}\left\|q_i(p_i) - q_i^{gt}(p_i)\right\|_F^2, \tag{7}$$

The second loss term $\mathcal{L}_{coarse}$ provides a coarse guidance to attach the matched joints. We use L2 distance between the matched pegs j_a^p and holes j_b^h. We use n_{joint} denotes the number of joint points,

$$\mathcal{L}_{coarse} = \sum_{\phi_i=1}^{M}||q_a(\bar{j}_a^p) - q_b(\bar{j}_b^h)||_2^2, \quad \bar{j}_a^p = \frac{1}{n_{joint}}j_a^p \tag{8}$$

The last loss component $\mathcal{L}_{fine}$ uses joint geometric cues to refine joint alignment. We use Chamfer Distance between the paired peg j_a^p and hole j_b^h with predicted poses applied,

$$\mathcal{L}_{fine} = d_{chamfer}(q_a(j_a^p), q_b(j_b^h)), \quad j_a^p \in p_a, j_b^h \in p_b \tag{9}$$

The latter two components of the joint losses are conditioned on a joint matching assignment $\Phi : \{\phi_i = (j_a^p, j_b^h)\}$. Since any arbitrary permutations in the congruent part class are also valid predictions, we cannot directly use the ground truth joint matching Φ_{gt} as our supervision signal. Therefore, to guarantee the order invariance of joint matching, we design a joint-matching algorithm with a graph traversal scheme to reassign matching of joints between congruent part classes (details in Alg. 1 in Supplementary Material).

Setting	Method	Joint Chamfer Distance ↓				Joint Matching Accuracy ↑			
		Chair	Table	Cabinet	Average	Chair	Table	Cabinet	Average
Original Setting	B-Global	0.712	0.847	0.667	0.780	13.4	15.8	10.7	14.5
	B-LSTM	0.756	0.728	0.651	0.733	17.0	13.2	14.8	14.8
	B-Complement	0.901	0.977	1.074	0.954	7.5	8.3	23.6	9.2
	B-GNN	0.725	0.855	0.683	0.791	24.4	30.0	18.6	26.9
	B-NSM	0.697	0.717	0.700	0.708	15.1	16.9	17.1	16.2
Our Full Setting (joint input and loss)	B-Global	0.513	1.268	0.488	0.912	12.7	4.0	6.9	7.6
	B-LSTM	0.394	0.875	0.467	0.655	20.3	7.7	13.8	13.1
	B-Complement	0.456	0.647	0.503	0.561	17.2	15.5	17.0	16.3
	B-GNN	0.379	0.786	**0.416**	0.598	21.5	10.3	20.0	15.4
	B-NSM	0.556	0.698	0.517	0.629	18.9	12.1	7.8	14.4
	B-Joinable	0.653	0.812	0.483	0.725	16.1	13.9	9.4	14.4
	Ours	**0.352**	**0.602**	0.620	**0.505**	**57.2**	**50.6**	**27.5**	**51.4**

Table 2. Quantitative comparison for the **Joint Matching Metrics** between our approach and the baseline methods under two settings. The two metrics reflects different aspect of the joint matching quality. Joint chamfer distance evaluates the average distance between matched peg-hole, but does not reflect whether joints are successfully aligned. Joint matching accuracy evaluates the number of aligned peg-hole pairs among all peg-hole joints for the shape. The bottom rows indicates that our joint input and loss setting helps baselines to lower the distance between all joints, but resulting in collapsed shapes and thus worse matching accuracy.

Remark. To tackle the intertwined problem of joint-centric part assembly, we propose an iterative hierarchical graph learning approach. We use two subgraph embeddings to focus on different aspects of the bilateral objectives. The part graph learns to predict and refine part poses to optimize the shape structure. The joint graph messaging passing discovers the joint-wise relationships. The two kinds of learned messages are combined to predict part poses to construct shapes that are both structurally sound and joint aligned.

4. Experiments

We introduce a joint-augmented part dataset as well as a set of evaluation metrics to examine both the shape-structure and joint-alignment aspects of the task performance. We compare with six re-purposed prior works to demonstrate that our proposed method is more effective. We conduct ablation studies to validate our design choices.

4.1. Dataset

We adapt the PartNet [36] for our task by augmenting the shapes parts with joint annotations. Following [30], we use the three largest furniture categories that requires real-world assembly, chairs, tables and cabinets, and adopt the PartNet official train/validation/test split. We use Furthest Point Sampling (FPS) to sample 1,000 points over each part mesh. All parts are canonicalized to be zero centered and rotated to be local axis aligned using PCA. We detect joint points by computing all pair-wise part Chamfer Distance (eq. 3) and take the closest 50 points between two connected parts with minimum distance less than 0.05. Following [15, 30], we use Level-3 granularity and filter out the shapes with more than 50 pairs of joints, which leaves us with 3736 chairs, 5053 tables, 719 cabinets.

4.2. Baseline Methods

Since our task is novel, there is no direct comparison from previous works that address the exact joint-alignment aspect of multi-part shape assembly. Instead, we re-purpose and adapt previous works as baselines to tackle our tasks as described below (implementation detail and additional baselines experiments can be found in Appendix 7.2, 7.4).
B-Complement: Sung et al. [42] tackles shape generation by retrieving part candidates from a large part repository. Following [15, 30], we modify it to tackle our task.
B-LSTM: Inspired by the sequential part generation work [12, 54], B-LSTM utilizes a LSTM backbone to sequentially decode part poses given previous part pose estimation.
B-Global: Inspired by [26, 39], B-Global augments part attributes with the global context when decoding part poses.
B-GNN: Previous works [12, 15] propose to use iterative graph neural network to assemble a variety of shapes given a part set. B-GNN follows [12, 15] to use dynamic graph learning for our joint-centric part assembly task.
B-NSM: Chen et al. [7] proposes a two part mating network by regressing part poses using transformer and adversarial training. B-NSM uses transformer with self-attention.
B-Joinable: Willis et al. [49] tackle the task of joint-centric assembly of *two* watertight volumetric parts by joint axis prediction. We adapt it for our task.

4.3. Evaluation Metric

Shape Evaluation Metric. Following [30], we adopt the two metrics of part accuracy *(Part Acc.)* and shape chamfer dis-

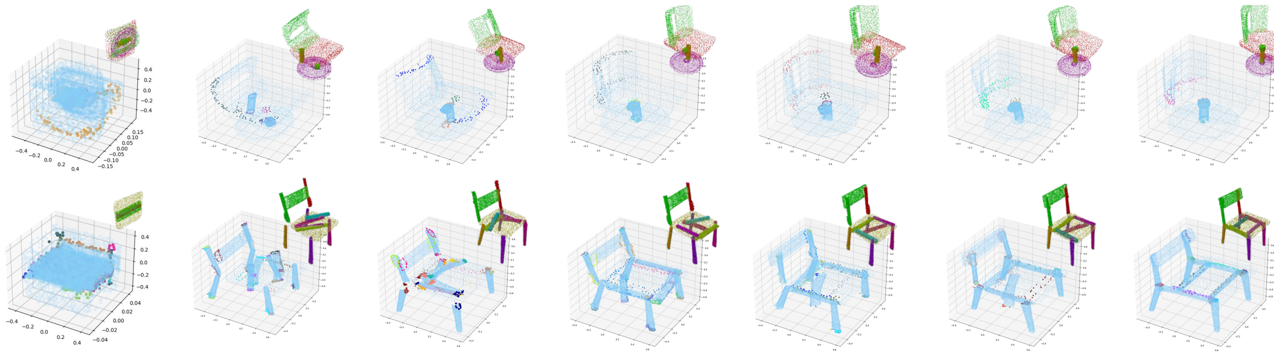

Figure 4. Examples of coarse-to-fine part pose prediction over different iterations of our network. The iterations are interleaved graph convolutions of part graph and joint graph.

tance *(Shape CD)* to evaluate the assembled shape structure. Part accuracy threshold τ_p is chosen to be 0.1.

$$\textit{Part Acc.} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[d_{chamfer}\left(q_i(p_i), q_i^{gt}(p_i)\right)\right] < \tau_p \tag{10}$$

$$\textit{shape CD} = \frac{1}{N \cdot n_{points}}\sum_{i=1}^{N} d_{chamfer}\left(q_i(p_i), q_i^{gt}(p_i)\right) \tag{11}$$

Joint Evaluation Metric. We propose two metrics for evaluating the second objective of the joint-centric part assembly task, joint accuracy *(Joint Acc.)* and joint chamfer distance *(Joint CD)*. Specifically, *Joint Acc.* evaluates how many joints are matched under the order-invariant joint matching algorithm defined by Alg. 1 (in Supplementary Material). It measures the percentage of joint pairs with Chamfer Distance under chosen threshold τ_j=0.01.

$$\textit{Joint Acc.} = \frac{1}{M}\sum_{\phi_i=1}^{M} \mathbb{1}\left[d_{chamfer}(q_b(j_b^h), q_a(j_a^p))\right] < \tau_j \tag{12}$$

where $\phi_i = (j_a^p, j_b^h)$ and $j_a^p \in p_a, j_b^h \in p_b$. Additionally, we also resort to joint Chamfer Distance metric to reflect the preciseness of part joint alignment by each method,

$$\textit{Joint CD} = \frac{1}{M}\sum_{\phi_i=1}^{M} d_{chamfer}\left(q_a(j_a^p), q_b(j_b^h)\right) \tag{13}$$

4.4. Results and Analysis

To guarantee the fairness of comparison, we devise two settings for the baseline experiments: (*I*) the setting in their original formulation, as also adopted in [12, 15, 30]; (*II*) baseline models trained with the same joint inputs and losses used in our model. The original task formulation of B-Joinable [49] also explicitly consider joints, so we show its results in setting II.

Setting I. By explicitly modeling joint connections, our method improves shape structure over previous methods that purely considers part geometries. Table 1 compares our proposed methods and baseline methods according to the shape structure metric. We can see from the table that our method consistently outperforms baseline methods on the shape metric. Qualitative evidence in Fig. 3 (shape structure in the top-right corner) also shows that our predictions are structurally similar to the ground truth, outperforming all the baselines. We can observe from the joint-centric view of the figure (blue shapes joint pairs are in the same color) that the baseline methods make flipped or upside-down predictions for parts with rotational and reflective symmetry, e.g. the chair armrest and seat poses predicted by B-GNN. Additionally, in their original proposed setting, the baseline methods only consider part geometries and can be confused by parts with similar geometries. Thus, they cannot determine the correct pose for these parts. For example, Fig. 3 shows, cabinets are made with boards of similar shapes. B-LSTM and B-Complement place all boards horizontally. Joints provide additional information for the functionality of each part. The two vertical side boards has multiple parallel joints, and the horizontal racks have joints at the two ends. Our method utilizes this information to achieve better shape structure.

Setting II. The performance of the baseline methods degrades significantly under the experimental setting II, compared with setting I. We can tell from Table 1 that the bottom section showing setting II is almost one-third of the performance of setting I, their original proposed setting, shown on the top section of the table. This demonstrates that our multi-part multi-joint task is challenging. Baseline methods cannot be directly adopted to tackle our task that involves two different objectives with two kinds of relationship reasoning. Similar phenomenon can be observed in Fig. 3, in which the bottom three rows show the baseline predictions under setting II. We can see that the baselines predict collapsed shapes. This is because when all parts are clustered

	SCD ↓	PA ↑	JCD ↓	JA ↑
w/o 1st iter of part graph	0.025	23.71	0.572	33.86
w/o joint embedding	0.019	33.36	0.258	36.90
w/o matching alg.	0.010	55.10	0.416	37.50
w/o $\mathcal{L}_{flip}$	0.007	68.44	0.436	39.71
w/o $\mathcal{L}_{coarse}$	0.006	66.65	0.381	44.82
w/o $\mathcal{L}_{fine}$	0.006	63.72	0.377	52.01
Ours full	0.006	72.81	0.352	57.18

Table 3. Ablation study conducted on the Chair category. SCD denotes Shape Chamfer Distance; PC denotes Part Accuracy; JCD denotes the Joint Chamfer Distance, and JA denotes Joint Accuracy. Arrows indicates the direction of better performance.

together, their joint distances are also decreased. This type of collapsed prediction is a local minimum for the joint losses.

This observation gives us insights into the loss landscape of the two task objectives. The shape structure objective aims to spread out the part geometries to various locations. The joint alignment objective aims to connect and contact parts together. For any inaccurate pose predictions, these two objectives conflict with each other. Shape structure wants to expand parts, whereas joint matching wants to contract parts. There exists only one set of pose predictions that simultaneously satisfy both task objectives–that is, the global optimum. This explains why most baseline methods fail when they are subjected to both objectives at the same time. The local minimum for the joint matching objective can easily trap pose predictions for baseline methods from achieving valid shape structures. However, our method maintains the valid shape structure using a coarse-to-fine scheme, and thus is less prone to stuck in the local minimum of collapsed shape.

In Figure 4, we show our coarse-to-fine assembly scheme by visualizing the predicted shape structures in the intermediate stages of our graph convolution. We observe that the first iteration of part graph learns to predict a rough shape structure. The subsequent joint graph iteration learns to modify part poses so more parts can connected with each other. Another iteration of part graph then learns to refine part poses with the corrected joint matching. Eventually, through these iterations of graph convolution, we can produce structurally sound and joint matched part assemblies. We use the part-joint-part-joint-joint graph combination, as we discovered that this combination works best empirically.

Ablation Study. We conduct three ablation experiments to demonstrate the effectiveness of different design choices of our proposed approach, as shown in Table 3. We first test our network design of using a part graph module as the first iteration of our network. We believe that the rough shape structure proposed by the first part graph module can serve as a shape structure heuristic to reduce the joint matching difficulty. As shown on the top row in Table. 3. Removing the first iteration of the part graph by directly having the joint graph to propose joint matching solutions significantly reduce our performance, and hence verifies our conjecture of shape structure heuristic on joint matching.

Our second ablation experiment aims to test the importance of the joint embedding, as shown on the second row in Table. 3. We remove the joint embedding step and applying the joint losses on the last stage part graph directly. The result shows a significant performance decrease in *Shape CD* and *Part Accu.*. This setting is similar to experimental setting II by directly adding joint losses to baseline methods. This shows that joint embedding is a non-trivial component to our network that provides more explicit joint-alignment pose editing signal.

We then test our method without the matching algorithm in the loss scheme, as described in Alg. 1 in Appendix. We observe that the performance decreases significantly across all metrics. This is because Hungarian-matching allows shape losses to be permutation invariant for geometrically congruent parts, the joint matching algorithm maintains this order-invariance property for the joint losses. Alg. 1 finds a new matching assignment considering permutations within the congruent part class, granting consistency between the two loss objectives. Without such consistency, the two losses are not synchronized and work in different directions, and thus result in problematic part pose predictions.

5. Conclusion and Future Work

We formulate a novel variant of the category-level multi-part 3D shape assembly problem by introducing the concept of joints. We focus on the peg-hole abstraction of part joints and proposed a hierarchical graph network approach that consists of a joint embedding and a part embedding for explicit hierarchical relationship reasoning to tackle the challenges. We introduce a joint-augmented multi-part assembly dataset along with evaluation metrics to set up the test bed for this task. We also provide extensive empirical evidence to demonstrate the effectiveness of our approach compared to the re-purposed prior works. We believe our work can on autonomous assembly systems.

As a start of the multi-part multi-joint assembly problem, we focus on the simple but commonly-used peg-hole joints. There are several possible scenarios that are not considered in our paper and are left for future work. One future direction is to extend to more complicated joints for this problem, and construct a general formulation for all possible joint types. Another future direction is programmatic or sequential planning for joint alignment, which would better enable vision algorithms to be deployed in autonomous systems.

6. Acknowledgement

We thank Jiayuan Mao, Liane Makatura, and Tongzhou Wang for the helpful writing suggestions. We also thank Karl Willis, Yunsheng Tian, and Xingchao Peng for the useful discussion. This work is done while the first author was at Stanford.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J Guibas. Learning representations and generative models for 3d point clouds. 2018.

[2] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. *ACM Trans. Graph.*, 30(4), 2011.

[3] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas J. Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *ACM SIGGRAPH*, 2011.

[4] Siddhartha Chaudhuri and Vladlen Koltun. Data-driven suggestions for creativity support in 3d modeling. In *ACM SIGGRAPH Asia 2010 papers*, pages 1–10. 2010.

[5] Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. Learning generative models of 3d structures. In *Comput. Graph. Forum*, volume 39, pages 643–666. Wiley Online Library, 2020.

[6] Vincent S Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2580–2590, 2019.

[7] Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

[9] Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomir Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. Learning generative models of shape handles. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 402–411, 2020.

[10] 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 Trans. Graph.*, 38(6), 2019.

[11] Songfang Han, Jiayuan Gu, Kaichun Mo, Li Yi, Siyu Hu, Xuejin Chen, and Hao Su. Compositionally generalizable 3d structure prediction. *2012.02493*, 2020.

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

[13] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Trans. Graph.*, 36(6):1–13, 2017.

[14] Haibin Huang, Evangelos Kalogerakis, and Benjamin Marlin. Analysis and synthesis of 3d shape families via deep-learned generative models of surfaces. In *Eurographics Symposium on Geometry Processing*, 2015.

[15] Jialei Huang, Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, and Hao Dong. Generative 3d part assembly via dynamic graph learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6315–6326. Curran Associates, Inc., 2020.

[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.

[17] Pablo Jiménez. Survey on assembly sequencing: a combinatorial and geometrical perspective. *Journal of Intelligent Manufacturing*, 24(2):235–250, 2013.

[18] 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 Trans. Graph.*, 39(6), 2020.

[19] 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.

[20] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9601–9611, 2019.

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[23] Manfred Lau, Akira Ohgawara, Jun Mitani, and Takeo Igarashi. Converting 3d furniture models to fabricatable parts and connectors. *ACM Trans. Graph.*, 30(4):1–6, 2011.

[24] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *ICRA*, 2019.

[25] Youngwoon Lee, Edward S Hu, Zhengyu Yang, Alex Yin, and Joseph J Lim. Ikea furniture assembly environment for long-horizon complex manipulation tasks. *1911.07246*, 2019.

[26] Jun Li, Chengjie Niu, and Kai Xu. Learning part generation and assembly for structure-aware shape synthesis. In *AAAI*, volume 34, pages 11362–11369, 2019.

[27] 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.

[28] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019.

[29] Yichen Li, Peter Yichen Chen, Tao Du, and Wojciech Matusik. Learning preconditioners for conjugate gradient pde solvers. In *International Conference on Machine Learning*, pages 19425–19439. PMLR, 2023.

[30] 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.

[31] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1261–1270, 2017.

[32] Minmin Lin, Tianjia Shao, Youyi Zheng, Niloy Jyoti Mitra, and Kun Zhou. Recovering functional mechanical assemblies from raw scans. *IEEE Transactions on Visualization and Computer Graphics*, 24(3):1354–1367, 2017.

[33] Katia Lupinetti, Jean-Philippe Pernot, Marina Monti, and Franca Giannini. Content-based cad assembly model retrieval: Survey and future challenges. *Computer-Aided Design*, 113:62–81, 2019.

[34] Marco Marconi, Giacomo Palmieri, Massimo Callegari, and Michele Germani. Feasibility study and design of an automatic system for electronic components disassembly. *Journal of Manufacturing Science and Engineering*, 141(2), 2019.

[35] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. StructureNet: Hierarchical graph networks for 3d shape generation. In *ACM SIGGRAPH Asia*, 2019.

[36] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 909–918, 2019.

[37] 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.

[38] Daniel Ritchie, Kai Wang, and Yu-an Lin. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6182–6190, 2019.

[39] Nadav Schor, Oren Katzir, Hao Zhang, and Daniel Cohen-Or. Componet: Learning to generate the unseen by part synthesis and composition. In *Int. Conf. Comput. Vis.*, pages 8759–8768, 2019.

[40] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012.

[41] Francisco Suárez-Ruiz, Xian Zhou, and Quang-Cuong Pham. Can robots assemble an ikea chair? *Science Robotics*, 2018.

[42] Minhyuk Sung, Hao Su, Vladimir G Kim, Siddhartha Chaudhuri, and Leonidas Guibas. Complementme: weakly-supervised component suggestions for 3d modeling. *ACM Trans. Graph.*, 36(6):1–12, 2017.

[43] Garrett Thomas, Melissa Chien, Aviv Tamar, Juan Aparicio Ojea, and Pieter Abbeel. Learning robotic assembly from cad. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3524–3531. IEEE, 2018.

[44] 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.

[45] Kai Wang, Paul Guerrero, Vladimir G. Kim, Siddhartha Chaudhuri, Minhyuk Sung, and Daniel Ritchie. The shape part slot machine: Contact-based reasoning fornbsp;generating 3d shapes fromnbsp;parts. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, page 610–626, Berlin, Heidelberg, 2022. Springer-Verlag.

[46] Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

[47] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8876–8884, 2019.

[48] Ziqi Wang, Peng Song, and Mark Pauly. State of the art on computational design of assemblies with rigid parts. 40(2), 2021.

[49] Karl DD Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Joinable: Learning bottom-up assembly of parametric cad joints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[50] Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Trans. Graph.*, 40(4), 2021.

[51] 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*, pages 14311–14320, 2023.

[52] Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models. In *Int. Conf. Comput. Vis.*, pages 6772–6782, October 2021.

[53] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. PQ-NET: A generative part seq2seq network for 3d shapes, 2019.

[54] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. *CVPR*, 2020.

[55] 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.

[56] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017.

[57] Xianghao Xu, Wenzhe Peng, Chin-Yi Cheng, Karl D.D. Willis, and Daniel Ritchie. Inferring cad modeling sequences using zone graphs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6062–6070, June 2021.

[58] Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen,

Oliver van Kaick, Hao Zhang, and Hui Huang. Rpm-net: Recurrent prediction of motion and parts from point cloud. 38(6):240:1–240:15, 2019.

[59] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018.

[60] Kangxue Yin, Zhiqin Chen, Siddhartha Chaudhuri, Matthew Fisher, Vladimir Kim, and Hao Zhang. Coalesce: Component assembly by learning to synthesize connections. *2008.01936*, 2020.

[61] Kevin Zakka, Andy Zeng, Johnny Lee, and Shuran Song. Form2fit: Learning shape priors for generalizable assembly from disassembly. 2019.

[62] Youyi Zheng, Daniel Cohen-Or, and Niloy J Mitra. Smart variations: Functional substructures for part compatibility. In *Computer Graphics Forum*, volume 32, pages 195–204. Wiley Online Library, 2013.

[63] Yang Zhou, Zachary While, and Evangelos Kalogerakis. Scenegraphnet: Neural message passing for 3d indoor scene augmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7384–7392, 2019.