3D Assembly Completion

Weihao Wang, Rufeng Zhang, Mingyu You*, Hongjun Zhou, Bin He

College of Electronic and Information Engineering, Tongji University, Shanghai 201804 China
{wwhtju,myyou,zhouhongjun,hebin}@tongji.edu.cn, cxrfzhang@foxmail.com

Abstract

Automatic assembly is a promising research topic in 3D computer vision and robotics. Existing works focus on generating assembly (*e.g.*, IKEA furniture) from scratch with a set of parts, namely 3D part assembly. In practice, there are higher demands for the robot to take over and finish an incomplete assembly (*e.g.*, a half-assembled IKEA furniture) with an off-the-shelf toolkit, especially in human-robot and multi-agent collaborations. Compared to 3D part assembly, it is more complicated in nature and remains unexplored yet. The robot must understand the incomplete structure, infer what parts are missing, single out the correct parts from the toolkit and finally, assemble them with appropriate poses to finish the incomplete assembly. Geometrically similar parts in the toolkit can interfere, and this problem will be exacerbated with more missing parts. To tackle this issue, we propose a novel task called 3D assembly completion. Given an incomplete assembly, it aims to find its missing parts from a toolkit and predict the 6-DoF poses to make the assembly complete. To this end, we propose FiT, a framework for **F**inishing the **i**ncomplete 3D assembly with **T**ransformer. We employ the encoder to model the incomplete assembly into memories. Candidate parts interact with memories in a memory-query paradigm for final candidate classification and pose prediction. Bipartite part matching and symmetric transformation consistency are embedded to refine the completion. For reasonable evaluation and further reference, we design two standard toolkits of different difficulty, containing different compositions of candidate parts. We conduct extensive comparisons with several baseline methods and ablation studies, demonstrating the effectiveness of the proposed method.

Introduction

Automatic assembly is a desirable capability of intelligent robot. Everyone dreams of owning *Andrew*, the android in *Bicentennial Man*, that can not only assemble a new Lego as your birthday surprise but also maintain your beloved toy which is broken, to mend your broken heart.

At present, for a robot, assembling a Lego or even furniture is coming true. Existing works have enabled the robot with the ability to assemble a set of parts into a whole from scratch, namely 3D part assembly (Zhan et al. 2020). Maintaining a broken toy, or in general, finishing an incomplete assembly somehow remains unexplored yet. It is a fundamental but essential task in human-robot and multi-agent cooperative assembly. Take as an example the robot cooperates with humans in assembling a complex IKEA furniture. Rather than replacing humans to complete the work, it will be more welcome to assist with the assembly. On the multi-agent assembly line, human-like cooperation requires the agent to take over various work-in-process assemblies which are half-assembled by other agents, and try to finish the assembly in its power. With such an ability, maintenance can be a piece of cake, *e.g.*, to repair a broken chair that lacks a leg, and the dream of *Andrew* will soon come true.

Despite its great significance, it is a challenging task to finish an incomplete assembly without any instruction manual or external guidance. Unlike 3D part assembly, which focuses on inter-part relation modeling, the robot needs a more detailed comprehension of the incomplete structure and infer part-to-structure relationships for the candidate parts. Moreover, for a particular missing part, the toolkit may contain multiple confusing candidates that are similar in geometry but distinct in functionality (*i.e.*, invalid for assembly), which can interfere with the correct selection. This issue can be further exacerbated when multiple parts are missing.

To tackle this issue, we propose 3D assembly completion, a novel task that aims to single out the correct missing parts from a toolkit of candidates and predict the 6-DoF poses to complete the assembly. To this end, we propose a framework named **FiT**, for **F**inishing the **i**ncomplete 3D assembly with **T**ransformer. We model the incomplete assembly into memories with the encoder and take the candidate parts modeled by the decoder as queries to interact with memories, completing multiple tasks of candidate classification and pose prediction.

During training, we design bipartite part matching to designate a unique candidate for each missing part, avoiding the conflict of multiple correct candidates in the toolkit. We also propose symmetric transformation consistency to approximate stability optimization for generating more stable and practical assemblies.

We design two standard toolkits for reasonable evaluation and comparison, namely the original toolkit and the blended toolkit. They contain candidate parts of different numbers, geometry, and thus difficulties. We conduct extensive experiments both qualitatively and quantitatively to demonstrate

*Corresponding author.

the effectiveness of the proposed method, which outperforms several baseline approaches by a large margin and formulates a strong baseline for the novel task of 3D assembly completion.

Related Work

Component suggestion and composition. Given a partial shape, a straightforward idea is to retrieve reasonable components from a large shape database (Xie et al. 2013; Vranic and Saupe 2004; Wang et al. 2012; Saupe and Vranić 2001). ComplementMe (Sung et al. 2017) learns the partial shape with components and non-components by contrastive (Khosla et al. 2020; Ghojogh et al. 2020) learning jointly in the embedding space. (Sung et al. 2018) shares a similar setting, which focuses on modeling complementarity and interchangeability of components in dual embedding spaces. Although common in partial input, the components to be retrieved are presented by part groups (*e.g.*, four legs of a chair as a component), which are more discriminative than finest-grained parts which denote the setting in 3D assembly completion. Moreover, only translation is considered which is naive. SCORES (Zhu et al. 2018) represents and completes missing parts as oriented bounding boxes (OBBs), which lose detailed geometric information. Since geometric-similar parts may share the same OBB, OBBs (Li et al. 2017) are not suitable for point-cloud-based part assembly. COALESCE (Yin et al. 2020) aims to synthesize well-connected shape by learning to translate and scale components from different shapes, while 3D assembly completion focus on the more complicated 6-DoF poses prediction.

3D part assembly. 3D part assembly is a typical task which aims to generate fully-assembled 3D shapes with a set of determined parts from scratch (Chaudhuri et al. 2011; Jaiswal, Huang, and Rai 2016; Kalogerakis et al. 2012). DGL (Zhan et al. 2020) is a pioneering work that adopts the graph neural network to reason about the part relations in a complete shape iteratively. Similar in task definition, RGL-NET (Harish, Nagar, and Raman 2022) leverages a recurrent graph learning paradigm to refine the relations of parts progressively for more accurate pose prediction. PQ-Net (Wu et al. 2020) proposes a part-wise Seq2Seq generative framework that models the parts by 3D shape reconstruction. PageNet (Li, Niu, and Xu 2020) is also a generative framework that consists a part generator based on VAE-GANs (Gulrajani et al. 2017; Adler and Lunz 2018) and a part assembler, to perform a two-stage part-wise assembly. There are also other works that utilize diverse sources of information as input, such as image (Li et al. 2020). Overall, these works generally take as input a full set of parts for an assembly, and predict the 6-DoF poses for them, where inter-part relation modeling is essential in the design of method. The novel task of 3D assembly completion naturally holds a different problem setup and remains unsolved in literature.

3D shape completion. Another related task is 3D shape completion, which mainly studies the problem of point-cloud-based completion (Yu et al. 2021; Wen et al. 2020; Yu et al. 2022). Previous methods like TopNet (Tchapmi et al. 2019), PCN (Yuan et al. 2018), and SA-Net (Wen et al. 2020) commonly adopt a generative solution with an encoder-decoder framework. An encoder (*e.g.*, PointNet (Qi et al. 2017)) is utilized to extract the global feature of incomplete shape. Accordingly, a decoder is designed to infer the complete point clouds. Other works like RL-GAN-Net (Sarmad, Lee, and Kim 2019) and Render4Completion (Hu et al. 2019) follow the framework of adversarial learning to improve the reality and consistency of the completion. In general, 3D shape completion aims to recover a set of point clouds without exact geometry or boundary. 3D assembly completion focuses on part-level completion, where a part is a set of point clouds with a specific geometry and semantic meaning. Recent works (Yan et al. 2022; Mittal et al. 2022) also adopt transformers (Vaswani et al. 2017) for global modeling of point distribution. Transformer shows effectiveness in point clouds relation modeling while we focus on a more coarse level of parts.

Method

Let $p \in \mathbb{R}^{n_d \times 3}$ denotes a set of 3D part point clouds. Given an incomplete assembly $\mathcal{A} = \{p_i^*\}_{i=1}^{N-k}$ with $N-k$ parts, where N is the total number of parts in the complete assembly, our goal is to single out its missing parts $\mathcal{M} = \{p_i\}_{i=1}^{k}$ from a toolkit $\mathcal{T} = \{p_i\}_{i=1}^{M}$ with M candidates, and predict the poses $\{(R_i, t_i)\}_{i=1}^{k}$ for the selected parts.

Here (R, t) denotes the 6-DoF pose of part in $SE(3)$ space. Individually, $R \in \mathbb{R}^4$ denotes the rigid rotation, represented by unit quaternion where $\|R\|_2 = 1$ and $t \in \mathbb{R}^3$ denotes the translation. Finally, the selected parts are assembled with the incomplete assembly as $\mathcal{A} \cup \left(\cup_i^k T_i\left(p_i\right)\right)$, where T_i represents the joint transformation of pose (R_i, t_i).

To tackle this problem, we propose a transformer-based framework called FiT, which contains three major components as illustrated in Figure 1. The encoder models the incomplete assembly into memories. The decoder takes candidates of the toolkit as queries to interact with memories. Finally, two predictors are designed for candidate classification and pose prediction to complete the assembly.

Toolkit Construction

Before providing detailed designs of FiT, we first introduce the toolkit, which is an indispensable component for 3D assembly completion. For android *Andrew*, he must select the appropriate parts from his magic box to maintain the broken toy. In 3D assembly completion, such a service kit is also required. We define it as toolkit. In this paper, a toolkit is a set of candidate parts, which must contain the ground-truth missing parts $\mathcal{M}$. Besides, it also contains some other parts. Considering practical situations, we propose two typical and reasonable toolkit configurations.

Original toolkit. Given a broken toy, the most intuitive solution is to find its missing parts from an original duplicate. Similarly, considering an assembly $\mathcal{S}$, we define its original toolkit as $\mathcal{T}_{orig} = \mathcal{M} \cup \{p_i | p_i \in \mathcal{S}\}_{i=1}^{m_1}$ which contains m_1 parts randomly sampled from $\mathcal{S}$ and k missing parts, thus $m_1 + k = M$. We adopt a balanced sampling strategy

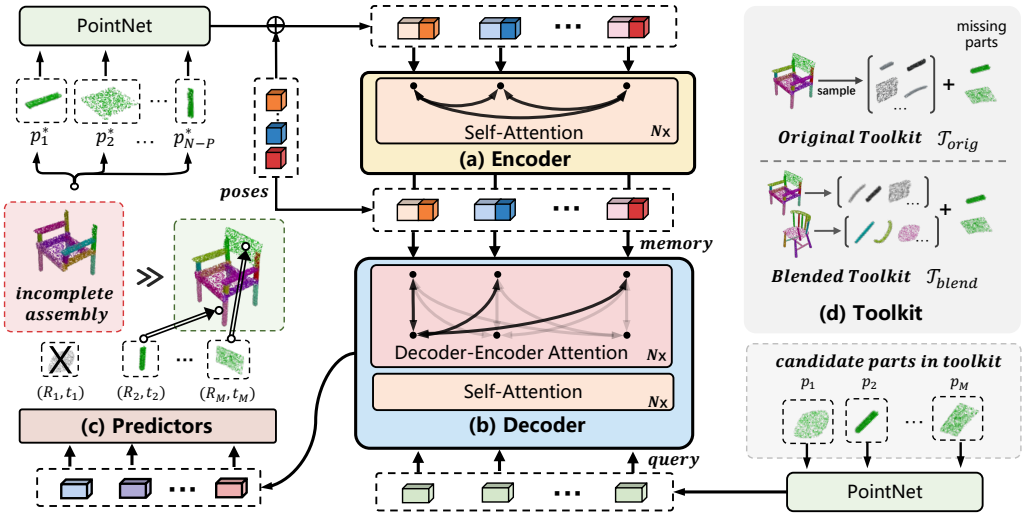

Figure 1: The pipeline of FiT, containing three major components:(a) an encoder to encode the parts of incomplete assembly, (b) an decoder for relation modeling between candidate parts and incomplete shape, (c) two predictors for candidate selection and pose prediction, individually.

according to the geometric class of parts, to avoid sampling on the major class repeatedly (*e.g.*, leg is the major part in number for chairs).

Blended toolkit. Besides taking a duplicate, it is more common cases to search in a box, blended with scattered parts that are taken from the toy itself, or other worn toys. Likewise, considering an assembly $\mathcal{S}$ and another assembly $\mathcal{Q}^-$, we define the blended toolkit as $\mathcal{T}_{blend} = \mathcal{M} \cup \{p_i | p_i \in \mathcal{S}\}_{i=1}^{m_1} \cup \{q_i | q_i \in \mathcal{Q}^-\}_{i=1}^{m_2}$, where p_i and q_i are randomly sampled from $\mathcal{S}$ and $\mathcal{Q}^-$ respectively, and $m_1 + m_2 + k = M$. $\mathcal{Q}^-$ denotes a negative assembly which introduces more disturbances. We consider total assemblies as a circular queue and designate the former of each assembly as its negative assembly.

Both toolkits consistently contains the correct missing parts and sometimes more than one. Figure 1(d) visualizes an example of toolkit construction. As is shown in the case, an assembly naturally contains lots of parts that are extremely similar in geometry (*e.g.*, the leg and the bar that supports the back, see Figure 1(d)), making it challenging to single out the correct ones. The blended toolkit can introduce more confusing parts and further increase the difficulty. We provide more detailed insights into the toolkits in the supplementary material.

Architecture of FiT

The transformer takes as input the part point clouds of incomplete shape $\mathcal{A}$ and toolkit $\mathcal{T}$. Since a part point cloud is a set of n_d points that are unorderly sampled, we first use a vanilla PointNet (Qi et al. 2017) to extract a global permutation-invariant feature for each part cloud. The PointNet features are sent to the encoder and decoder for further relation modeling. Note that the part clouds of incomplete assembly are transformed with ground-truth poses before being sent to the PointNet, representing they are pre-assembled and the candidate part clouds remain in canonical space.

Encoder. The structure of incomplete assembly provides essential evidence for reasoning about the missing parts. We use the transformer encoder to capture such structural information. Detailed, the encoder takes as input the PointNet features of the parts in incomplete assembly. We employ self-attention mechanism computed by Eq.1 to model both geometry and pose relations of the parts, where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{(N-k)\times 256}$ denote query, key and value, respectively. The part-to-part interaction is conducted through multiple self-attention layers in a progressive manner. Finally, the parts of incomplete shape are encoded into memories to formulate a memory bank for further interaction.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\mathbf{Q}\mathbf{K}^T / \sqrt{d_k}\right)\mathbf{V} \quad (1)$$

Worth noting that the PointNet is more sensitive to variation in geometry but less in pose. This may cause pose information degradation during self-attention modeling. To this end, we concatenate the parts of incomplete shape with their poses at the input of self-attention layers in encoder and the memory (*i.e.*, the output of encoder), to reinforce pose-aware information.

Decoder. The decoder takes as input the PointNet features of the candidate parts in toolkit. Different toolkits may contain different numbers of candidates, with different distributions of geometry. We first enable self-attention mechanism among the candidate parts, which capture the relations (*i.e.*, geometry distribution) of candidates. Then the candidate parts serve as queries to calculate multi-head attention with memories from the encoder to model the relationships between the candidates and the incomplete structure. Multi-head attention is computed by Eq.1 in a similar way, with $\mathbf{Q} \in \mathbb{R}^{M\times 256}$ obtained from candidates and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{(N-k)\times 256}$ from

memories. With such interaction mechanism, we incorporate the global information of incomplete assembly into the queries, which is essential in the prediction of correct candidates. We apply multi-head attention layers of standard transformer for both self- and decoder-encoder attention.

Positional encoding. Since the transformer is permutation-invariant to the input parts, we adopt position encoding as a supplementary. Following (Zhang et al. 2022), we design part-aware position encoding, a vector that contains intra-class encoding and inter-class encoding. Instead of the vanilla position encoding, we concatenate the input features with the part-aware position encoding. Detailed computation is described in the supplementary material.

Predictors. We design two predictors for the final tasks. One predictor is designed to perform binary classification for the candidate parts. we use $\mathrm{sigmoid}$ operation to output a classification score in $(0, 1)$ for each candidate, which indicates the possibility of being the correct missing part. Another predictor is used to predict 6-DoF poses for the candidates. We employ a 3-layer feed-forward network for both predictors, including two fully-connected layers with a hidden size of 256 and ReLU activation function and a linear projection layer.

Multi-Task Learning of FiT

We consider 3D assembly completion as multiple tasks of classifying the candidates into the correct missing parts or not and predicting poses for the correct candidates. During training, we propose bipartite part matching and symmetrical transformation consistency to refine the completion.

Bipartite part matching. The assembly model infers k parts from a fixed-size toolkit of M candidates in candidate classification, where M is usually more extensive than k. A straightforward idea is to supervise all correct candidates with label 1 and the others 0. However, such indiscriminate supervision may confuse with multiple missing parts. Take as an example a chair that misses a leg and a back. If the toolkit contains several correct leg candidates, the model possibly overfits selecting legs repeatedly. Instead, it is desirable that only one of the leg candidates is matched for supervision.

Inspired by DETR (Carion et al. 2020), we propose bipartite part matching to designate the best-matched candidate for each missing part, before computing the classification loss. Given the ground-truth missing parts $\mathcal{M}$ and a set of candidate parts $\mathcal{C}$, bipartite part matching searches for a subset of candidates $\mathcal{C}^*$ to minimize the cost:

$$\mathcal{C}^* = \underset{\hat{\mathcal{C}} \subseteq \mathcal{C}}{\arg\min} \sum_{i}^{k} \mathcal{L}_{match} \left(\mathcal{M}_i, \mathcal{C}_i^* \right). \tag{2}$$

Here, we employ Chamfer distance (CD) as the matching cost $\mathcal{L}_{match}$ to obtain the optimal assignment between the parts of two sets. The optimal assignment is efficiently computed with the Hungarian algorithm following (Stewart, Andriluka, and Ng 2016). Chamfer distance is calculated between two part clouds $\mathcal{X}$ and $\mathcal{Y}$, formally,

$$d_c(\mathcal{X}, \mathcal{Y}) = \sum_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \|x - y\|_2^2 + \sum_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \|x - y\|_2^2. \tag{3}$$

Finally, we supervise the candidate parts with the binary cross-entropy loss:

$$\mathcal{L}_c = -\sum_{i}^{M} y_i^* \log y_i + (1 - y_i^*) \log (1 - y_i), \tag{4}$$

where y denotes the prediction and y^* denotes the ground-truth label. For candidate part p_i, $y_i^* = 1$ if $p_i \in \mathcal{C}^*$; else, $y_i^* = 0$.

Approximate stability optimization. Stability is the primary principle in the design of engineering structures. So is in assembly. The stability criteria is established in order to determine whether a structure is in stable equilibrium with a given set of loadings. Based on energy methods (Yoo and Lee 2011), given a virtual displacement, the structure is stable equilibrium if and only if $\Delta\Pi > 0$, where Π denotes the total potential energy and $\Delta\Pi$ the change of total potential energy. We provide formula expression and an example of a 2-DoF system in the supplementary material.

However, it is infeasible to apply potential energy analysis for stability evaluation on 3D point-cloud-based assembly. Alternatively, we find that a symmetric skeletal structure is proven to retain its geometric stability while its geometry or connectivity is altered (Chen, Sareh, and Feng 2015). In other words, a symmetric structure tends to be more stable in nature and robust to disturbances. In practice, most functional objects (*e.g.*, chairs and tables) obey a design principle of symmetry to maintain structure-aware stability.

Inspired by this, we propose symmetric transformation consistency to approximate stability optimization in training. It ensures the assembled shape remains symmetrical in structure and can be easily achieved by measuring the Chamfer distance in Eq.3 between the assembled shape $\mathcal{S}$ and its mirror image $\Phi(\mathcal{S})$, formally,

$$\mathcal{L}_{sym} = d_c \left(\mathcal{S}, \Phi(\mathcal{S}) \right), \tag{5}$$

where Φ denotes the mirror transformation along the ground-truth symmetry plane which is pre-computed.

Loss components. We apply straightforward supervision on pose prediction. Each predicted pose can be decomposed into translation t and rotation R. Given the ground-truth translation t_i^* of the missing parts, the translation is supervised with an $\mathcal{L}2$ loss:

$$\mathcal{L}_t = \sum_{i=1}^{k} \|t_i - t_i^*\|_2^2. \tag{6}$$

Rotation is supervised by Chamfer distance between the part clouds with predicted rotation $R_i(p_i)$ and ground-truth rotation $R_i^*(p_i)$:

$$\mathcal{L}_r = \sum_{i=1}^{k} d_c \left(R_i(p_i), R_i^*(p_i) \right). \tag{7}$$

We use shape Chamfer distance between the completed assembly $\mathcal{S}$ and the ground-truth assembly $\mathcal{S}^*$ to measure the overall quality of assembly completion:

$$\mathcal{L}_s = d_c \left(\mathcal{S}, \mathcal{S}^* \right). \tag{8}$$

k	Toolkit	MA↑			SCD↓			PA↑			CA↑		
		Chair	Table	Lamp	Chair	Table	Lamp	Chair	Table	Lamp	Chair	Table	Lamp
1	original	92.93	91.14	90.22	0.0069	0.0049	0.0250	70.71	76.78	30.39	62.22	71.83	41.61
	blended	81.04	84.89	54.55	0.0086	0.0052	0.0339	64.91	74.40	21.90	59.29	75.32	39.23
2	original	61.36	66.54	59.19	0.0240	0.0123	0.0424	44.16	52.61	20.17	36.47	50.22	31.11
	blended	49.46	45.58	24.86	0.0286	0.0164	0.0556	39.51	36.34	8.732	34.93	39.03	26.75
3	original	60.41	65.26	-	0.0232	0.0126	-	41.81	46.74	-	32.01	42.37	-
5	original	59.47	62.97	-	0.0149	0.0076	-	35.21	43.14	-	28.82	45.13	-

Table 1: Quantitative results on different toolkits with k missing parts.

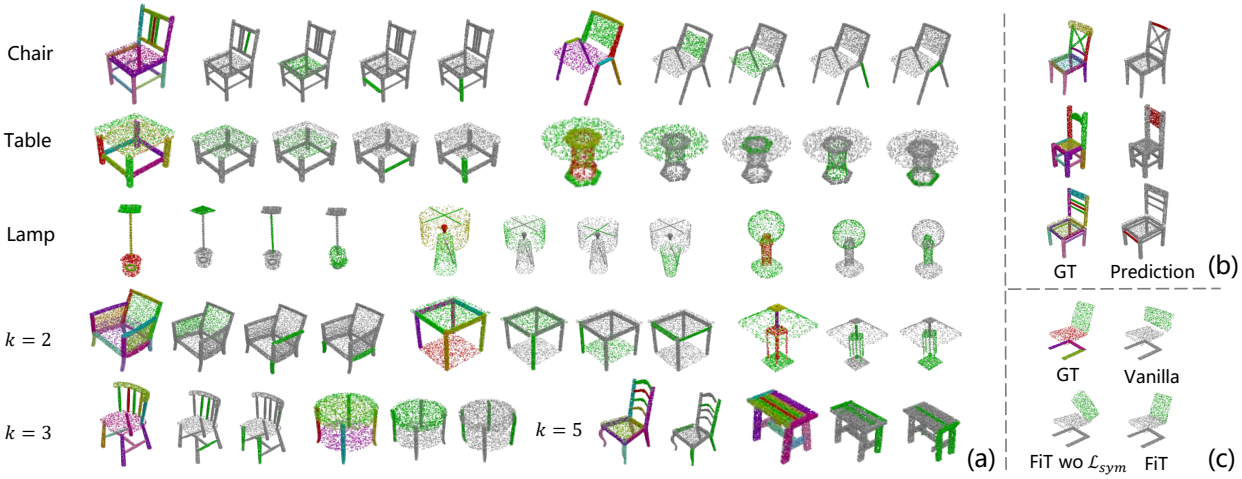

Figure 2: Qualitative results. (a) Completion results on the original toolkit. Multi-colored assembly denotes the ground truth and two-color assemblies denote completion results. Predicted parts are visualized in green. (b) Plausible predictions with wrongly assembled parts visualized in red. (c) Qualitative analysis of pose concatenation and $\mathcal{L}_{sym}$.

The overall loss function is defined as a weighted combination of the translation loss $\mathcal{L}_t$, rotation loss $\mathcal{L}_r$, shape Chamfer distance loss $\mathcal{L}_s$, symmetrical transformation loss $\mathcal{L}_{sym}$ and classification loss $\mathcal{L}_c$:

$$\mathcal{L} = \lambda_t \mathcal{L}_t + \lambda_r \mathcal{L}_r + \lambda_s \mathcal{L}_s + \lambda_{sym} \mathcal{L}_{sym} + \lambda_c \mathcal{L}_c, \quad (9)$$

where λ_* denote the weights of losses.

Evaluation Metrics

We define match accuracy (MA) as $\mathcal{MA} = c/k$ to evaluate the accuracy of candidate classification, where k denotes the number of missing parts and c denotes the number of parts that are correctly matched with the ground-truth parts.

Overall quality of assembly completion can be measured by shape Chamfer distance (SCD) defined as Eq. 3. Part accuracy (PA) is defined based on SCD, which indicates the percentage of parts within a certain Chamfer distance threshold. Formally,

$$\mathcal{PA} = \frac{1}{k} \sum_{i=1}^{k} \left(d_c \left(T_i \left(p_i \right), T_i^* \left(p_i^* \right) \right) < \tau_p \right), \quad (10)$$

where the threshold τ_p is set to 0.01.

Following (Zhan et al. 2020), we adopt connectivity accuracy (CA) to evaluate how well the assembled parts are connected in the assembly. We define connectivity accuracy as:

$$\mathcal{CA} = \frac{1}{|\mathcal{C}|} \sum_{\{c_{ij}, c_{ji} \in \mathcal{C}\}} \left(\|T_i(c_{ij}) - T_j(c_{ji})\|_2^2 < \tau_c \right), \quad (11)$$

where c_{ij} denotes the point of part p_i which is closest to part p_j (*i.e.*, contact point). $\mathcal{C}$ is the set of contact point pairs between incomplete shape and candidate parts. The threshold τ_c is set to 0.01.

Experiments

Dataset

We evaluate the proposed method on the PartNet (Mo et al. 2019) dataset, a large-scale synthetic dataset of 3D shapes annotated with instance-level and hierarchical 3D part information. We choose the three largest categories of 6,323 chairs, 8,218 tables, and 2,207 lamps with the most fine-grained level of segmentation and follow the default train/val/test splits of 70%/20%/10%. The shapes are filtered with the number of parts limited to $(k, 20]$, where k is the number of missing parts. We sample 1,000 points with Furthest Point Sampling (FPS) for each part component and transform the part point clouds into canonical space with PCA (Pearson 1901).

Baseline	MA↑	SCD↓	PA↑	CA↑
Complement	77.69	0.0332	14.87	15.21
Single Image	81.75	0.0141	34.72	29.80
DGL	91.05	0.0090	62.13	46.21
PoinTr	-	0.0368	-	-
AutoSDF	-	0.0411	-	-
FiT (ours)	**92.93**	**0.0069**	**70.71**	**62.22**

Table 2: Comparison with baseline methods.

Category	Identical	MA↑	SCD↓	PA↑	CA↑
Chair	✓	84.88	0.0176	46.81	21.09
	✗	57.50	0.0154	47.34	49.98
Table	✓	87.58	0.0093	53.80	19.02
	✗	57.68	0.0077	50.71	42.64

Table 3: Different combinations of missing parts with $k = 2$.

Encoder	Memory	MA↑	SCD↓	PA↑	CA↑
		88.45	0.0113	52.93	43.27
✓		90.67	0.0095	62.68	54.24
	✓	92.67	0.0083	65.24	55.04
✓	✓	**92.93**	**0.0069**	**70.71**	**62.22**

Table 4: Importance of pose concatenation.

$\mathcal{L}_t$	$\mathcal{L}_r$	$\mathcal{L}_s$	$\mathcal{L}_{sym}$	MA↑	SCD↓	PA↑	CA↑
✗				86.16	0.0112	49.34	44.19
	✗			91.03	0.0076	67.56	58.06
		✗		91.52	0.0081	67.95	58.24
			✗	89.63	0.0082	66.96	56.31
✓	✓	✓	✓	**92.93**	**0.0069**	**70.71**	**62.22**

Table 5: Effect of loss components.

Implementation Details

We train FiT with the AdamW optimizer with an initial learning rate of 1.5×10^{-4} for 500 epochs on 8 GPUs. Batch size is set to 64. During training, we randomly sample k parts as missing parts for each assembly. We adopt a blended toolkit a with a fixed size of $M = 30$, where $m_1 = 10 - k$ parts are sampled from the assembly itself and $m_2 = 20$ parts from the other shapes in the same batch. During evaluation, we iterate over all possible combinations of k missing parts (maximum combinations set to 500 when $k > 2$). We set the original toolkit with a size of $M = 10$. For the blended toolkit, we set $M = 30, m_1 = 10 - k$ and $m_2 = 20$.

Baseline Methods

We compare FiT with several baseline methods, including assembly-based (Sung et al. 2017; Li et al. 2020; Zhan et al. 2020) and generation-based (Yu et al. 2021; Mittal et al. 2022) completion methods described as follows.

- **Complement** (Sung et al. 2017) learns the partial shape with components and non-components by contrastive learning jointly in the embedding space. We reserve the PointNet-based embedding networks and pose decoder. We perform contrastive learning by sampling negative parts from the toolkit and adapt the decoder to select candidates and predict 6-DoF poses.
- **Single Image** (Li et al. 2020) focuses on constructing 3D assembly guided by a single image. We adapt this method by taking out the image encoders. We model the incomplete shape as global feature and candidate parts as local feature with PointNet and concatenate them for prediction.
- **DGL** (Zhan et al. 2020) takes iterative graph neural network (GNN) as the core for part relation modeling in 3D part assembly. We replace self-attention and decoder-encoder attention modules in FiT with GNNs. The predictors are adopted from FiT.
- **PoinTr** (Yu et al. 2021) is a transformer-based framework designed for point-aware shape completion, which takes partial point clouds as input and generates missing regions of points without any toolkits. We test PoinTr with a dense prediction in our setting.
- **AutoSDF** (Mittal et al. 2022) uses signed distance field (SDF) as 3D representation for auto-regressive shape completion per voxel. We evaluate AutoSDF by transferring incomplete assembly represented in point clouds to SDFs and sample the completion results into point clouds to compute the metrics.

Main Results and Comparison

Table 1 summarizes the results with different numbers of missing parts and compositions of toolkit. On the original toolkit, we obtain 92.93% and 91.14% in match accuracy, and 70.71% and 76.78% in part accuracy on chair and table with $k = 1$. Table performs better in pose prediction. The blended configuration is more difficult, which contains more disturbances and has more impact on match accuracy. On the blended toolkit, we observe a decline of 6.25% in match accuracy and 2.38% in part accuracy on table. Figure 2(a) visualizes qualitative results on the original toolkit. Figure 2(b) provides some wrong completion results which remain plausible in structure.

Comparison with baselines. We compare FiT with baseline methods on the original toolkit with $k = 1$. Results are summarized in Table 2. As shown, FiT outperforms all baseline methods by a significant margin in most metrics, especially in part accuracy. FiT has an improvement of 8.58% and 1.88% in part accuracy and match accuracy compared with DGL, indicating the efficiency of transformer in part relation modeling. Single Image has an obvious decline in performance without feature augmentation from part segmentation. Complement drops drastically in part accuracy. This is because geometrically similar parts introduce undiscriminating positive and negative features, which may confuse contrastive learning. PoinTr tends to learn an interpolation-like completion prior and can not precisely catch the region of missing parts. We report more results on the other categories in supplementary material.

Multiple missing parts. As shown in Table 1, FiT is also capable of completing assembly with multiple missing parts.

Part match	Pose pred	MA↑	SCD↓	PA↑	CA↑
✓		84.67	-	-	-
	✓	-	0.0139	51.05	42.62
✓	✓	92.93	0.0069	70.71	62.22

Table 6: Ablation for multiple tasks.

Encoder	Decoder	MA↑	SCD↓	PA↑	CA↑
2	2	90.75	0.0080	67.99	57.29
3	3	**92.93**	**0.0069**	**70.71**	**62.22**
6	6	87.44	0.0086	67.42	57.87

Table 7: Influence of transformer layers.

When $k = 2$, we obtains a match accuracy of 61.36% and part accuracy of 44.14% on chair with original toolkit. We observe a relatively robust performance while increasing the number of missing parts by $k = 3, 5$. It is rather challenging since each assembly of chair/table/lamp has 10.2/9.1/4.6 parts in average. For chair and table, $k = 5$ means approximately half of the assembly components are taken away. In this case, the incomplete assembly provides less information and it remains a larger solution space. This may be destructive for lamp with an average number of parts less than 5. We also explore the combination of two missing parts in Table 3, where *identical* denotes a pair of geometrically identical parts, *prop.* denotes the proportion of identical combination. We report the results on chair and table, which contains more identical parts. As shown, an identical combination tends to be easier to select but more difficult to assemble.

Ablation Study

Pose concatenation. We first study the effect of pose concatenation at different stages of transformer. The results are reported in Table 4. *Encoder* denotes concatenation at the input of each self-attention layer in encoder. *Memory* denotes concatenation with the memories (*i.e.*, the output of encoder). As shown, pose concatenation in both encoder and memory gives the best result. It serves as a direct and effective strategy, which largely facilitates pose prediction with an improvement over 17% on part accuracy compared with no concatenation.

Loss components. We evaluate the influence of loss functions by removing each loss component individually. Table 5 summarizes the results. Each component contributes to improvement in performance. Especially, translation loss $\mathcal{L}_t$ is dominant in the supervision of pose prediction. We obtain quite a bit of improvement with $\mathcal{L}_{sym}$, which indicates that the designed symmetrical transformation consistency is effective in guiding the completion of assembly. Figure 2(c) visualizes an example, where *Vanilla* denotes FiT without pose concatenation and $\mathcal{L}_{sym}$. As shown, the result of vanilla has a large deviation in pose estimation, and pose concatenation alleviates this issue to some extent. The result can be further polished with $\mathcal{L}_{sym}$, indicating its effectiveness.

Multi-task learning. 3D assembly completion naturally forms a multi-task learning setup. We compare multi-task learning with single-task learning in Table 6. *Part match* denotes the task of candidate classification, where only $\mathcal{L}_c$ is enabled. *Pose pred* denotes the task of pose prediction and the ground-truth missing parts are assumed to be known. As shown, multi-task learning obtains better results compared with single-task learning, which indicates that the two tasks have a mutually reinforcing effect rather than conflict.

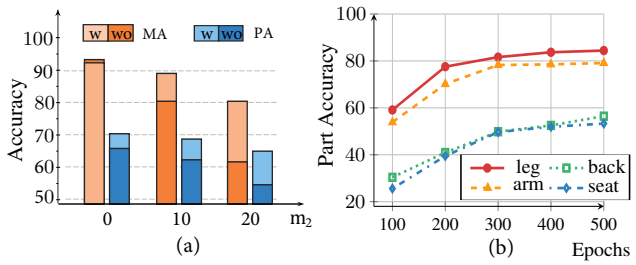

Figure 3: Exploration of (a) self-attention in decoder and (b) different semantic classes.

Encoder-decoder layers. We also examine the influence of encoder-decoder layers in Table 7. A 3-layer structure gives the best result. Stacking more layers does not work well in our task and on the contrary, may increase time consumption in training and inference. We choose a 3-layer configuration as default, considering both performance and inference speed.

Self-attention of queries. We further explore the function of self-attention mechanism adopted for candidate parts in the decoder. We achieve this by varying m_2, the number of parts that are sampled from another shape in blended toolkit. We set $m_2 = 0, 10, 20$. The results are demonstrated in Figure 3(a). Compared on any configuration, the transformer with self-attention (*w* for short in figure) generally has advantages in pose prediction. When m_2 varies from 0 to 20, decoder without self-attention (*wo* for short in figure) has a sharper decline in both metrics than that with self-attention. This indicates that the transformer with self-attention in decoder obtains a more robust performance with more candidates added to the toolkit, in terms of both tasks.

Different semantic classes. We also explore the learning results of parts with different semantic classes. Figure 3(b) shows the variation of part accuracy of four semantic classes on chair during training. Leg and arm tend to converge more quickly and obtain more accurate results, while the center components back and seat remain a gap of nearly 25%.

Conclusions

In this paper, we propose a novel task of 3D assembly completion, which aims to finish an incomplete assembly with the toolkit. We propose FiT, a transformer-based framework to tackle this problem. We also design two typical toolkits which are common cases in practical for reasonable evaluation. Extensive experiments and analysis demonstrate the effectiveness of FiT in various evaluation configurations and also set a strong baseline for reference. Further work may deal with 3D assembly completion with unfixed number of missing parts and test the model in real environment with real scanned data.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62073244, Shanghai Innovation Action Plan under Grant 20511100500 and Innovation Program of Shanghai Municipal Education Commission (202101070007E00098).

References

Adler, J.; and Lunz, S. 2018. Banach wasserstein gan. *Advances in neural information processing systems*, 31.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Chaudhuri, S.; Kalogerakis, E.; Guibas, L.; and Koltun, V. 2011. Probabilistic reasoning for assembly-based 3D modeling. In *ACM SIGGRAPH 2011 papers*, 1–10.

Chen, Y.; Sareh, P.; and Feng, J. 2015. Effective insights into the geometric stability of symmetric skeletal structures under symmetric variations. *International Journal of Solids and Structures*, 69: 277–290.

Ghojogh, B.; Sikaroudi, M.; Shafiei, S.; Tizhoosh, H. R.; Karray, F.; and Crowley, M. 2020. Fisher discriminant triplet and contrastive losses for training siamese networks. In *2020 international joint conference on neural networks (IJCNN)*, 1–7. IEEE.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Harish, A. N.; Nagar, R.; and Raman, S. 2022. Rgl-net: A recurrent graph learning framework for progressive part assembly. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 647–656. IEEE.

Hu, T.; Han, Z.; Shrivastava, A.; and Zwicker, M. 2019. Render4Completion: Synthesizing multi-view depth maps for 3D shape completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Jaiswal, P.; Huang, J.; and Rai, R. 2016. Assembly-based conceptual 3D modeling with unlabeled components using probabilistic factor graph. *Computer-Aided Design*, 74: 45–54.

Kalogerakis, E.; Chaudhuri, S.; Koller, D.; and Koltun, V. 2012. A probabilistic model for component-based shape synthesis. *Acm Transactions on Graphics (TOG)*, 31(4): 1–11.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673.

Li, J.; Niu, C.; and Xu, K. 2020. Learning part generation and assembly for structure-aware shape synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 11362–11369.

Li, J.; Xu, K.; Chaudhuri, S.; Yumer, E.; Zhang, H.; and Guibas, L. 2017. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4): 1–14.

Li, Y.; Mo, K.; Shao, L.; Sung, M.; and Guibas, L. 2020. Learning 3d part assembly from a single image. In *European Conference on Computer Vision*, 664–682. Springer.

Mittal, P.; Cheng, Y.-C.; Singh, M.; and Tulsiani, S. 2022. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 306–315.

Mo, K.; Zhu, S.; Chang, A. X.; Yi, L.; Tripathi, S.; Guibas, L. J.; and Su, H. 2019. 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*, 909–918.

Pearson, K. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Sarmad, M.; Lee, H. J.; and Kim, Y. M. 2019. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5898–5907.

Saupe, D.; and Vranić, D. V. 2001. 3D model retrieval with spherical harmonics and moments. In *Joint Pattern Recognition Symposium*, 392–397. Springer.

Stewart, R.; Andriluka, M.; and Ng, A. Y. 2016. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2325–2333.

Sung, M.; Dubrovina, A.; Kim, V. G.; and Guibas, L. 2018. Learning fuzzy set representations of partial shapes on dual embedding spaces. In *Computer Graphics Forum*, volume 37, 71–81. Wiley Online Library.

Sung, M.; Su, H.; Kim, V. G.; Chaudhuri, S.; and Guibas, L. 2017. ComplementMe: Weakly-supervised component suggestions for 3D modeling. *ACM Transactions on Graphics (TOG)*, 36(6): 1–12.

Tchapmi, L. P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; and Savarese, S. 2019. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 383–392.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vranic, D. V.; and Saupe, D. 2004. *3D model retrieval.* Ph.D. thesis, Citeseer.

Wang, M.; Gao, Y.; Lu, K.; and Rui, Y. 2012. View-based discriminative probabilistic modeling for 3D object retrieval and recognition. *IEEE Transactions on Image Processing*, 22(4): 1395–1407.

Wen, X.; Li, T.; Han, Z.; and Liu, Y.-S. 2020. Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1939–1948.

Wu, R.; Zhuang, Y.; Xu, K.; Zhang, H.; and Chen, B. 2020. Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 829–838.

Xie, X.; Xu, K.; Mitra, N. J.; Cohen-Or, D.; Gong, W.; Su, Q.; and Chen, B. 2013. Sketch-to-design: Context-based part assembly. In *Computer Graphics Forum*, volume 32, 233–245. Wiley Online Library.

Yan, X.; Lin, L.; Mitra, N. J.; Lischinski, D.; Cohen-Or, D.; and Huang, H. 2022. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6239–6249.

Yin, K.; Chen, Z.; Chaudhuri, S.; Fisher, M.; Kim, V. G.; and Zhang, H. 2020. Coalesce: Component assembly by learning to synthesize connections. In *2020 International Conference on 3D Vision (3DV)*, 61–70. IEEE.

Yoo, C. H.; and Lee, S. 2011. *Stability of structures: principles and applications*. Elsevier.

Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; and Zhou, J. 2021. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12498–12507.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19313–19322.

Yuan, W.; Khot, T.; Held, D.; Mertz, C.; and Hebert, M. 2018. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, 728–737. IEEE.

Zhan, G.; Fan, Q.; Mo, K.; Shao, L.; Chen, B.; Guibas, L. J.; Dong, H.; et al. 2020. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems*, 33: 6315–6326.

Zhang, R.; Kong, T.; Wang, W.; Han, X.; and You, M. 2022. 3D Part Assembly Generation with Instance Encoded Transformer. *IEEE Robotics and Automation Letters*.

Zhu, C.; Xu, K.; Chaudhuri, S.; Yi, R.; and Zhang, H. 2018. SCORES: Shape composition with recursive substructure priors. *ACM Transactions on Graphics (TOG)*, 37(6): 1–14.