PART: Learning 3D Part Assembly and Retrieval with Transformers

RUCHAO BAO, University of Science and Technology of China, China
WENZHENG WU, University of Science and Technology of China, China
CHUCHENG XIANG, University of Science and Technology of China, China
ZHONGYUAN LIU, Tencent, China
YUAN LIU, Hong Kong University of Science and Technology, China
JINXIN DONG, Tencent, United States of America
LIGANG LIU, University of Science and Technology of China, China
ZIQI WANG*, Hong Kong University of Science and Technology, China



Fig. 1. Results of our PART framework. Given a monolithic target shape (left, visualized as a mesh and represented as a point cloud for model input), PART retrieves appropriate components from a candidate part library (middle) and reconstructs the target as a part assembly by predicting the 6-DoF pose of each selected component (right).

3D assembly is fundamental to modern manufacturing and digital content creation. In this paper, we present PART, a unified transformer-based framework for 3D part retrieval and assembly: given a target shape and a part library, PART automatically selects the appropriate parts and predicts their 6-DoF poses to reconstruct the target. While prior work has achieved impressive progress on assembling a pre-defined set of parts, this more practical retrieval-based setting remains largely unexplored. The task faces three key challenges: (i) a combinatorially explosive search space that grows exponentially with library size; (ii) variable-length outputs, as different targets require different numbers of parts; and (iii) continuous 6-DoF pose estimation for part assembly. To address these, we formulate retrieval and assembly as a set prediction problem and design a novel transformer-based framework that retrieves parts and regresses their poses with variable-length output. Additionally, we exploit the duality between part pose estimation and target segmentation through joint training and a novel segmentation-enhanced optimization module. Finally, We curate a large-scale dataset of 80K+ shapes, and the results show that PART generalizes to scene layouts, image targets, and real-world scans. Project Page: https://iambrc.github.io/PART-project-page/.

CCS Concepts: • **Computing methodologies** → **Point-based models**; *Shape analysis*; Neural networks.

Additional Key Words and Phrases: 3D Part Assembly, Point Cloud Understanding, Part-based Modeling, Transformers

ACM Reference Format:
Ruchao Bao, Wenzheng Wu, Chucheng Xiang, Zhongyuan Liu, Yuan Liu, Jinxin Dong, Ligang Liu, and Ziqi Wang. 2026. PART: Learning 3D Part Assembly and Retrieval with Transformers. In *SIGGRAPH Asia 2026 Conference Papers (SA Conference Papers '26), December 01–04, 2026, Kuala Lumpur, Malaysia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3829340.3842152

*Corresponding author.

Authors' Contact Information: Ruchao Bao, University of Science and Technology of China, China, iambrc@mail.ustc.edu.cn; Wenzheng Wu, University of Science and Technology of China, China, wuwzh@mail.ustc.edu.cn; Chucheng Xiang, University of Science and Technology of China, China, xcc2020@mail.ustc.edu.cn; Zhongyuan Liu, Tencent, China, lockliu@tencent.com; Yuan Liu, Hong Kong University of Science and Technology, China, yuanly@ust.hk; Jinxin Dong, Tencent, United States of America, jxdong@global.tencent.com; Ligang Liu, University of Science and Technology of China, China, lgliu@ustc.edu.cn; Ziqi Wang, Hong Kong University of Science and Technology, China, ziqiw@ust.hk.

This work is licensed under a Creative Commons Attribution 4.0 International License.
SA Conference Papers '26, Kuala Lumpur, Malaysia
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2842-6/2026/12
https://doi.org/10.1145/3829340.3842152

1 Introduction

Assemblies are fundamental to 3D content creation, gaming, and manufacturing, where complex objects are often represented as compositions of smaller and simpler parts. Traditionally, modeling 3D assemblies has been a labor-intensive process, requiring designers to manually arrange hundreds of components in CAD software. Recent advances in artificial intelligence have shown a promising path toward automating this process, enabling 3D assembly generation from text prompts [Pun et al. 2025] or reference images [Zhao et al. 2025a]. Although recent 3D generative models can already synthesize plausible, high-quality geometry from scratch [Lai et al. 2025], many CAD, industrial design, gaming, and asset-production workflows require more than plausible geometry: objects should be built from known, reusable components so that part identity, editability, and compatibility with existing libraries are preserved. Assembling target shapes from such components library therefore remains an important problem.

Extensive prior work on part assembly has achieved impressive results [Xu et al. 2025b; Zhan et al. 2020; Zhao et al. 2025a]. However, these methods primarily focus on arranging a predefined set of components, assuming that the provided parts exactly match those required to reconstruct the target shape, which limits their applicability in practical settings where component libraries often contain missing, redundant, or interchangeable parts. In both real-world manufacturing and virtual asset creation, the ability to automatically select and reuse parts from a component library is crucial for scalable assembly automation, yet this retrieval-based assembly setting remains largely unexplored. Motivated by this gap, we target a more challenging problem to automate the assembly pipeline: given a target shape and a part library, our system jointly selects appropriate parts and estimates their 6-DoF poses to reconstruct the target geometry.

Part retrieval and assembly face three main challenges. **(i) Combinatorial explosion:** selecting appropriate parts from a large part library induces a vast search space whose complexity grows rapidly with library size. **(ii) Variable-length outputs:** different target shapes require different numbers of components, posing a significant challenge for conventional deep networks that typically assume fixed-size outputs. **(iii) Continuous 6-DoF pose estimation:** unlike modular systems like LEGO [Xu et al. 2025a], general part assembly requires reasoning over both semantic and geometric properties of parts to estimate poses in the continuous, non-Euclidean space $SE(3)$.

We address these challenges through a unified framework built on two key insights. For the first two challenges, we cast part retrieval and assembly as a **set prediction** problem (i.e., predicting an unordered collection of items whose exact number can vary). Inspired by DETR [Carion et al. 2020], we adopt a query-based transformer architecture in which a fixed set of learnable query vectors attends to both the target shape and the candidate parts in a shared feature space. Each query independently decides which part to retrieve and where to place it, while unmatched queries are suppressed, enabling variable-length output within a fixed-capacity model. Our approach can handle part libraries containing over 100 parts within seconds at inference time, since the model processes all queries in parallel in a single forward pass. However, directly training on large libraries is both memory- and time-intensive due to the quadratic complexity of the Transformer architecture. To address this, we adopt a *train-short-test-long* strategy: we first train the model on small libraries augmented with randomly sampled distractor parts, and then apply it at inference time to full-sized libraries (see subsection 4.2).

For the third challenge, our key observation is that **assembly and segmentation are dual problems**: the part pose implicitly determines which region of the target it occupies, while a per-part segmentation of the target directly constrains where and how that part is placed. We exploit this duality by jointly training our framework on retrieval, assembly, and segmentation. At test time, we further introduce a segmentation-enhanced optimization module that uses the predicted segmentation masks to refine the initial pose estimates via part-to-region alignment. Moreover, we adopt a continuous rotation representation (A-Matrix) for singularity-free orientation regression and design a symmetry-aware loss that accounts for geometric ambiguities in symmetric parts, jointly mitigating the difficulty of pose estimation in $SE(3)$.

Our contributions are summarized as follows: (1) We innovatively formulate 3D part retrieval and assembly as a set prediction problem, and propose PART, a unified transformer-based framework that jointly addresses retrieval and assembly with variable-length output. (2) We introduce a novel segmentation-enhanced optimization module and a symmetry-aware loss to improve assembly accuracy. (3) We curate a large-scale and diverse dataset of 80K+ shapes for general part assembly task, and demonstrate that PART generalizes to scene layouts, image targets, and real-world scans.

2 Related Work

2.1 3D Part Assembly

The 3D part assembly task aims to assemble multiple parts into a coherent and complete model. Early works employ intelligent scissoring techniques [Funkhouser et al. 2004] and probabilistic graphical models [Chaudhuri et al. 2011; Jaiswal et al. 2016; Kalogerakis et al. 2012] to address the challenges associated with the semantic and geometric relationships among parts. With the advent of deep learning and the availability of large-scale part-level datasets with detailed annotations [Chang et al. 2015; Mo et al. 2019], recent methods have increasingly focused on fully automated assembly through neural networks. ImagePA [Li et al. 2020] first proposes single-image-based unlabeled 3D part assembly, and DGL [Zhan et al. 2020] innovatively proposes an iterative graph neural network to learn parts' relationships and poses. Building on these advances, RGL [Narayan et al. 2022] and SPAFormer [Xu et al. 2025b] incorporate input part order to improve the assembly performance; IET [Zhang et al. 2022] presents an instance-encoded Transformer framework to better distinguish geometrically similar parts; ScorePA [Cheng et al. 2023] proposes a score-based framework to learn the conditional probability distribution of part poses; 3DHPA [Du et al. 2024] develops a part-whole-hierarchy message passing network; and Imagine [Wang et al. 2025b] utilizes a structural knowledge graph to guide the assembly process; The Shape Part Slot Machine [Wang et al. 2022] formulates assembly through explicit contact-based reasoning, and CFPA [Zhang et al. 2026] introduce a coarse-to-fine framework

utilizing semantic super-parts and symmetry-aware pose estimation. A parallel line of research explores integrating physical constraints (PhysFit [Wang et al. 2024] and JointPA [Li et al. 2024]), segmentation-based assembly (GPAT [Li et al. 2023]), and retrieval-based assembly (UPRA [Xu et al. 2023]). More recent works focus on advanced models or real-world automated assembly. For example, Assembler [Zhao et al. 2025a] utilizes an anchor point diffusion model, Rectified Point Flow [Sun et al. 2025] and GARF [Li et al. 2025] use flow matching models to indirectly predict the pose of each part. Meanwhile, ManualPA [Zhang et al. 2025], Manual2Skill [Tie et al. 2025b], Manual2Skill++ [Tie et al. 2025a] explore part assembly schemes guided by instruction manuals.

Our method is most related to UPRA [Xu et al. 2023] and GPAT [Li et al. 2023]. UPRA couples a pre-trained VAE with latent optimization to perform retrieval and assembly jointly, yet relies on hyperparameters or complex rules to control the number of retrieved parts, and does not explicitly model inter-part or part-whole relationships. GPAT leverages a segmented target point cloud to facilitate the assembly process. While it demonstrates the capability to successfully assemble general unseen parts, it still relies on a predefined set of input parts and does not consider the retrieval stage. In contrast, our approach jointly reasons about segmentation, retrieval, and assembly with explicit part-part and part-whole consistency, while supporting variable-length parts output.

2.2 3D Part Retrieval and Fitting

The 3D part retrieval and fitting problem aims to reconstruct the target shape by retrieving compatible components from a given library and fitting them to the target. Notably, significant research has focused on 3D primitive fitting and shape abstraction, which utilize basic geometric primitives (e.g., cuboids, cylinders) to represent complex shapes. Researchers have explored various fitting approaches, including optimization-based fitting [Li et al. 2011; Liu et al. 2023; Schnabel et al. 2007], network-based fitting (unsupervised [Paschalidou et al. 2019; Tulsiani et al. 2017; Yang and Chen 2021] and supervised [Zou et al. 2017]), and more recently, Large Language Model (LLM) based fitting [Tian et al. 2025]. While primitive representations offer advantages such as low storage and fast rendering, retrieving and assembling components from real-world part libraries more closely mirrors practical manufacturing pipelines and can produce more complex, higher-quality assemblies. However, general part retrieval and assembly remains largely underexplored [Uy et al. 2021; Xu et al. 2023], and these approaches do not inherently support variable-length retrieval and depend on hyperparameters or post-processing. LLM-Primitives [Tian et al. 2025] addresses this via LLM-based approach, but has only been validated on simple primitives, falling short of handling realistic part complexity. Our method focuses on reconstructing complete target shapes by retrieving and assembling parts from a given library and can produce variable-length parts output based on given target.

2.3 Transformers and Set Prediction

The Transformer architecture [Vaswani et al. 2017], originally designed for sequence modeling, has been widely used in various 2D and 3D tasks, e.g., Vision Transformer [Dosovitskiy 2020] and Point Transformer [Wu et al. 2024, 2022; Zhao et al. 2021]. Set prediction requires a model to output a variable-size, unordered set of elements, which is inherently difficult for canonical deep learning networks. A representative milestone is DETR [Carion et al. 2020], which pioneered a query-based transformer framework for variable-length set prediction without hand-crafted heuristics such as anchors or non-maximum suppression (NMS). Building on DETR, this query-based paradigm has been extended to various 3D tasks, such as 3D object detection [Liu et al. 2021; Misra et al. 2021] and instance segmentation [Schult et al. 2022; Sun et al. 2023]. Inspired by DETR, we adopt a query-based transformer framework to jointly address part retrieval and assembly.

3 Method

Given a part library containing canonical-pose candidate part point clouds and a target point cloud (e.g., obtained from image-to-3D models or real scans), our goal is to retrieve a subset of parts from the part library (with repetition allowed) and predict a rigid transformation for each selected part, such that the resulting assembly reconstructs the target shape. In the following, we first describe our network architecture in subsection 3.1, followed by the training losses in subsection 3.2. Finally, we introduce a segmentation-enhanced post-processing optimization module in subsection 3.3.

3.1 Network Architecture

Our PART network consists of three stages: the feature extraction, the transformer decoder, and the prediction; see Figure 2.

Features Extraction. For both input part point clouds $\mathcal{C} = \{C_i\}_{i=1}^{M}$ and target point cloud $\mathcal{P}$, we use point transformer v3 [Wu et al. 2024], a state-of-the-art backbone network for point cloud deep learning, to extract global features $\in \mathbb{R}^D$ of each part and point-wise features $\in \mathbb{R}^{N_p \times D}$ of the target. Here $N_p = 10000$ represents the total number of target points, and D denotes the hidden dimension. Detailed PTv3 network hyperparameter configuration can be found in the supplementary material. The following self- and cross-attention blocks then exchange information between parts and target, yielding the contextualized part features $f^{\mathcal{C}} = \{f_i\} \in \mathbb{R}^{M \times D}$ and target shape feature $f^{\mathcal{P}} \in \mathbb{R}^{N_p \times D}$.

Transformer Decoder. The number of parts in different assemblies within the same category may vary. For example, chairs can be decomposed into four parts in one design and seven parts in another. This makes the retrieval and assembly task a variable-length set prediction problem. To overcome this challenge, inspired by DETR [Carion et al. 2020], we employ a query-based transformer decoder architecture. Specifically, we utilize a fixed number of N learnable instance queries $Q^{(0)} = \{Q_j\} \in \mathbb{R}^{N \times D}$ to act as agents to retrieve appropriate parts. Each query is trained to either match with a candidate part or fall into a dedicated *empty* class. This formulation naturally supports variable-length outputs, as queries are allowed to select the *empty* class when no corresponding part exists. Moreover, parts and instance queries are processed in parallel, enabling efficient inference for the input part library. The queries are refined by L transformer decoder layers. Within each layer, a query first attends to the other queries (modeling inter-dependencies among

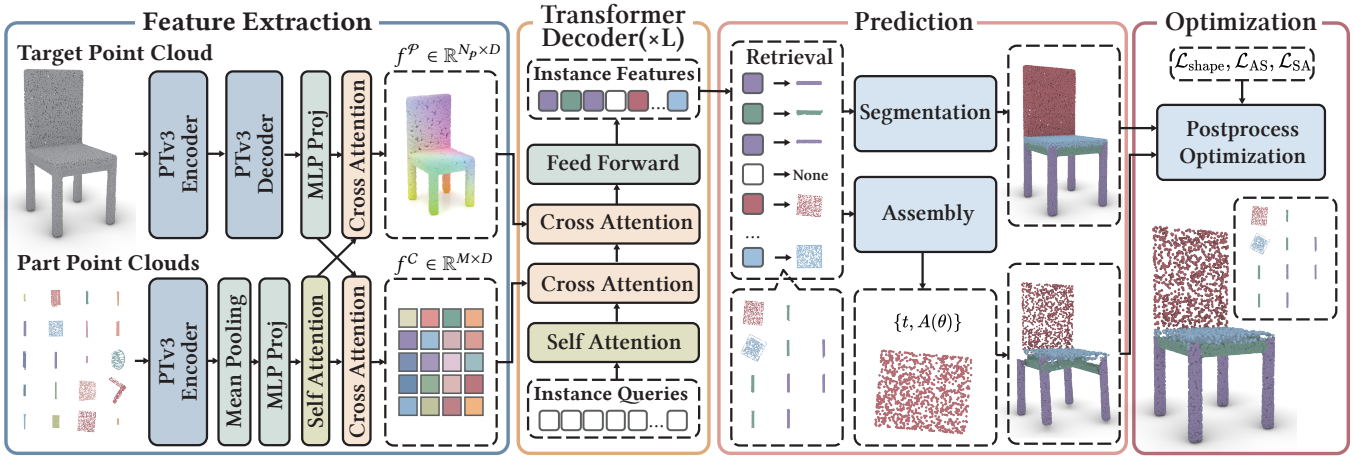


Fig. 2. **Overview of our framework.** Given a target point cloud and a set of candidate parts, we employ Point Transformer v3 layers followed by self- and cross-attention mechanisms to extract point-wise features $f^{\mathcal{P}}$ and part global features f^{C}. These representations serve as conditions for the Transformer decoder layers, where a set of learnable instance queries interacts with the encoded features. The decoded instance features are then fed into parallel prediction heads to determine part assignments (classifying into candidate indices or empty class), estimate 6-DoF assembly poses, and generate segmentation masks. Finally, a segmentation-enhanced optimization module refines the assembly configuration to yield the precise result.

different instances) and then sequentially cross-attends to the part features f^{C} and to the target features $f^{\mathcal{P}}$:

$$Q^{(\ell)} = \text{CrossAttn}(\text{CrossAttn}(\text{SelfAttn}(Q^{(\ell-1)}), f^{C}), f^{\mathcal{P}}). \quad (1)$$

We write the final per-instance features as $f^{Q} = \{Q_j^{(L)}\} \in \mathbb{R}^{N \times D}$, which jointly encode the geometry of candidate parts and the structural context of the target.

Prediction. The per-instance features f^{Q} are decoded into three coupled outputs in our prediction head, including part retrieval, target point-cloud segmentation, and assembly pose estimation. First, the **retrieval head** R classifies each query instance f_j^{Q} as one of the M candidate parts plus an *empty* class.

$$R = \text{MLP}_{R_Q}(f^{Q})\,\text{MLP}_{R_C}(f^{C*})^{\top}/\sqrt{D} \in \mathbb{R}^{N \times (M+1)}, \quad (2)$$

where the extended part features are $f^{C*} = f^{C} \cup \{e\}$ and $e \in \mathbb{R}^{D}$ is a learnable token for representing the *empty* class. Second, the **segmentation head** S classifies the target point $f^{\mathcal{P}}$ as one of the N query instances:

$$S = \text{MLP}_{S_{\mathcal{P}}}(f^{\mathcal{P}})\,\text{MLP}_{S_Q}(f^{Q})^{\top}/\sqrt{D} \in \mathbb{R}^{N_p \times N}. \quad (3)$$

Lastly, the **assembly head** predicts a translation t_j and a rotation quaternion q_j per query. The pose head utilizes the instance feature f_j^{Q} and a soft max summary of the extended part features:

$$\begin{aligned} [t_j, A_j] &= \text{MLP}_{\text{pose}}\left(f_j^{Q} \parallel \text{softmax}(R_j)\, f^{C*}\right) \in \mathbb{R}^{3+10}, \\ q_j &= \underset{\|q\|=1,\, q \in \mathbb{R}^4}{\arg\min}\; q^{\top} A_j\, q, \end{aligned} \quad (4)$$

where we follow the A-matrix parameterization [Lin et al. 2023; Peretroukhin et al. 2020] for predicting the rotation quaternion to avoid suboptimality, a common issue that arises in SO(3) rotation regression [Geist et al. 2024; Xiang and Li 2020; Zhou et al. 2019]. Specifically, the network first predicts a symmetric matrix A_j with 10 parameters. Then, the predicted quaternion q_j is recovered as the eigenvector of A_j associated with the smallest eigenvalue.

3.2 Training and Losses

This subsection introduces our training procedure and loss terms. During training, we are given the ground-truth set of parts C^{g} along with their translations t_k^{g} and rotation quaternions q_k^{g}, where $k = 1, \cdots, N^{g}$. To enable a stable training, we perform a bipartite matching to assign each instance query to the most suitable ground-truth part, thereby eliminating the permutation ambiguity inherent in set prediction. Note that we have more queries than the number of ground-truth parts ($N^{g} \le N$). Queries that are not assigned to any ground-truth part are matched to an *empty* class. Our bipartite matching algorithm minimizes the sum of the matching cost c_{ij} between the query Q_i and the ground-truth part C_j^{g}.

$$c_{ij} = -\lambda_1 c_{i,j}^{R} + \lambda_2 \|t_i - t_j^{g}\|_2 - \lambda_3 c_{i,j}^{S}. \quad (5)$$

The first term $c_{i,j}^{R}$, measures the retrieval match cost. As each ground truth part C_j^{g} must have a corresponding part $C_{j'}$ in the input part library C, this cost term encourages maximizing the retrieval probability $c_{i,j}^{R} = R(Q_i, C_{j'})$ of assigning the part $C_{j'}$ to the query Q_i. The second term penalizes the distance between the predicted and ground-truth part translations. The third term $c_{i,j}^{S}$, measures the segmentation match cost. For the j-th ground-truth part C_j^{g}, it must correspond to a set of points $\mathcal{P}_j$ in the target $\mathcal{P}$. We define the segmentation cost between query Q_i and ground-truth part C_j^{g} as:

$$c_{i,j}^{S} = \frac{1}{|\mathcal{P}_j|} \sum_{p \in \mathcal{P}_j} S(p, Q_i) - \frac{1}{|\mathcal{P} \setminus \mathcal{P}_j|} \sum_{p \notin \mathcal{P}_j} S(p, Q_i), \quad (6)$$

where two terms respectively encourage maximizing the segmentation probability of Q_i on points inside $\mathcal{P}_j$ and minimizing it on

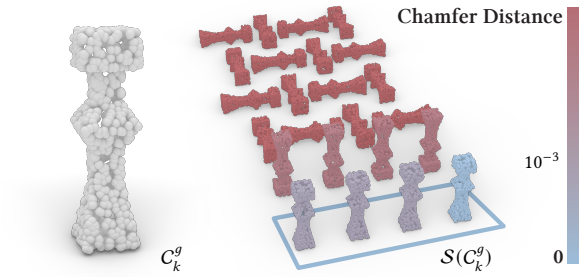


Fig. 3. Illustration of part symmetry handling. For a given part C_k^g, we evaluate all 24 discrete rotations. Valid transformations $\mathcal{S}(C_k^g)$ are selected based on a low Chamfer Distance (colored blue) to the original shape.

points outside. The hyperparameters are set to $\lambda_1 = 3, \lambda_2 = \lambda_3 = 1$. More details are provided in the supplementary material. For the k-th ground-truth part C_k^g, we denote its best match query as $Q_{[k]}$ in the followings.

The overall loss contains three tasks: part retrieval, 6-DoF poses for assembly, and point-wise segmentation. The **pose losses** include translation and rotation loss. Translation is supervised with a standard ℓ_2 loss:$\mathcal{L}_T = \frac{1}{N^g} \sum \|t_{[k]} - t_k^g\|_2^2$. For rotation loss, we notice that parts might exhibit discrete symmetries (e.g., a cube looks identical under 90° rotations). To account for such symmetries, we construct a valid set of rotation quaternions $\mathcal{S}(C_k^g)$ by applying the 24 cubic symmetry rotations and retaining only those that yield a small Chamfer distance (i.e., 10^{-3}) with respect to the ground-truth part, as illustrated in Figure 3. The training loss for part rotation is:

$$\begin{aligned} \mathcal{L}_Q &= \frac{1}{N^g} \sum_k \min_{q_k^s \in \mathcal{S}(C_k^g)} \left(\lambda_{\text{CD}} \mathcal{L}_{\text{CD}} + \lambda_{\text{Geo}} \mathcal{L}_{\text{Geo}}\right), \\ \mathcal{L}_{\text{CD}} &= \text{CD}(q_{[k]} q_k^s C_k^g, q_k^g C_k^g), \\ \mathcal{L}_{\text{Geo}} &= \frac{1}{\pi} \arccos(2\langle q_{[k]} q_k^s, q_k^g \rangle^2 - 1). \end{aligned} \tag{7}$$

The loss term $\mathcal{L}_{\text{CD}}$ measures the Chamfer Distance between the point clouds transformed using the predicted rotation $q_{[k]} q_k^s$ and the ground truth rotation q_k^g. The loss term $\mathcal{L}_{\text{Geo}}$ directly penalizes the geodesic distance of the two rotations. The two terms are complementary: $\mathcal{L}_{\text{CD}}$ offers a smooth, spatially grounded gradient, while $\mathcal{L}_{\text{Geo}}$ prevents convergence to orientations that are metrically close in point-cloud space but far in rotation space.

In addition, in order to provide supervision for the overall shape of each part and the part retrieval scores, we introduce the following soft chamfer loss:

$$\mathcal{L}_C = \frac{1}{N^g} \sum_{k=1}^{N^g} \sum_{j=1}^{M+1} \hat{S}_{[k],j} \text{CD}(q_{[k]} C_j + t_{[k]}, C_k^g), \tag{8}$$

where $\hat{S}_{[k],j}$ is the softmax retrieval probability over M candidate parts $\{C_j\}_{j=1}^M$ plus an *empty* class. This loss back-propagates shape reconstruction error through the retrieval distribution, encouraging the network to assign high probability to the candidate whose transformed geometry best matches the ground truth.

Since both retrieval and segmentation can be formulated as classification tasks, we employ cross-entropy loss to directly supervise the loss between the predicted scores R, S and the corresponding matched ground truth labels R^g, S^g: $\mathcal{L}_R = \text{CE}(R, R^g)$, $\mathcal{L}_S = \text{CE}(S, S^g)$. Finally, the overall loss function is computed as:

$$\mathcal{L} = \lambda_T \mathcal{L}_T + \mathcal{L}_Q + \lambda_C \mathcal{L}_C + \lambda_R \mathcal{L}_R + \lambda_S \mathcal{L}_S, \tag{9}$$

here we set hyperparameters as $\lambda_T = 1, \lambda_{\text{CD}} = 10, \lambda_{\text{Geo}} = 0.1, \lambda_C = 10, \lambda_R = 1, \lambda_S = 0.1$.

3.3 Inference and Post-process Optimization

This subsection describes our inference procedure and the subsequent post-processing optimization. During inference, we omit the bipartite matching step and directly assign each instance query to either a candidate part or the *empty* class by selecting the maximum-probability index from the retrieval head output. Each selected part is additionally associated with a predicted 6-DoF pose and point-wise segmentation logits for the following post-process optimization.

Suppose the network outputs segmentation logits S, along with a set of M_p selected parts $\hat{C}_k$ and their predicted poses $(\hat{t}_k, \hat{q}_k)$. We denote the transformed point cloud of each selected part as $\hat{\mathcal{P}}_k = \hat{q}_k \hat{C}_k + \hat{t}_k$. Our key observation is an *assembly–segmentation duality*. On the one hand, transforming the selected parts using the predicted poses induces a refined segmentation of the target point cloud, denoted by $\hat{S}$. On the other hand, the predicted point-wise segmentation logits define a set of segmented target point clouds $\{\mathcal{P}_k^S\}$, which provides strong geometric cues about the position and orientation of the corresponding selected part $\hat{C}_k$. We therefore refine poses and segmentation jointly, driving them towards mutual consistency while keeping the assembled shape faithful to the target:

$$\begin{aligned} &\min_{\hat{t}_k, \hat{q}_k, S} \ \lambda_{\text{Shape}} \mathcal{L}_{\text{Shape}} + \lambda_{SA} \mathcal{L}_{SA} + \lambda_{AS} \mathcal{L}_{AS}, \\ &\mathcal{L}_{\text{Shape}} = \text{CD}\left(\cup \hat{\mathcal{P}}_k, \mathcal{P}\right), \\ &\mathcal{L}_{SA} = \frac{1}{M_p} \sum_k \text{CD}(\hat{\mathcal{P}}_k, \mathcal{P}_k^S), \quad \mathcal{L}_{AS} = \text{CE}(S, \hat{S}), \end{aligned} \tag{10}$$

where we set $\lambda_{\text{Shape}} = \lambda_{\text{SA}} = 10$, $\lambda_{\text{AS}} = 0.1$. The first objective term, $\mathcal{L}_{\text{Shape}}$, minimizes the Chamfer distance between the union of transformed parts and the target shape. The second term, $\mathcal{L}_{SA}$, leverages the predicted segmentation to refine pose estimation, where $\mathcal{P}_k^S$ denotes the segmented target point cloud associated with part k according to the predicted segmentation logits S. The final term, $\mathcal{L}_{AS}$, uses the predicted part poses to refine the segmentation. $\hat{S}$ is computed as the segmentation induced by assigning each target point to the transformed part with the nearest point correspondence.

To mitigate local minima on the non-convex rotation manifold, we run the optimization in parallel from all 24 symmetry-equivalent rotations of $\hat{q}_k$ and keep the one with the lowest final loss. Each branch runs for 250 steps with AdamW [Loshchilov and Hutter 2017], a learning rate of 0.1, and cosine annealing.

Table 1. Quantitative comparisons with assembly methods. We report the assembly conditions for each method, highlighting the **best** and second best results.

		Chair				Table				Lamp			
Method	Condition	SCD ↓	PA ↑	CA ↑	SR ↑	SCD ↓	PA ↑	CA ↑	SR ↑	SCD ↓	PA ↑	CA ↑	SR ↑
DGL [Zhan et al. 2020]	-	9.62	37.50	22.56	8.06	5.23	47.44	36.65	20.13	7.84	32.70	38.23	14.18
RGL [Narayan et al. 2022]	Sequence	10.10	44.25	28.90	7.98	5.29	49.80	39.45	20.97	8.71	31.30	48.58	14.90
GPAT [Li et al. 2023]	Point Cloud	7.78	58.41	35.39	20.38	7.73	51.79	26.55	20.37	8.23	60.58	34.41	26.12
ScorePA [Cheng et al. 2023]	-	7.36	41.76	28.25	8.72	4.53	51.52	39.82	17.71	8.66	31.21	48.45	12.29
3DHPA [Du et al. 2024]	-	5.52	62.34	47.18	20.24	4.85	54.36	48.95	28.75	7.65	36.35	57.28	18.92
SPAFormer [Xu et al. 2025b]	Sequence	6.23	55.83	38.73	24.57	3.95	64.43	61.01	39.69	11.34	36.92	47.19	18.14
Assembler [Zhao et al. 2025a]	Image	7.21	66.17	52.20	24.92	4.72	70.44	57.72	43.46	16.24	51.42	57.44	33.24
Ours-PN	Point Cloud	6.63	59.26	43.05	14.25	4.44	61.61	58.26	29.55	10.24	42.57	55.64	16.62
Ours	Point Cloud	**1.19**	**86.44**	**69.90**	**49.86**	**0.72**	**84.71**	**75.95**	**59.96**	**1.63**	**75.74**	**58.37**	**52.24**

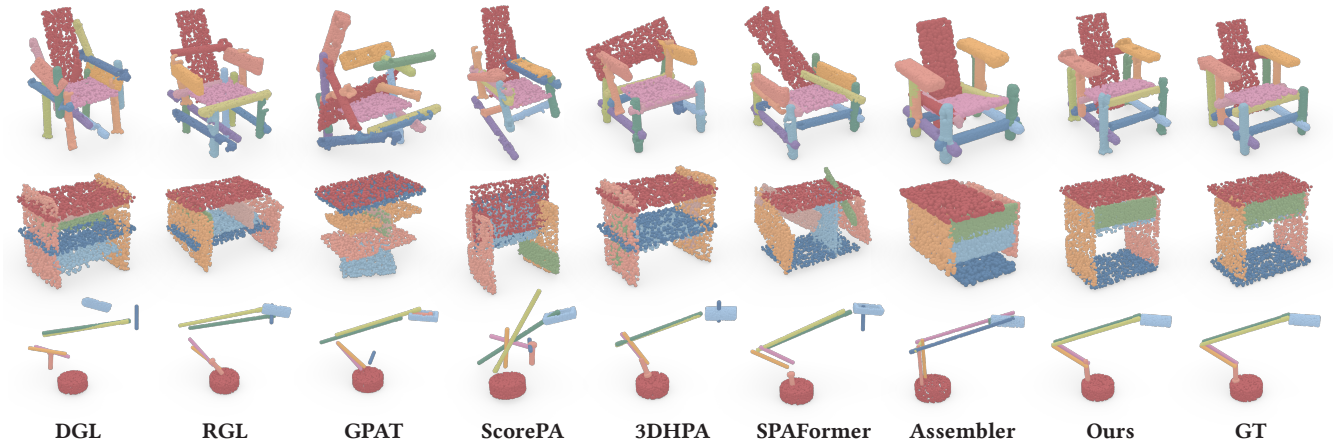


Fig. 4. Qualitative Comparisons with assembly methods.

4 Experiments

4.1 Datasets

PartNet. Following previous works [Xu et al. 2025b; Zhan et al. 2020], we benchmark our method on PartNet [Mo et al. 2019] across the *Chair*, *Table*, and *Lamp* categories at the finest-grained segmentation level, filtering for objects with 2 to 20 parts. We adopt the official train/validation/test splits of 70%/10%/20% and report results under the same evaluation protocol for fair comparison.

Curated Data. To extend our framework beyond these canonical categories to general object retrieval and assembly, we collect and process 80K+ data from multiple part-annotated 3D datasets, including PartNet [Mo et al. 2019], PartNeXt [Wang et al. 2025a], 3DCoMPaT++ [Slim et al. 2025] and PartVerse-XL [Ding et al. 2025].

3D-FRONT. We also validate our model on the 3D-FRONT [Fu et al. 2021] dataset to evaluate its performance on scene-level assembly, where the goal is to arrange multiple furniture objects into a coherent room layout.

4.2 Implementation Details

Configurations. We provide two model configurations tailored to different task scales. For category-specific benchmarks on PartNet and scene-level assembly on 3D-FRONT, we set the hidden dimension D to 384, yielding a model of 51.9M parameters. For general object assembly on the curated data, we set D to 768, resulting in 113M parameters. All models are trained on 8 NVIDIA A800 GPUs using AdamW [Loshchilov and Hutter 2017] with a batch size 16 per GPU, learning rate of 8×10^{-5}, weight decay of 10^{-4}, and cosine annealing schedule. Further details on datasets and training are provided in the supplementary material.

Train Short, Test Long. A key practical question is how to choose the training-time library size M. A large M inflates the quadratic attention cost, slows each iteration, and makes convergence noticeably harder. We therefore adopt a *train-short-test-long* strategy: during training each shape is paired with a fixed-size library of M=20 parts, comprising its own ground-truth parts (deduplicated by geometric identity) and distractor parts randomly sampled from other shapes in the training set; at inference time the same model is applied to libraries of arbitrary size $M' \gg 20$ (by default we set $M' = 100$ in the experiments). This strategy is effective for two reasons. First,

candidate parts are encoded as an unordered set without positional encodings, making the number of attention keys a runtime variable that the architecture is inherently agnostic to [Kazemnejad et al. 2023; Kool et al. 2018; Press et al. 2021]. Second, resampling distractors at every iteration exposes the model to thousands of distinct library compositions, covering a negative pool far larger than M. We empirically verify this strategy in subsection 4.5.

4.3 Assembly Comparisons on PartNet Dataset

Comparisons with Assembly Methods. We first evaluate our method against existing approaches that assemble a fixed set of given parts. Following previous works, we conduct the experiments on *Chair*, *Table*, *Lamp* categories from the PartNet [Mo et al. 2019] with fixed parts as input, using Shape Chamfer Distance (SCD), Part Accuracy (PA), Connectivity Accuracy (CA), and Success Rate (SR) as evaluation metrics (details are provided in the supplementary material). Since previous approaches mostly use PointNet/PointNet++ as their backbone and do not employ post-processing optimization, we additionally provide a variant (Ours-PN) using PointNet++ without optimization for fair comparison. Figure 4 visualizes the qualitative results, and Table 1 summarizes the quantitative performance. The results demonstrate that our 3D target point cloud prior based approach achieves superior assembly accuracy compared to other fixed-part assembly methods. Moreover, Ours-PN outperforms earlier methods (such as DGL and RGL), though it does not show a significant advantage over recent methods like 3DHPA and SPAFormer. We conjecture that these methods benefit from exploiting the fixed, known part set through super-part relation reasoning or order encoding, a prior unavailable in our library-based harder setting. Notably, our full method surpasses all baselines, particularly on the SCD metric, as the optimization enables parts to accurately fit the target shape.

Comparisons with Retrieval and Assembly Methods. In the second setting, we evaluate our full pipeline on the task of part retrieval and assembly. We compare our approach against two distinct baselines: a Genetic Algorithm (GA) based method, which employs heuristic search and brute-force optimization to align parts onto the target shape, and another baseline UPRA [Xu et al. 2023]. Since UPRA does not release its training data or pretrained checkpoints, we retrain its model on PartNet using the official codebase. For a fair comparison, all methods use the same candidate library consisting of 1,000 unique parts from the corresponding category's test split (1,000 points per part) and the same 10,000-point target point clouds. As shown in Table 2 and Figure 5, our method achieves lower assembly error while requiring significantly less inference time per shape. All methods are benchmarked on a single NVIDIA A800 GPU. Note that the reported inference time does not include data loading.

4.4 Ablation Study

We conduct ablation studies on the PartNet *Chair* category to validate each component, with results summarized in Table 3, where SmIoU (Shape mIoU) serves as the evaluation metric for segmentation quality. We compare our full model against the following baselines: *w/o Optimization* omits the joint optimization stage where assembly and segmentation mutually reinforce each other; *w/o A-Matrix* replaces A-Matrix rotation representation with direct quaternion regression; *w/o Symmetry* removes symmetry matching and uses direct rotation loss; *w/o Fusion* removes the self- and cross-attention based feature fusion between target and part clouds; *Segmentation+PCA* discards all assembly losses and recovers pose indirectly by aligning PCA axes of the predicted segmentation with canonical part poses. Results show that each component contributes positively. In particular, the comparison with *w/o Optimization* confirms that the joint optimization stage improves both assembly and segmentation performance, while *Segmentation+PCA* demonstrates that indirect pose recovery via independent segmentation is inferior to our unified framework, underscoring the necessity of joint training for assembly and segmentation. More ablations are provided in the supplementary material.

Table 2. Quantitative comparisons with retrieval and assembly methods.

	SCD ↓			Average Time (s)		
Method	Chair	Table	Lamp	Chair	Table	Lamp
GA	8.27	6.57	7.81	99.0	71.4	101.6
UPRA [Xu et al. 2023]	4.46	2.29	3.24	227.6	296.0	99.1
Ours	**1.84**	**0.70**	**3.16**	**6.43**	**6.35**	**4.89**

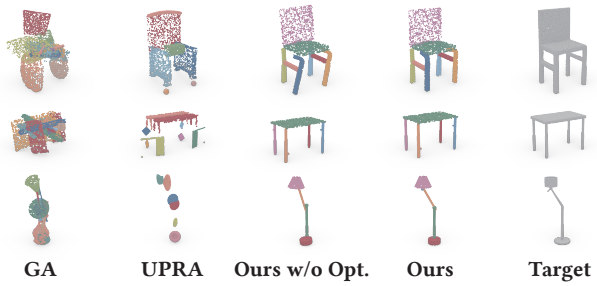


Fig. 5. Qualitative comparisons with retrieval and assembly methods. Each shape is reconstructed from a library of 1,000 parts. Our method adaptively handles the variable part count and produces more accurate assembly results combined with segmentation-enhanced optimization.

Table 3. Ablation studies on PartNet Chair category. Segmentation metrics are omitted for *w/o A-Matrix* and *w/o Symmetry*, as these are designed for assembly task.

	SCD ↓	PA ↑	CA ↑	SR ↑	SmIoU ↑
w/o Optimization	3.96	80.16	61.31	33.65	74.88
w/o A-Matrix	4.43	78.04	56.45	31.12	–
w/o Symmetry	4.15	79.90	61.27	32.98	–
w/o Fusion	4.09	80.23	60.76	33.93	75.15
Segmentation+PCA	5.95	63.22	32.01	23.06	64.96
Ours	**1.19**	**86.44**	**69.90**	**49.86**	**77.16**

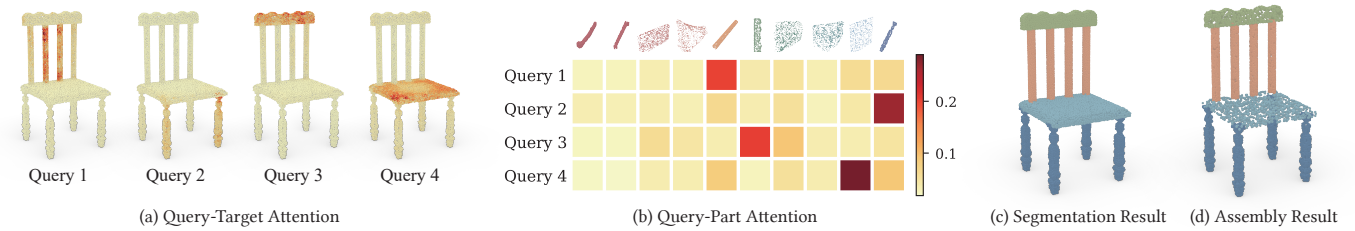


Fig. 6. Attention visualization. (a): query-target attention weights. (b): query-parts attention weights. (c)(d): the final segmentation and assembly result.

Effect of Part Library Size. As discussed before, our query-based transformer architecture naturally accepts a variable number of input parts. To evaluate the effect of library size, we randomly sample 100 shapes from the PartNet *Chair* test set and measure assembly quality, inference time, and GPU memory under library sizes of 20, 50, 100, 200, 500, and 1000 unique parts on a single NVIDIA A800 GPU. As shown in Figure 7, Chamfer distance and inference time increase only marginally as the library grows. Although GPU memory grows due to the quadratic complexity inherent in the transformer's attention mechanism, they remain within a practical range (approximately 3.2GB per shape even at 1,000 parts), which is sufficient for most application scenarios. Qualitatively, despite the increasing retrieval difficulty posed by larger libraries, the assembled shapes maintain strong global consistency with the target, with only minor local deviations. This validates the effectiveness of our train-short-test-long strategy and demonstrates the robustness of the model to varying numbers of input parts.

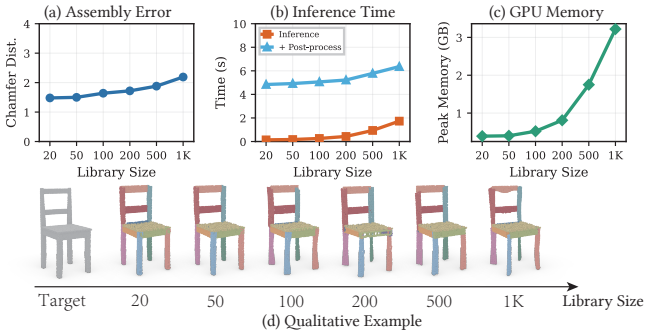


Fig. 7. Effect of part library size on assembly performance. (a) Assembly error (SCD). (b) Average inference time. (c) Average GPU memory usage across different library sizes. (d) A qualitative example showing assembly results under varying library sizes.

4.5 More Analysis

Attention Map Visualization. We visualize representative cross-attention weights from the transformer decoder layers in Figure 6. The query-target attention (left) shows that each query attends to a localized region of the target point cloud, indicating that the model learns to decompose the shape into semantically meaningful segments. The query-part attention (middle) reveals that each query identifies the most geometrically compatible part in the library. Through iterative cross-attention, each valid query progressively establishes correspondence between a part and its target placement, yielding consistent segmentation and assembly results (right).

Additional Results. Figure 12 visualizes more assembly results on our curated test data. Our framework adaptively retrieves an appropriate number of parts based on the target, producing faithful assemblies ranging from simple to complex geometries.

Assembly w/o Perfect Parts & Failure Cases. We further evaluate our model under the challenging setting where the part library does not contain the perfect matched parts (i.e., ground-truth parts of the target shape). As shown in Figure 8, the model can still retrieve geometrically plausible substitutes that closely approximate the target (column 3). However, failure cases arise when the parts in the library differ significantly in scale from the target (column 5) or when there are no geometrically similar parts available (column 4).

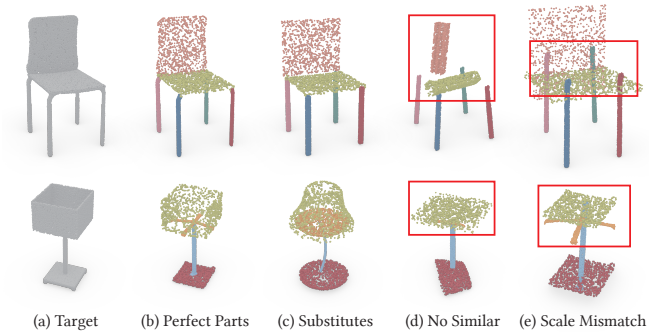


Fig. 8. Assembly with/without perfect parts and failure cases. (a): Input target point cloud. (b) Assembly with perfect parts. (c) Assembly without perfect parts, our method still retrieves plausible substitutes. (d) No geometrically similar parts in the library. (e) The parts and target differ in scale.

4.6 Exploratory Extensions

We further demonstrate the scalability of our framework through three exploratory extensions.

Image-based Assembly. First, we extend it to the image-based retrieval and assembly by leveraging recent advances in 3D generation (Figure 10). Specifically, we utilize Hunyuan3D [Lai et al. 2025; Zhao et al. 2025b] to lift single-view images into 3D meshes, from which we sample 10,000 surface points as the input target point cloud. To

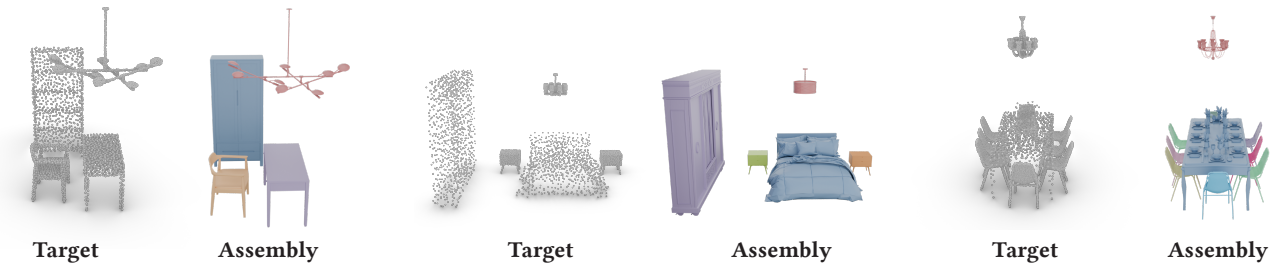


Fig. 9. Scene assembly visualizations. Left: Input target point cloud. Right: Our assembled result.

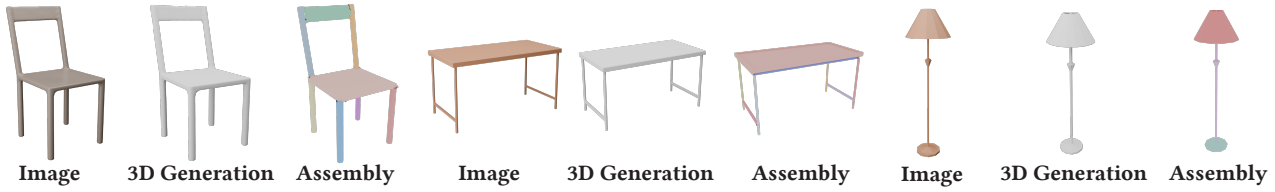


Fig. 10. Examples of image-based assemblies. Given a single image as reference (left), we first lift it to a 3D model using Hunyuan3D (middle). This model is then sampled into a point cloud and utilized as the target to obtain the final assembly (right).

handle the scale inconsistency between the generated target and the input parts, we normalize the target to a unit sphere and incorporate object scale as an optimizable parameter during post-processing.

Scene-level Assembly. Second, for scene-level assembly on the 3D-FRONT dataset, given an input scene point cloud and a set of candidate assets, our model automatically retrieves and places the appropriate assets to reconstruct the target scene (Figure 9).

Generalization to Real-world Scans. Third, to evaluate the real-world applicability, we conduct a zero-shot generalization experiment using scanned point clouds from Redwood dataset [Choi et al. 2016]. For each scan, we randomly sample 100 parts from PartNet as the library. As shown in Figure 11, our method retrieves appropriate parts and produces plausible assemblies that approximate the scanned inputs, demonstrating the zero-shot generalizability of our framework to real-world inputs. Additional results are provided in the supplementary material.

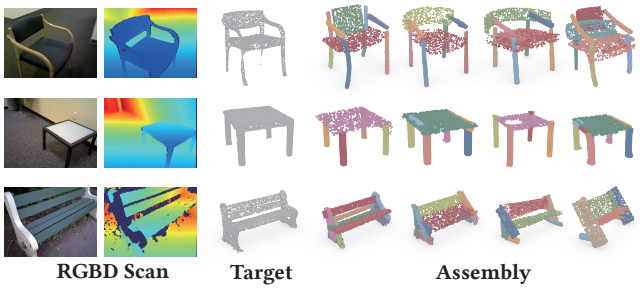


Fig. 11. Generalization to real-world scans. Left: Raw RGBD captures from Redwood dataset. Middle: Target point cloud sampled from the scanned model. Right: Diverse assembly results using parts retrieved from the PartNet library to reconstruct the targets.

5 Conclusion

We present PART, a unified framework for 3D part retrieval and assembly. By leveraging a transformer-based architecture, our method jointly retrieves parts from a candidate library and estimates their 6-DoF poses in a single forward pass, successfully handling variable-length outputs. Moreover, the joint learning of assembly and segmentation, together with a novel segmentation-enhanced optimization module, further improves the assembly accuracy. Extensive experiments on scene assembly and single-view image-guided assembly further demonstrate the scalability and potential of our framework.

Our work has several limitations. For example, we do not consider inter-part connections and physical constraints such as collision avoidance and structural stability. Additionally, our method assumes rigid transformations and predicts only 6-DoF poses for each part, without accounting for part scaling or non-rigid deformations. For future work, incorporating physics-aware assembly constraints, retrieving and adapting parametric CAD parts, and leveraging generative models or large language models (LLMs) are promising directions to explore.

Acknowledgments

We thank the anonymous reviewers for their constructive comments and valuable suggestions. This work was supported by the National Natural Science Foundation of China (62025207), the BYD-HKUST Joint Lab Research Project (BYD26IS02), and the Shenzhen Loop Area Institute (FPF10120250006).

References

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 213–229.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).



Fig. 12. Additional retrieval and assembly results across various categories with varying number of parts. For each shape, the part library contains 100 parts randomly sampled from the test split (including the needed parts). PART is able to retrieve an appropriate number of parts conditioned on the target and assemble them accurately.

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

Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. 2023. Score-PA: Score-based 3D Part Assembly. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA. https://papers.bmvc2023.org/0438.pdf

Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. 2016. A Large Dataset of Object Scans. *arXiv:1602.02481* (2016).

Lihe Ding, Shaocong Dong, Yaokun Li, Chenjian Gao, Xiao Chen, Rui Han, Yihao Kuang, Hong Zhang, Bo Huang, Zhanpeng Huang, et al. 2025. FullPart: Generating each 3D Part at Full Resolution. *arXiv preprint arXiv:2510.26140* (2025).

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

Bi'an Du, Xiang Gao, Wei Hu, and Renjie Liao. 2024. Generative 3d part assembly via part-whole-hierarchy message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20850–20859.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 2021. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10933–10942.

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

Andreas René Geist, Jonas Frey, Mikel Zhobro, Anna Levina, and Georg Martius. 2024. Learning with 3D rotations, a hitchhiker's guide to SO(3). In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*. PMLR, 15331–15350.

Prakhar Jaiswal, Jinmiao Huang, and Rahul Rai. 2016. Assembly-based conceptual 3D modeling with unlabeled components using probabilistic factor graph. *Computer-Aided Design* 74 (2016), 45–54.

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

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems* 36 (2023), 24892–24928.

Wouter Kool, Herke Van Hoof, and Max Welling. 2018. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475* (2018).

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. 2025. Hunyuan3D 2.5: Towards High-Fidelity 3D Assets Generation with Ultimate Details. *arXiv preprint arXiv:2506.16504* (2025).

Sihang Li, Zeyu Jiang, Grace Chen, Chenyang Xu, Siqi Tan, Xue Wang, Irving Fang, Kristof Zyskowski, Shannon P McPherron, Radu Iovita, Chen Feng, and Jing Zhang. 2025. GARF: Learning Generalizable 3D Reassembly for Real-World Fractures. In *International Conference on Computer Vision (ICCV)*.

Yichen Li, Kaichun Mo, Yueqi Duan, He Wang, Jiequan Zhang, and Lin Shao. 2024. Category-level multi-part multi-joint 3d shape assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3281–3291.

Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. 2020. Learning 3d part assembly from a single image. In *European conference on computer vision*. Springer, 664–682.

Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. 2011. Globfit: Consistently fitting primitives by discovering global relations. In *ACM SIGGRAPH 2011 papers*. 1–12.

Yulong Li, Andy Zeng, and Shuran Song. 2023. Rearrangement Planning for General Part Assembly. In *Conference on Robot Learning*. PMLR.

Chen Lin, Andrew J Hanson, and Sonya M Hanson. 2023. Algebraically rigorous quaternion framework for the neural network pose estimation problem. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14097–14106.

Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. 2023. Marching-primitives: Shape abstraction from signed distance function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8771–8780.

Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. 2021. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2949–2958.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

Ishan Misra, Rohit Girdhar, and Armand Joulin. 2021. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2906–2917.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. 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.

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

Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. 2019. Superquadrics Revisited: Learning 3D Shape Parsing beyond Cuboids. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Valentin Peretroukhin, Matthew Giamou, David M. Rosen, W. Nicholas Greene, Nicholas Roy, and Jonathan Kelly. 2020. A Smooth Representation of SO(3) for Deep Rotation Learning with Uncertainty. In *Proceedings of Robotics: Science and Systems (RSS'20)* (2020-07-12/2020-07-16).

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409* (2021).

Ava Pun, Kangle Deng, Ruixuan Liu, Deva Ramanan, Changliu Liu, and Jun-Yan Zhu. 2025. Generating physically stable and buildable brick structures from text. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14798–14809.

Ruwen Schnabel, Roland Wahl, and Reinhard Klein. 2007. Efficient RANSAC for point-cloud shape detection. In *Computer graphics forum*, Vol. 26. Wiley Online Library, 214–226.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. 2022. Mask3d: Mask transformer for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105* (2022).

Habib Slim, Xiang Li, Yuchen Li, Mahmoud Ahmed, Mohamed Ayman, Ujjwal Upadhyay, Ahmed Abdelreheem, Arpit Prajapati, Suhail Pothigara, Peter Wonka, et al. 2025. 3dcompat++: An improved large-scale 3d vision dataset for compositional recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. 2023. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 2393–2401.

Tao Sun, Liyuan Zhu, Shengyu Huang, Shuran Song, and Iro Armeni. 2025. Rectified Point Flow: Generic Point Cloud Pose Estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kuan Tian, Zhihao Hu, Yonghang Guan, and Jun Zhang. 2025. LLM-Primitives: Large Language Model for 3D Reconstruction with Primitives. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–12.

Chenrui Tie, Shengxiang Sun, Yudi Lin, Yanbo Wang, Zhongrui Li, Zhouhan Zhong, Jinxuan Zhu, Yiman Pang, Haonan Chen, Junting Chen, et al. 2025a. Manual2Skill++: Connector-Aware General Robotic Assembly from Instruction Manuals via Vision-Language Models. *arXiv preprint arXiv:2510.16344* (2025).

Chenrui Tie, Shengxiang Sun, Jinxuan Zhu, Yiwei Liu, Jingxiang Guo, Yue Hu, Haonan Chen, Junting Chen, Ruihai Wu, and Lin Shao. 2025b. Manual2Skill: Learning to Read Manuals and Acquire Robotic Skills for Furniture Assembly Using Vision-Language Models. In *Proceedings of Robotics: Science and Systems (RSS)*.

Shubham Tulsiani, Hao Su, Leonidas J Guibas, Alexei A Efros, and Jitendra Malik. 2017. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2635–2643.

Mikaela Angelina Uy, Vladimir G Kim, Minhyuk Sung, Noam Aigerman, Siddhartha Chaudhuri, and Leonidas J Guibas. 2021. Joint learning of 3d shape retrieval and deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11713–11722.

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

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

Penghao Wang, Yiyang He, Xin Lv, Yukai Zhou, Lan Xu, Jingyi Yu, and Jiayuan Gu. 2025a. Partnext: A next-generation dataset for fine-grained and hierarchical 3d part understanding. *arXiv preprint arXiv:2510.20155* (2025).

Weihao Wang, Yu Lan, Mingyu You, and Bin He. 2025b. Imagine: Image-Guided 3D Part Assembly with Structure Knowledge Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 7889–7897.

Weihao Wang, Mingyu You, Hongjun Zhou, and Bin He. 2024. PhysFiT: Physical-aware 3D Shape Understanding for Finishing Incomplete Assembly. *ACM Transactions on Graphics* 44, 1 (2024), 1–16.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. 2024. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4840–4851.

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. 2022. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems* 35 (2022), 33330–33342.

Sitao Xiang and Hao Li. 2020. Revisiting the continuity of rotation representations in neural networks. *arXiv preprint arXiv:2006.06234* (2020).

Boshen Xu, Sipeng Zheng, and Qin Jin. 2025b. Spaformer: Sequential 3d part assembly with transformers. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 1317–1327.

Hao Xu, Yuqing Zhang, Yiqian Wu, Xinyang Zheng, Yutao Liu, Xiangjun Tang, Yunhan Yang, Ding Liang, Yingtian Liu, Yuanchen Guo, et al. 2025a. LegoACE: Autoregressive Construction Engine for Expressive LEGO® Assemblies. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–11.

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

Kaizhi Yang and Xuejin Chen. 2021. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM Transactions On Graphics (TOG)* 40, 4 (2021), 1–11.

Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, Hao Dong, et al. 2020. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems* 33 (2020), 6315–6326.

Jiahao Zhang, Anoop Cherian, Cristian Rodriguez, Weijian Deng, and Stephen Gould. 2025. Manual-pa: Learning 3d part assembly from instruction diagrams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6304–6314.

Rufeng Zhang, Tao Kong, Weihao Wang, Xuan Han, and Mingyu You. 2022. 3d part assembly generation with instance encoded transformer. *IEEE Robotics and Automation Letters* 7, 4 (2022), 9051–9058.

Xinyi Zhang, Bingyang Wei, Ruixuan Yu, and Jian Sun. 2026. Coarse-to-Fine 3D Part Assembly via Semantic Super-Parts and Symmetry-Aware Pose Estimation. *Advances in Neural Information Processing Systems* 38 (2026), 171393–171420.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. 2021. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*. 16259–16268.

Wang Zhao, Yan-Pei Cao, Jiale Xu, Yuejiang Dong, and Ying Shan. 2025a. Assembler: Scalable 3D Part Assembly via Anchor Point Diffusion. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–11.

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. 2025b. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025).

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5745–5753.

Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 2017. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 900–909.