MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Manual-PA: Learning 3D Part Assembly from Instruction Diagrams

Zhang, Jiahao; Cherian, Anoop; Rodriguez, Cristian; Deng, Weijian; Gould, Stephen

TR2025-139 October 01, 2025

Abstract

Assembling furniture amounts to solving the discretecontinuous optimization task of selecting the furniture parts to assemble and estimating their connecting poses in a physically realistic manner. The problem is hampered by its combinatorially large yet sparse solution space thus making learning to assemble a challenging task for current machine learning models. In this paper, we attempt to solve this task by leveraging the assembly instructions provided in diagrammatic manuals that typically accompany the furniture parts. Our key insight is to use the cues in these diagrams to split the problem into discrete and continuous phases. Specifically, we present Manual-PA, a transformer-based instruction Manual-guided 3D Part Assembly framework that learns to semantically align 3D parts with their illustrations in the manuals using a contrastive learning backbone towards predicting the assembly order and infers the 6D pose of each part via relating it to the final furniture depicted in the manual. To validate the efficacy of our method, we conduct experiments on the benchmark PartNet dataset. Our results show that using the diagrams and the order of the parts lead to significant improvements in assembly performance against the state of the art. Further, Manual-PA demonstrates strong generalization to real-world IKEA furniture assembly on the IKEA-Manual dataset.

IEEE International Conference on Computer Vision (ICCV) 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Manual-PA: Learning 3D Part Assembly from Instruction Diagrams

Jiahao Zhang[1] Anoop Cherian[2] Cristian Rodriguez[3] Weijian Deng[1] Stephen Gould[1]
[1]The Australian National University, [2]Mitsubishi Electric Research Labs
[3]The Australian Institute for Machine Learning
[1]{first.last}@anu.edu.au [2]cherian@merl.com [3]crodriguezop@gmail.com

Abstract

Assembling furniture amounts to solving the discrete-continuous optimization task of selecting the furniture parts to assemble and estimating their connecting poses in a physically realistic manner. The problem is hampered by its combinatorially large yet sparse solution space thus making learning to assemble a challenging task for current machine learning models. In this paper, we attempt to solve this task by leveraging the assembly instructions provided in diagrammatic manuals that typically accompany the furniture parts. Our key insight is to use the cues in these diagrams to split the problem into discrete and continuous phases. Specifically, we present Manual-PA, a transformer-based instruction Manual-guided 3D Part Assembly framework that learns to semantically align 3D parts with their illustrations in the manuals using a contrastive learning backbone towards predicting the assembly order and infers the 6D pose of each part via relating it to the final furniture depicted in the manual. To validate the efficacy of our method, we conduct experiments on the benchmark PartNet dataset. Our results show that using the diagrams and the order of the parts lead to significant improvements in assembly performance against the state of the art. Further, Manual-PA demonstrates strong generalization to real-world IKEA furniture assembly on the IKEA-Manual dataset.

1. Introduction

With the rise of the DIY trend, nowadays it is the consumers who are increasingly assembling products like IKEA furniture [2, 3, 21, 57, 58], LEGO structures [47, 49], and exercise equipments [45, 61]. This paradigm shift can reduce the product cost, but often poses challenges for untrained consumers, as an incorrect assembly in any step may lead to irreparable outcomes. Thus, automating the assembly or providing active help in the process is often desired.

Automated assembly faces the dual challenge of large combinatorial solution space, parameterized by the number of discrete parts in the assembly, and the need for a precise 6D pose estimation that correctly connects parts between each other. The limited number of feasible, stable assembly sequences makes this discrete-continuous optimization problem extremely challenging, even for advanced neural networks, especially as the number of parts increases [54]. Fortunately, most assembly tasks include language-agnostic instruction manuals, such as diagrams, delineating the assembly. While such manuals help shrink the solution search space, the problem of correctly matching the 2D manual diagrams with their corresponding 3D object parts and inferring their assembly pose and order remains a significant challenge for automated systems.

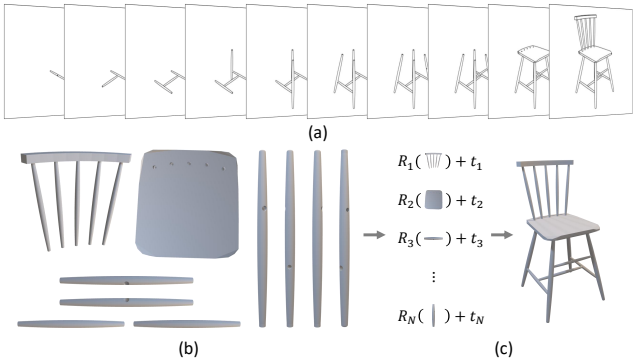

Figure 1. An illustration of the manual-guided 3D part assembly task. Given (a) a diagrammatic manual book demonstrating the step-by-step assembly process and (b) a set of texture-less furniture parts, the goal is to (c) infer the order of parts for the assembly from the manual sequence and predict the 6DoF pose for each part such that the spatially transformed parts assembles the furniture described in the manual.

This paper introduces a method to enable machines to learn how to assemble shapes by following a visual instruction manual, a task we call *manual-guided 3D part assembly*. As illustrated in Fig. 1, given a set of parts and a manual, our approach interprets the manual's step-by-step diagrams to predict the precise 6D pose of each part, assembling them into the target shape. There are two lines of prior approaches that have similarities to our proposed task, namely methods that focus on the geometry of individual parts and inter-part relationships [8, 9, 23, 56, 59, 60],

and methods such as MEPNet [49] that learns the assembly in specific conditions, such as LEGO shapes. The former rely on priors like peg-hole joints [25] or sequence generation [54] to reduce search complexity but are generative and may yield unstable assemblies due to occlusions [23]. The latter often simplifies the task by assuming that parts are provided step-by-step, e.g., LEGO manuals. However, furniture manuals, such as IKEA, may not have such information, and it is necessary to determine which parts are used at each step before assembly. Additionally, while LEGO tasks feature standardized "stud" joints, general furniture assembly is more complex and lacks such constraints.

Based on the above observations, we identify two key challenges in our task. First, how to detect which part or parts are added at each step in the assembly diagrams. Given that these diagrams depict incremental assembly stages, the core challenge is to recognize newly added parts at each step. This requires aligning 2D assembly diagrams with the 3D parts to establish a correct assembly sequence. Second, how to effectively align the assembly process with the learned order. A straightforward approach is to directly use the learned sequence as the order for predicting poses, *e.g.*, by adopting an auto-regressive prediction method similar to MEPNet [49]. However, this approach carries the risk of accumulating errors—if a mistake occurs in an earlier step, subsequent steps are likely to fail as well. On the other hand, completely disregarding the sequence forces the assembly model to implicitly find correspondences, which makes both learning and prediction significantly more difficult. Therefore, the inferred sequence should serve as a *soft guidance* for the assembly process, helping each part focus more on the relevant step diagram.

To address these challenges, we present Manual-guided Part Assembly (Manual-PA), a novel transformer-based neural network that predicts correspondences between step diagrams and parts by computing their semantic similarity, which is learned using contrastive learning. Using this similarity matrix, we establish an assembly order via solving an optimal assignment problem between the aligned features of the 3D object parts and the ordered manual diagram steps, followed by permuting each part's positional encoding according to the learned order. Finally, a transformer decoder fuses the two feature modalities to predict the final 6DoF pose for each part. Notably, the cross-attention between step diagrams and parts is guided by the positional encoding, ensuring that each step diagram receives a higher attention score for its corresponding part.

To empirically validate the effectiveness of Manual-PA, we conduct experiments on the standard PartNet benchmark dataset [27], on the categories of assembling the chair, table and storage classes. Our experiments and ablation studies clearly show that Manual-PA leads to state-of-the-art results on multiple metrics, including success rate, outperforming prior methods by a significant margin. Results further showcase the strong generalizability of our approach to real-world IKEA furniture assembly through experiments using the IKEA-Manual dataset [50].

We summarize our key contributions below:

1. We introduce a new problem setup of assembling 3D parts using diagrammatic manuals for guidance.
2. We present a novel framework, Manual-PA, for solving this task that learns the assembly order of the 3D parts, which is then used as soft guidance for the assembly.
3. We present experiments on the benchmark PartNet and IKEA-Manual datasets demonstrating state-of-the-art performances and cross-category generalization.

2. Related Works

3D Part Assembly involves reconstructing complete shapes from sets of 3D parts, a task originally assisted by leveraging semantic part information [6, 11, 15, 43, 52, 55]. With the introduction of the PartNet dataset [27], recent works [8, 9, 28, 56, 59, 60] leverage geometric features. For instance, 3DHPA [9] uses part symmetry for a hierarchical assembly, SPAFormer [54] circumvents combinatorial explosion by generating a part order, and Joint-PA [25] introduces a peg-hole constraint to facilitate assembly. However, these methods are generative towards producing plausible final shapes without aiming for preciseness. Approaches like GPAT [24] and Image-PA [23] add guidance from point clouds or rendered images, respectively. Our work extends this direction by introducing guidance through instruction manuals, similar to how humans would do.

Permutation Learning is often used in self-supervised methods for learning image representations [4, 16, 31, 35, 40]. However, in this work, we seek to align the 3D parts to instruction steps, for which we adopt a contrastive learning [14, 17, 22, 39] using a novel cross-modal alignment setup.

6DoF Object Pose Estimation methods can be classified into instance-level [36, 38, 53], category-level [44, 48], and unseen [20, 51]. Our approach relates to the unseen object category as furniture typically comprises novel parts. When the 3D models of these parts are available, many methods [7, 13, 26, 30, 34] rely on feature matching followed by the PnP algorithm to infer pose. However, IKEA-style manuals are represented as line drawings making accurate feature extraction for matching to 3D parts challenging. Alternatively, some approaches [20, 51] use end-to-end models to directly predict the 6D pose. A common approach is to use CNOS [29] for object segmentation. However, in assembly diagrams, where multiple parts are connected and depicted together, SAM-based methods [18] often struggle with accurate segmentation. Our approach addresses these challenges using an end-to-end framework that uses a cross-attention scheme to enable each part to attend to its relevant region in the step diagram.

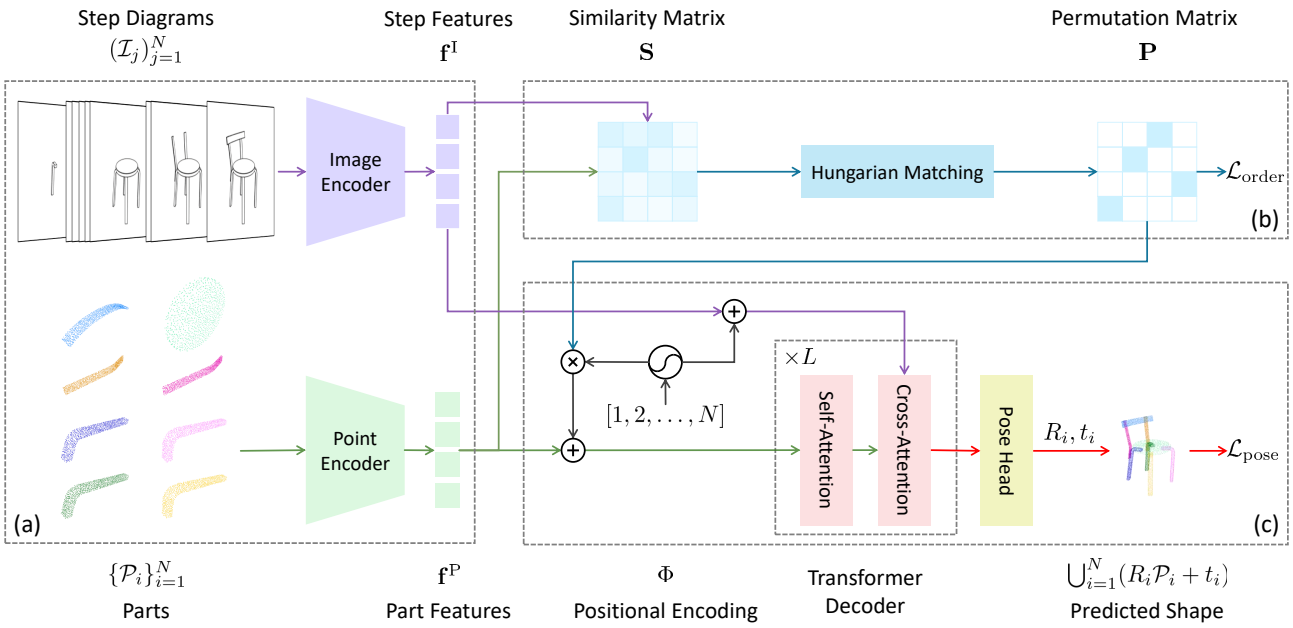

Figure 2. Overview of our proposed method Manual-PA. (a) Feature extraction (Sec. 3.2): we extract semantic and geometrical features from both the step diagrams of the assembly manual and the corresponding part point clouds using the image encoder and point encoder, respectively. (b) Manual-guided part permutation learning (Sec. 3.3): we compute a similarity matrix $\mathbf{S}$ between the two modalities, and subsequently apply the Hungarian algorithm to obtain the permutation matrix $\mathbf{P}$ for the parts. (c) Manual-guided part pose estimation (Sec. 3.4): we add positional encodings (PE) Φ to both the step diagrams and parts, where the PE order for parts is determined by the order predictions from (b). This is followed by a transformer decoder to enable multimodal feature fusion and interaction, along with a pose prediction head to determine the rotation R_i and translation t_i of each part. The predicted poses are then applied to the corresponding parts to obtain the final assembled shape $\mathbf{S}$.

3. Manual-Guided Part Assembly

In this section, we first provide a concrete formal definition of the *manual-based 3D part assembly* task in Sec. 3.1. We then present a detailed explanation in Secs. 3.2 to 3.4 of the proposed learning pipeline as depicted in Fig. 2, followed by a description of the training process and the loss functions used for each stage in Sec. 3.5.

3.1. Problem Formulation

Our task involves two different input modalities: a set of 3D parts and a corresponding instruction manual. The 3D parts are represented as a multiset[1] of unordered point clouds, denoted by $\{\mathcal{P}_i\}_{i=1}^N$. Note that the number of parts N may differ from furniture to furniture. The manual consists of an ordered sequence of N step diagram images, denoted by $(\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N)$; each diagram depicting an incremental step in the assembly process of a 3D furniture shape $\mathcal{S}$. These step diagrams are 2D line drawings, which are 2D projections of textureless 3D CAD models of these parts captured from a camera $\mathcal{C}$, and we assume for each step, only one additional part is attached to the furniture assembled thus far. Our goal is to align the 3D parts with their delineations in the respective manual steps towards predicting the assembly order of the parts and the part poses $\{(R_i, t_i) \in SE(3)\}_{i=1}^N$ for the final assembled furniture, where R_i, t_i represent the 3×3 rotation matrix and the three-dimensional translation vector respectively for the i-th part in the SE(3) transformation space. That is, these parts undergo rigid transformations such that their combination constructs the complete 3D furniture shape $\mathcal{S}$ as seen from the viewpoint of camera $\mathcal{C}$.

[1] We allow multiple identical parts, *e.g.*, chair legs, to appear in the set.

3.2. Feature Extraction

We first use off-the-shelf encoders to extract semantic latent features from the inputs. For the 3D part point clouds $\{\mathcal{P}_i\}_{i=1}^N$, we utilize a lightweight version of PointNet [37], similar to the one used in Image-PA [23], followed by a linear layer to map the features to a common dimension D, resulting in geometric part features $\mathbf{f}^{\mathrm{P}} \in \mathbb{R}^{N \times D}$. For sequences of step diagrams $(\mathcal{I}_j)_{j=1}^N$, given that the assembly instructions are provided step-by-step in an incremental manner, our primary focus is on the differences between consecutive steps. For any two consecutive diagram images $\mathcal{I}_j$ and $\mathcal{I}_{j+1}$, for $j = 1, 2, \ldots, N-1$, we compute the difference image given by $|\mathcal{I}_j - \mathcal{I}_{j+1}|$ that shows which new part is added to the thus-far assembled furniture. Next, this

difference image is patchified into K image patches, that are then fed to DINOv2 image encoder [5, 33] to extract features, followed by a linear layer to map these features to the same dimensionality D, resulting in semantic image features $\mathbf{f}^{\text{I}} \in \mathbb{R}^{N\times K\times D}$.

3.3. Manual-Guided Part Permutation Learning

In our task, we seek to learn the order of a set of parts conditioned on an ordered sequence of step diagrams. This problem is similar to learning the feature correspondences between two modalities in order to align them, with an ideal alignment corresponding to a permutation matrix. Specifically, our objective is to derive a similarity matrix between the two modalities such that each modality feature is unambiguously assigned to its corresponding feature in the other modality and we desire this assignment matches with the ground truth order in the step sequence in the manual. Formally, we construct the similarity matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ as:

$$\mathbf{S}_{ij} = \text{sim}(\mathbf{f}^{\text{P}}_i, \mathbf{g}^{\text{I}}_j), \tag{1}$$

where $\mathbf{g}^{\text{I}} \in \mathbb{R}^{N\times D}$ is derived from the $N \times K \times D$-dimensional step-diagram features $\mathbf{f}^{\text{I}}$ by performing a max-pooling operation on the patch dimension K. We measure the similarity in the feature space using the dot product as $\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a} \cdot \mathbf{b}$.

Next, we obtain the permutation matrix $\mathbf{P} \in \{0, 1\}^{N\times N}$ by solving a bipartite matching optimization problem using the Hungarian matching algorithm [19] with cost $\mathbf{C} = -\mathbf{S}$:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{C}_{ij}\mathbf{P}_{ij} \\ \text{subject to} \quad & \textstyle\sum_{j=1}^{N} P_{ij} = 1, \quad \forall i \in \{1, \dots, N\} \\ & \textstyle\sum_{i=1}^{N} P_{ij} = 1, \quad \forall j \in \{1, \dots, N\} \\ & P_{ij} \in \{0, 1\}, \quad \forall i, j \in \{1, \dots, N\}, \end{aligned} \tag{2}$$

where we minimize the cost, subject to constraints ensuring $\mathbf{P}$ is a permutation matrix, i.e., each row and column of $\mathbf{P}$ contains exactly one entry equal to 1, with all other entries being 0. The final part order can be obtained by $\sigma = \text{argmax}(\mathbf{P})$ where argmax is applied columnwise.

3.4. Manual-Guided Part Pose Estimation

Positional Encoding. Since the attention mechanism in the transformer is permutation-invariant, we follow Vaswani et al. [46] in using positional encoding to convey the order of both step diagrams and parts to the model. Formally, the positional encoding is defined as a function that maps a scalar to a sinusoidal encoding, $\phi : \mathbb{N} \to \mathbb{R}^D$:

$$\phi(x)_{2i} = \sin\left(\frac{x}{\tau_p^{2i/d}}\right), \ \phi(x)_{2i-1} = \cos\left(\frac{x}{\tau_p^{2i/d}}\right), \tag{3}$$

where $i = 1, \dots, d/2$ indexes the dimension and τ_p is a temperature hyperparameter. By applying ϕ to encode the sequential order of step diagrams as they appear in the manual (i.e., using x as a simple increasing sequence $1, 2, \dots, N$), we obtain a positional encoding $\Phi \in \mathbb{R}^{N\times D}$, where $\mathbf{p}^{\text{I}} := \Phi$. Given the permutation matrix $\mathbf{P}$ that corresponds to the order of the parts, we can compute the positional encoding for the parts as $\mathbf{p}^{\text{P}} = \mathbf{P}^T\Phi$. During training, we use the ground truth part order, whereas at inference time, we replace it with the predicted permutation matrix $\hat{\mathbf{P}}$, yielding $\hat{\mathbf{p}}^{\text{P}} = \hat{\mathbf{P}}^T\Phi$. These positional encodings are then added to the respective features before being fed into the transformer decoder and provide soft guidance on the sequence order. We adopt RoPE [42] for better performance.

Transformer Decoder. The modality interaction and fusion between step diagrams and parts occur through a stack of L transformer [46] decoder layers. In each layer, the self-attention module receives the output from the previous layer (initially, the part features $\mathbf{f}^{\text{P}} + \mathbf{p}^{\text{P}}$ in the first layer) to perform part-to-part interactions. This is followed by a cross-attention module that injects and fuses the step diagram features $\mathbf{f}^{\text{I}} + \mathbf{p}^{\text{I}}$ with the part features processed by the self-attention module, enabling step-to-part interactions. Notably, for the step diagram features, we concatenate the features from all diagrams from the same manual along the patch dimension, resulting in a tensor of shape $\mathbb{R}^{NK\times D}$.

Pose Prediction Head. Finally, the output O_{Dec} from the transformer decoder is fed into a pose prediction head, which predicts the translation and rotation for each part, denoted as $(\hat{R}_i, \hat{t}_i)$. In our implementation, the rotation matrix $\hat{R}_i$ is represented as a unit quaternion $\hat{q}_i \in \mathbb{R}^4$. We initialize its bias to an identity quaternion $(1, 0, 0, 0)$ and normalize the prediction to ensure a unit norm.

3.5. Training and Losses

First, we train the manual-guided part order learning model until it successfully aligns the two modalities and provides an optimal permutation for the parts. Subsequently, we train the transformer decoder-based part pose estimation model using the predicted order from the first model to infer the part poses. Our empirical results indicate that having an accurate part order is crucial for enhancing the inference performance of the pose estimation network.

Loss for Permutation Learning. Borrowing the notation from Secs. 3.2 and 3.3, we define the contrastive-based order loss as:

$$\mathcal{L}_{\text{order}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\text{sim}(\mathbf{f}^{\text{P}}_{\sigma(i)}, \mathbf{g}^{\text{I}}_i)/\tau)}{\sum_{j=1}^{B}\exp(\text{sim}(\mathbf{f}^{\text{P}}_{\sigma(i)}, \mathbf{g}^{\text{I}}_j)/\tau)}, \tag{4}$$

where B is the mini-batch size, τ is the temperature, and $\sigma(i)$ denotes the index for the $\sigma(i)$-th part that corresponds to the j-th step according to the ground truth part order σ. Here, mini-batches are constructed by sampling pairs

of step-diagram and part features, $(\mathbf{f}^{\text{P}}_{\sigma(i)}, \mathbf{g}^{\text{I}}_j)$ for optimizing the InfoNCE loss [12, 32]. Note that we randomly sample one part from their geometric-equivalent group. In this setup, pairs $(\mathbf{f}^{\text{P}}_{\sigma(i)}, \mathbf{g}^{\text{I}}_i)$ are considered as positive, otherwise negative. This loss encourages positive pairs to have higher similarity while negative pairs have lower similarity, allowing the similarity matrix to be used for generating the permutation matrix via Hungarian matching, as discussed in Sec. 3.3.

Losses for Pose Estimation. Given a set of furniture parts, there may be subsets of parts that are geometrically identical to each other, such as the four legs of a table, the armrests on both sides of an armchair, or the support rods on the back of a chair. We denote geometrically equivalent parts as a part group $\mathcal{G}$. Within each group, we calculate the chamfer distance between these parts and the ground truth. Then, we apply Hungarian Matching (see Sec. 3.3) again with cost:

$$\mathbf{C}_{ij} = \text{CD}(\hat{R}_i\mathcal{P}_i + \hat{t}_i, R_j\mathcal{P}_j + t_j), \tag{5}$$

where CD denotes the chamfer distance measuring the similarity between unordered two point clouds [10], i and j index the parts within $\mathcal{G}$, and $\hat{R}$ and $\hat{t}$ denote the respective predicted rotation and translation. The optimal matching sequence is denoted as $\mathcal{M}$.

For the 3D part assembly task, following Li et al. [23], we supervise translation and rotation separately. We compute the ℓ_2-distance for translation as follows:

$$\mathcal{L}_T = \frac{1}{N}\sum_{i=1}^{N}\|\hat{t}_{[i]} - t_i\|_2, \tag{6}$$

where $[i]$ denotes the index of the matched part corresponding to the i-th part under optimal matching $\mathcal{M}$.

In addition to the geometric similarity between the parts mentioned above, each part itself may also have intrinsic symmetries. For instance, a cylindrical table leg has axial symmetry. Thus, we cannot simply use the absolute distance (*e.g.*, ℓ_2) on rotation as a supervision signal. Instead we use chamfer distance to supervise the rotation per part:

$$\mathcal{L}_C = \frac{1}{N}\sum_{i=1}^{N}\text{CD}(\hat{R}_{[i]}\mathcal{P}_i, R_i\mathcal{P}_i). \tag{7}$$

However, some parts may not be perfectly symmetric, *e.g.*, due to small holes in different locations. To address these cases, we still encourage the model to penalize the ℓ_2 distance for the rotation as a regularization term, formally expressed as:

$$\mathcal{L}_E = \frac{1}{N}\sum_{i=1}^{N}\|\hat{R}_{[i]}\mathcal{P}_i - R_i\mathcal{P}_i\|_2. \tag{8}$$

Additionally, to ensure that the predicted assembled shape closely matches the ground truth, we compute the chamfer distance between the predicted and ground truth assembled shapes:

$$\mathcal{L}_S = \text{CD}\left(\bigcup_{i=1}^{N}(\hat{R}_{[i]}\mathcal{P}_i + \hat{t}_{[i]}), \bigcup_{i=1}^{N}(R_i\mathcal{P}_i + t_i)\right), \tag{9}$$

where $\cup_{i=1}^{N}$ indicates the union of N parts to form the assembled shape.

Finally, the overall loss for pose estimation is computed as a weighted sum of the aforementioned losses:

$$\mathcal{L}_{\text{pose}} = \lambda_T\mathcal{L}_T + \lambda_C\mathcal{L}_C + \lambda_E\mathcal{L}_E + \lambda_S\mathcal{L}_S. \tag{10}$$

4. Experiments

4.1. Datasets

Following [8, 9, 23, 28, 54, 56, 59, 60], we use PartNet [27], a comprehensive 3D shape dataset with fine-grained, hierarchical part segmentation, for both training and evaluation. In our work, we adopt the train/validation/test split used by Li et al. [23], where shapes with more than 20 parts are filtered out, and we use the finest granularity, *Level-3*, of PartNet. We focus on the three largest furniture categories: chair, table and storage, containing 40,148, 21,517 and 9,258 parts, respectively. Besides that, to demonstrate the generalizability of our method, we also evaluate its zero-shot capabilities on the IKEA-Manual [50] dataset, which features real-world IKEA furniture at the part level that are aligned with IKEA manuals. For consistency with PartNet, we evaluate both chair category with 351 parts, and table, with 143 parts. Storage is excluded due to only 3 samples.

For each individual part, we sample 1,000 points from its CAD model and normalize them to a canonical space. Geometrically-equivalent part groups are identified based on the size of their Axis-Aligned Bounding Box (AABB). To generate the assembly manual, we render diagrammatic images of parts using Blender's Freestyle functionality. More details of the preprocessing and rendering process are provided in the supplementary material.

4.2. Evaluation Metrics

To evaluate the effectiveness of 3D part assembly of different models, we employ three key metrics: Shape Chamfer Distance (SCD) [23], Part Accuracy (PA) [23], and Success Rate (SR) [24]. SCD quantifies the overall chamfer distance between predicted and ground truth shapes, providing a direct measure of assembly quality, scaled by a factor of 10^3 for interpretability. PA assesses the correctness of individual part poses by determining whether the chamfer distance for a part falls below a threshold of 0.01, reflecting the accuracy of each part's alignment. Finally, SR evaluates whether all parts in an assembled shape are accurately predicted, offering a strict criterion for complete assembly success.

Table 1. 3D part assembly results on the PartNet test split and Ikea-Manual. †: We re-trained the Image-PA model using diagrams (2D line drawing images) as the conditioning input instead of the original RGB images. The values in bold represent the best results, while underlined indicate the second. ‡: Average over five runs.

Method	Condition	SCD↓			PA↑			SR↑		
		Chair	Table	Storage	Chair	Table	Storage	Chair	Table	Storage
Fully-Supervised on PartNet [27]										
$DGL_{NIPS'20}$ [56]	-	9.1	5.0	-	39.00	49.51	-	-	-	-
$IET_{RA\text{-}L'22}$ [59]	-	5.4	3.5	-	62.80	61.67		-	-	-
Score-$PA_{BMVC'23}$ [8]	-	7.4	4.5	-	42.11	51.55	-	8.320	11.23	-
$CCS_{AAAI'24}$ [60]	-	7.0	-	-	53.59	-	-	-	-	-
$3DHPA_{CVPR'24}$ [9]	-	5.1	2.8	-	64.13	64.83	-	-	-	-
$RGL_{WACV'22}$ [28]	Sequence	8.7	4.8	-	49.06	54.16	-	-	-	-
$SPAFormer_{ArXiv'24}$ [54]	Sequence	6.7	3.8	4.5	55.88	64.38	56.11	16.40	33.50	**7.850**
Joint-$PA_{CVPR'24}$ [25]	Joint	6.0	7.0	-	72.80	67.40	-	-	-	-
Image-$PA_{ECCV'20}$ [23]	Image	6.7	3.7	5.0	45.40	71.60	40.20	-	-	-
Image-$PA^{\dagger}_{ECCV'20}$	Diagram	5.9	3.9	3.7	62.67	70.10	47.79	19.97	32.83	4.730
Manual-PA w/o Order	Manual	3.0	3.6	4.2	79.07	74.03	48.54	34.13	37.71	4.054
Manual-PA (Ours)	Manual	**1.7**	**1.8**	**2.9**	**89.06**	**87.41**	**56.51**	**58.03**	**56.66**	4.730
Manual-PA (Ours)	Multi-Part	3.4	3.5	3.3	84.63	78.95	53.41	38.31	40.15	5.405
Manual-PA (Ours)	Multi-View‡	3.6	3.2	3.0	82.04	75.49	54.37	38.31	39.96	4.054
Zero-Shot on IKEA-Manual [50]										
$3DHPA_{CVPR'24}$	-	34.3	37.8	-	1.914	4.027	-	0.000	0.000	-
Image-$PA^{\dagger}_{ECCV'20}$	Diagram	17.3	14.7	-	19.07	36.74	-	0.000	10.53	-
Manual-PA w/o Order	Manual	12.8	8.9	-	38.36	42.01	-	1.754	10.53	-
Manual-PA (Ours)	Manual	**11.4**	**4.8**	-	**42.51**	**49.72**	-	**3.509**	**15.79**	-

4.3. Main Results

Fully Supervised on PartNet. In this setting, we train our model on PartNet for each category and test it on unseen shapes in the test split, which share similar patterns with the training set. As shown in Tab. 1, our method, Manual-PA, significantly outperforms all previous approaches in all three categories. Specifically, Manual-PA achieves superior performance compared to all other conditioned methods. Notably, when compared to Image-PA, we observe improvements of 26, 17 and 8 part accuracy for the chair, table and storage categories, respectively. This shows that for the 3D Part Assembly task, instruction manuals provide far better guidance than a single image or other conditioning inputs. "Manual-PA w/o Order" refers to our model without access to the part assembly order information, requiring it to learn this implicitly. Although the instruction manual provides strong visual guidance, performance improvements remain limited in this case, highlighting the challenges of implicit learning. By explicitly learning the correspondences between step diagrams and parts and applying positional encoding as soft guidance, we achieve PA improvements of approximately 10, 13 and 8 in the chair, table and storage categories, respectively. Note that all other methods, except for Image-PA and ours, are generative and follow the setting proposed by [56], where multiple assembly results are generated with varying Gaussian noise for the same set of parts. Only the most faithful assembly (based on Minimum Matching Distance [1]) is evaluated.

Extension to More Realistic Assembly Settings. Our problem setting arises from the lack of a standardized format in real-world instruction manuals, which makes it difficult to define a universal learning framework. Our model is designed with scalability and generalizability in mind, allowing it to be extended to more realistic assembly scenarios. To demonstrate this, we implement and evaluate our framework under two challenging settings: 1) Multi-Part Assembly, where multiple new parts are introduced in a single step. In this scenario, parts are grouped based on their geometric similarity, and their positional encodings are adjusted accordingly to reflect their associations. 2) Multi-View , where each step diagram is captured from a randomly chosen viewpoint among eight predefined angles, simulating the viewpoint variations commonly observed in real manuals. To handle these variations, we modify the vision encoder to encode consecutive step diagrams separately before concatenating their features as input. As shown in Tab. 1, while these more complex settings introduce additional challenges and result in some performance degradation compared to the original setup, our method still significantly outperforms baseline approaches. These results highlight the robustness and versatility of our framework, demonstrating its ability to generalize beyond the

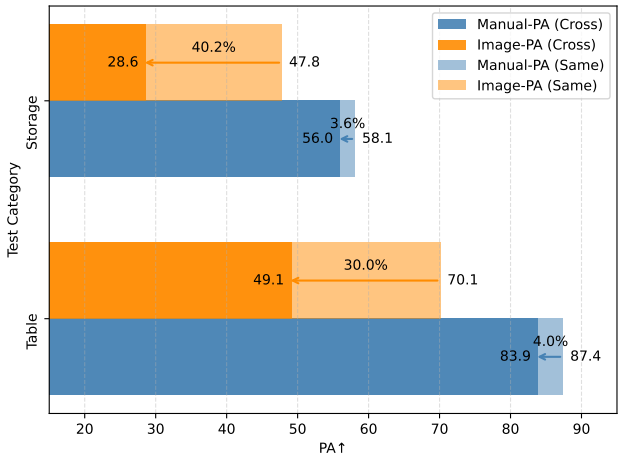

Figure 3. Results trained on chair and tested table and storage. Solid bars: cross-category; lighter bars: same-category. Arrows show performance drop with percentages above.

single-part-per-step assumption and handle more practical assembly tasks. Further details on these experiments are provided in the supplementary material.

Cross-Category Generalization. We evaluate cross-category generalization by training both Manual-PA and Image-PA on the chair category and testing on chair, table, and storage. As shown in Fig. 3, the performance drop of Manual-PA when tested on unseen categories is minimal compared to that of Image-PA, indicating stronger cross-category generalization. This result suggests that Manual-PA does not solely rely on shape priors but effectively extracts assembly-relevant information from manuals, enabling strong performance even on unseen shapes. These findings reinforce the advantage of using stepwise instruction diagrams as guidance for 3D part assembly.

Zero-shot on IKEA-Manual. We use the same model, pre-trained on PartNet, to directly estimate the poses of furniture in IKEA-Manual, where all shapes are 3D models of real-world IKEA furniture, and each part corresponds to illustrations in their official assembly manuals. As shown in Tab. 1, our method, significantly outperforms strong competitors such as 3DHPA and Image-PA, demonstrating the strong generalizability of our approach to real world.

4.4. Ablation Studies

Effect of the Order for 3D Part Assembly. We conduct experiments to explore various methods for incorporating order information in a 3D part assembly model, as shown in Tab. 2. First, by comparing lines 1 and 2, we observe that adding sequential information solely to part features improves assembly performance. We attribute this improvement to two factors: (1) distinguishing geometrically equivalent components, and (2) mitigating the combinatorial explosion problem, as claimed in SPAFormer [54]. Second, by comparing lines 2 and 5, where the latter provides both the step diagram and part features with correct correspondence, we achieve a significant performance boost. This indicates that explicitly establishing correspondence between steps and parts is beneficial, as learning these correspondences implicitly remains challenging for the model. Last, our empirical results show that, when position information is solely relevant to self-attention (lines 2 and 3), both PE and OneHot encodings yield similar performance. This observation aligns with prior transformer-based studies, which also utilize OneHot encodings [9, 54, 59]. However, in cases requiring cross-attention (line 4 and 5), PE outperforms OneHot encodings, highlighting its advantages in cross-attentive contexts.

Table 2. Ablation results for various order designs on the PartNet chair test split. We use GT order for both training and inference.

#	Type	Step	Parts	SCD↓	PA↑	SR↑
1	PE			4.3	72.22	22.50
2	PE		✓	3.0	79.31	33.88
3	OneHot		✓	3.0	79.07	34.13
4	OneHot	✓	✓	2.7	81.22	36.92
5	PE (Ours)	✓	✓	1.7	95.38	73.07

Table 3. Ablation results for various component designs in permutation learning on the PartNet chair test split. KT denotes the Kendall-tau coefficient that measure the ordinal correlation between predicted and GT order.

Variants	KT↑	SCD↓	PA↑	SR↑
Manual-PA (Ours)	0.789	1.7	89.06	58.03
w/o batch-level sampling	0.410	2.3	56.93	15.68
w/ Sinkhorn	0.774	1.7	88.25	56.26
w/ GT Order	1.000	1.7	95.38	73.07

Analysis of Permutation Learning Choices. We conduct an ablation study for permutation learning, as shown in Tab. 3. First, we find that batch-level sampling is essential compared to applying contrastive learning individually within each manual book. This is primarily because contrastive learning benefits from a larger pool of negative samples. Second, while we employ the Sinkhorn [41] algorithm on the similarity matrix to enforce that rows and columns sum to one, empirical results indicate limited performance improvement. Last, by providing the model with the ground truth assembly order, we establish an upper performance bound for our assembly model.

4.5. Visualization and Analysis

Demonstration of Soft Guidance. We manually add different positional encodings to the feature of chair's back, as shown in Fig. 5. With PE(5), the model focuses on the final position, aligning well with the correct step diagram.

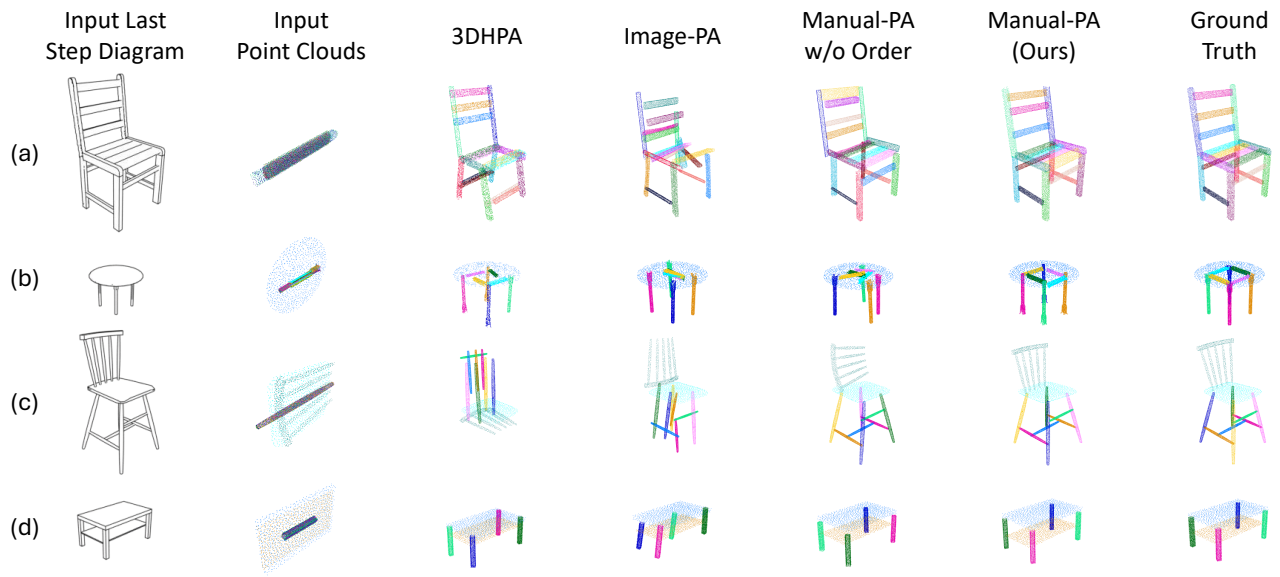

Figure 4. Qualitative comparison of various 3D part assembly methods. Four examples are shown: chair (a) and table (b) from the PartNet dataset, and chair (c) and table (d) from the IKEA-Manual dataset.

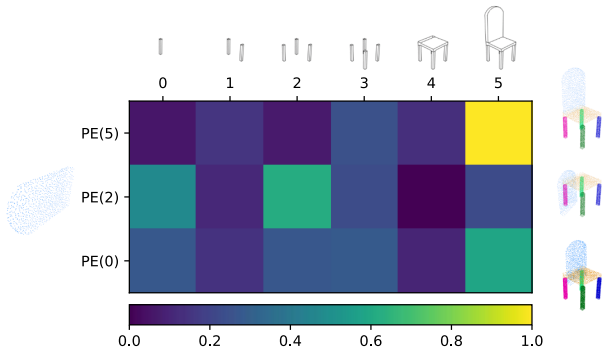

Figure 5. Visualization of the cross attention scores between step diagrams (top) and a part (left, chair's back) with PE at different positions. The final assembly results are displayed on the right.

However, when PE(2) is applied, drawing the model's attention to the second step diagram, the model struggles to accurately predict the chair's back pose. Notably, even with PE(0), the model manages to shift its attention toward the correct final step, demonstrating the fault tolerance and robustness of soft guidance: the model can still identify the correct match even when given an incorrect sequence.

Qualitative Comparison. As shown in Fig. 4, our proposed Manual-PA produces the most faithful part pose prediction for the shapes illustrated in the step diagrams. For "Manual-PA w/o Order", some parts do not align well with the assembly instructions, indicating the importance of providing precise correspondences between the step diagram and the parts for accurate assembly. In the case of Image-PA, the input consists only of the final image, making it challenging to locate the poses of parts that are occluded, such as the supporting rods beneath the table top in (b). 3DHPA lacks visual input and operates in a generative mode, relying solely on shape priors. Consequently, it generates shapes that are reasonable, as seen in (a), but there are noticeable discrepancies between its output and the ground truth shapes. For zero-shot results on the IKEA-Manual dataset, both 3DHPA and Image-PA exhibit degraded performance, while Manual-PA still produces faithful assembly results, demonstrating its superior generalization ability.

5. Conclusions and Future Work

We introduce a novel 3D part assembly framework that leverages diagrammatic manuals for guidance. Our proposed approach, Manual-PA, learns the sequential order of part assembly and uses it as a form of soft guidance for 3D part assembly. Manual-PA achieves superior performance on the PartNet dataset compared to existing methods and demonstrates strong generalizability to real-world IKEA furniture, as validated on the IKEA-Manual dataset.

Future work could aim to bridge the gap between synthetic and real-world IKEA manuals. Although we have addressed adaptability in detecting a varying number of new parts introduced in each step and robust handling of differing perspectives, other visual elements such as arrows, detailed close-ups, and part labels remain challenging. Moreover, manuals often include sub-module assemblies, resulting in a tree-like, non-linear assembly structure.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, pages 40–49. PMLR, 2018. 6

[2] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *CVPR*, pages 847–859, 2021. 1

[3] Yizhak Ben-Shabat, Jonathan Paul, Eviatar Segev, Oren Shrout, and Stephen Gould. Ikea ego 3d dataset: Understanding furniture assembly actions from ego-view 3d point clouds. In *WACV*, pages 4355–4364, 2024. 1

[4] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, pages 2229–2238, 2019. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 4

[6] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *SIGGRAPH*, pages 1–10, 2011. 2

[7] Jianqiu Chen, Mingshan Sun, Tianpeng Bao, Rui Zhao, Liwei Wu, and Zhenyu He. Zeropose: Cad-model-based zero-shot pose estimation. *arXiv preprint arXiv:2305.17934*, 2023. 2

[8] Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. Score-pa: Score-based 3d part assembly. *BMVC*, 2023. 1, 2, 5, 6

[9] Bi'an Du, Xiang Gao, Wei Hu, and Renjie Liao. Generative 3d part assembly via part-whole-hierarchy message passing. In *CVPR*, pages 20850–20859, 2024. 1, 2, 5, 6, 7

[10] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, 2017. 5

[11] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM TOG*, 23(3):652–663, 2004. 2

[12] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742. IEEE, 2006. 5

[13] Junwen Huang, Hao Yu, Kuan-Ting Yu, Nassir Navab, Slobodan Ilic, and Benjamin Busam. Matchu: Matching unseen objects for 6d pose estimation from rgb-d images. In *CVPR*, pages 10095–10105, 2024. 2

[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pages 4904–4916. PMLR, 2021. 2

[15] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM TOG*, 31(4):1–11, 2012. 2

[16] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *WACV*, pages 793–802. IEEE, 2018. 2

[17] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, pages 5583–5594. PMLR, 2021. 2

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 2

[19] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4

[20] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *CoRL*, 2022. 2

[21] Youngwoon Lee, Edward S Hu, and Joseph J Lim. Ikea furniture assembly environment for long-horizon complex manipulation tasks. In *ICRA*, pages 6343–6349. IEEE, 2021. 1

[22] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *NeurIPS*, 34:9694–9705, 2021. 2

[23] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3d part assembly from a single image. In *ECCV*, pages 664–682. Springer, 2020. 1, 2, 3, 5, 6

[24] Yulong Li, Andy Zeng, and Shuran Song. Rearrangement planning for general part assembly. In *7th Annual Conference on Robot Learning*, 2023. 2, 5

[25] Yichen Li, Kaichun Mo, Yueqi Duan, He Wang, Jiequan Zhang, and Lin Shao. Category-level multi-part multi-joint 3d shape assembly. In *CVPR*, pages 3281–3291, 2024. 2, 6

[26] Jiehong Lin, Lihua Liu, Dekun Lu, and Kui Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *CVPR*, pages 27906–27916, 2024. 2

[27] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, pages 909–918, 2019. 2, 5, 6

[28] Abhinav Narayan, Rajendra Nagar, and Shanmuganathan Raman. Rgl-net: A recurrent graph learning framework for progressive part assembly. In *WACV*, pages 78–87, 2022. 2, 5, 6

[29] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *CVPR*, pages 2134–2140, 2023. 2

[30] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *CVPR*, pages 9903–9913, 2024. 2

[31] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016. 2

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4

[34] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomáš Hodaň. Foundpose: Unseen object pose estimation with foundation features. *ECCV*, 2024. 2

[35] Kaiyue Pang, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In *CVPR*, pages 10347–10355, 2020. 2

[36] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *CVPR*, pages 4561–4570, 2019. 2

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 3

[38] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, pages 3828–3836, 2017. 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2

[40] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Deeppermnet: Visual permutation learning. In *CVPR*, pages 3949–3957, 2017. 2

[41] Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967. 7

[42] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 4

[43] Minhyuk Sung, Hao Su, Vladimir G Kim, Siddhartha Chaudhuri, and Leonidas Guibas. Complementme: Weakly-supervised component suggestions for 3d modeling. *ACM TOG*, 36(6):1–12, 2017. 2

[44] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *ECCV*, pages 530–546. Springer, 2020. 2

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

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[47] Yuxuan Wan, Kaichen Zhou, Hao Dong, et al. Scanet: Correcting lego assembly errors with self-correct assembly network. *arXiv preprint arXiv:2403.18195*, 2024. 1

[48] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, pages 2642–2651, 2019. 2

[49] Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Chin-Yi Cheng, and Jiajun Wu. Translating a visual lego manual to a machine-executable plan. In *ECCV*, pages 677–694. Springer, 2022. 1, 2

[50] Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Ran Zhang, Chin-Yi Cheng, and Jiajun Wu. Ikea-manual: Seeing shape assembly step by step. *NeurIPS*, 35:28428–28440, 2022. 2, 5, 6

[51] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, pages 17868–17879, 2024. 2

[52] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *CVPR*, pages 829–838, 2020. 2

[53] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2017. 2

[54] Boshen Xu, Sipeng Zheng, and Qin Jin. Spaformer: Sequential 3d part assembly with transformers. *arXiv preprint arXiv:2403.05874*, 2024. 1, 2, 5, 6, 7

[55] Kangxue Yin, Zhiqin Chen, Siddhartha Chaudhuri, Matthew Fisher, Vladimir G Kim, and Hao Zhang. Coalesce: Component assembly by learning to synthesize connections. In *2020 International Conference on 3D Vision (3DV)*, pages 61–70. IEEE, 2020. 2

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

[57] Jiahao Zhang, Anoop Cherian, Yanbin Liu, Yizhak Ben-Shabat, Cristian Rodriguez, and Stephen Gould. Aligning step-by-step instructional diagrams to video demonstrations. In *CVPR*, pages 2483–2492, 2023. 1

[58] Jiahao Zhang, Frederic Z Zhang, Cristian Rodriguez, Yizhak Ben-Shabat, Anoop Cherian, and Stephen Gould. Temporally grounding instructional diagrams in unconstrained videos. *WACV*, 2025. 1

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

[60] Ruiyuan Zhang, Jiaxiang Liu, Zexi Li, Hao Dong, Jie Fu, and Chao Wu. Scalable geometric fracture assembly via co-creation space among assemblers. In *AAAI*, pages 7269–7277, 2024. 1, 2, 5, 6

[61] Xinghao Zhu, Devesh K Jha, Diego Romeres, Lingfeng Sun, Masayoshi Tomizuka, and Anoop Cherian. Multi-level reasoning for robotic assembly: From sequence inference to contact selection. In *ICRA*, pages 816–823. IEEE, 2024. 1