2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# Manual2Skill++: Connector-Aware General Robotic Assembly from Instruction Manuals via Vision–Language Models

Chenrui Tie*[1] Shengxiang Sun*[2] Yudi Lin[1] Yanbo Wang[3] Zhongrui Li[1] Zhouhan Zhong[1]
Jinxuan Zhu[1] Yiman Pang[1] Haonan Chen[1] Junting Chen[1] Ruihai Wu[4] Lin Shao†[1]

***Abstract*— Assembly hinges on reliably forming connections between parts; yet most robotic approaches plan assembly sequences and part poses while treating connectors as an afterthought. Connections represent the foundational physical constraints of assembly execution; while task planning sequences operations, the precise establishment of these constraints ultimately determines assembly success. In this paper, we treat connections as explicit, primary entities in assembly representation, directly encoding connector types, specifications, and locations for every assembly step. Drawing inspiration from how humans learn assembly tasks through step-by-step instruction manuals, we present Manual2Skill++, a vision-language framework that automatically extracts structured connection information from assembly manuals. We encode assembly tasks as hierarchical graphs where nodes represent parts and subassemblies, and edges explicitly model connection relationships between components. A large-scale vision-language model parses symbolic diagrams and annotations in manuals to instantiate these graphs, leveraging the rich connection knowledge embedded in human-designed instructions. We curate a dataset containing over 20 assembly tasks with diverse connector types to validate our representation extraction approach, and evaluate the complete task understanding-to-execution pipeline across four complex assembly scenarios in simulation, spanning furniture, toys, and manufacturing components with real-world correspondence. More detailed information can be found at https://nus-lins-lab.github.io/Manual2SkillPP/**

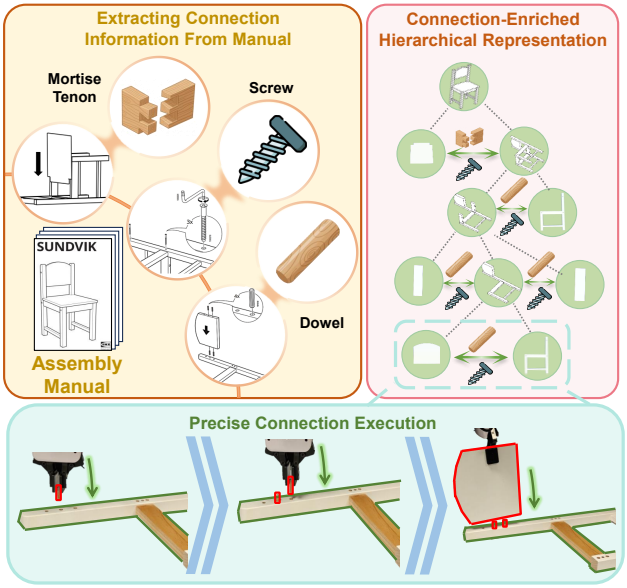


Fig. 1: **Manual2Skill++.** We extract connection information from assembly manuals, build connection-enriched hierarchical graphs encoding part relationships and connector details, then execute precise robotic assembly guided by these connection relations.

## I. Introduction

Assembly fundamentally concerns the reliable connection of individual components into a coherent whole. Over the past decades, significant progress has been made in robotic execution and spatial reasoning, ranging from learning precise insertion policies [1] to optimizing part-level motions and poses [2]–[5]. However, these methods predominantly focus on the geometry and dynamics of assembly, often treating the underlying connection relationships as predefined or secondary concerns. In practical scenarios, such as furniture construction or industrial manufacturing, the success of a task depends on more than just spatial alignment; it requires the informed selection and application of specific connectors (*e.g.*, adhesives, mortise-tenon joints, or screws). There remains a critical gap in interpreting and utilizing structured connection knowledge: identifying the exact type, quantity, and placement of connectors needed for structural integrity. This gap severely limits the applicability of existing methods to practical assembly scenarios, where connector selection and usage are indispensable.

Incorporating connector selection and usage into an assembly framework presents significant challenges that compound the inherent complexity of robotic assembly. First, connectors vary widely in form and function; mortise-tenon joints require precise peg insertion, while screws demand controlled rotation and torque, and a single task may involve multiple connection types simultaneously (As shown in Figure 1, the assembly of a simple chair involves three types of connection). Second, each connection adds a decision sequence: selecting the connector, locating its placement, and executing the specific insertion or fastening motion. Third, assembly is inherently long-horizon, involving many parts and sub-assemblies, and explicit connector relationships create complex dependencies across the process [6]. These challenges highlight the need for a structured representation that systematically organizes parts, sub-assemblies, and connectors for comprehensive task understanding and planning.

* Equal contribution.
† Corresponding author: Lin Shao (linshao@nus.edu.sg).
[1] School of Computing, National University of Singapore, Singapore.
[2] University of Toronto, Toronto, Canada.
[3] Zhejiang University, Zhejiang, China.
[4] Peking University, Beijing, China.

In everyday life, humans most naturally learn assembly procedures by consulting instruction manuals, whether for IKEA furniture, LEGO models, or hobbyist models. These manuals encode information of connector selection and placement locations, providing connection knowledge that humans intuitively interpret. By parsing sketched illustrations and symbolic annotations, a person can identify not only which parts to assemble, but precisely how they connect through specific connector types and operations. Prior work has investigated extraction of task structure and part relationships from manuals [7], but has largely overlooked this connector-level information, leaving a critical gap in automated assembly understanding. Extracting structured connection information from manuals requires overcoming the ambiguity of sketches and symbolic annotations. To address this, we propose a hierarchical graph representation for assembly tasks. In this graph, leaf and intermediate nodes denote parts and sub-assemblies, while edges explicitly encode connector types, quantities, and spatial constraints. This abstraction bridges high-level task understanding and low-level robotic execution.

Building on this representation, we introduce Manual2Skill++, a general framework that leverages vision-language models to parse instruction manuals into these hierarchical graphs. By identifying connector correspondences and placement features in a single pipeline, Manual2Skill++ enables the direct computation of relative part poses via geometric constraint optimization, achieving significantly higher accuracy than prior learning-based pose estimation methods [4], [7], thereby providing the precision necessary for robust robotic execution in real-world assembly tasks.

To validate our approach, we curate a dataset of 20+ complex tasks (IKEA furniture, toys, and industrial parts) featuring precise 3D models with annotated connector attachment sites and ground-truth assembly sequences. Furthermore, we develop a simulation benchmark spanning four long-horizon scenarios (As shown in Figure 3). Unlike previous works using simplified object models [8], [9], our benchmark incorporates full-scale assets with realistic connection modalities, such as mortise-tenon joints, dowels, and screws, providing a robust testbed for task planning and contact-rich control under realistic assembly constraints with explicit connector complexity.

In summary, we make the following contributions:

- We propose a graph representation that explicitly models connection relations between assembly components across all assembly tasks, effectively capturing the complexity of assembly tasks while enabling direct computation of part poses through connection constraints.
- We introduce Manual2Skill++, which extracts structured assembly representations from instruction manuals, supported by a curated dataset of over 20 diverse assembly tasks with fine-grained assets and annotations.
- We develop simulation environments featuring long-horizon assembly tasks with diverse connector types. Our benchmark introduces connectors and supports multiple connection operations.

## II. RELATED WORK

### A. Part Assembly

Part assembly represents a fundamental challenge in robotics, with extensive research exploring how to construct complete objects from individual components [2]–[5], [10]–[12]. Assembly tasks span diverse domains including furniture construction [11], toy building [9] and industrial manufacturing [10], each presenting unique challenges in terms of part complexity, connection mechanisms, and assembly sequences. Broadly, we categorize part assembly approaches into *geometric assembly* and *semantic assembly*. *Geometric assembly* relies primarily on geometric cues such as surface compatibility, edge features, or shape complementarity to determine part relationships [12]–[14]. These methods excel at puzzle-like tasks where parts fit together based purely on geometric constraints. *Semantic assembly* leverages high-level semantic understanding of parts and their functional relationships to guide the assembly process [2], [4], [10], [11]. This approach is particularly effective for structured objects like furniture, where parts have predefined roles and follow intuitive assembly logic.

Previous research has addressed various aspects of assembly, including motion planning [15], multi-robot coordination [16], pose estimation [4], [17], [18], and sequence planning [2], [3]. Several datasets and simulation environments have been developed to facilitate research: IKEA furniture datasets [6] provide 3D models and structured procedures; simulation environments [8], [11] enable reproducible evaluation; and specialized benchmarks [9] offer standardized evaluation protocols. However, existing approaches predominantly focus on individual subproblems, such as pose estimation or motion planning, while overlooking the critical role of physical connections between parts. Most methods assume simplified part relationships [9] or rely on oracle mechanics [11], neglecting the diverse connection types that determine assembly success in practice. This limits applicability to real-world scenarios where connection selection and execution are paramount.

### B. VLM-Guided Robot Learning

Vision Language Models (VLMs) [19] have been employed in robotics for environment understanding [20], human–robot interaction [21], and high-level task planning [22]. While end-to-end Vision Language Action models can directly generate robot actions from multimodal inputs [23]–[25], they demand large datasets and often struggle with long-horizon, precise manipulation. An alternative is to use VLMs as reasoning engines, providing structured guidance for task decomposition [26], scene interpretation [27], and control interfaces [28], [29]. Additionally, VLMs have been applied to robot learning tasks such as instruction parsing [30] and assembly design assistance [31]. Building on this paradigm, we introduce Manual2Skill++, a novel application of VLMs to parse symbolic assembly manuals and extract detailed connection relationships, including types, quantities, and placement constraints, enabling precise, connection-aware robotic assembly operations.

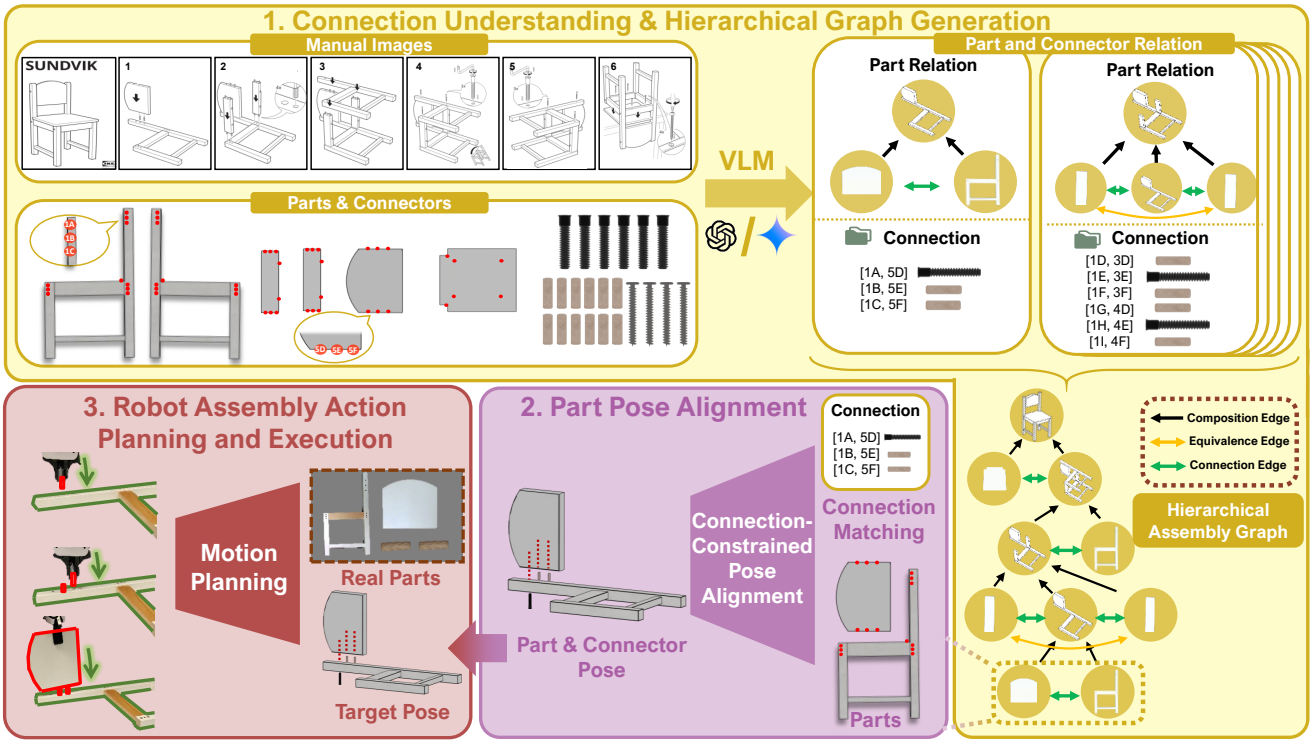


Fig. 2: **Framework Overview.** (1) A VLM processes manual images to extract part and connector relations, generating a connection-enriched hierarchical assembly representation. (2) The extracted connection constraints guide a geometric optimization process to compute precise target poses for parts and connectors. (3) The system executes the assembly by planning and performing robotic actions based on the connection-enriched hierarchical graph and aligned poses.

## III. Problem Formulation

Given the 3D models of all involved parts and connectors in an assembly, and its assembly manual, our goal is to generate a physically feasible sequence of robotic assembly actions to aggregate all parts and connectors into a whole object. We define the manual pages as a set of $N$ images. $\mathcal{I} = \{I_1, I_2, \cdots, I_N\}$, where each image $I_i$ illustrates a specific step in the assembly process, such as the merging of certain parts or sub-assemblies using certain connectors.

The assembly consists of $M$ individual parts $\mathcal{P} = \{P_1, P_2, \cdots, P_M\}$ and $Q$ connectors $\mathcal{C} = \{C_1, C_2, \cdots, C_Q\}$. A *subassembly* is any partially or fully assembled structure that forms a proper subset of $\mathcal{P} \oplus \mathcal{C}$ (*e.g.*, $\{P_1, P_2, C_1\}$ denotes $P_1, P_2$ are connected via connector $C_1$). A *component* may refer to a part or a subassembly. We define an *attachment point* as the geometric location on a part designated for connector placement, such as protrusions, pin holes, or screw holes (see red points in Figure 2 (1)). Here, we assume access to precise 3D models of all parts, allowing us to know the exact positions of every attachment point. In practice, these high-fidelity models can be obtained via 3D scanning of physical parts or retrieved from manufacturer CAD files [32], making this assumption realistic in real-world assembly settings. Then all the connections can be formulated as binary pairs of attachment points and corresponding connectors.

## IV. Method

Our methodology centers on connection modeling as the key to comprehensive assembly understanding. We introduce a hierarchical graph representation that explicitly encodes component connections as a structured task plan (Section IV-A), and we propose Manual2Skill++, a VLM framework that automatically constructs these hierarchical graphs from instruction manuals (Section IV-B). Leveraging the extracted connection constraints, we directly compute relative part poses via optimization, eliminating the need for separate multi-part pose estimation module (Section IV-C). This task plan then guides downstream policies for precise insertion, screwing, or snapping operations. To validate the end-to-end pipeline, we develop an Isaac Lab benchmark featuring four complex assembly tasks with diverse connector modalities, demonstrating the necessity and effectiveness of our connector-aware representation (Section IV-D).

### A. General Representation for Assembly Tasks

Our assembly representation employs a hierarchical graph structure $\mathcal{G}$ where leaf nodes correspond to atomic parts, intermediate nodes represent sub-assemblies at various stages, and the root node represents the final assembled product. The graph organizes assembly information across multiple abstraction levels, with each level capturing the granularity appropriate for different reasoning tasks.

We define three types of edges that capture distinct assembly relationships. Composition edges connect parent-child node pairs, indicating that a parent subassembly is formed by aggregating its child components. Equivalence edges link nodes representing identical parts ($e.g.$, the four legs of a table), signifying that these components are interchangeable during assembly. Most importantly, connection edges between sibling nodes represent physical connections that must be established between components through specific connectors, this constitutes the key innovation of our representation. These connection edges carry attributes that encode how the connected nodes are physically joined. Formally, we have $\mathcal{G} = \{\mathcal{V}, \mathcal{E}_{\text{comp}}, \mathcal{E}_{\text{eqv}}, \mathcal{E}_{\text{conn}}\}$, where $\mathcal{V}$ denotes the node set, $\mathcal{E}_{\text{comp}}, \mathcal{E}_{\text{eqv}}, \mathcal{E}_{\text{conn}}$ denote three kinds of edge sets. Each connection edge $e_{\text{conn}} \in \mathcal{E}_{\text{conn}}$ comprises one or more connection instances, since real assemblies often use multiple connectors ($e.g.$, several screws) between the same parts. Each connection instance corresponds to an attachment feature pair joined by a connector. We define the feature of each connection instance as follow:

TABLE I: Features of Connection Instance

| **Feature Name** | **Description** |
|---|---|
| Connection Type | Type of connection, such as mortise-tenon joint, dowel, or screw. |
| Connector | Connector identifier (empty for mortise-tenon joints). |
| Attachment Feature on Part 1 | Position and normal vector of an attachment point on Part 1's local frame. |
| Attachment Feature on Part 2 | Position and normal vector of an attachment point on Part 2's local frame. |

An example of the three kinds of edges can be found at Figure 2(1), where the parent-child nodes are linked by composition edge (denoted by black arrow), physically jointed nodes are linked by connection edge (denoted by green arrow), and equivalent nodes are linked by equivalence edge ($e.g$, two identical rods are connected by orange arrow).

### *B. VLM-Guided Hierarchical Graph Generation*

To initialize the hierarchical graph's nodes, composition edges, and equivalence edges, we employ Manual2Skill's VLM generation stage [7], which uses the manuals to identify parts needed at every assembly step. Manual2Skill++ then introduces the novel framework of generating the connection edges. Since we assume access to the 3D model of each part, constructing the connection edges reduces to identifying the correct pairs of attachment points for each connector. This is challenging as it requires correlating low-detail, abstract manual sketches with highly detailed real-world assembly scenes. Large parts like panels are straightforward to match, but connectors and attachment points are much smaller and often drawn with minimal detail. Furthermore, each step typically involves multiple connectors, which must be matched with their corresponding attachment point pairs simultaneously. Determining these pairings is a complex combinatorial problem: from a large set of candidate attachment points, the correct subset needs to be selected and organized into binary pairs, all while handling visual clutter and occlusions in the unstructured manual images.

We address these challenges by leveraging the reasoning VLM [33]. In our pipeline, the VLM iteratively processes each assembly step. For each iteration, the VLM receives two types of visual inputs. The first type is a manual image that encodes the ground-truth attachment point pair for every connector through abstract sketches. The second type is a set of high-fidelity 2D images of the involved components. Rendered from 3D models via Blender, these images present all candidate attachment points and connectors. The visual inputs are combined with a text input providing domain-specific knowledge about the connector types. From this combined input, the VLM outputs the precise pairs of attachment points to be linked by each connector.

For example, consider the first assembly step in Figure 2(1). The manual input is the first page of the assembly instructions (denoted with step number 1 on the top left). The two components involved are the h-shaped side frame (10 red dots representing attachment points) and the curved backrest panel (6 red dots). All attachment points are uniquely indexed, though only some are shown in Figure 2 for readability. These component inputs show 16 candidate attachment points in total. Given this information, the VLM first infers from the manual that the step requires two wooden dowels. It then reasons over the candidate points and predicts the exact pairs for each dowel: [1B, 5E] and [1C, 5F]. The VLM repeats this for all subsequent steps to predict attachment point pairings for the other required connectors. These predicted pairings collectively form the connection edges of the hierarchical graph.

We implement this input-output process via a two-stage prompting strategy. In Stage 1, the VLM uses the manual input to estimate the number and type of connectors that are placed on each component. For the chair's first step, the output is {"h-shaped side frame": [2, "dowel"], "curved backrest panel": [2, "dowel"]}, indicating that two attachment points on the side frame will be paired with two attachment points on the backrest using 2 dowels. Obtaining this high-level information restricts the subsequent search of a connector's possible attachment points to a specific component rather than all components in the assembly, drastically reducing the combinatorial complexity of matching attachment points into binary pairs. In Stage 2, the VLM identifies precise attachment point pairs for each connector using the output of Stage 1, along with the same manual and component images as input. For example, in the chair's first assembly step, after determining that there should be two dowels connecting the two components, Stage 2 outputs the two exact pairs, [1B, 5E] and [1C, 5F]. Repeating both stages across all assembly steps yields the complete hierarchical graph.

By extracting the connector information from manuals and formulating the assembly process into a structured hierarchical graph, Manual2Skill++ achieves general task understanding for a spectrum of assembly tasks and empowers downstream assembly execution.

### C. Part Pose Alignment via Connection Matching

In our framework, we address part pose estimation through direct connection matching. Given that precise attachment points are known and their pairwise correspondences are extracted from assembly manuals, we can directly compute the relative poses between connected components through geometric constraint satisfaction. This approach significantly improves pose alignment accuracy, achieving the precision necessary for reliable assembly operations. Formally, for each connection edge $e_{\text{conn}}$, the attachment features are denoted as $[(x_1^a, n_1^a), (x_1^b, n_1^b)], ..., [(x_k^a, n_k^a), (x_k^b, n_k^b)]$ , where $k$ is the number of connection instances ($e.g.$, if two parts are connected by 3 screws, $k = 3$) between part $a$ and part $b$, $x \in \mathbb{R}^3$ is the position of attachment, $n \in \mathbb{R}^3$ is the normal unit vector of attachment. We formulate the pose alignment as the following optimization problem:

$$\min_{R,t} \quad \sum_{i=1}^{k} \left( |Rx_i^a + t - x_i^b|^2 + \alpha |Rn_i^a + n_i^b|^2 \right) \tag{1}$$

where $R \in SO(3)$ is the rotation matrix and $t \in \mathbb{R}^3$ is the translation vector that transforms part $a$ to align with part $b$. The first term ensures attachment positions coincide, while the second term enforces that attachment normals are collinear and opposite ($i.e.$, $Rn_i^a = -n_i^b$). The weight $\alpha$ balances position and orientation alignment. This constraint-based approach achieves millimeter-level accuracy required for reliable connection operations.

### D. Robotic Assembly Benchmark in Isaac Lab

To address the importance of explicit connector-modeling, we present four complex, long-horizon assembly tasks in Isaac Lab, featuring full-scale IKEA furniture and authentic toy models with diverse connection mechanics.

*1) Task Selection and Complexity:* As illustrated in Figure 3, our benchmark encompasses the complete assembly of an IKEA chair, shoe shelf, airplane model, and LEGO figure. These tasks span varying complexity levels with different numbers of parts, assembly steps, and connector types, as summarized in Table II. Here the *step* is defined as the number of connection operations, as the joint of two parts may involve multiple connection operations.

TABLE II: Summary of Four Assembly Tasks

| | | | Connector Types | | |
|---|---|---|---|---|---|
| **Task Name** | **Parts** | **Steps** | **Mortise-tenon** | **Dowels** | **Screws** |
| Shoe Shelf | 4 | 11 | ✓ | × | ✓ |
| Chair | 6 | 22 | ✓ | ✓ | ✓ |
| LEGO Person | 9 | 8 | ✓ | × | × |
| Plane Model | 11 | 12 | ✓ | ✓ | × |

*2) Connection Mechanics Implementation:* We design simulation mechanics for three primary connector types. **Mortise-tenon joints** are realized through proximity-based fixed joint creation when relative pose alignment falls within specified thresholds. **Dowels** similarly trigger fixed joint formation upon correct part-to-dowel pose alignment. Similar to mortise-tenon and dowel connections, **screws** require precise pose alignment to initiate a connection. However, instead of forming a fixed joint immediately, we employ a multi-stage D6 joint mechanism: initial proximity creates a constrained joint that permits only rotation along its central axis. Incremental translation along this same axis is then unlocked as rotational thresholds are exceeded, realistically simulating screw tightening without complex friction modeling. This connector-aware simulation provides a faithful testbed for evaluating assembly planning and control under realistic connection constraints.

## V. EXPERIMENT

In this section, we perform a series of experiments aimed at addressing the following questions.

- Q1: Can our hierarchical graph representation generalize across diverse assembly domains and complexity levels? (Section V-A)
- Q2: How effectively can Manual2Skill++ extract the representation from manuals? (Section V-A)
- Q3: Can our connection-matching pose alignment achieve the precision required for practical assembly tasks? (Section V-B)
- Q4: Are our graph representation and pose alignment sufficient to guide downstream connection execution? (Section V-C)

### A. Hierarchical Graph Generation

*1) Dataset:* To validate the universality of our assembly representation and generalization ability of Manual2Skill++, we curate a dataset covering 21 diverse assembly tasks. It includes 11 IKEA furniture items sampled from the IKEA-Manuals dataset [6], and 10 toy models, LEGO sets, and manufacturing components sourced from the Internet. For furniture items, we use the original IKEA manual, while for other assembly tasks, we generate instruction manuals using Blender's Freestyle functionality.

For each task, we manually annotate the 3D positions of all attachment points on every part. To prepare VLM inputs, we projected the 3D models of components and their annotated attachment points into 2D renderings for each assembly step, and we assign unique IDs to each attachment point. The final ground truth for each step consists of the correct pairings of these attachment points for every connector.

Our dataset features a wide range of assembly tasks requiring complex reasoning. On average, a task involves assembling 6 parts over 4 steps. For each step, the model must correctly select and pair around 6 attachment points from a set of 12 possible choices. The dataset also includes high-complexity instances, such as tasks with up to 11 parts, 8 connection steps, or the need to select and pair 12 points from a set of 84 choices. The fact that every task can be fully described by our connection-enriched hierarchical representation demonstrates its broad applicability and expressiveness. Figure 3 shows 6 examples from our collection.

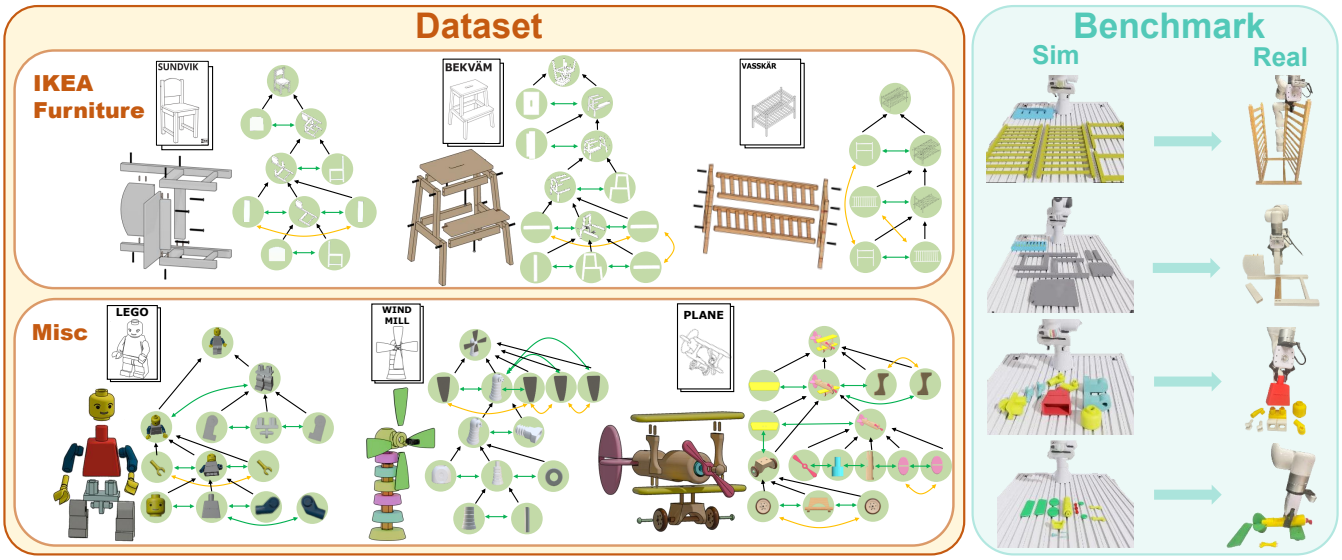


Fig. 3: **Overview of Our Dataset and Benchmark.** The dataset comprises 21 representative assembly tasks, each with high-fidelity 3D part models, manual pages for every step, and fully annotated connection information. From this collection, we selected four tasks and implemented them in Isaac Lab, designing connection mechanics to enable long-horizon assembly with explicit connector operations that directly correspond to real-world procedures.

*2) Evaluation Metric:* We evaluate the VLM's ability to understand assembly manuals by the accuracy of its predicted attachment point pairs from Stage 2, which are critical for downstream pose estimation and connection execution tasks. Predictions are compared against ground-truth annotations for each assembly step under two complementary settings:

- Set Matching (Points Selected): We treat the task as identifying the correct *set of attachment points* that receives connectors. A prediction is correct if it identifies the right points, regardless of how they are paired.
- Pair Matching (Connections Made): This setting evaluates the task as identifying the correct *unordered pairs of attachment points* that form individual connections. This is a stricter measure of whether the exact connections are inferred.

For each setting, We report the *success rate*, where a single assembly step is considered successful if the predicted set and pairs perfectly match their ground-truth counterparts. We also calculate standard *F1-score* to quantify the overlap between predicted and ground-truth sets or pairs. An attachment point from the predicted set or an attachment point pair from the predicted pairs is a true positive if it matches a ground-truth entry otherwise it is a false positive; any ground-truth entry not predicted is a false negative. Parts with equivalent edge are considered as identical. Final performance is reported as the F1-score and success rate averaged across all 85 assembly steps in the dataset.

*3) Baseline:* We compare our VLM-based method against two heuristic approaches, and two ablation settings.

- Random Sampling: Randomly generates connection pairs from available choices. This evaluates the proportion of trivial assembly steps in our dataset.
- Geometric Matching: An OpenCV-based heuristic that identifies attachment points in manuals by referencing manually chosen template images and thresholds. It can only perform Set Matching and completely fails at Pair Matching due to a lack of relational reasoning.
- Incomplete Manual: To evaluate how the performance of our method is affected by incomplete or skipped manuals, we randomly replace 1 assembly step with a blank manual image and let VLM infer assembly representation from its prior knowledge and past context.
- GPT: We compare the performance of reasoning VLM (Gemini-2.5-Pro [33]), against non-reasoning models that use much fewer tokens (GPT-4o [34]).

TABLE III: Result of Assembly Graph Extraction (↑)

| **Method** | **Pairs** | | **Set** | |
|---|---|---|---|---|
| | F1 | Success | F1 | Success |
| Geometric Matching | 0.00 | 0.00 | 0.02 | 0.02 |
| Random Sampling | 11.99 | 2.35 | 53.51 | 9.41 |
| Incomplete Manual (GPT) | 36.06 | 22.35 | 75.82 | 41.18 |
| Ours (GPT) | 38.58 | 28.24 | 74.55 | 45.88 |
| Incomplete Manual (Gemini) | 56.71 | 41.18 | 82.68 | 63.53 |
| **Ours (Gemini)** | **63.42** | **49.41** | **86.06** | **69.41** |

*4) Result:* Table III shows that our method substantially outperforms the random sampling heuristic, with over 51% and 47% gains in F1 and success rate for Pair Matching, and over 32% and 60% for Set Matching. These results highlight that the assembly steps in our dataset are largely non-trivial and cannot be solved by chance. Similarly, the near-zero success of Geometric Matching highlights that traditional methods fail to generalize across diverse geometries, even with carefully chosen thresholds and templates.

Manual2Skill++ bridges this gap by leveraging VLM-derived commonsense reasoning to accurately extract complex assembly relationships from abstract manuals. Despite longer runtimes (averaging 2-3 minutes per assembly step) for reasoning VLMs, they exceed non-reasoning VLMs by over 20% across most metrics, underscoring the dataset's demand for advanced visual-spatial reasoning. Finally, strong performance under the "Incomplete Manual" condition showcases our approach's robustness: even when a manual step is omitted, both models correctly infer the missing connection information to complete the assembly graph. These findings confirm the viability of our framework to advance automated assembly understanding in a spectrum of assembly tasks.

### B. Pose Alignment

*1) Baseline:* We evaluate the performance of our method on our dataset. We compare our method with two baselines: SingleImage [4] and Manual2Skill [7].

Since our method assumes access to precise attachment feature locations, we provide both baselines with masked point clouds where attachment points are highlighted, ensuring fair comparison under equivalent conditions.

*2) Evaluation Metric:* We adopt comprehensive evaluation metrics to assess the pose alignment, following [7], we report Geodesic Distance (GD), Root Mean Squared Error (RMSE), Chamfer Distance (CD) and Part Accuracy (PA).

TABLE IV: Result of Pose Alignment

| | GD↓ | RMSE↓ | CD↓ | PA↑ |
|---|---|---|---|---|
| SingleImage | 1.86 | 0.2653 | 0.418 | 0.043 |
| Manual2skill | 0.76 | 0.0880 | 0.065 | 0.254 |
| **Ours** | **0.03** | **0.0013** | **0.005** | **0.944** |

*3) Result:* The pose alignment results (Table IV) reveal that the strongest baseline methods still incur centimeter-level errors, which are insufficient for precise connection operations. In contrast, our constraint-matching approach consistently achieves millimeter-level alignment accuracy, meeting the stringent precision demands of everyday assembly tasks. This substantial improvement underscores the effectiveness of our connection-aware representation: by explicitly modeling physical joint constraints, we enable dramatically more accurate part alignment, thereby bridging the gap between high-level assembly planning and execution.

### C. Connection Execution

*1) Experiment Setup:* To evaluate whether our hierarchical graph representation and constraint-based pose alignment suffice for end-to-end assembly, we decompose each of the four benchmark tasks (Chair, Shoe Shelf, Plane, LEGO Figure) into a sequence of connection operations (insertion for mortise-tenon and dowel; screw tightening for screw) by traversing the corresponding hierarchical graph. We enforce a top-down insertion constraint to ground the relative poses into the world frame: each operation fixes one part on the table and initializes the other part or connector 2 cm above its aligned pose along the global Z-axis.

To quantitatively assess each connection operation, we define a trial as successful if the pose error of the held component relative to its ground-truth pose falls below two thresholds: a rotation error of $\varepsilon_R = 0.05$ rad and a translation error of $\varepsilon_t = 0.2mm$. These thresholds were carefully calibrated to guarantee that any trial meeting both criteria corresponds to a true insertion or tightening event, $i.e.$, the peg is fully inserted into its designated hole rather than merely contacting or bypassing it. By enforcing this strict metric, we ensure that every recorded success reflects a genuinely correct peg insertion or screw-tightening operation.

*2) Connection Strategy Evaluation:* Under this setup, we compare three connection strategies:

- Random Search: The held part is lowered vertically until the distance to the target hole stops decreasing, then constant downward pressure with small lateral perturbations.
- Grid Search: A 1 cm side-length grid with 2 mm resolution is traversed in an S-shape over the target slot. Insertion is attempted at each grid point, with small perturbations to correct misalignment or tilt-induced friction upon jamming.
- Force-Position Hybrid: Extends Grid Search by using a tri-axial force sensor to detect contact forces, triggering a fine insertion phase where displacement compensation based on magnitude and direction of lateral force guide targeted corrections, greatly improving success in tight tolerance scenarios.

TABLE V: Result of Connection Policy (↑)

| | Chair | Shoe Shelf | Plane | LEGO Figure |
|---|---|---|---|---|
| Random Search | 0.039 | 0.033 | 0.015 | 0.157 |
| Grid Search | 0.733 | 0.760 | 0.687 | 0.707 |
| **Force-Position hybrid** | **0.767** | **0.816** | **0.737** | **0.750** |

*3) Result:* The results in Table V represent averages connection success rate across all sub-tasks within each item, each connection operation is repeated 100 times with 3mm uniform initial pose perturbation. Random Search consistently fails across all tasks. Grid Search demonstrates moderate performance with success rates between 68-76%, but remains sensitive to pose uncertainties and connector tolerances. Force-Position Hybrid strategy achieves the highest reliability, with success rates ranging from 73.7% to 81.6% across all tasks. These results demonstrate two key findings: first, our hierarchical graph representation correctly determines the assembly sequence for complex multi-step tasks. More importantly, our connection-constrained pose alignment achieves millimeter-level precision that enables robust execution through simple local search strategies. The consistent performance across diverse connection scenarios validates that our framework provides a reliable foundation for automating complex assembly tasks.

## VI. CONCLUSION

This paper presents Manual2Skill++, a vision-language framework that explicitly models connector relationships in robotic assembly. By extracting structured hierarchical graphs from manuals, our approach achieves millimeter-level pose alignment, enabling robust execution across complex tasks. We bridge the gap in connection understanding by curating a diverse dataset with explicit annotations and establishing a realistic simulation benchmark with multiple connector modalities. This work provides the community with the tools to develop and evaluate methods on authentic, contact-rich assembly scenarios. Ultimately, Manual2Skill++ lays the groundwork for future advances in connector-aware manipulation, multimodal task understanding, and the development of truly generalizable assembly systems.

## REFERENCES


[1] O. Spector, V. Tchuiev, and D. Di Castro, "Insertionnet 2.0: Minimal contact multi-step insertion using multimodal multiview sensory input," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6330–6336.

[2] Y. Tian, J. Xu, Y. Li, J. Luo, S. Sueda, H. Li, K. D. Willis, and W. Matusik, "Assemble them all: Physics-based planning for generalizable assembly by disassembly," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–11, 2022.

[3] Y. Tian, K. D. Willis, B. Al Omari, J. Luo, P. Ma, Y. Li, F. Javid, E. Gu, J. Jacob, S. Sueda *et al.*, "Asap: Automated sequence planning for complex robotic assembly with physical feasibility," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4380–4386.

[4] Y. Li, K. Mo, L. Shao, M. Sung, and L. Guibas, "Learning 3d part assembly from a single image," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer, 2020, pp. 664–682.

[5] G. Scarpellini, S. Fiorini, F. Giuliari, P. Moreiro, and A. Del Bue, "Diffassemble: A unified graph-diffusion model for 2d and 3d reassembly," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 098–28 108.

[6] R. Wang, Y. Zhang, J. Mao, R. Zhang, C.-Y. Cheng, and J. Wu, "Ikea-manual: Seeing shape assembly step by step," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 428–28 440, 2022.

[7] C. Tie, S. Sun, J. Zhu, Y. Liu, J. Guo, Y. Hu, H. Chen, J. Chen, R. Wu, and L. Shao, "Manual2skill: Learning to read manuals and acquire robotic skills for furniture assembly using vision-language models," *arXiv preprint arXiv:2502.10090*, 2025.

[8] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," *arXiv preprint arXiv:2305.12821*, 2023.

[9] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine, "Fmb: a functional manipulation benchmark for generalizable robotic learning," *The International Journal of Robotics Research*, vol. 44, no. 4, pp. 592–606, 2025.

[10] B. Jones, D. Hildreth, D. Chen, I. Baran, V. G. Kim, and A. Schulz, "Automate: A dataset and learning approach for automatic mating of cad assemblies," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–18, 2021.

[11] Y. Lee, E. S. Hu, and J. J. Lim, "Ikea furniture assembly environment for long-horizon complex manipulation tasks," in *2021 ieee international conference on robotics and automation (icra)*. IEEE, 2021, pp. 6343–6349.

[12] R. Wu, C. Tie, Y. Du, Y. Zhao, and H. Dong, "Leveraging se (3) equivariance for learning 3d geometric shape assembly," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 311–14 320.

[13] S. Sellán, Y.-C. Chen, Z. Wu, A. Garg, and A. Jacobson, "Breaking bad: A dataset for geometric fracture and reassembly," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 885–38 898, 2022.

[14] B. Du, X. Gao, W. Hu, and R. Liao, "Generative 3d part assembly via part-whole-hierarchy message passing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 850–20 859.

[15] F. Suárez-Ruiz, X. Zhou, and Q.-C. Pham, "Can robots assemble an ikea chair?" *Science Robotics*, vol. 3, no. 17, p. eaat6385, 2018.

[16] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "Ikeabot: An autonomous multi-robot coordinated furniture assembly system," in *2013 IEEE International conference on robotics and automation*. IEEE, 2013, pp. 855–862.

[17] M. Yu, L. Shao, Z. Chen, T. Wu, Q. Fan, K. Mo, and H. Dong, "Roboassembly: Learning generalizable furniture assembly policy in a novel multi-robot contact-rich simulation environment," *arXiv preprint arXiv:2112.10143*, 2021.

[18] Y. Li, K. Mo, Y. Duan, H. Wang, J. Zhang, and L. Shao, "Category-level multi-part multi-joint 3d shape assembly," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3281–3291.

[19] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A survey on multimodal large language models," *arXiv preprint arXiv:2306.13549*, 2023.

[20] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "Copa: General robotic manipulation through spatial constraints of parts with foundation models," *arXiv preprint arXiv:2403.08248*, 2024.

[21] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn, "Yell at your robot: Improving on-the-fly from language corrections," *arXiv preprint arXiv:2403.12910*, 2024.

[22] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *IEEE Access*, 2024.

[23] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "pi_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[24] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[25] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[26] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[27] H. Jiang, B. Huang, R. Wu, Z. Li, S. Garg, H. Nayyeri, S. Wang, and Y. Li, "Roboexp: Action-conditioned scene graph via interactive exploration for robotic manipulation," *arXiv preprint arXiv:2402.15487*, 2024.

[28] X. Li, M. Zhang, Y. Geng, H. Geng, Y. Long, Y. Shen, R. Zhang, J. Liu, and H. Dong, "Manipllm: Embodied multimodal large language model for object-centric robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 061–18 070.

[29] Z. Zhao, W. S. Lee, and D. Hsu, "Large language models as commonsense knowledge for large-scale task planning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[30] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[31] A. Goldberg, K. Kondap, T. Qiu, Z. Ma, L. Fu, J. Kerr, H. Huang, K. Chen, K. Fang, and K. Goldberg, "Blox-net: Generative design-for-robot-assembly using vlm supervision, physics simulation, and a robot with reset," *arXiv preprint arXiv:2409.17126*, 2024.

[32] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, "Abc: A big cad model dataset for geometric deep learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[33] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[34] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.