Supplementary Materials for IKEA-Manual: Seeing Shape Assembly Step by Step

First, we introduce our annotation methodology for all the annotations in our dataset in Section 1. Then we provide more details about the manual plan generation experiments in Section 2. Section 3 presents additional experiments results with standard error. Section 4 presents the documentation of IKEA-Manual following the Datasheets for Datasets [1] standards.

We have also included two videos that show the annotation process of our web-based interface for manual-related annotation.

1 Annotation Methodology

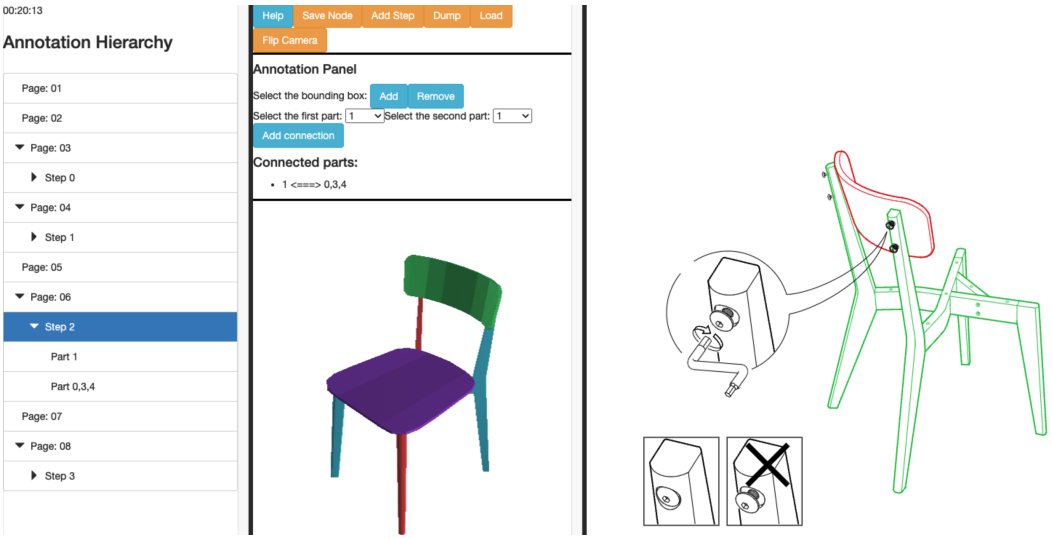

Figure 1: Our web-based interface for manual-related annotation. It provides functionality to annotate the assembly plan, segmentation and 2D-3D correspondences in a unified environment.

1.1 Overview

For each 3D model, we first move it so that its Axis-Aligned-Bounding-Boxes (AABB) are centered at the origin and then scale the shape to make the diagonal of the AABB have a unit length. Then we use Blender to decompose the 3D model into assembly parts by grouping the faces of the meshes into different objects. There may be some missing parts in the original collected 3D model and the sizes of parts may be inaccurate compared with the assembly manuals, which we also manually fix. After the assembly part decomposition is finished, we assign an index to each assembly part for the preparation of the following annotations.

Submitted to the 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks. Do not distribute.

Based on the annotation system of PartNet [2], we build a web-based interface for annotations related to visual manuals as shown in Fig. 1. The annotator is presented with the decomposed 3D shape and the visual manual. First, steps that contain connecting actions are preserved. Then the annotator is asked to label the connection action in each step by comparing the parts in the manual with the indexed 3D assembly parts. For segmentation, as all IKEA manuals are available in vector graphics format, we design an annotating tool that enables the annotator to directly assign labels to line strokes by clicking or box selection without dense pixel-level labeling. The pixel-level segmentation is achieved by a flood fill algorithm. Finally, correspondences are built between assembly parts in each step and manual image by annotating keypoints. The annotation interface allows the annotator to manually assign correspondence keypoints between 3D parts and 2D keypoints and see the pose estimation results from the EPnP [3] algorithm as feedback. Additionally, we annotate the connection and geometric equivalence relationship between assembly parts.

1.2 Annotation Details

Here we provide more details about the processes for different types of annotations.

Part connection relation. To efficiently annotate the connection relations between primitive assembly parts, for every object, we first compute the Chamfer distances between primitive assembly parts, and record the pairs of parts whose Chamfer distance is within a threshold $5e-6$. This can be treated as a coarse annotation of the connection relations. As close parts are not necessarily connected, we manually check the original assembly manuals to update the connection relationships to obtain the final annotation.

Part geometrical equivalence relation. Similarly to the annotation process of part connection relation, we compute a coarse version of the relation automatically. For each part, we first center and normalize its mesh, then we sample 2000 points from it by furthest point sampling and perform PCA on the sampled point cloud to put them in canonical space. Then we compute the Chamfer distance between point clouds of different parts in an object, and record the pairs whose Chamfer distance is within 0.06. Based on this information, we manually check the visual manuals to obtain the geometrical equivalence relations between parts.

Pixel segmentation. We annotate the pixel segmentation of manuals based on the line segmentation. Since the line segmentation only outlines the boundary of each part, we need to fill the internal areas of parts to obtain the pixel segmentation. As the assembly parts are usually non-convex and abundant with holes, to fill the correct areas, we need to mark whether each area belongs to a part or not. We implemented a tool where the annotator only needs to specify the background area by clicking a seed point. Then flood fill algorithm is used to propagate the annotation to the other pixels that have the same color with the seed point, until a boundary (either an image border or a line with other colors) is hit. Once we have annotated all background areas. We treat the remaining areas as the internal areas of parts, and use flood fill algorithms to fill them.

2D-3D alignment between parts and manuals. Similar to previous works [4, 5], given a list of 3D keypoints and their corresponding 2D keypoints, we align parts and manuals by finding the camera parameters and 3D poses that minimize the reprojection error of keypoints. We parameterize the camera intrinsic matrix as:

$$\begin{bmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{bmatrix}$$

where f is the focal length, and w, h correspond to the width and height of the image. This is under the assumption that the camera is zero-skew, has square pixels and the optical center is at the center of the image. As the EPnP algorithm does not optimize for the focal length, we search this parameter from 300 to 4000 with a step size of 10. For each candidate, we compute the corresponding 3D pose by EPnP. Then we select the focal length with minimum reprojection error.

For each part in a step, we ask the annotator to first annotate keypoints on the 3D mesh, and then annotate the corresponding 2D keypoints on the manual. The number of keypoints for each part is

	IKEA-Manual (Chair)	
	Shape CD	Part Accuracy
RGL-Net [8]	$0.0583_{\pm 0.00002}$	$2.01_{\pm 0.00000}$
Huang et al. [9]	$0.0151_{\pm 0.00006}$	$6.90_{\pm 0.00000}$

Table 1: Quantitative results of RGL-Net [8] and Huang et al. [9], evaluated on the chair category of IKEA-Manual. We report the mean and standard error of all metrics across 3 runs.

between 5 and 18. After all the keypoints are annotated, the annotation tool will provide feedback about the quality of the keypoints by rendering the 3D part with the estimated camera parameter and pose. Then the user can decide whether to refine the keypoints.

For quality control, after estimating the poses, we also compute the projected mask of each part on the image, and compare the Intersection-over-Union(IoU) with the annotated pixel segmentation masks. Then we refine the keypoints of parts with low IoU scores. In the end, we achieve an average IoU of 71.3 for all the 1056 annotated poses.

2 Details for Manual Plan Generation

Algorithm 1 GeoCluster

Input: Parts $\{P_1, \cdots, P_n\}$
Output: An assembly tree T
  procedure GROUP(I)
    partition I into clusters $\{C_1, \cdots, C_m\}$ based on DGCNN [6] features of $\{P_i\}_{i \in I}$
    if clustering fails **then**
      return a node with children I
    end if
    choose cluster C^* with minimum in-cluster feature variance
    return a node with children $C^* \bigcup \{\text{GROUP}(I \setminus C^*)\}$
  end procedure
  return GROUP($\{1, \cdots, n\}$)

Here, we provide more details about the GeoCluster baseline used in the manual plan generation task. The pseudocode for GeoCluster is shown in Algorithm 1. We use outputs from the intermediate layers of DGCNN [6] as the features of parts for clustering. To determine a clustering, we first use KMeans to perform multiple passes of clustering with different number of clusters. Then we select the clustering with maximum Silhouette Coefficient [7], which measures the quality of the clustering. If the maximum Silhouette Coefficient is lower than a threshold, we treat the clustering process as failed. After determining the clustering, we select the cluster with minimum in-cluster variance, which is the mean of distances between every pairs of points in the cluster. By this procedure, we expect to group parts with similar geometries in a node during the construction of the assembly plan.

Implementation details. We leverage the 1024-dim vector after the final global max pooling layer of DGCNN as the feature for each part. And the DGCNN model is pretrained for the classification task in the original paper. During the clustering process, we select between clusterings of 2 and 3 clusters. Finally, the threshold for clustering failure is set to be the negative of in-cluster variance treating all the points as a single cluster.

3 Additional Experiment Results

In this section, we report expeimental results with standard error. All the models are run on a single Titan RTX. The training time for the part segmentation experiment is around 1 hour.

For Section 4.2, we train the model from Li et al. [10] with 3 different random seeds and obtained an IoU with mean 25.58 and standard error 0.95.

For Section 4.3, evaluation results with standard error are shown in Table 1. The standard error for PartNet are not shown in this table since the results are taken from the original paper with no standard error reported.

4 Datasheets For Dataset

In this section, we present the documentation for IKEA-Manual following the Datasheets for Datasets [1] standards.

4.1 Motivation

- **For what purpose was the dataset created?** To study the shape assembly problem with human-designed assembly manuals from perspectives of manual generation and visual manual interpretation.
- **Who created the dataset?** The authors listed on this paper, which include researchers from Stanford, MIT and Autodesk.
- **Who funded the creation of the dataset?** The creation of IKEA-Manual is funded by Stanford and Autodesk.

4.2 Composition

- **What do the instances that comprise the dataset represent and how many instances are there?** As our dataset is hierarchical, there are multiple types of instances involved. On the highest level, IKEA-Manual includes 102 instances. Each instance contains 1) the 3D model of an IKEA object and 2) a visual assembly manual comprised of a series of vector graphic images. For each object, IKEA-Manual includes the step-by-step information about how an object is assembled. Each step can be treated as an instance that contains: an image specifying the assembly process via the projection of the assembly parts. There are 393 step instances in total.
- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** It is a sampled subset.
- **What data does each instance consist of?** For each object instance, we provide:
  - Assembly part decomposition (specified as `.obj` files).
  - Connection and geometric equivalence relationships between assembly parts. (specified as a `.json` file).

  For each step instance, we provide :
  - Assembly plan information. (specified as tree structures in a `.json` file).
  - Segmentation information. (specified as `.png` files as well as serialized information that can be decoded to `numpy.array`).
  - Pose estimation information. (specified as matrices in a `.json` file).
- **Is any information missing from individual instances?** No.
- **Are relationships between individual instances made explicit?** There is no relationship between object instances. For step instances, different steps may belong to the same object. We include this information in the dataset.
- **Are there recommended data splits?** We include the data splits for the three tasks presented in the main paper. For other possible tasks, we do not include recommended data splits.
- **Are there any errors, sources of noise, or redundancies in the dataset?** Because the annotation is collected by humans, we expect there may be errors and noises for shape decomposition, segmentation and pose estimation in the dataset.
- **Is the dataset self-contained, or does it link to or otherwise rely on external resources?** The original assembly manuals need to be downloaded from external websites [11]. We will

provide links and download scripts for these resources. And we will maintain the availability of these links. The copyright of the visual manuals is owned by IKEA.

- **Does the dataset contain data that might be considered confidential?** No.
- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** No.
- **Does the dataset relate to people?** No.

4.3 Collection Process

- **How was the data associated with each instance acquired? What sampling strategies/mechanisms were used to collect the data?** The 3D models of IKEA objects are downloaded from online repositories. The original IKEA assembly manuals are downloaded from IKEA's official website [11]. We randomly select IKEA objects from IKEA's website, and check if there are available 3D models online. If so, we check if they match the 2D projections in the assembly manuals. We include samples that meet this criterion. We ensure the collection processes comply with the terms of use for all assets.
- **Who was involved in the data collection process?** Students in the author list.
- **Over what timeframe was the data collected?** The data as well as the annotation are collected from February 2022 to May 2022.
- **Were any ethical review processes conducted?** N/A.
- **Does the dataset relate to people?** No.

4.4 Preprocessing/cleaning/labeling

- **Was any preprocessing/cleaning/labeling of the data done?** Yes, we ask students to manually annotate the shape decomposition, segmentation and pose estimation information. We refer readers to Section 1 for more details.
- **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?** Yes. We will include this information in the the dataset.
- **Is the software used to preprocess/clean/label the instances available?** Yes. We will make the software publicly available along with the dataset.

4.5 Uses

- **Has the dataset been used for any tasks already?** Yes, we include manual plan generation, part segmentation and part assemlby tasks in our main paper.
- **Is there a repository that links to any or all papers or systems that use the dataset?** No other papers have used IKEA-Manual yet.
- **What (other) tasks could the dataset be used for?** The dataset can also be used for a variety of tasks like 3D reconstruction and pose estimation.
- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** No.
- **Are there tasks for which the dataset should not be used?** No.

4.6 Distribution

- **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** Yes. It will be publicly available for download.
- **How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?** We will distribute our dataset through Zeonodo [12], which can provide long-term availability and a digital object identifier for IKEA-Manual.
- **When will the dataset be distributed?** IKEA-Manual will be publicly available before September 2022.

- **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** We licensed the annotations of visual manuals under `CC BY-NC-SA 4.0`*. Copyright of the original visual manuals and 3D models are owned by their creators respectively. More detailed breakdown about license will be included in the metadata of the dataset.
- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** The copyright of the visual manuals is owned by IKEA. And the copyright of 3D models is owned by their creators. We bear all responsibility in case of violation of rights.
- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No.

4.7 Maintenance

- **Who will be supporting/hosting/maintaining the dataset?** The authors of the paper will be maintaining the dataset.
- **How can the owner/curator/manager of the dataset be contacted?** The correspondence author of the main paper can be contacted via email listed in the main paper.
- **Will the dataset be updated?** We will update the dataset promptly to keep external resources available and correct annotation errors. New versions of the dataset will be shown on the Zenodo page.
- **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?** N/A.
- **Will older versions of the dataset continue to be supported/hosted/maintained?** Yes, the platform Zenodo where we will host our dataset provides this feature.
- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** We provide `CC BY-NC-SA 4.0` license for annotations so others can freely contribute to annotations in our dataset as well as build downstream applications based on our dataset.

*https://creativecommons.org/licenses/by-nc-sa/4.0/

References

[1] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. 1, 4

[2] 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*, 2019. 2

[3] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *IJCV*, 81(2):155, 2009. 2

[4] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013. 2

[5] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3D: Dataset and methods for single-image 3D shape modeling. In *CVPR*, 2018. 2

[6] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. In *ACM TOG*, 2019. 3

[7] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 3

[8] Abhinav Narayan, Rajendra Nagar, and Shanmuganathan Raman. RGL-NET: A recurrent graph learning framework for progressive part assembly. In *WACV*, 2022. 3

[9] Jialei Huang, Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas Guibas, and Hao Dong. Generative 3D part assembly via dynamic graph learning. In *NeurIPS*, 2020. 3

[10] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3D part assembly from a single image. In *ECCV*, 2020. 4

[11] IKEA. https://www.ikea.com/us/en/. 4, 5

[12] Zenodo. https://zenodo.org. 5