LEARNING TO ASSEMBLE FURNITURE FROM THEIR 2D DRAWINGS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

DENGE UZEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

DECEMBER 2023

Approval of the thesis:

LEARNING TO ASSEMBLE FURNITURE FROM THEIR 2D DRAWINGS

submitted by **DENGE UZEL** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. M. A. Sahir Arıkan
Head of Department, **Mechanical Engineering** _______________

Assoc. Prof. Dr. Ahmet Buğra Koku
Supervisor, **Mechanical Engineering, METU** _______________

Prof. Dr. Sinan Kalkan
Co-supervisor, **Computer Engineering, METU** _______________

Examining Committee Members:

Assoc. Prof. Dr. Ali Emre Turgut
Mechanical Engineering, METU _______________

Assoc. Prof. Dr. Ahmet Buğra Koku
Mechanical Engineering, METU _______________

Prof. Dr. Sinan Kalkan
Computer Engineering, METU _______________

Assoc. Prof. Dr. Erol Şahin
Computer Engineering, METU _______________

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering, TED University _______________

Date:07.12.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Denge Uzel

Signature :

ABSTRACT

LEARNING TO ASSEMBLE FURNITURE FROM THEIR 2D DRAWINGS

Uzel, Denge
M.S., Department of Mechanical Engineering
Supervisor: Assoc. Prof. Dr. Ahmet Buğra Koku
Co-Supervisor: Prof. Dr. Sinan Kalkan

December 2023, 57 pages

Prior work on learning furniture assembly assumes the availability of precise 3D information about the target furniture. This thesis elevates this assumption by learning to assemble furniture given a 2D drawing of its assembled form. To this end, the thesis introduces a novel network that can learn the similarity (conformity) between a 2D furniture drawing and a 3D point cloud representing the current state of the assembly. The proposed network is then used to formulate a reward signal for assembly learning using reinforcement learning.

To ensure real-world applicability, a simulation environment generates a visually similar representation of the assembled furniture based on IKEA assembly instructions. The research encompasses three furniture classes: bookcase, chair, and table. A dedicated dataset is presented, including 2D furniture drawings resembling IKEA instructions and a 3D mesh model dataset encompassing various furniture assembly scenarios. The AssembleRL-2D model is trained using positive and negative input pairs from the 2D drawing and 3D mesh datasets, demonstrating proficiency across the three furniture classes. Notably, the model achieves accurate final furniture assembly, even in various assembly combinations where the parts of a chair model are assembled in different orders.

AssembleRL-2D marks promising progress in furniture assembly learning, representing the inaugural application of a reinforcement learning model grounded in 2D final furniture assembly knowledge as a reward. The significance of this model in robotic assembly is highlighted by its capability to solve previously unencountered problems, showcasing its potential impact on addressing novel challenges in the field.

Keywords: Assembly Learning, 2D Manual-Like Image Guided 3D Furniture Assembly, 2D Drawing - 3D Assembly Similarity Network, Reinforcement Learning

ÖZ

2B ÇİZİMDEN MOBİLYA MONTAJI ÖĞRENME

Uzel, Denge
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Ahmet Buğra Koku
Ortak Tez Yöneticisi: Prof. Dr. Sinan Kalkan

Aralık 2023, 57 sayfa

Mobilya montajını öğrenmeye dair daha önceki çalışmalar, hedef mobilya hakkında 3B bilginin varlığına dayanmaktadır. Bu tez, nihai montajlanmış mobilyanın 2B çiziminden yola çıkarak mobilya monte etmeyi öğrenen yeni bir yöntem sunması itibariyle literatürdeki çalışmalardan ayrılır. Bu doğrultuda tez, 2B mobilya çizimi ile montajın mevcut durumunu temsil eden 3B nokta bulutu arasındaki benzerliği (uygunluk) öğrenebilen yeni bir ağı tanıtmaktadır. Önerilen ağ, pekiştirmeli öğrenme kullanılarak montaj öğrenimi için bir ödül sinyali formüle edilmesi için kullanılmaktadır.

Gerçek hayata uygunluğu adına, mobilyalara ait IKEA montaj talimatlarında yer alan görseller baz alınarak, montajlanmış mobilya çizimine ait benzer bir görüntü simülasyon ortamında oluşturulur. Kitaplık, masa ve sandalye olmak üzere 3 sınıf temel alınarak, IKEA montaj talimatı benzeri 2B mobilya çizim veriseti ile olası mobilya montaj senaryolarından oluşan 3B ağ modeli veriseti sunmaktayız.

Üç boyutlu nokta bulutları ile temsillenen montajlanmış mobilya ağ veriseti ile iki boyutlu çizim verisetinden positif ve negatif girdi çiftleri ile AssembleRL-2B mimarisi beslenerek, üç mobilya sınıfı üzerinde bu yapının öğrenim başarısı incelenmiştir. Önermekte olduğumuz model, bir sandalye modeline ait parçaların değişik sırayla birleştirildiği farklı montaj kombinasyonlarında bile doğru nihai mobilya montajını gerçekleştirmektedir.

AssembleRL-2B, ilk defa 2B nihai mobilya montaj bilgisininin ödül olarak baz alındığı bir pekiştirmeli öğrenme modeli olmasıyla, mobilya montaj öğrenminde umut vadeden bir ilerleme olma özelliği taşımaktadır. Bu modelin daha önce karşılaşmadığı problemleri çözebilmesinin gösterilmesi ile birlikte robotik montajı için önemi kanıtlanmaktadır.

Anahtar Kelimeler: Montaj Öğrenimi, 2B Manuel Benzeri Görüntü Kılavuzlu 3B Mobilya Montajı, 2B Çizim - 3B Montaj Benzerlik Ağı, Pekiştirmeli Öğrenme

To My Family

ACKNOWLEDGMENTS

I extend my heartfelt appreciation to my advisor, Assoc. Prof. Dr. Ahmet Buğra Koku, for his unwavering support throughout my academic journey from undergraduate to this thesis. His dedication and encouragement have been instrumental in shaping my academic career, and I deeply admire his fair and benevolent approach to all students. Special thanks go to my co-advisor, Prof. Sinan Kalkan, whose valuable guidance, comments, and support have played a crucial role in developing this thesis. I appreciate his patience in assisting and guiding me. I am indebted to Assoc. Prof. Dr. Erol Şahin for accepting the project and providing continuous support and patience throughout its execution. Working with professors like them, whose passion for academia is genuinely commendable, has been an honor.

A sincere expression of gratitude goes to Özgür Aslan and Burak Bolat for their unwavering support, guidance, and friendship during the research. Collaborating with them as a team has been a valuable opportunity, and I felt privileged to work together on a daily basis. My thanks also extend to Ege Meyvecioğlu and Yunus Talha Erzurumlu for their contributions to our collaborative efforts.

I want to express my appreciation to my dear friend Mehmet Şahin for his loyal support in various forms and for our continued academic conversations since undergraduate school. Additionally, I thank my friends Umutcan Üstündaş, Giray Önür, Cem Bilaloğlu, Atakan Botasun, İlyas Eren Yılmaz, Ege Keyvan, Umut Kurt, and Anıl Gülbeden, with whom I created lasting memories within METU ROMER. Their support has been invaluable.

My gratitude extends to Harika Firuzan Kaya, Çiğdem Karatepe, and all members of METU ROMER for fostering an admirable working environment.

I express my thanks to Dilay Güleryüz, Elif Gülen, Elif Tola, Hatun Taşkaya, Hatem Ak, Hümeyra Beyan, Murat Çakır, Öznur Uşaklılar and Sıla Yüksel for their encouragement and support. Special thanks to my cousin Seda Nur Uzel for her constant

presence. I deeply appreciate my friends Merve Saraç and Zozan Sarı, who have stood by me for years. Ultimately, I would like to thank Feyza Pirim, whose reason and goodwill have been a guiding light in overcoming difficulties.

Lastly, my deepest gratitude goes to my parents, Saadet Uzel and Hüseyin Uzel, for their boundless love, unwavering support, and existence in my life.

This study is part of the KALFA project "New Methods for Assembly Scenarios with Collaborative Robots" supported by TUBITAK 1001 (Project No: 120E269).

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
BCE	Binary Cross Entropy
CAD	Computer Aided Design
CNN	Convolutional Neural Network
FL	Focal Loss
MLP	Multi Layer Perceptron
OFF	Object File Format
OS	Object State
PIL	Python Imaging Library
PPFNet	Point Pair Feature Network
ResNet	Residual Neural Network
RL	Reinforcement Learning
STL	Stereolithography / Standard Triangle Language (?)
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Throughout human history, individuals have constructed structures and crafted tools to safeguard and enhance their lives. Despite evolving lifestyles and opportunities, the fundamental nature of human existence revolves around production and consumption. Modern technological advancements and production methodologies have empowered individuals to create increasingly complex products easily. The journey from conveyor belts and mass production to today's digital era has substantially reduced industry costs [8]. The Internet's accessibility, virtual simulation capabilities, and the integration of intelligent concepts have catalyzed a transformation in various sectors, leading to a surge in companies utilizing modern production techniques and the widespread adoption of industrial robots [9]. While human labor persists in roles demanding flexibility, creativity, adaptability, or decision-making skills, robots are now frequently employed in tasks such as inspection, painting, welding, and assembly lines that involve repetitive and ergonomically taxing movements [10, 11]. Notably, the number of industrial robots installed has risen by approximately 135% over the past decade, and projections suggest that over half a million will be in operation by 2024 in Figure 1.1 [2].

This widespread implementation is anticipated to streamline assembly lines and reduce production lead times significantly. As industrial robots have advanced to perceive and respond to environmental factors, they are no longer confined to isolated areas away from human interaction. Collaborative robots, known as "cobots," have emerged to work alongside humans in joint tasks, both physically and synchronously

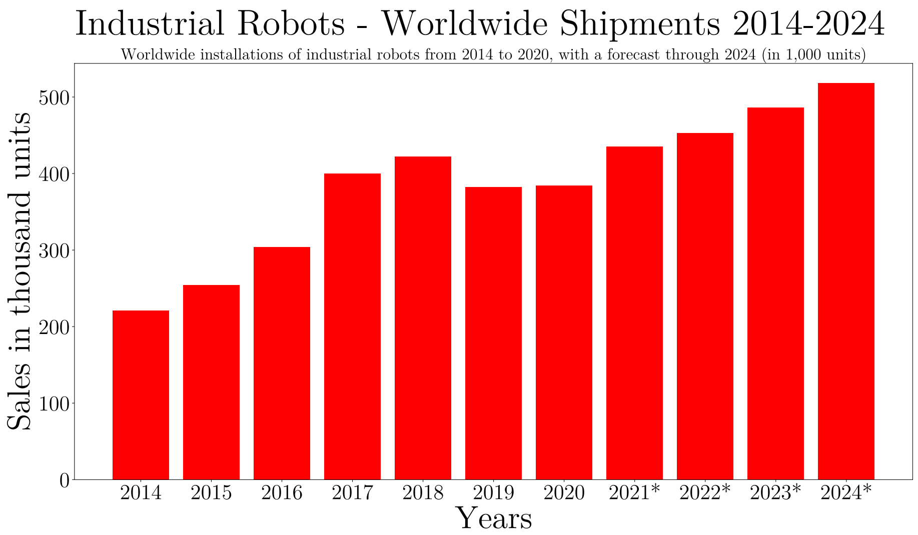


Figure 1.1: The worldwide shipments of industrial robots from 2014 to 2024. The data in the graph is taken from [2].

[12]. Integrating cobots into assembly lines serves to help the physical and cognitive burden on humans, simultaneously enhancing production quality and productivity [13]. Consequently, the collaboration between humans and robots has become a crucial focal point in industrial advancements.

The ÇIRAK (apprentice) and KALFA (journeyman) initiatives involve collaborative projects where humans and cobots work together on assembly lines. In this dynamic, the human takes on a superior role while the cobot is an assistant. In the ÇIRAK project, the cobot observes and mimics the master human's movements, providing the necessary tools during the assembly process [14]. Building on this, the KALFA project further refines human-robot interaction using Disney animation principles, positioning the cobot as a knowledgeable figure in the assembly line process, similar to a journeyman rather than a mere apprentice. To simulate real-world challenges, the assembly scenario chosen for both projects involves the assembly of IKEA furniture. The complexity of this task, coupled with its physical accessibility and the availability of CAD models and information on furniture, makes it a suitable problem for exploration. The succeeding KALFA project extends the capabilities of the cobot, enabling it to collaborate with humans in assembling IKEA furniture. A model is developed within this project that can learn the sequential installation of parts using

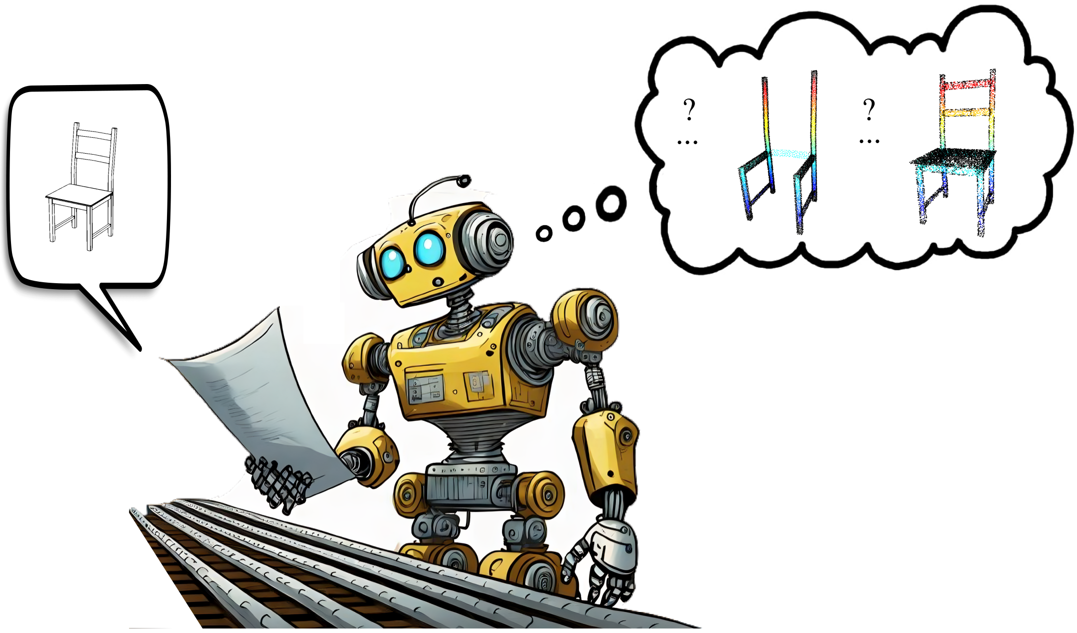


Figure 1.2: The robot holds a manual-like image and tries to imagine an assembled chair in the assembly line. The robot and assembly line image generated using Adobe Firely from the prompt "a cartoon robot is thinking while looking at a paper on a cartoon conveyor belt."

three-dimensional data [1]. However, traditional learning methods for assembly rely on product assembly manuals. In practical situations, individuals commonly rely on the final assembled visual of the furniture provided in assembly instructions to effectively address assembly challenges. This thesis introduces a model suitable for collaborative efforts between robots and humans on a production line. The proposed model empowers the robot to accurately establish the correct assembly process by examining the two-dimensional assembly drawing of the product, mirroring the depicted scenario illustrated in the Figure 1.2.

1.2 Contributions

In this thesis, we study the problem of IKEA furniture assembly using learning-based methods. Our goal is to solve the assembly problem given the target assembled object as a 2D drawing, in a similar setting to how an IKEA consumer would perform assembly by following 2D drawings.

More specifically, we make the following main contributions:

- The novel dataset has been generated, encompassing both 2D manual drawings and 3D representations of potential assembly actions for IKEA furniture.
- We address assembly learning as a reinforcement learning problem where the reward is based solely on the 2D drawing of the target assembled object. To the best of our knowledge, we are the first to approach the problem in this fashion.
- In order to be able to learn from 2D drawings, we devised a network which can learn the similarity between a 2D drawing and a 3D object so that during the assembly process, the robot can identify whether the assembled object matches the target 2D drawing.
- The proposed model demonstrates the ability to generalize to previously unseen furniture objects.

1.3 Outline of the Thesis

In Chapter 2, an extensive review of existing assembly methodologies in the literature will be conducted. Subsequently, an examination of available datasets and simulation environments related to IKEA furniture, the focus of this study, will be undertaken.

Moving on to Chapter 3, we will delve into the data collection process for both 2D and 3D datasets created specifically for the AssembleRL-2D architecture, the novel model proposed in this thesis. A detailed exploration of our network architecture and a comprehensive explanation of the learning process for furniture assembly steps will follow.

Within Chapter 4, we will clarify the utilization of dataset data in our network architectures as input. Additionally, we will provide insights into the training details employed in the experiments conducted.

Chapter 5 will commence with an investigation into the most suitable 3D feature extractor for the proposed architecture. Subsequently, parameters determined through hyperparameter optimizations will be outlined. We conducted a comprehensive evaluation of the outcomes generated by our proposed method across three fundamental classes. Ultimately, the performance results of using our approach in assembly learning the furniture assembly scenario were thoroughly scrutinized.

Finally, in Chapter 6, a summary and discussion of the thesis will be presented, alongside a mention of planned future studies.

CHAPTER 2

BACKGROUND AND RELATED WORK

The challenge of furniture assembly has been a longstanding problem that researchers have attempted to address over numerous years. Existing literature establishes various methods that have been explored to solve this intricate problem. Furthermore, a wealth of simulation environments and datasets has been developed to investigate scenarios involving collaborative work between humans and robots on assembly lines.

2.1 Furniture Assembly

While it has become commonplace for humans and robots to collaborate on assembly lines, creating assembly plans remains a task managed mainly by people. A relatively unexplored aspect is the collaborative assembly involving both robots and humans [1]. The assembly learning based on the parts of the object or the 3D model of the target object has been studied before.

2.1.1 Assembly Learning Based on 3D Model Knowledge

Huang *et al.* [15] introduced a dynamic graph learning structure that predicts assembly based on the point clouds of parts from 3D objects within the PartNet dataset. This study incorporates the 6-degree-of-freedom poses of the parts. In a different study [16], a model with two network modules is proposed, utilizing learned relations to estimate the positions and scales of the parts. Li *et al.* [17] conducted a study using PartNet [18] objects, employing two network modules that extract assembly information from both the mesh rendering of the 3D model and the point cloud information of the 3D object. Another similar study [19], utilizing both the point cloud information of the 3D model and 2D image information, proposes a network architecture based on a two-stage encoder and decoder structure.

While these studies assume knowledge of the 3D model, they also presume a need for more manual information or instructional videos demonstrating the assembly steps. Despite contributing to the field, these architectures primarily focus on establishing relationships between parts, overlooking variations among the parts. This limitation restricts their generalizability to furniture with repetitive components. Addressing this challenge, Zhang *et al.* [20] are developing a self-attention mechanism to identify part relationship constraints and poses. Notably, their use of PartNet in this study struggles to identify contact points on furniture with round shapes.

In a different study conducted by Aslan *et al.* [1] within the KALFA project, the approach is likely rooted in reinforcement learning. This methodology involves training an agent by learning a policy that guides the agent's actions to maximize a reward signal, aiming to assemble furniture from point cloud 3D data. Here, the target point cloud serves as a guiding reference.

2.1.2 Assembly Learning Based on 2D Instruction Manual Knowledge

All of the above studies hinge on the assumption of having a pre-existing final assembled 3D model, which diverges from real-world scenarios. In practical settings, assembly information for a product on a production line is available, necessitating planning based on this information. This thesis specifically addresses the challenge of preparing an assembly plan for IKEA furniture. The tasks of strategizing the assembly process by consulting the IKEA manual and identifying potential errors in robot assembly according to the manual represent ongoing research challenges that still require investigation [14].

The assembly instructions provided by IKEA are typically extensive, featuring numerous pages, symbols, and diagrams. In contrast to traditional 2D drawings with clear general rules, IKEA manuals depict how parts should be manipulated using various methods, such as arrows or images of individuals holding the product, as shown

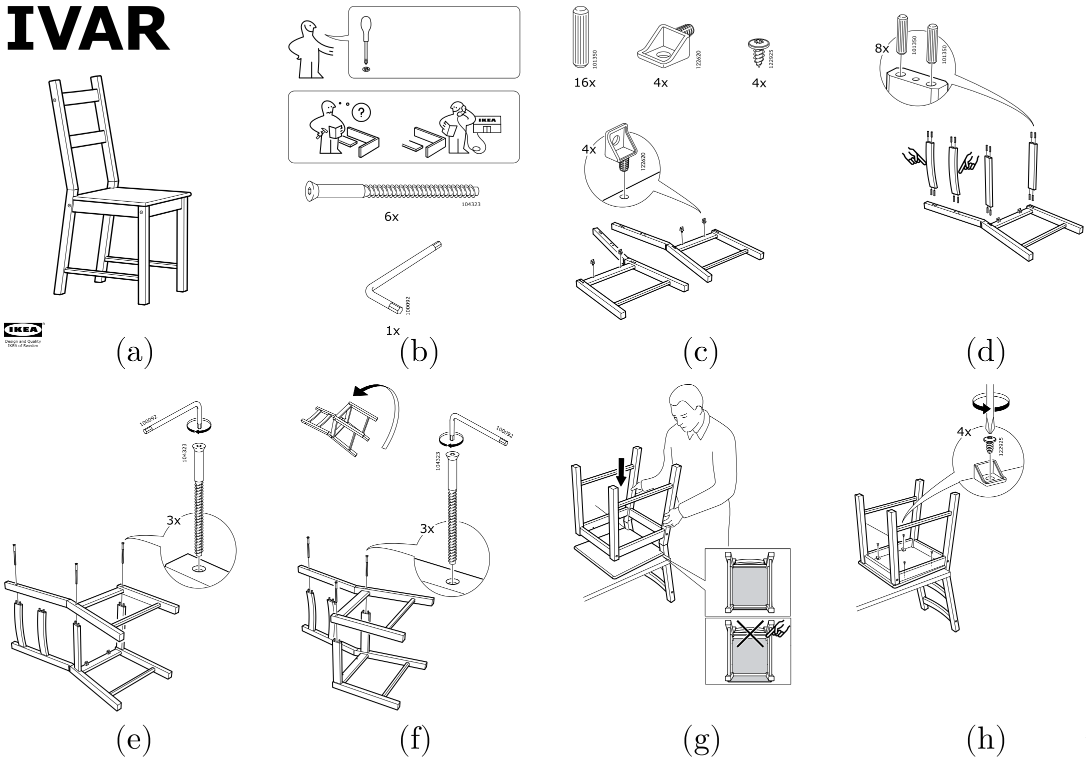


Figure 2.1: The original guidance manual for the IKEA ivar chair can be found at [3]. Here is a breakdown of key instructions: a) The visual representation of a fully assembled ivar chair. b) Image sequence: Prepare tools, read instructions, contact IKEA for assistance if there are installation questions or missing/defective parts using part numbers, and verify parts (6 countersunk head bolts, 1 allen key). c) Confirm inclusion of 16 dowels, 4 chair mounting bracket support parts, and 4 screws. Caption shows correct bracket mounting with a speech bubble. d) Dowel placement and indication of chair back curvature using fingers. e) and f) step-by-step assembly of 6 countersunk head bolts with an allen wrench. Arrows denote tightening and turning directions. g) Invert partially assembled chair for proper center alignment of the seating part with edges. h) Attach 4 screws to previously installed mounting brackets, turning in the arrow direction using a screwdriver.

in Figure 2.1. Intermediate elements required for assembly are sometimes illustrated with bubbles, but these lack detailed information about the entire process. While the manuals occasionally highlight actions to avoid, these warnings may not always be apparent. Given these complexities, utilizing the IKEA assembly manual for assembly planning poses a significant and challenging problem. Individuals typically do not strictly adhere to a sequential, step-by-step approach when assembling with a manual as guidance. Instead, they draw upon their general understanding based on past experiences. While seeking assistance for verification at specific points, many find that inspecting the final version of the product is often sufficient.

Consequently, studies like [21], which leverage the IKEA manual by tracking intermediate steps, prove valuable for detecting tools and elements used in assembly. However, these studies lack extensions for part identification and detailed assembly steps. A two-stage network architecture proposed by Wang *et al.* [22] demonstrates an ability to identify potential locations for installing a new part based on 2D manual information and the current 3D Lego shape. Similar to this approach, the design of a network architecture in which the status of furniture assembly is monitored by comparing the final assembled form from the 2D IKEA manual to its instantaneous 3D assembly status is elucidated in this thesis.

2.2 Datasets and Simulation Environments

In assembly learning studies, simulation environments are favored due to the inaccessibility and high costs associated with using real robots in assembly lines. Numerous simulation environments for robot learning, particularly in assembly scenarios, are well-documented in the literature. The Robosuite study [23] presents a versatile simulation environment tailored for various robot learning tasks, facilitating the comparison of different robot learning algorithms, extending beyond furniture assembly. In contrast, the IKEA furniture assembly environment [24] and RoboAssembly [25] specifically concentrate on furniture assembly tasks. Unlike the more general-purpose Robosuite, these environments are designed to realistically model the relationships between furniture pieces. The IKEA furniture assembly environment [24] is dedicated to the assembly scenarios of IKEA furniture parts and encompasses a dataset of over 80 distinct IKEA furniture models within its environment.

Apart from simulation environments, datasets containing IKEA furniture information can be valuable resources for furniture assembly studies. Notably, the IKEA ASM dataset [28] and the IKEA Object State (OS) dataset [27] serve as distinct sources,

Table 2.1: A comparison of IKEA 2D image and 3D part level datasets

	Shape			Manual		
	Part Segmentation	Partial Assembly Steps	Assembled Object	Real-World Object Images	Manual-Like View	2D-3D Relation
PartNet [18]	✓	✗	✓	✗	✗	✗
Pix3D [26]	✗	✗	✗	✗	✗	✓
IKEA OS [27]	✓	✗	✓	✗	✗	✗
IKEA ASM [28]	✓	✓	✓	✓	✗	✗
IKEA Objects [29]	✗	✗	✓	✓	✗	✓
IKEA Furniture [24]	✓	✗	✓	✗	✗	✗
IKEA Manual [30]	✓	✗	✓	✗	✓	✓
OUR DATASET	✓	✓	✓	✗	✓	✓

each with its unique focus. The IKEA ASM dataset [28] centers on part poses and their relations during furniture assembly. It was generated through imitation learning from videos capturing people assembling IKEA furniture, making it particularly suitable for research on human actions. On the other hand, the IKEA OS dataset [27] is primarily employed to assess 6-degree-of-freedom object pose estimation algorithms, containing data for only five furniture models. In contrast to the IKEA OS dataset, which comprises only 5 objects, the IKEA Manual dataset is more extensive, encompassing 3D models and establishing a relation with 2D manual information for over 100 objects [30]. Additionally, the Pix3D dataset [26] provides relation both 2D images of furniture and their corresponding 3D CAD models, explicitly focusing on IKEA furniture [29]. However, it does not include information about the furniture instruction manual or assembly steps.

Considering all the information presented, the IKEA furniture assembly simulation environment [24] emerges as the most suitable choice for study. This environment provides a simulation platform and includes a dataset, making it a comprehensive resource. The inclusion of a large number of furniture models, along with the segmentation of pieces based on the IKEA manual, adds to its significance. Notably, this environment has been utilized in the work of [1] and is well-suited for generating scenarios involving both fully assembled and partially assembled furniture for furniture assembly studies. This thesis centers around a scenario for a furniture assembly with cobots and humans on an assembly line. All 3D datasets in Table 2.1 have assembly scenarios that contain the correct actions. A 3D dataset is essential, encompassing assembly scenarios that depict all possible correct and incorrect actions. 3D dataset was generated using AssembleRL [1].

Some models of IKEA furniture are outdated, leading to unavailability of their manuals. Additionally, there is a requirement for a new 2D dataset to generate adjustable manual data for each furniture, particularly for custom products or other brand furniture lacking manuals. Despite the IKEA ASM dataset meeting the 3D data prerequisites, it falls short in providing intermediate steps necessary for action recognition and tracking on video streaming, as indicated in Table 2.1.

Taking into account the requirements in Table 2.1, there exists a literature gap for a dataset encompassing images resembling manuals, explicit 2D-3D relationships, segmented furniture parts, partial assembly steps, and information regarding the final assembled furniture's 3D shape. Consequently, our dataset serves as a significant contribution to addressing this crucial gap in the existing literature. The IKEA furniture essential for 2D and 3D data collection was sourced from the furniture model library within the IKEA furniture assembly simulation environment [24].

CHAPTER 3

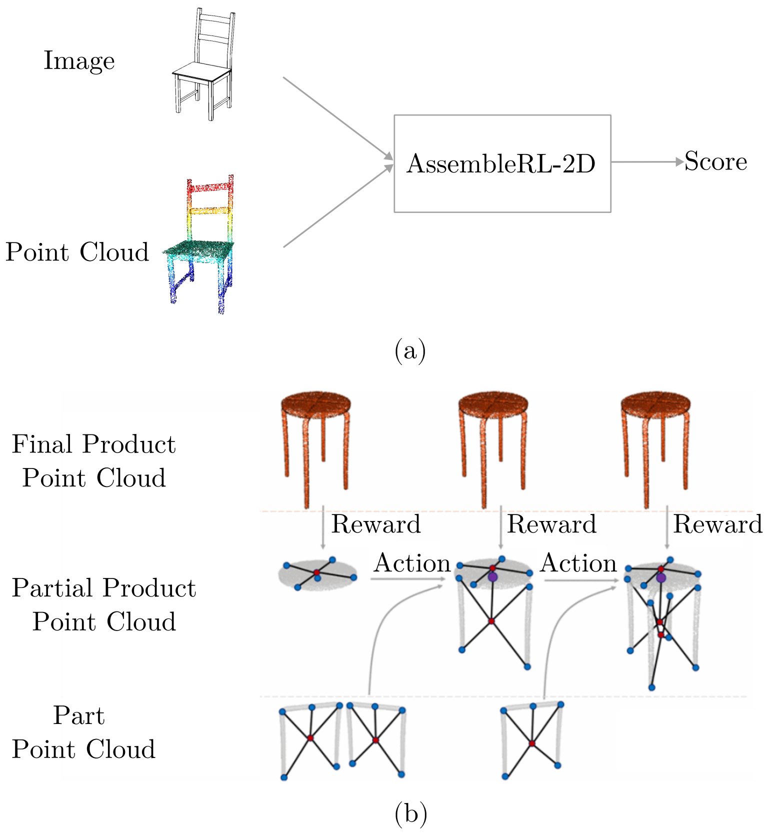
PROPOSED MODEL: ASSEMBLERL-2D

3.1 Overview

In this section, we examine the AssembleRL-2D model, an extension of the AssembleRL model [1] to learn furniture assembly from 2D drawings of furniture. This chapter provides an overview of the AssembleRL-2D and AssembleRL architectures, as depicted in the Figure 3.1. We will delve into the details of these architectures in the subsequent sections. The AssembleRL-2D network needs both 2D and 3D data. For this reason, we had to collect 2D manual data and 3D data consisting of point clouds having furniture assembly steps to create 2D and 3D datasets. This data collection process is also described in this chapter.

3.2 Data Collection

To serve as inputs for AssembleRL-2D, it is imperative to create 2D and 3D datasets. Despite the existence of manuals for assembling IKEA furniture, there might be instances where accessing these manuals is not feasible, or the product in question is outdated and no longer available. Consequently, we generate 2D data that closely resembles the IKEA instruction manuals for the furniture models in our possession. In the context of this assembly scenario, we also generate 3D data, representing a 3D point cloud of the fully assembled furniture.



(a)

(b)

Figure 3.1: (a) AssembleRL-2D establishes a similarity between the 2D image and the furniture assembly status, which is depicted as a 3D point cloud. (b) AssembleRL is a reinforcement learning architecture developed for acquiring correct furniture assembly steps.

3.2.1 2D Data

XML files of furniture models and mesh files of furniture parts were obtained from the IKEA furniture assembly environment [24]. According to the XML file, the target locations of each part to be connected were obtained. We obtain the assembled version of the assembled furniture using AssembleRL [1] study. The fully assembled furniture mesh is saved in STL format. The furniture created is imported into the Blender environment in STL format. Subsequently, essential scene objects and tools are generated using the Blender Python API to create manual-like images.

The maximum dimension in the x, y, and z axes of the imported furniture are determined. Based on this length, the longest dimension of the piece is resized to $0.5m$.

Subsequently, the object is repositioned in the coordinate system at $(0\ m, 0\ m, 0.5\ m)$ beacuse of the object location is based on the object's center of gravity. The object is always as close as possible of the origin with this reloaction values. A material is assigned to the created object to white color the shadow method set to none.

In IKEA manuals, furniture is typically depicted in white on a white background, with the lines of the furniture drawn in black. To replicate this, black lines are created using the Grease Pencil object to emphasize details around the furniture. The brush radius of the Grease Pencil object is set to 1. The object input type is designated as a line art object (LRT_OBJECT) to be added to the active object, i.e., the furniture. Additionally, the value, i.e. use_in_front, True is input to ensure that the line art grease pencil appears in front of everything else. The thickness of the pen is set to 3, and back face culling is employed to showcase all the line details. A time offset of 5 frames is established to fully load this object before moving on to subsequent steps. Similar to the assembly instruction, when capturing images of the object, a plane with a size of $1000\ m$ is incorporated under the furniture to create a white background. The plane color and the background color attached to the world are also configured to white, with shadows turned off. The camera object is situated at $(2\ m, 2\ m, 2\ m)$, a constraint is established to follow the furniture, and the target is set as the furniture. Subsequently, the scene is created, and the camera is added. The desired image format to be generated is set to PNG. Following this setup, the camera is rotated around the object at user-defined angles between $0°$ and $360°$, as shown in Figure 3.2, and the rendering is captured with the assistance of the camera, saving the output to a file. These steps can be examined in Figure 3.3.

In this study, the recorded images were captured at $1°$ intervals. For each furniture in both the train and validation sets, a total of 360 images were generated, each at $1°$ intervals. As for the furniture in the test set, there are 90 images captured at $1°$ degree intervals, specifically featuring only front views of the furniture, similar to those found in manuals.

The post-processes are performed to remove the unnecessary shadows, creating a guide-like image. Figure 3.4 shows the generated and the IKEA manual images. It can be seen that the images obtained are very similar to the images in the IKEA

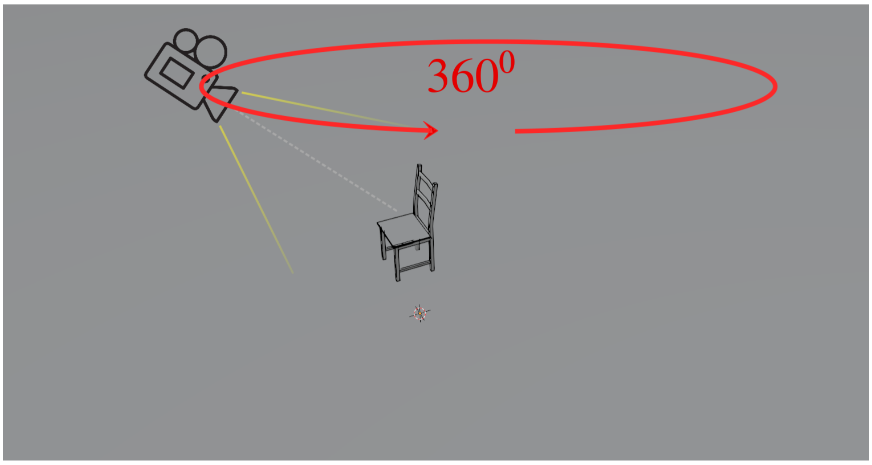


Figure 3.2: The image captured by rotating the camera $360°$ in the counter-clockwise direction to track the object.

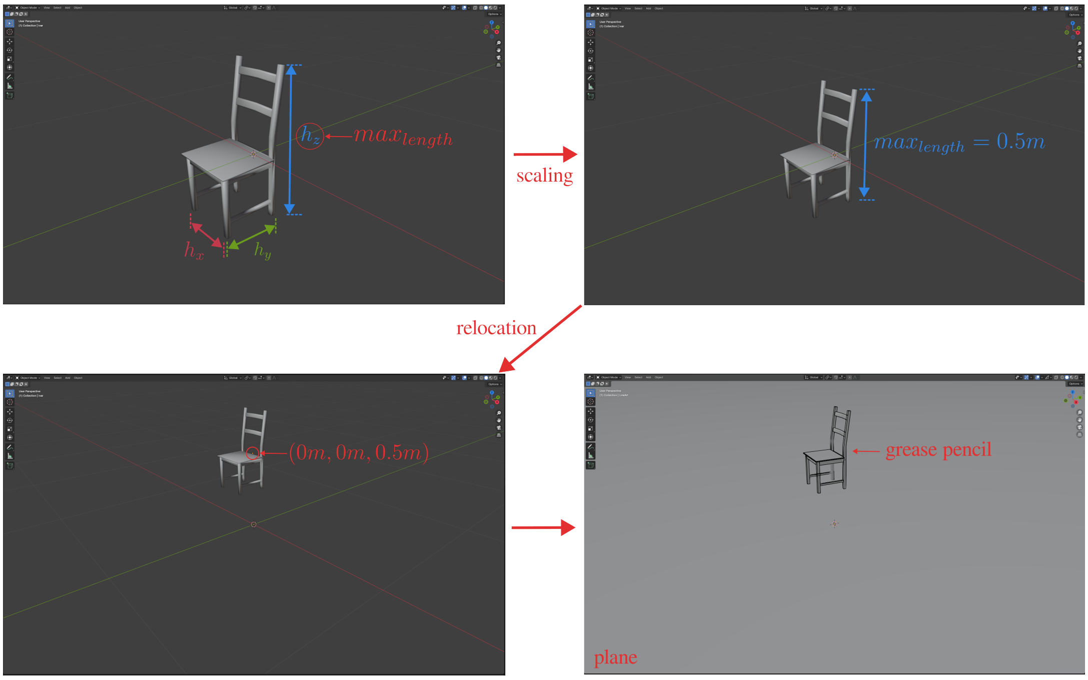
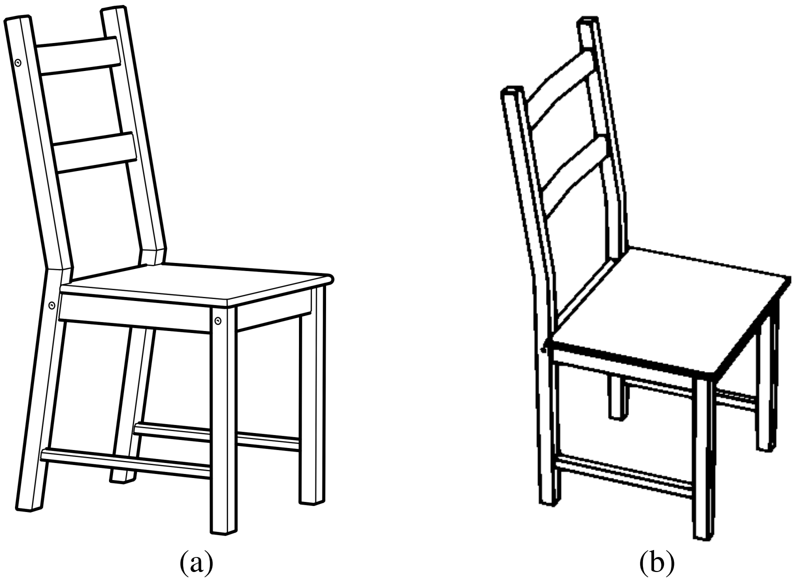


Figure 3.3: The stages and images of the process are given to create a manual-like image in the Blender environment.

manuals.



(a) (b)

Figure 3.4: (a) The original IKEA manual image of the assembled Ivar furniture and (b) a sample image from the generated 2D data are given.

3.2.2 3D Data

3D geometry information can be represented as a mesh, voxel, or point cloud. The voxel data type has an input space around the object, while the mesh data type has a sparse structure, as shown in Figure 3.5. Also, the mesh format only involves the surface information about the object. On the other hand, when we receive data from a real-world object with the help of a sensor, such as a lidar or camera, the raw data we get is similar to a point cloud. In addition, the point cloud can be easily converted to other representations.

The voxels have a regular form, but mesh and point cloud are not. Convolutional architectures generally require regular information like 3D voxel. For this reason, data with mesh or point cloud representations could be transformed into a voxel format and used in the network. This situation causes a more significant need for computational power and also a lot of unnecessary data storage. For all these reasons, it would be a more accurate approach to use the point cloud directly. With the discovery of the PointNet [5] architecture, it has been made possible to use the point cloud in convolutional architectures directly. Hence, we collected our 3D data in point cloud format.

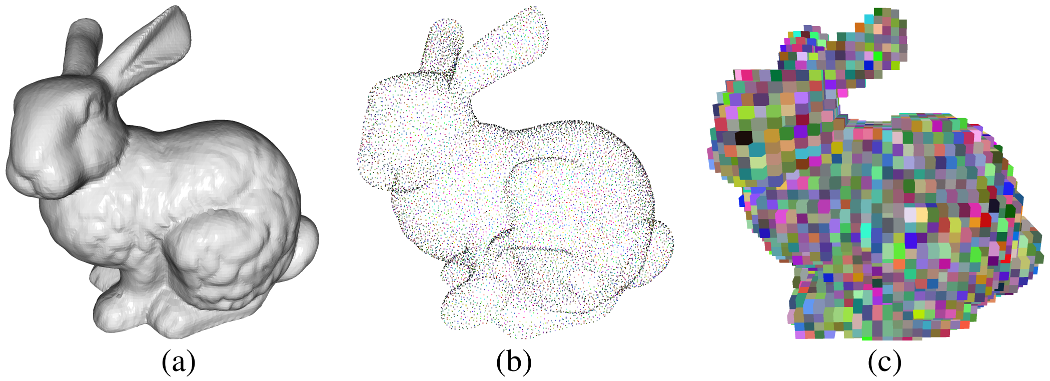


Figure 3.5: Some of the 3D geometric representations: (a) mesh, (b) point cloud, and (c) voxelization.

Within the scope of the KALFA project, the method recommended in [1] is used for 3D data collection. Point clouds of assembly scenarios are created by using 11 pieces of IKEA furniture taken from the IKEA furniture assembly environment. It is assumed that the target poses of the furniture are known. They used P^T to denote the point cloud of fully assembled furniture, P^0 to denote the point cloud of the seed part at the start of assembly, and P^t to denote partially assembled furniture at step t of assembly. The current assembly's point cloud, P^t, graph is rendered by convolutional layers, followed by fully connected layers. The selected action is rewarded by comparing it with the updated assembly (P^{t+1}) target P^T. Deep reinforcement learning was used to find a policy (π) for successful furniture assembly and, more specifically, Proximal Policy Optimization. Graphs are used to code the environment status, and a Graph Convolutional Network calculates probability distribution on actions. A reward function consisting of two measures is used to train the network: incompleteness and incorrectness, as shown in Figure 3.6.

Furniture scenarios encompassing all potential correct and incorrect actions in three dimensions were generated as a result of this study. The creation of the 3D dataset was a collaborative effort conducted in partnership with Özgür Aslan, the author of the AssembleRL [1] study. The approach employed in this study involves generating and capturing mesh models of potential assembly scenarios. Subsequently, these mesh models will be treated as point samples, serving as the basis for creating and utilizing point clouds. With the help of this study, 3D data collection involving negative and

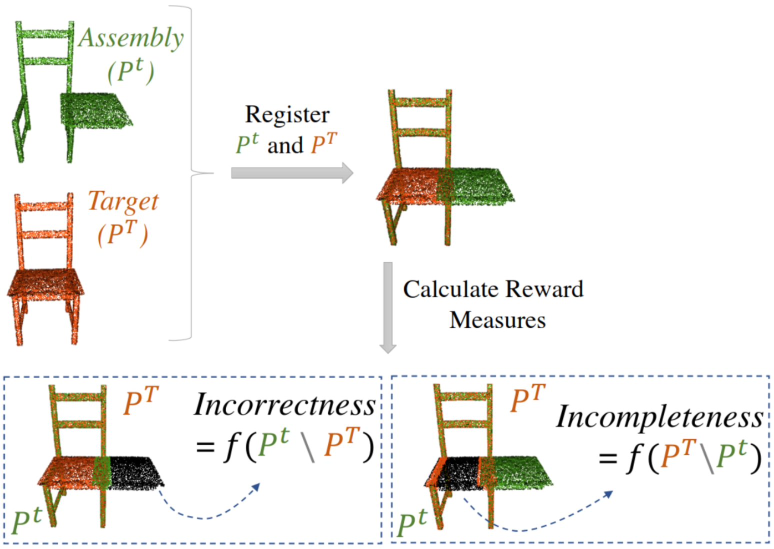


Figure 3.6: Incorrectness and incompleteness measurement with the partial and fully assembled furniture point clouds. [Figure Source: [1]]

positive data was made. A fully assembled furniture is called "positive" 3D data. On the other hand, "negative" 3D data, consisting of incorrectly assembled parts or mated with correct steps but not yet finished, are created, illustrated in Figure 3.7. Position information containing all points (x, y, z) coordinates and their normal vectors are saved. The correct assembly state is measured according to the proximity of the coordinates of all points.

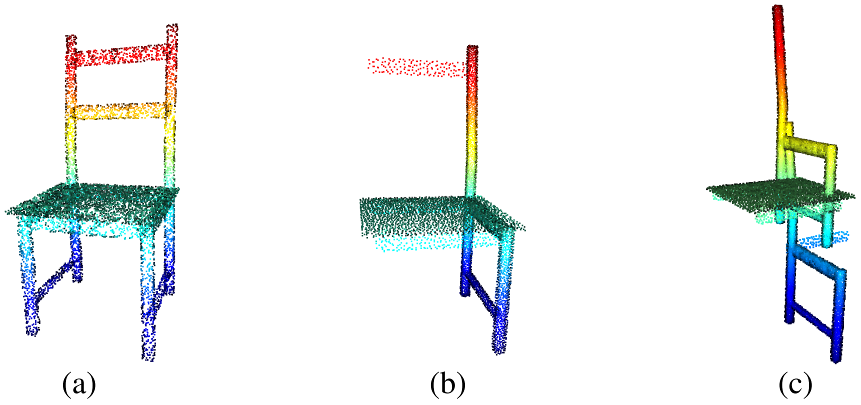


Figure 3.7: (a) The complete fully assembled furniture point cloud creates positive 3D Data. (b) The point cloud that is assembled correctly but has an assembly that is incomplete, and (c) the misassembled furniture point cloud creates negative 3D data.

3.3 A Novel Network 2D-3D Similarity

As mentioned in Section 3.1, 2D and 3D datasets must be created for the network. Although there are manuals for assembling IKEA furniture, we may not always be able to reach these manuals, or we may have an out-of-date product that is no longer available. For this reason, we produce 2D data similar to the assembly manuals of the furniture models we have. In this assembly scenario, we also create 3D data containing a 3D mesh of the assembled furniture.

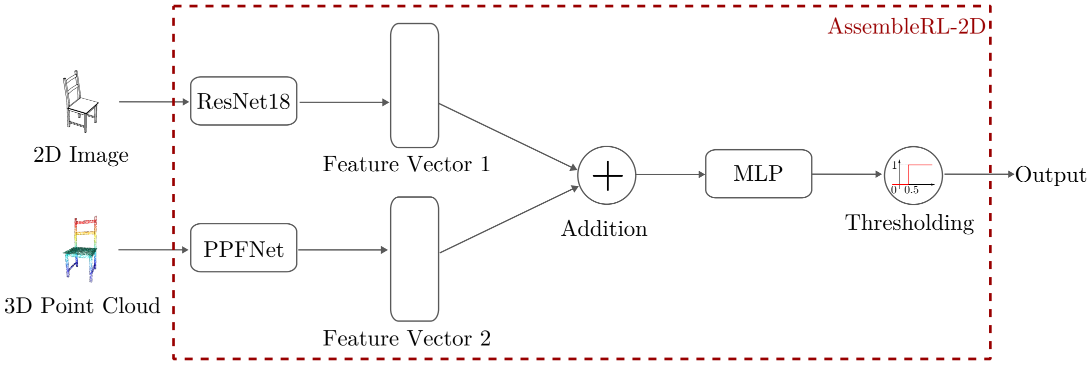


Figure 3.8: The proposed AssembleRL-2D architecture.

In this task, one is expected to learn the assembly plan of the three-dimensional furniture model of the mesh structure by giving a two-dimensional picture. The main problem addressed here is to calculate the similarity between two-dimensional and three-dimensional inputs. The proposed network architecture is given in 3.8. The first feature vectors to the AssembleRL-2D network structure will be obtained by training the residual neural network, ResNet18, in the two-dimensional image dataset. The other feature vector will be created from training with PPFNet using three-dimensional point clouds. ResNet and PPFNet outputs will be trained to be the same size. These created feature vectors will enter the proposed network structure separately and pass through the same parameters and stages, and similarity learning in two dimensions and three dimensions will be provided.

3.3.1 Input Features and Outputs

The proposed AssembleRL-2D structure takes a 2D manual image and a 3D point cloud indicating the assembly state of the furniture as an input pair. The 2D manual input is a 3-channel RGB image, which we use as a $(3 \times 224 \times 224)$ shape. The 3D input contains the mesh models of the furnitures. As these models are matched as pair inputs, they undergo a conversion process into point cloud format by being transformed into point samples. The coordinates of the point clouds (x, y, z) belonging to the furniture's completed or incomplete assembly scenarios. This input also carries the information of the normal vectors of these points. Two feature vectors are obtained by passing 2D and 3D inputs separately through ResNet18 and structures. The resulting 512-dimensional feature vectors are collected. This resulting vector is the input of an MLP. The network output passes through the thresholding operation. The network scores 0 or 1, depending on whether it is above or below the threshold value of 0.5.

3.3.2 Loss Function

The network takes the 2D image and 3D cloud as input pairs and calculates a score as 0 or 1. In other words, this structure makes a binary classification. The 2D and 3D data inputs are converted to vectors and fed into the MLP layer, and the output reduces to one-dimensional. This output goes through the Sigmoid function:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{3.1}$$

A decision boundary needs to be determined with the value range of the function [0,1]. Therefore, a 0.5 value is selected as thresholding, and a decision is made on whether the score of the network is above or below;

$$p \geq 0.5 \longrightarrow class = 1,$$
$$p < 0.5 \longrightarrow class = 0.$$

An error measurement is made between this value calculated during the training and the final score value, indicating that 2D and 3D data are completed correctly and on the same furniture classes.

3.3.2.1 Binary Cross Entropy Loss

Binary Cross Entropy Loss is used in this classification since there is no effect between classes in the decision process. Since the binary classification problem is considered, the class number is $C = 2$, and the Binary Cross Entropy Loss formula is given in Eqn. 3.2,

$$BCE = \sum_{i=1}^{C} t_i \log(f(s_i)), \tag{3.2}$$

where f is the Sigmoid function, t_i and s_i are the ground-truth label for the correct furniture class and the score is taken from the network output. For binary classification, $t_2 = 1 - t_1$ and $s_2 = 1 - s_1$ are written for the C_2 class. Then Eqn. 3.2 can be rewritten as follows:

$$BCE = \begin{cases} -\log(f(s_1)) & \text{if } t_1 = 1, \\ -\log(1 - f(s_1)) & \text{if } t_1 = 0. \end{cases} \tag{3.3}$$

3.3.2.2 Focal Loss

Focal Loss, proposed by Lin *et al.* [31], is an alternative loss calculation method fundamentally rooted in the structure of Cross Entropy Loss. It addresses a specific issue relevant to our context. Focal Loss is employed to mitigate the imbalance between classes in datasets where classification is not evenly distributed. Essentially, it assigns different weights to the contribution of each class to the loss, aiming to create a more balanced class distribution. As it utilizes the Sigmoid activation function, it can be conceptualized as a form of Binary Cross Entropy Loss:

$$FL = -\sum_{i=1}^{C} (1 - s_i)^{\gamma} t_i \log(s_i), \tag{3.4}$$

Focal Loss is equal to the Binary Cross Entropy Loss when $\gamma = 0$ and $C = 2$. γ is the focusing parameter. The tunable focusing parameter range is $\gamma >= 0$. To address the issue of class imbalance, a weighting factor $\alpha \in [0, 1]$ is typically employed, where the frequencies of positive and negative samples are specified. However, in our case, we have the flexibility to adjust the number of positive and negative samples provided to the network directly in our code. Therefore, we prefer to use it without setting any specific α hyperparameter. Focal loss was chosen for this study due to its effective handling of class imbalance situations, which aligns well with the characteristics of our case.

3.3.3 Network Layers

As can be seen in Figure 3.8, feature vectors of 2D Data with pre-trained ResNet18 and 3D Data with pre-trained PointNet++ are calculated from the input pair. The score is calculated by summing the two feature vectors obtained and passing them through an MLP layer. This structure is called the AssembleRL-2D.

3.3.3.1 2D Feature Extractor

Residual Network architecture [4] uses redundant blocks with multiple layers to reduce training errors. ResNet18, one of the Residual Network architectures, is now used to extract the 2D feature vector from the input pairs like a 2D manual image. For this, the network is first pre-trained with the ImageNet dataset [32]. This way, the network is used in this problem by tuning approximately 11.7M parameters.

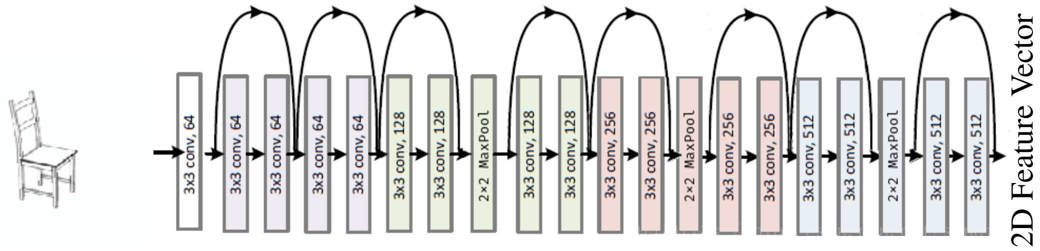


Figure 3.9: The ResNet18 architecture used as feature extractor by removing the average pooling layer. [Part of the figure from: [4]]

The average pooling layer, of the pretrained ResNet18 is removed. In this way, the feature vector size of the 2D data becomes 512. The modified ResNet18 structure can be examined in Figure 3.9.

3.3.3.2 3D Feature Extractor

The study initially employed PointNet++ for 3D feature extraction. However, inconsistencies were noted in the results when the same point cloud transformed. Consequently, the decision was made to transition to PPFNet. Although PPFNet is based on the PointNet++ architecture, it addresses the variability issue by incorporating the normal information of points in its feature extraction process.

PointNet++. We obtain feature vectors from 3D Data using PointNet++ [6]. PointNet [5] was the first method to use the point cloud directly, while PointNet++ is an advanced version using the PointNet architecture where neighboring points are also included as local features. It calculates the features of 3D furniture assembly point clouds pre-trained on ModelNet10 and ModelNet40 [33].

Furthermore, PointConv [34] is the Deep Convolution Network structure in which the convolution operation used in image convolution is extended for point cloud. MLP is used in Convolution filters. PointNet and PointNet++ also use PointConv.

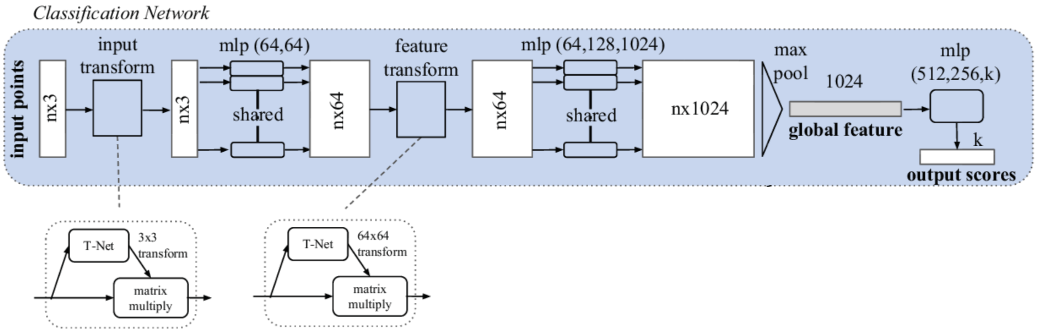


Figure 3.10: The PointNet classification network. [Figure Source: [5]]

The PointNet++ architecture contains PointNet layers, as shown in Figure 3.11. The PointNet layer, which belongs to PointNet++, has a 3-layer MLP layer, in Figure 3.10. A feature vector output of 512 is obtained by subtracting the last two layers here.

PPFNet. PointNet++ is not a rotation-independent construct. When the same point clouds are rotated, and their transformed forms are given to the same network, the fea-

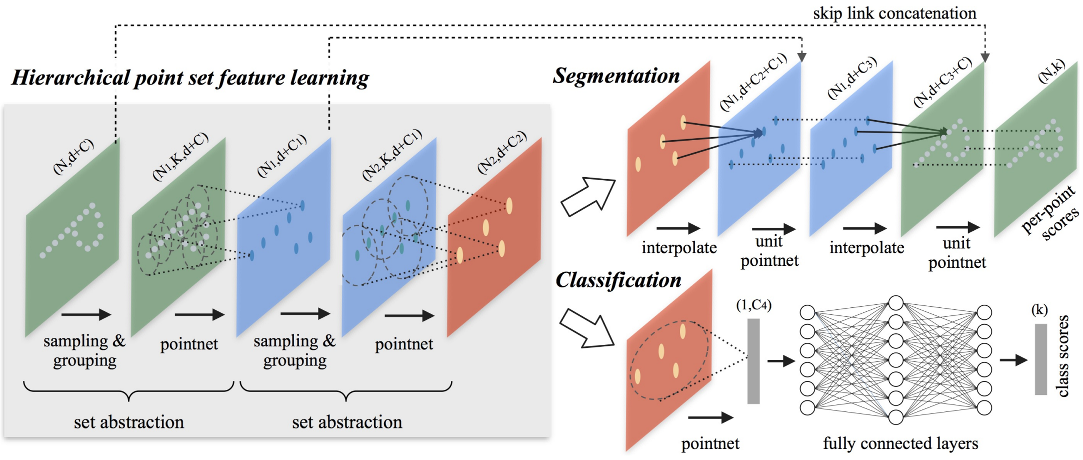


Figure 3.11: The PointNet++ architecture for point cloud classification and segmentation which uses based on PointNet. [Figure Source: [6]]

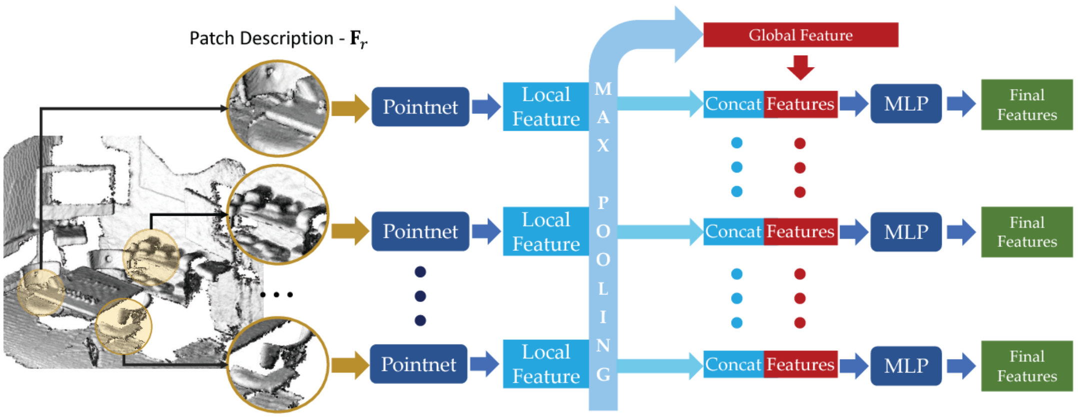


Figure 3.12: The PPFNet architecture which is consists of multiple PointNet [7].Max pooling aggregation and the output back integration to local feature is involved to encompass the global context. [Figure Source: [7]]

ture vector values obtained change. For this reason, the same study is also repeated by using the rotation invariant PPFNet (Point Pair Feature Network) [7] instead of the PointNet++ structure in the AssembleRL-2D architecture. While PointNet++ uses only the position information of the points, PPFNet also uses the information of the normal vectors of the points in addition to this information. Like PointNet++, PPFNet also includes the 3-layer MLP, as shown in Figure 3.12. The last two layers of the MLP layer, which is in the form of $[1024, 512, 256, num\_class]$ (the last layer, num_class, is 10 or 40 depending on the pretrained model, ModelNet10 or

ModelNet40), are removed and the PPFNet output is used as 512.

3.3.3.3 Multi-layer Perceptron (MLP)

The feature vectors of the 2D and 3D data, which have an equal size of 512 obtained are collected. The process involves the addition of these two feature vectors, resulting in a 512 dimensional vector. This vector serves as the initial input for the 3-layer MLP, as shown in Figure 3.13, where the hidden layer's input size is set at 128, and the MLP's output size is configured to be 1. This output is mapped through the Sigmoid between [0,1], then after thresholding, the network score 0 or 1 is output.

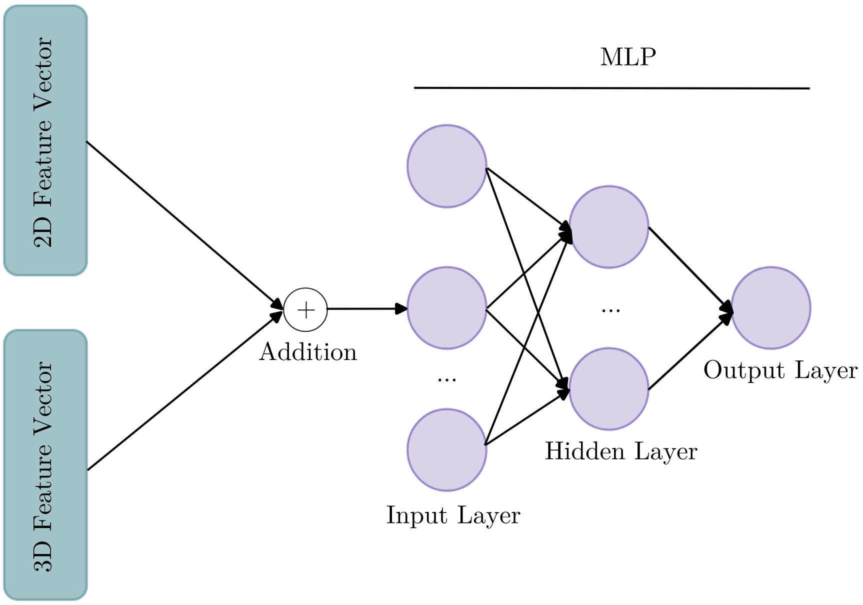


Figure 3.13: The added feature vectors are passed through three-layer MLP.

3.4 Assemble Learning with 2D-3D Similarity Network

We employed the AssembleRL-2D to learn the furniture assembly scenario under the guidance of Özgür Aslan in a collaborative effort based on the AssembleRL study. Assuming the availability of point cloud information for the final furniture assembly, the AssembleRL structure compared this ultimate model with the partial assembly resulting from each action it executed. However, we substitute AssembleRL-2D for

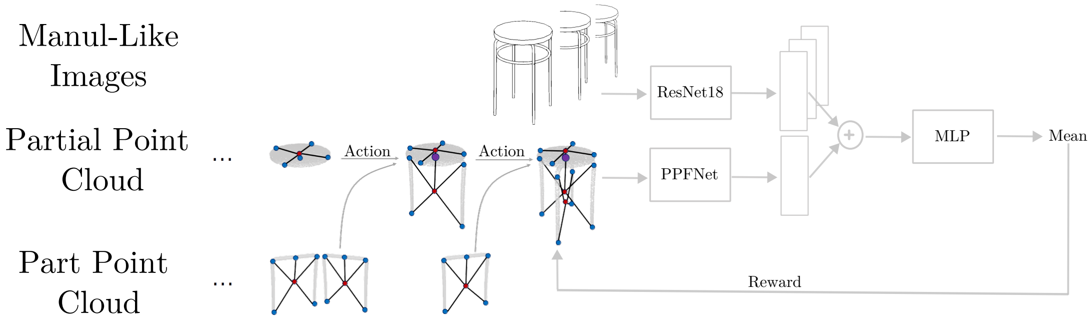


Figure 3.14: The integration of the AssembleRL and AssembleRL-2D demonstrating with agne chair assembly steps.

the final point cloud information in this work.

The dataset is split train and testset, undergoes training in AssembleRL-2D and is subsequently employed when the AssembleRL takes deterministic actions. The agent is iteratively updated by collecting data from 512 actions within the AssembleRL environment. Upon episode completion, the outcome of the taken action is recorded. Using this recorded model, 2D manual images are passed through AssembleRL-2D, and the network result is considered a mean without employing any thresholding, serving as the reward signal, outlined in Figure 3.14.

CHAPTER 4

EXPERIMENTAL SETUP

Within this section, we will scrutinize the datasets employed in the AssembleRL-2D similarity network architecture and elucidate their utilization. We will discuss the datasets selected for pretraining feature extractors. Subsequently, we will explore the datasets we generated and the methodology employed to utilize them as input pairs to feed our AssembleRL-2D network. Lastly, we will delve into the parameters applied in the context of this network architecture.

4.1 Datasets

The model performs binary classification by utilizing a 2D manual image of the furniture to determine whether the furniture has been correctly assembled or not. The objective is to leverage the similarity in 2D and 3D features by utilizing datasets composed of manual images and mesh models of assembled IKEA furniture. Therefore, it is crucial to extract accurate features. To achieve this, employing pre-trained networks as 2D and 3D feature extractors will yield improved results. Furthermore, the networks, ResNet18 and PPFNet, used in the AssembleRL-2D were pre-trained on ImageNet 1K and ModelNet datasets, respectively.

In our experiments, the generated datasets were used by separating them as train, validation and test sets. Due to the inadequacy of furniture and an imbalanced class distribution, as shown in Table 4.1 in the AssembleRL objects [1], the dataset was augmented by incorporating furniture from the [24] study. The types of furniture utilized in this study, along with their distribution, are detailed in Table 4.2 below.

Table 4.1: The furniture distribution in AssembleRL [1].

Chair	Table	Shelf	Tv Unit
Agne	Klubbo	Liden	Tvunit
Bernhard	Lack	Sivar	
Bertil	Mikael		
Ivar			
Swivel			

Table 4.2: The dataset distribution of the proposed model. Furniture models are listed according to their furniture class. The number of pieces the furniture has is shown in parentheses. Trainset includes 9 furniture in total, validationset 3 and testset 4.

	Bookcase	Chair	Table
Trainset	Agerum (10)	Agam (10)	Hemnes (11)
	Besta (9)	Agne (4)	Lack (5)
	Expedit (7)	Bernhard (3)	
		Ingolf (5)	
Validationset	Billy (11)	Ivar (5)	Klubbo (5)
Testset	Hensvik (10)	Balser (8)	Benno (8)
			Dalom (5)

4.1.1 Datasets for Pretraining Feature Extractors

The ResNet18, which serves as the 2D feature extractor, undergoes pretraining on ImageNet, while the PPFNet, which functions as the 3D feature extractor, is pretrained on ModelNet40. AssembleRL employs an input pair derived from the 2D and 3D datasets we generate, as illustrated in Figure 4.1.

4.1.1.1 ImageNet Dataset

ImageNet dataset [32] contains 1000 classes of images. ImageNet relies on a WordNet [35] that groups words according to a hierarchical structure. The nouns, adjec-

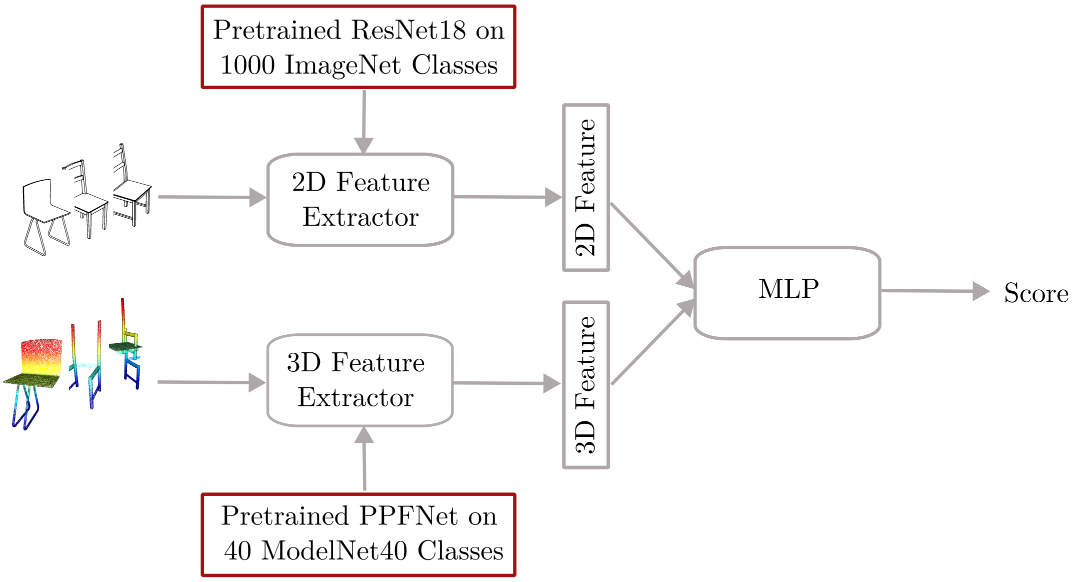

Pretrained ResNet18 on 1000 ImageNet Classes
2D Feature Extractor
2D Feature
MLP
Score
3D Feature Extractor
3D Feature
Pretrained PPFNet on 40 ModelNet40 Classes

Figure 4.1: The training pipeline of the proposed AssembleRL-2D similarity network.

tives, adverbs, and verbs cluster as cognitive synonyms, called synset, in WordNet. This structure is likened to a tree, so the synset words are at the same level, as shown in Figure 4.2. Similarly, ImageNet includes 12 subtrees: mammal, bird, fish, reptile, amphibian, vehicle, furniture, musical, instrument, geological formation, tool, flower, and fruit.

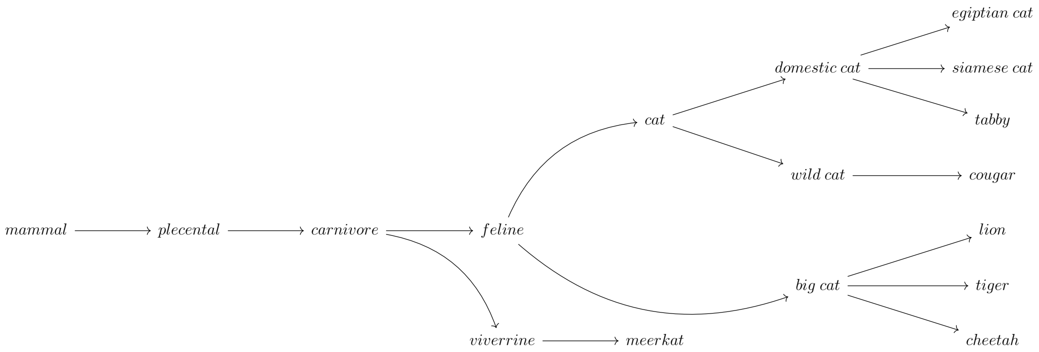


Figure 4.2: A snippet from the ImageNet dataset to illustrate the hierarchical structure. Domestic and wild cats are called synset, and they are on the same level. The hyponyms of domestic cats are Egyptian cat, Siamese cat, and tabby, while the hypernym is the cat. In this way, we can achieve the mammal level, one of the 12 subtrees of ImageNet, from a cat species according to their hypernyms.

The ResNet18 model is used with obtained parameters trained on the ImageNet dataset.

The accuracy values of the model, which is trained on ImageNet-1K in torchvision [36], were 69.758% for the highest probability of top class score and 89.078% for the top 5 probability.

4.1.1.2 ModelNet10 Dataset

ModelNet10 dataset [33] consists of 3D CAD models of objects. The CAD models in the dataset are in OFF, which kept the surface geometries in ASCII format. ModelNet10 contains more than 150K 3D CAD models belonging to 660 object categories. In this study, we pretrain 3D feature extractor using the ModelNet10 dataset, which has ten classes: bathtub, bed, chair, desk, dresser, monitor, nightstand, sofa, table, and toilet, are given in Figure 4.3.

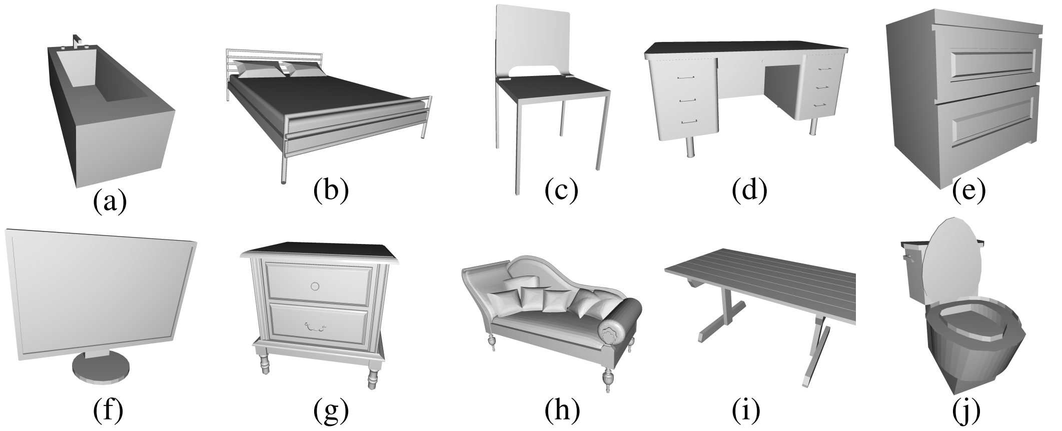


Figure 4.3: Samples of the ModelNet10 dataset classes are given: (a) bathtub, (b) bed, (c) chair, (d) desk, (e) dresser, (f) monitor, (g) nightstand, (h) sofa, (i) table, and (j) toilet.

4.1.1.3 ModelNet40 Dataset

ModelNet40 is a larger dataset than ModelNet10, which contains 40 object classes, shown in Table 4.3. Since it contains more objects, it is planned to be used in pretraining to achive better results.

Table 4.3: The class distribution of ModelNet40 dataset.

Class Name	Samples	Class Name	Samples	Class Name	Samples	Class Name	Samples
airplane	726	cup	99	laptop	169	sofa	780
bathtub	156	curtain	158	mantel	384	stairs	144
bed	615	desk	286	monitor	565	stool	110
bench	193	door	129	night stand	286	table	492
bookshelf	672	dresser	286	person	108	tent	183
bottle	435	plower pot	169	piano	331	toilet	444
bowl	84	glass box	271	plant	340	tv stand	367
car	297	guitar	255	radio	124	vase	575
chair	989	keyboard	165	range hood	215	wardrobe	107
cone	187	lamp	144	sink	148	xbox	12

4.1.2 2D Dataset

To facilitate the assembly task, certain furniture items within the [24] environment underwent simplification. Consequently, a selection process was implemented, aligning the number of pieces in the furniture with the original IKEA manuals. To achieve a more balanced class distinction, three primary categories were chosen: table, bookcase, and chair. The resultant images closely resemble those found in the original IKEA manual. Figures 4.4, 4.5 and 4.6 provide a comparative examination of the original and generated images for furniture belonging to the table, bookcase, and chair classes, respectively.

4.1.3 3D Dataset

The 3D data is acquired through the procedures outlined in Chapter 3.2.2. For furniture generated as a correctly assembled mesh model, it is categorized as positive. Positive mesh has one model for every furniture. Models where furniture parts are assembled accurately, but the overall assembly is incomplete or incorrect, are designated as negative. The number of negative samples depends on the number of furniture pieces, as shown in Table 4.4. The resulting 3D dataset, encompassing both positive and negative examples, comprises the mesh formats of these objects. This dataset is partitioned into three sets: training, validation, and test sets.

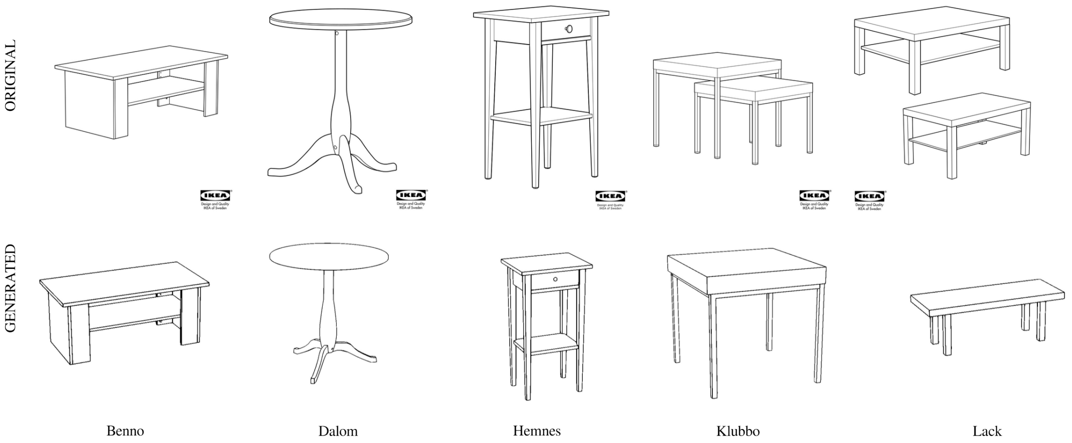


Figure 4.4: The original IKEA manual screenshots of furniture belonging to the table class are above; below are the sample images of the produced manual-like dataset.

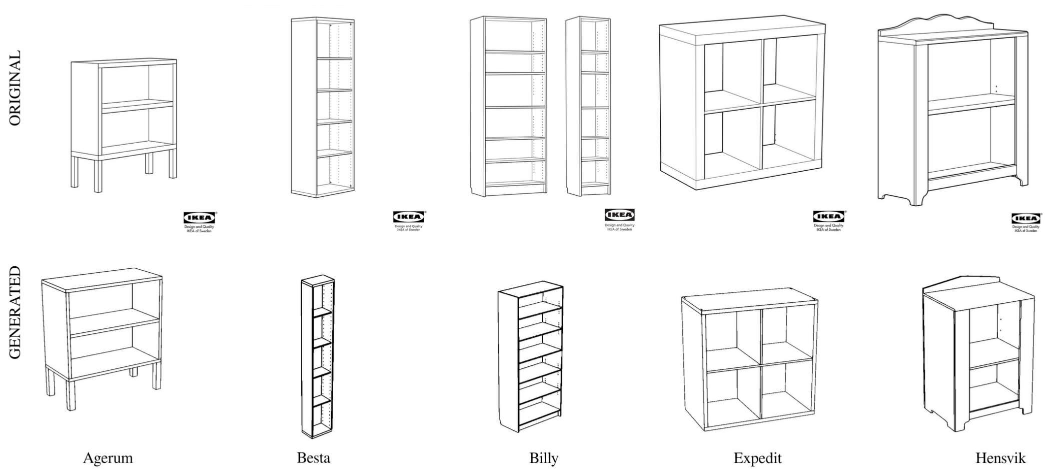


Figure 4.5: The original IKEA manual screenshots of furniture belonging to the bookcase class are above; below are the sample images of the produced manual-like dataset.

4.2 Paired Input Generation

The images of the 2D dataset are made suitable for the ResNet18 input form. First, the 2D manual images are resized to 256×256 pixels, then cropped to the center at 224×224 pixels. Some random transformations such as rotation, scaling, shearing, and flipping are applied to the image. Finally, it is converted to tensor format and normalized. These operations are done using the torchvision package.

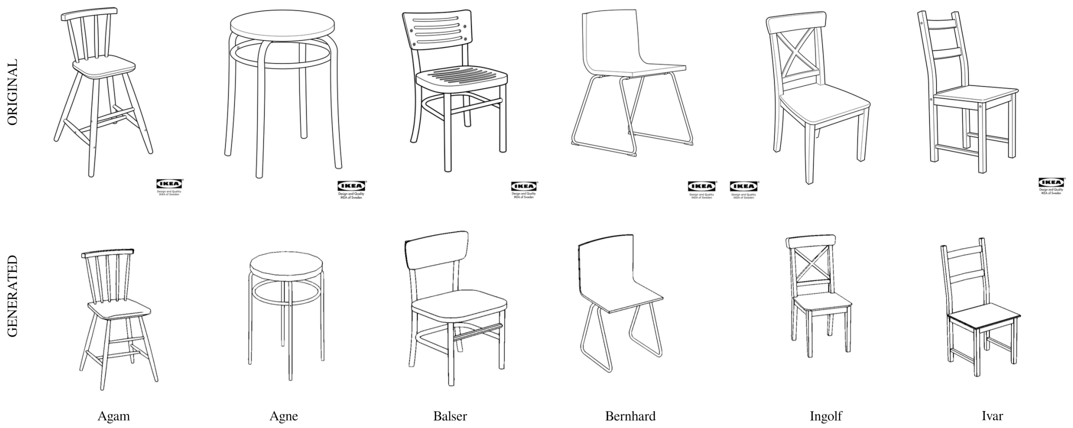


Figure 4.6: The original IKEA manual screenshots of furniture belonging to the chair class are above; below are the sample images of the produced manual-like dataset.

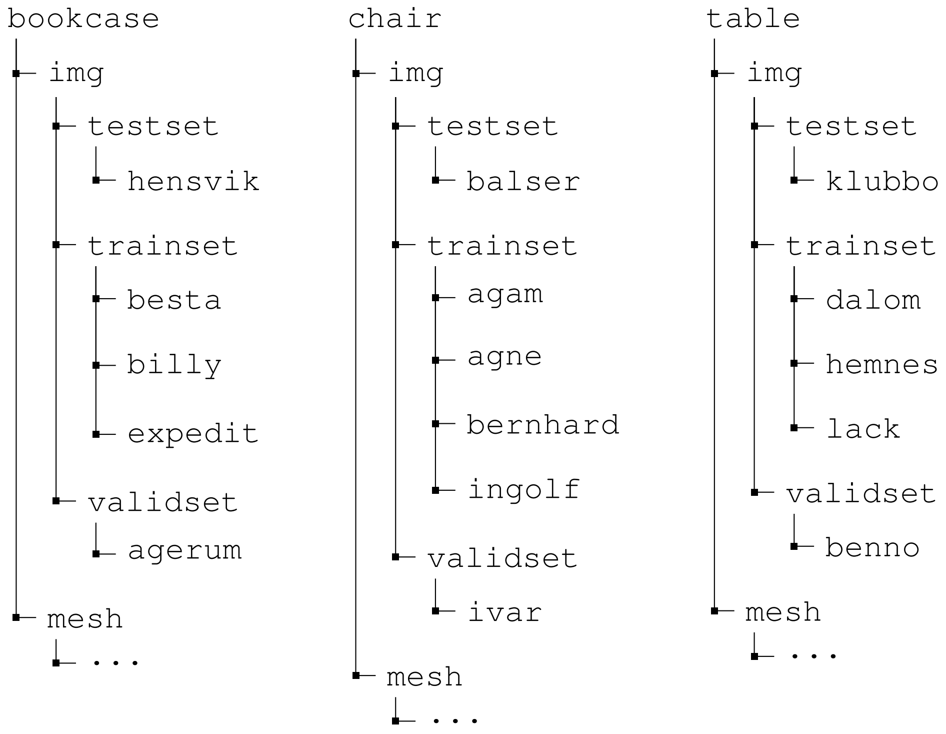


Figure 4.7: The direction trees for bookcase, chair and table classes

The 2D dataset images serve as inputs for ResNet18. These images are loaded using the DatasetFolder dataloader from the torchvision package, utilizing the PIL loader function. The file structure adheres to the displayed Figures 4.7. The AssembleRL-2D structure is trained by pairing models belonging to the same furniture class. The dataset loader specifies the furniture class, i.e., chair, bookcase, or table, when loading

Table 4.4: The variation of the number of negative samples according to the number of parts of the furniture models.

	Bookcase	Part Number	Number of Negative Samples
Trainset	Besta	9	1314
	Billy	11	2061
	Expedit	7	2034
Validationset	Agerum	10	2079
Testset	Hensvik	10	2000

	Chair	Part Number	Number of Negative Samples
Trainset	Agam	10	2104
	Agne	4	1680
	Bernhard	3	7
	Ingolf	5	1912
Validationset	Ivar	5	1913
Testset	Balser	8	2115

	Table	Part Number	Number of Negative Samples
Trainset	Dalom	5	194
	Hemnes	11	1017
	Lack	5	130
Validationset	Benno	8	1062
Testset	Klubbo	5	188

the dataset for training, validation, or testing. The torchvision dataset functions are applied to create a dataset with image transforms. Initially, the 2D manual images are resized to 256×256 pixels and then center-cropped to 224×224 pixels. Image transforms include random scaling between $(0.5, 1)$, 10 random shear, and a fill value of 255 to ensure a white background. Finally, the image is converted to tensor format and normalized, with mean values of $[0.485, 0.456, 0.406]$ and a standard deviation of $[0.229, 0.224, 0.225]$.

The models in the 3D dataset are categorized as positive or negative. Positive models consist of one for each furniture model, while negative data varies based on the furniture piece. Multi-piece furniture entails more negative patterns. The 3D dataset produces a tuple list containing the necessary mesh information—position, face, and the number of furniture pieces (n). The furniture model is initially normalized using this rescaled mesh information, which is then added to the positive mesh library. Negative data undergoes a similar process, if the number of negative samples greater than 1000, only 1000 random samples are taken from the 3D data due to its dependence on the number of parts. The first n elements, representing furniture pieces, are retained, and the remaining $1000 - n$ samples are randomly selected and added to the negative mesh library.

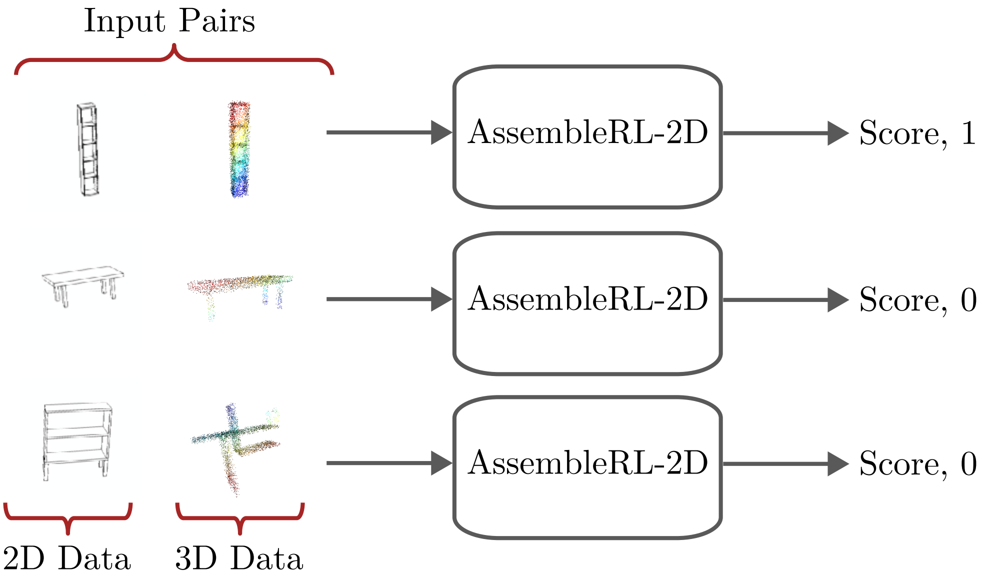


Figure 4.8: The overview of the proposed solution. AssembleRL-2D learns the similarity (conformity) score between a 2D drawing and a point cloud representing the current state of the assembly.

The pairing process involves randomly selecting a furniture model from the 2D dataset, followed by determining a random image among the images of this model. When an item is called from the dataset, the 2D data is matched with positive or negative 3D data of the same class at a rate of $1/2$. If the dataset element pair comes from a 2D image and matches the positive 3D data, the target score is set to 1. The input pair generated serves as the input for AssembleRL-2D, as illustrated in Figure 4.8. If the dataset element pair comes from a 2D image and matches the positive 3D data, the target score is set to 1, otherwise 0. The loss function compares the target score with the network's predicted score.

4.3 Implementation and Training Details

In Chapter 5, the experiments outlined in Table 4.5 will be conducted. The optimizer of choice is AdamW [37]. Learning rates were individually explored for each network, as detailed in Chapter 5.3. The selected loss function is Sigmoid Focal [31], where the weighting factor was not utilized in our study, and the ignored value was set to -1. After investigating the focusing parameter in Chapter 5.3, the optimal value for each class was determined as 2. In the same section, we established the 3-layer MLP and the input size of the hidden layer as 128. The results for AssembleRL, as presented in Chapter 5.4, were obtained over 20 epochs. The global seed value for all these experiments, excluding Chapter 5.5, was set at 8.

Table 4.5: The hyperparameters used in AssembleRL network

Hyperparameter	Selection
Optimizer	AdamW
Learning Rates:	
ResNet18	0.0001
PPFNet	0.001
MLP	0.0001
Loss Function	Sigmoid Focal Loss
Weighting Factor, α	-1 (ignored)
Focusing Parameter, γ	2
MLP Number of Layers	3
MLP Hidden Layer Input Size	128
Number of Epochs	20
Seeds	8

CHAPTER 5

EXPERIMENTS AND RESULTS

In Chapter 3, we introduced the network architecture of AssembleRL-2D, which is proposed in this thesis, and provided insights into the creation of the 2D and 3D datasets produced. Chapter 4 elucidated how we integrate datasets into our architecture and presented implementation details. This section focuses on the experiments conducted with AssembleRL-2D. Initially, we present the outcomes of the study conducted for dataset selection for the 3D feature extractor. Subsequently, we provide the research results for hyperparameters. Another experiment encompasses the results of our AssembleRL-2D architecture across three classes. Finally, we present the outcomes of the furniture assembly scenario obtained by combining the AssembleRL and AssembleRL-2D studies. These experiments were conducted using the PyTorch library [38]. Our training process done with NVIDIA GeForce RTX 3060Ti GPU.

5.1 Experiments and Architecture Details

The proposed AssembleRL-2D architecture is composed of ResNet18 as a 2D feature extractor, PPFNet as the 3D feature extractor, and an MLP layer that aggregates and processes the acquired features. The AssembleRL-2D structure undergoes separate training procedures for three furniture classes. The model is evaluated on previously unseen furniture models not included in the training and validation sets, ensuring that it encounters new and unfamiliar data during testing.

AssembleRL-2D architecture consists of ResNet18, PPFNet and MLP layers. PPFNet and MLP use ReLu as the layer activation function. The outputs of ResNet18 and PPFNet are 512-dimensional feature vectors. These are collected and passed through the MLP layer. MLP consists of 3 layers which have the input size is 512, and the output size is 1. AdamW was used as an optimizer because it can converge better and generalize [37]. Parameter update is performed as follows:

$$\theta_{t+1,i} = \theta_{t,i} - \eta \left(\frac{1}{\sqrt{\hat{v}_t + \epsilon} \times \hat{m}_t} + w_{t,i}\theta_{t,i} \right), \tag{5.1}$$

where w_t is the rate of the weight decay at time t, θ stands for parameters.

5.2 Experiment 1: Quantitative Analysis of Assembly

The networks employed as 2D and 3D feature extractors are pretrained models. The ResNet18 architecture, pretrained with the ImageNet dataset, is utilized directly from the torchvision package. Initially, PointNet++ was considered as the 3D feature extractor, but due to its limitations as mentioned in Chapter 3, PPFNet was chosen as a more accurate alternative. PPFNet was trained with both ModelNet10 and ModelNet40 datasets, and the results were compared. For training on the ModelNet dataset, random rotations between $[-180, 180]$ and point sampling were applied. The training was conducted for 200 epochs with a batch size of 32, utilizing the Adam optimizer. The learning rate was set to 0.001, and the learning rate annealing.

To better comprehend this, the outcomes of the AssembleRL-2D Network trained using 3D feature extractors pretrained on both ModelNet10 and ModelNet40 during 200 epochs are presented. While both networks, trained on ModelNet10, have accuracy values of approximately 85% and above, PPFNet's loss value is closer to 0, as shown in Figure 5.1. For ModelNet40, although the accuracy value decreases for both, PPFNet converges to 80% faster, in Figure 5.2. The model, pretrained with PPFNet using the ModelNet40 dataset as a 3D feature extractor, exhibits lower loss and higher accuracy values rather than using PointNet++. In the evaluation of training results between ModelNet10 and ModelNet40, ModelNet40 was selected despite the fact that PPFNet exhibited lower loss values and higher accuracy. This decision was based on the consideration that pretraining with ModelNet40, which contains more diverse class examples, would likely enhance the performance of AssembleRL-2D.

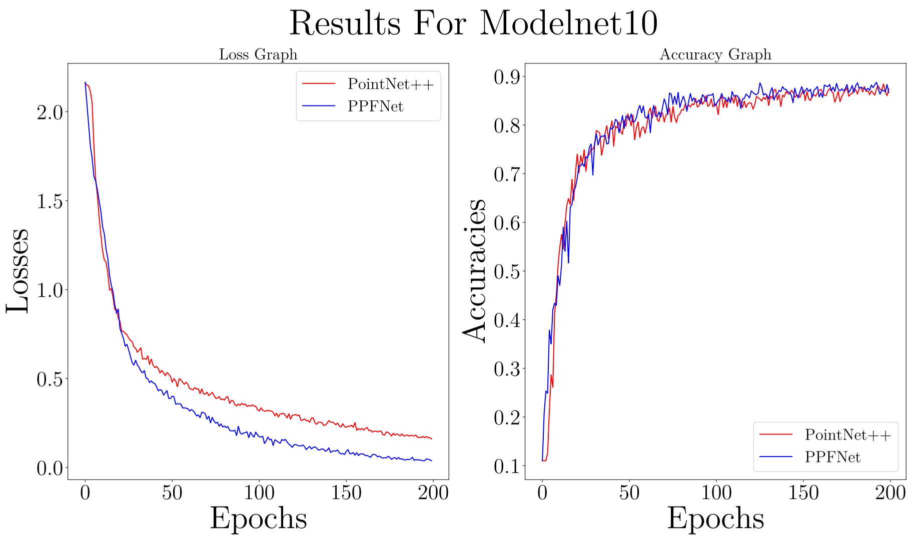


Figure 5.1: Results of PointNet++ vs PPFNet trained on ModelNet10. Train loss is shown at the left, and the accuracy graph is displayed at the right.

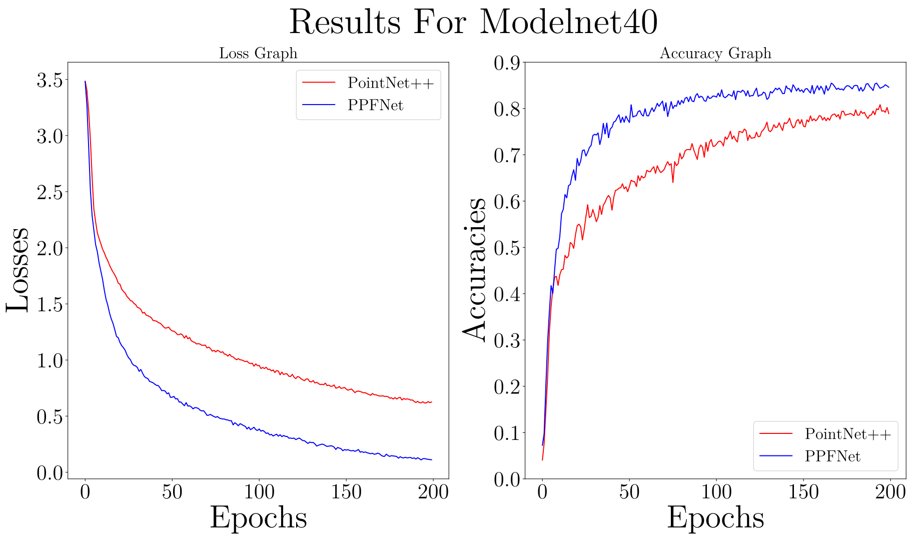


Figure 5.2: Results of PointNet++ vs PPFNet trained on ModelNet40. Train loss is shown at the left, and the accuracy graph is displayed at the right.

5.3 Experiment 2: Ablation and Hyperparameter Analyses

The AssembleRL-2D architecture underwent separate training for the table, bookcase, and chair classes. The learning rate annealing was implemented through scheduling, reducing the learning rate in each epoch by a fixed value determined based on the total number of epochs. The epoch number starts from 1. The learning rate annealing formulation is:

$$\textit{learning rate} = \left(1 - \frac{(\textit{epoch} - 1)}{\textit{total number of epochs}}\right) \times \textit{initial learning rate}. \tag{5.2}$$

The learning rate setting was conducted by varying the learning rate values independently for ResNet18, PPFNet, and MLP layers for chair class. A total of 27 combinations were tested for each network, utilizing learning rate values of 0.0001, 0.001, and 0.0001, respectively. The results of 10 epochs AssembleRL-2D training were analyzed to determine the optimal learning rate values. In this investigation, the selected learning rate values for ResNet18, PPFNet, and MLP were determined to be 0.0001, 0.001, and 0.0001, respectively. The loss and the accuracy graph is represented in Figure 5.3 for obtained learning rates.

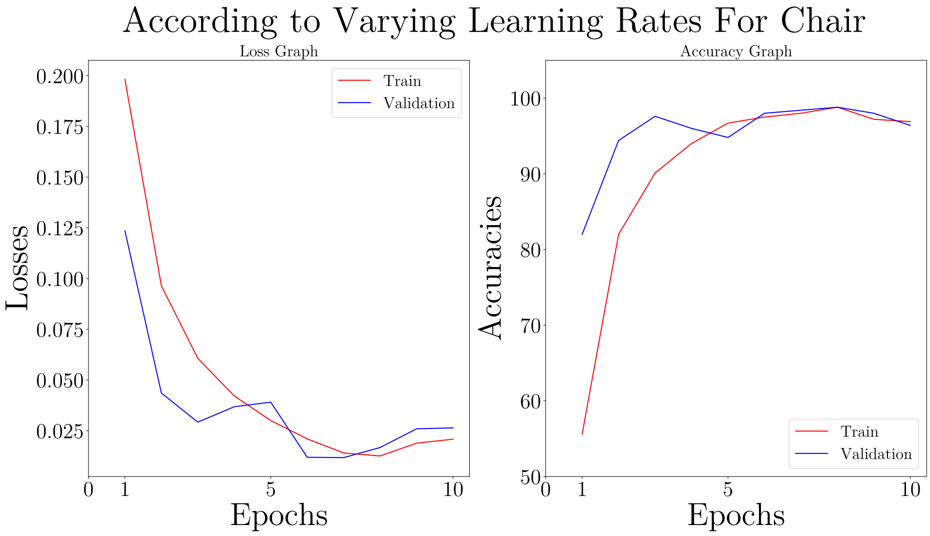


Figure 5.3: Results of AssembleRL-2D with learning rate values for ResNet18: 0.0001, PPFNet: 0.001, MLP: 0.0001.

The configuration of the MLP Layer was explored to determine the optimal structure for the AssembleRL-2D Network. Different combinations of hidden layer sizes and depths were tested for 20 epochs. Various configurations, including 2 and 3 layers with hidden layer input sizes of 256, 128, 64, 32, and 16 were experimented. It was observed that a 3-layer MLP with an input size of the hidden layer is 128 achieved faster convergence to zero loss and accuracy values above 95%, in Figure 5.4.

In this study, focal loss was utilized as the loss function. To address the class imbal-

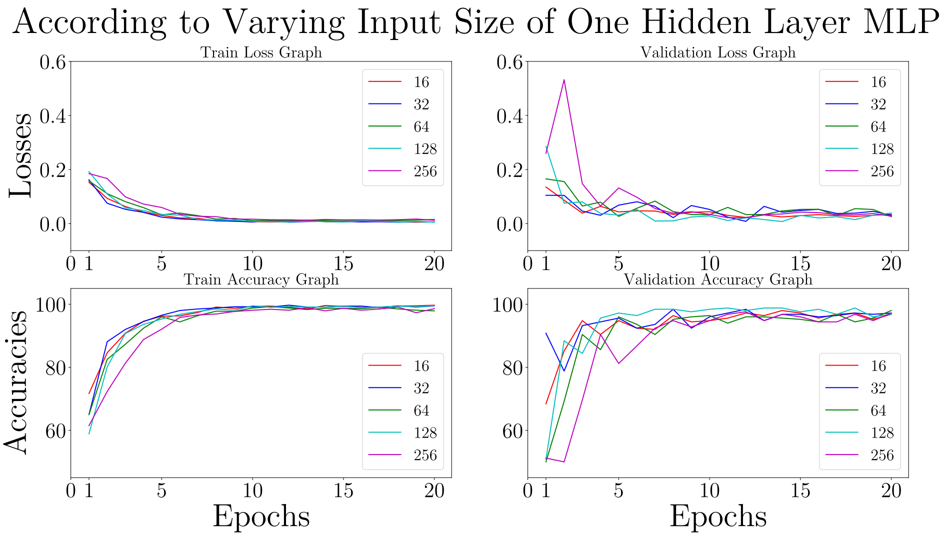


Figure 5.4: Loss and accuracy values for the training and validation sets with a 3-layer MLP with the changing hidden layer input size.

ance issue, the values of α and γ needed to be tuned. Given that α can be adjusted when extracting data from the dataset, it was set to -1 and ignored. The recommended value for γ was 2 [31]. In our experiments, we observed that setting $\gamma = 2$ resulted in higher accuracy and lower loss values during 20 epochs for all classes, as shown in Figures 5.5, 5.6 and 5.7. For $\gamma = 0$, the both accuracy and loss curves are more smooth, but the loss values start grater values. Therefore, $\gamma = 2$ is selected.

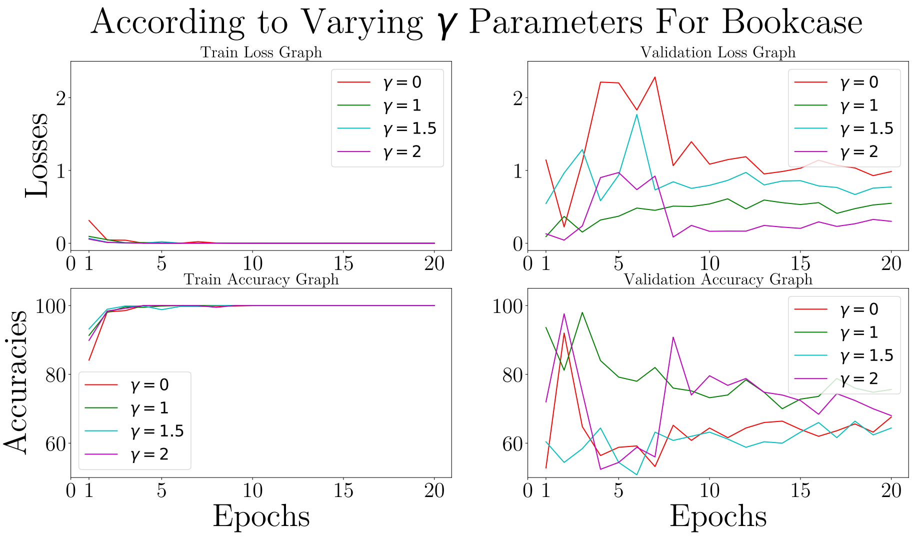


Figure 5.5: Loss and accuracy graphs for bookcase class changing gamma values.

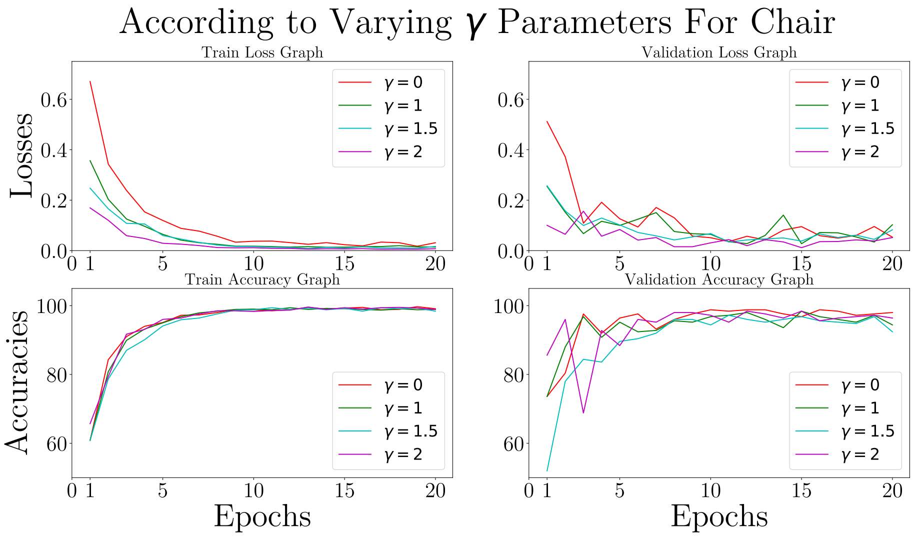


Figure 5.6: Loss and accuracy graphs for chair class changing gamma values.

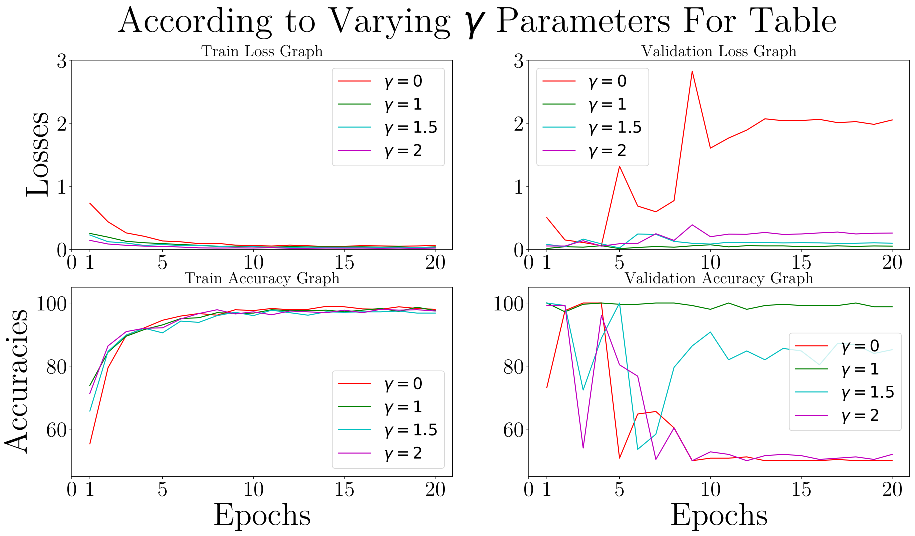


Figure 5.7: Loss and accuracy graphs for table class changing gamma values.

5.4 Experiment 3: Visual Results

This section presents results for the bookcase, chair, and bookcase classes. As the training progresses for AssembleRL-2D, the loss steadily approaches zero across all classes, accompanied by training accuracy values consistently exceeding 95%, as shown in Figures 5.8, 5.10 and 5.12. The success of AssembleRL-2D's predictions for correct assembly can be discerned from the confusion matrices provided in Figure

5.9, 5.11, and 5.13. Given that the model was supplied with one positive and one negative input, the results demonstrate notable success. Unfortunately, a comparable success rate, as observed in other classes, is not evident in the bookcase and table. This discrepancy arises from the diverse nature of furniture models and the inadequate number of furniture within the bookcase and table class. The test accuracies of the AssembleRL-2D are nearly 80% for the bookcase and table classes. Moreover, validation accuracies for these classes are above 90%. Expanding the dataset would enhance AssembleRL-2D's capacity to generalize across various furniture types by providing exposure to a more extensive range of examples.

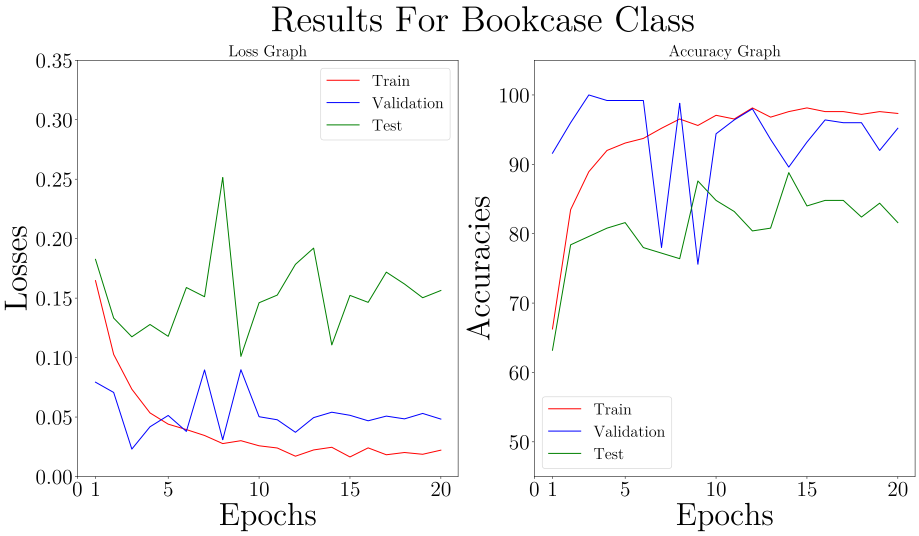


Figure 5.8: Loss and accuracy graphs for the bookcase class based on learning rate values of 0.0001, 0.001, 0.0001 for ResNet18, PPFNet, MLP Layers, seed number 8, $\alpha = -1$, $\gamma = 2$ for focal loss.

5.5 Experiment 4: Learning Furniture Assembly Stages with AssembleRL-2D

In this integrated study, combining AssembleRL and AssembleRL-2D structures, an assembly learning scenario was conducted using the ivar furniture model. As given in Table 5.1, the dataset was divided into a training set and a test set, and AssembleRL-2D was trained using this dataset. The trained model was then utilized during deterministic actions in the AssembleRL environment.

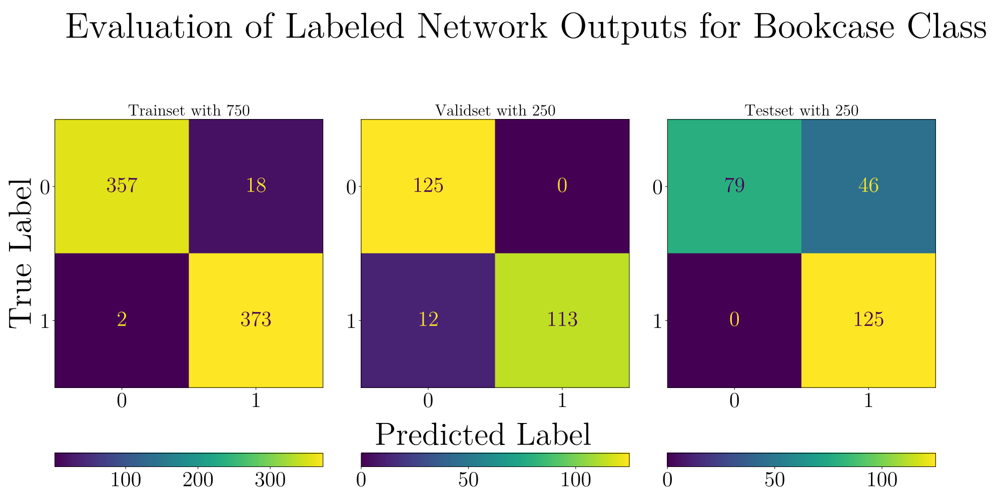


Figure 5.9: Comparison between predicted and true labels for training results of network for the bookcase class.

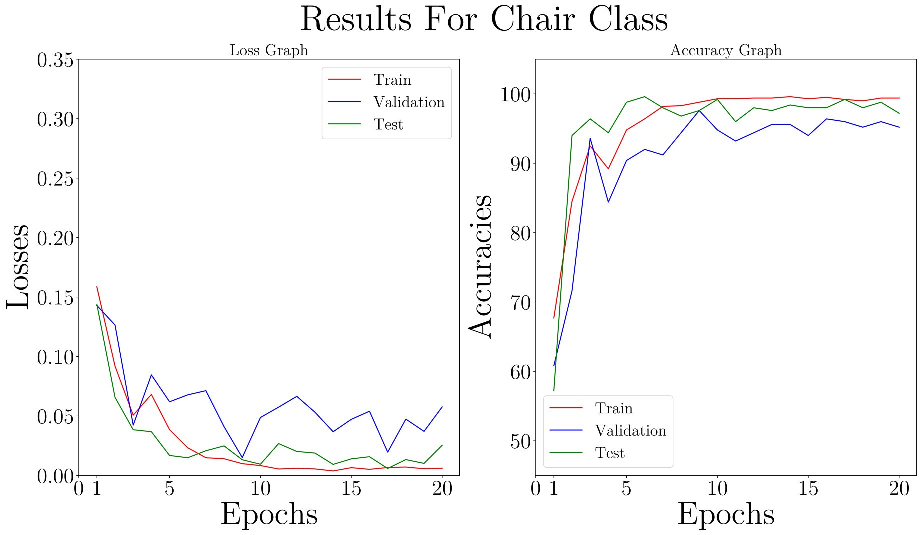


Figure 5.10: Loss and accuracy graphs for the chair class based on learning rate values of 0.0001, 0.001, 0.0001 for ResNet18, PPFNet, MLP Layers, seed number 8, $\alpha = -1$, $\gamma = 2$ for focal loss.

When an action is taken in the AssembleRL environment, the $next\_data$ and $info$ are returned. Instead of a reward, the value known as $final\_info$ is returned, which keeps all rewards received until the end of the episode. This $final\_info$ value is included in the returned $info$. The agent was updated using data collected from

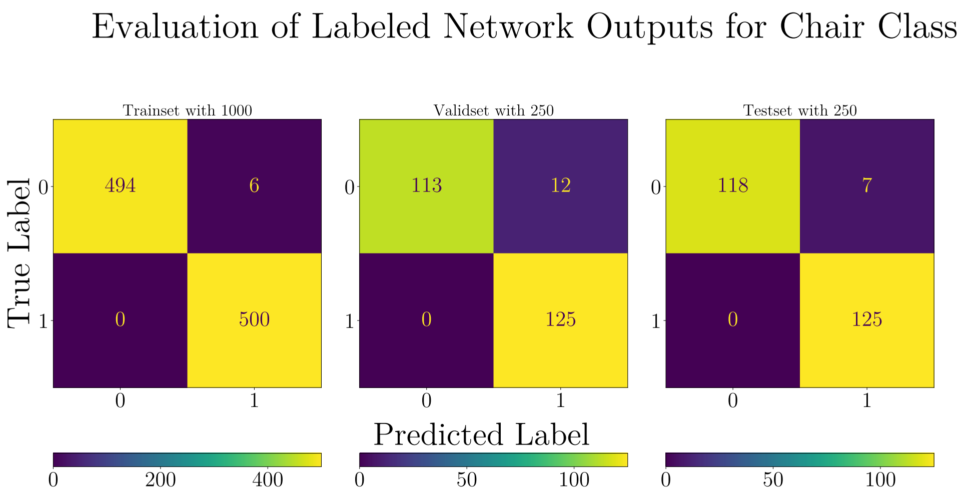


Figure 5.11: Comparison between predicted and true labels for training results of network for the chair class.

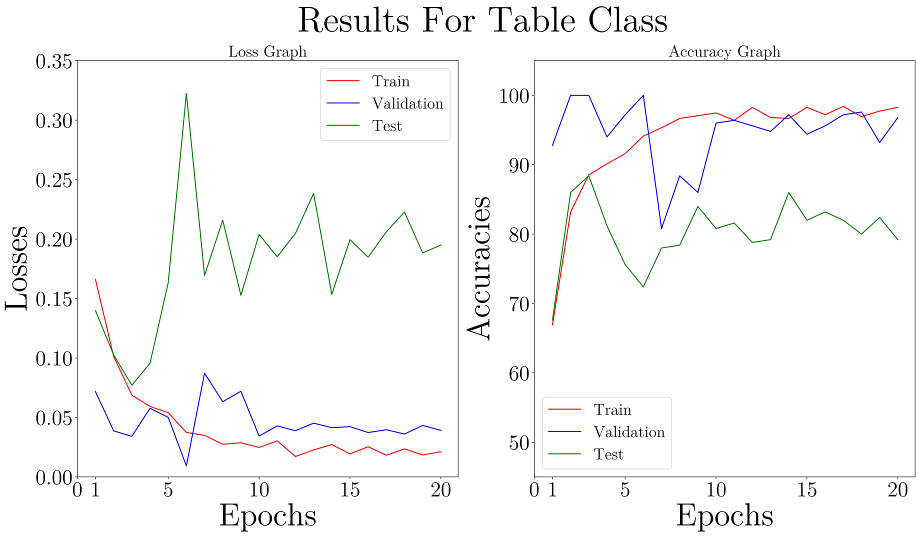


Figure 5.12: Loss and accuracy graphs for the table class based on learning rate values of 0.0001, 0.001, 0.0001 for ResNet18, PPFNet, MLP Layers, seed number 8, $\alpha = -1$, $\gamma = 2$ for focal loss.

512 actions and rewards during training. If an episode concludes due to the actions taken, it is recorded. Essentially, a model is saved based only on the actions taken by AssembleRL.

After recording with 512 data points, the agent is run deterministically, which means

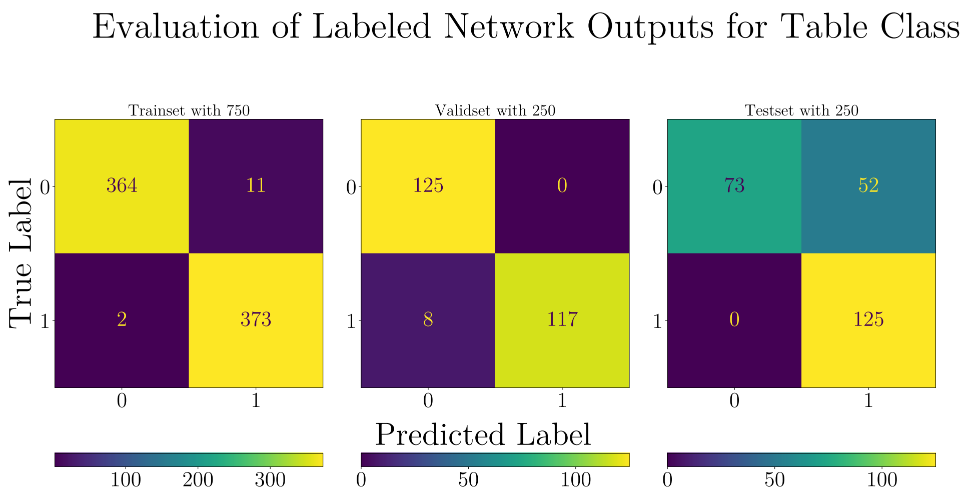


Figure 5.13: Comparison between predicted and true labels for training results of network for the table class.

Table 5.1: The dataset distribution of the AssembleRL-2D for furniture assembly task.

	Chair
Trainset	Agam
	Agne
	Balser
	Bernhard
	Ingolf
Testset	Ivar

with a trained AssembleRL-2D model, to take actions 10 times. The average of these 10 actions is recorded. The environment learns according to the changing seed values. For correctly assembled furniture, the agent's returns converge around a value of 3.5, as illustrated in Figure 5.14. However, achieving an episodic return value of 3.5 does not imply that the agent consistently learns. The learning process is verified by inspecting the recorded assembled furniture models. By examining the saved models, it was observed that the agent successfully learned furniture assembly. Despite learning different actions for each seed, the agent could correctly combine them.

Analyzing the value and policy loss graphs reveals that the loss values tend to approach zero, as shown in Figures 5.15 and 5.16. Although the value loss decreased, the inability of the variance to approach 1 indicates that the value function did not learn the reward effectively, in Figure 5.17.

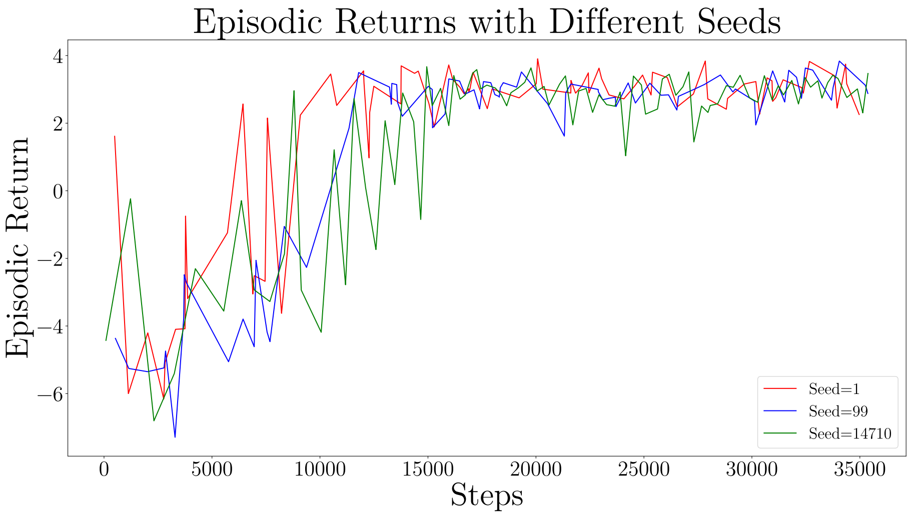


Figure 5.14: The episodic return values depicts assembling the ivar chair under different seeds 1, 99, and 14710.

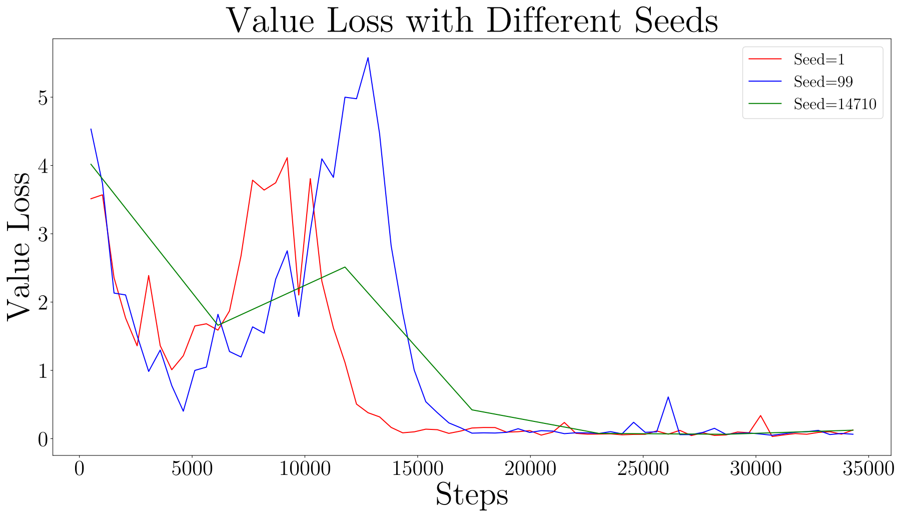


Figure 5.15: The value loss with different seeds 1, 99, and 14710.

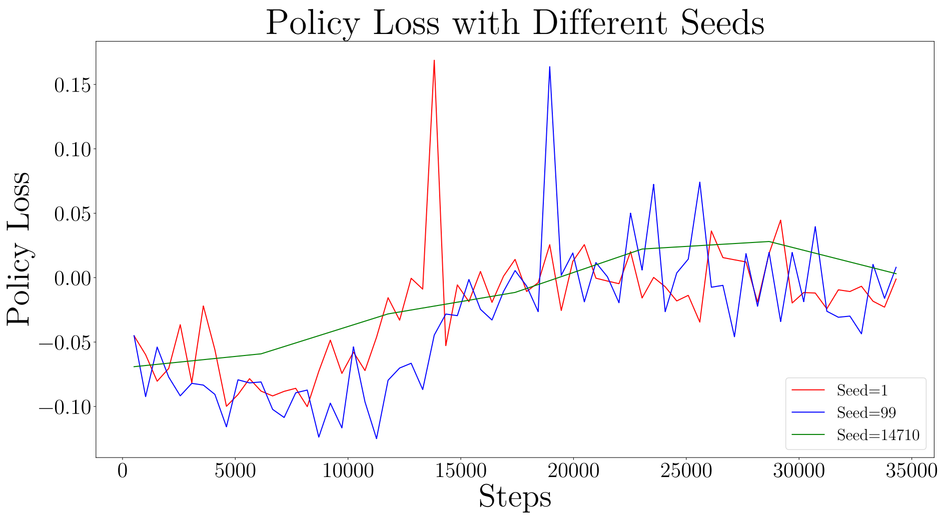


Figure 5.16: The policy loss with different seeds 1, 99, and 14710.

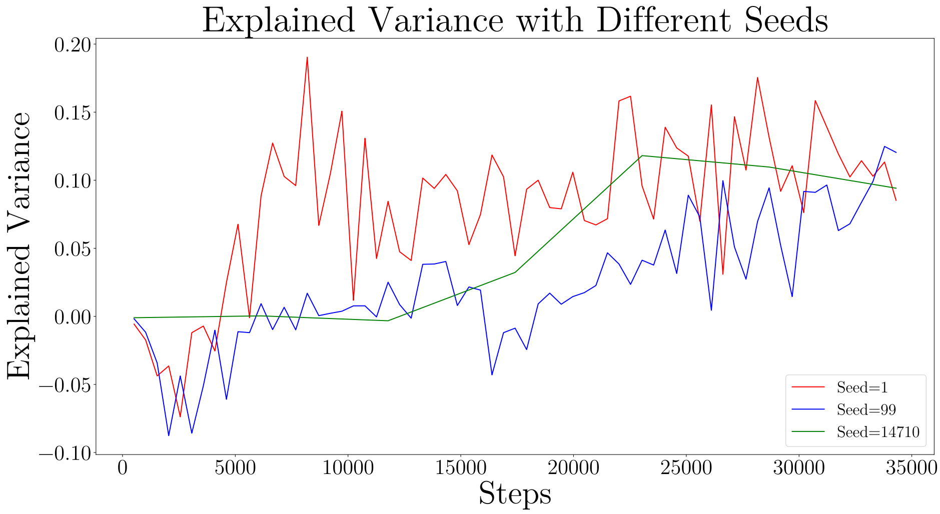


Figure 5.17: The explained variance with different seeds 1, 99, and 14710.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This thesis introduces the novel AssembleRL-2D model, a departure from existing literature that predominantly relies on 3D assembled furniture knowledge in furniture assembly learning. In addition to presenting this model, we present unique 2D and 3D datasets employed as inputs for AssembleRL-2D. The 2D dataset comprises images resembling the 2D drawing of the final assembled furniture, while the 3D dataset encompasses mesh models portraying various assembly scenarios for the furniture.

The AssembleRL-2D architecture, fed with positive and negative input pairs created using these datasets, incorporates ResNet18 as a 2D feature extractor, PPFNet as a 3D feature extractor, and an MLP layer that aggregates the feature vectors obtained from these extractors. The proposed AssembleRL-2D architecture has been individually tested for three furniture classes: bookcase, chair, and table. Exceptional training accuracy values exceeding 95% have been achieved for all classes, with validation accuracy values surpassing 90%, and loss values approaching zero. The unambiguous success of our proposed architecture is evident. Testing the model with unseen furniture models demonstrates its conceivable extension to previously unseen furniture. Furthermore, the AssembleRL-2D model, when combined with AssembleRL, exhibits the capability to assemble furniture accurately, even when parts are assembled in different orders. Our experiments showcase the effectiveness of the learned similarity metric with AssembleRL-2D between 2D manual-like image information and assembled furniture mesh models represented as a 3D point cloud.

In summary, there was a notable absence in the existing literature of a dataset encompassing both 2D manual information and 3D partial assembly steps. This thesis addresses this gap by introducing a novel dataset. Furthermore, we present a groundbreaking AssembleRL-2D architecture designed to learn the similarity between these two types of data, enabling the assembly of furniture using a model that utilizes this similarity as a reward signal.

6.1 Future Work

The dataset utilized in this thesis comprises a total of 16 furniture models distributed among bookcase, chair, and table classes with 5, 6, and 5 representations, respectively. However, the distribution of furniture models in the dataset lacks sufficient diversity. To address potential learning limitations, especially in the bookcase and table classes, expanding the dataset with additional furniture models is recommended. The study has successfully demonstrated that cross-class learning and dataset expansion hold the potential to enhance the generalizability of the AssembleRL-2D network across various classes. Furthermore, testing AssembleRL-2D with objects beyond furniture can be explored to assess the generalizability of the proposed method. Given that the dataset collection process is adaptable to desired objects, the expectation is to achieve consistent results with AssembleRL-2D for objects other than furniture.

The combined AssembleRL and AssembleRL-2D study identifies correct final assemblies even with different assembly orders. However, the explained variance value does not approach 1, suggesting a potential issue with the accuracy of the value function. Improving the value function could lead to more consistent results.

The choice of representing furniture models using 3D point clouds aims to mirror the data received from the real world using sensors or cameras. Consequently, the proposed model is well-suited for real-life scenarios. Further research could delve into exploring the consistency of the proposed method using 3D data derived from creating real-world furniture assembly scenarios.

REFERENCES

[1] O. Aslan, B. Bolat, B. Bal, T. Tumer, E. Sahin, and S. Kalkan, “Assemblerl: Learning to assemble furniture from their point clouds,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2748–2753, 2022.

[2] “Worldwide installations of industrial robots from 2004 to 2020, with a forecast through 2024 (in 1,000 units) [graph].” `https://www.statista.com/statistics/264084/worldwide-sales-of-industrial-robots/`, Dec. 2021.

[3] “Ivar,” assembly instructions, IKEA. `https://www.ikea.com/gb/en/assembly_instructions/ivar-chair-pine__AA-908032-2-2.pdf`.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017.

[6] C. R. Q. Li, Y. Hao, S. Leonidas, and J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, p. 5105–5114, Curran Associates, Inc., 2017.

[7] H. Deng, T. Birdal, and S. Ilic, “Ppfnet: Global context aware local features for robust 3d point matching,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 195–205, IEEE Computer Society, Dec. 2018.

[8] M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, and M. F. Talib, “Industry

4.0: A review on industrial automation and robotic," *Jurnal Teknologi*, vol. 78, pp. 137–143, 2016.

[9] A. Rojko, "Industry 4.0 concept: Background and overview," *International Journal of Interactive Mobile Technologies*, vol. 11, pp. 77–90, 2017.

[10] C. Weckenborg, K. Kieckhäfer, C. Müller, M. Grunewald, and T. S. Spengler, "Balancing of assembly lines with collaborative robots," *Business Research*, vol. 13, pp. 93–132, Apr. 2020.

[11] A. M. Djuric, J. L. Rickli, and R. J. Urbanic, "A framework for collaborative robot (cobot) integration in advanced manufacturing systems," *SAE International Journal of Materials and Manufacturing*, vol. 9, pp. 457–464, Apr. 2016.

[12] F. Vicentini, "Collaborative robotics: A survey," *Journal of Mechanical Design*, vol. 143, p. 040802, Apr. 2021.

[13] A. Cherubini, R. Passama, A. Crosnier, A. Lasnier, and P. Fraisse, "Collaborative manufacturing with physical human-robot interaction," *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 1–13, Aug. 2016.

[14] Y. Terzioglu, O. Aslan, B. Bolat, B. Bal, T. Tumer, F. C. Kurnaz, S. Kalkan, and E. Sahin, "APPRENTICE: Towards a cobot helper in assembly lines"," in *ICRA2021 Workshop on Unlocking the Potential of HRC for Industrial Applications*, 2021.

[15] J. Huang, G. Zhan, Q. Fan, K. Mo, L. Shao, B. Chen, L. Guibas, H. Dong, and P. C. Laboratory, "Generative 3d part assembly via dynamic graph learning," in *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS 2020) Datasets and Benchmarks Track*, Dec. 2020.

[16] J. Li, C. Niu, and K. Xu, "Learning part generation and assembly for structure-aware shape synthesis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 11362–11369, Apr. 2020.

[17] Y. Li, K. Mo, L. Shao, M. Sung, and L. Guibas, "Learning 3d part assembly from a single image," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), pp. 664–682, Springer International Publishing, 2020.

[18] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 909–918, June 2019.

[19] P. Mandikal, K. L. Navaneet, M. Agarwal, and R. V. Babu, "3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image," July 2018.

[20] R. Zhang, T. Kong, W. Wang, X. Han, and M. You, "3d part assembly generation with instance encoded transformer," vol. 7, pp. 9051–9058, Oct. 2022.

[21] J. Lee, S. Lee, S. Back, S. Shin, and K. Lee, "Object detection for understanding assembly instruction using context-aware data augmentation and cascade mask r-cnn," Jan. 2021.

[22] R. Wang, Y. Zhang, J. Mao, C.-Y. Cheng, and J. Wu, "Translating a visual lego manual to a machine-executable plan," in *European Conference on Computer Vision (ECCV)*, 2022.

[23] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, Sept. 2020.

[24] 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)*, vol. 2021-May, pp. 6343–6349, Institute of Electrical and Electronics Engineers Inc., 2021.

[25] 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*, Dec. 2021.

[26] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman, "Pix3d: Dataset and methods for single-image 3d shape modeling," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2974–2983, IEEE Computer Society, Dec. 2018.

[27] Y. Su, M. Liu, J. Rambach, A. Pehrson, A. Berg, and D. Stricker, "Ikea object state dataset: A 6dof object pose estimation dataset and benchmark for multi-state assembly objects," *CoRR*, vol. abs/2111.08614, 2021.

[28] Y. Ben-Shabat, X. Yu, F. Saleh, D. Campbell, C. Rodriguez-Opazo, H. Li, and S. Gould, "The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 846–858, 2021.

[29] J. J. Lim, H. Pirsiavash, and A. Torralba, "Parsing ikea objects: Fine pose estimation," in *2013 IEEE International Conference on Computer Vision*, pp. 2992–2999, Institute of Electrical and Electronics Engineers Inc., 2013.

[30] R. Wang, Y. Zhang, J. Mao, R. Zhang, C.-Y. Cheng, G. Research, and J. Wu, "Ikea-manual: Seeing shape assembly step by step," in *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022) Datasets and Benchmarks Track*, Sept. 2022.

[31] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, Feb. 2020.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009.

[33] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, June 2015.

[34] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June, pp. 9613–9622, IEEE Computer Society, June 2019.

[35] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press, May 1998.

[36] S. Marcel and Y. Rodriguez, “Torchvision the machine-vision package of torch,” in *Proceedings of the 18th ACM International Conference on Multimedia*, MM ’10, (New York, NY, USA), p. 1485–1488, Association for Computing Machinery, 2010.

[37] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019.

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. D. Facebook, A. I. Research, Z. Lin, A. Desmaison, L. Antiga, O. Srl, and A. Lerer, “Automatic differentiation in pytorch,” in *Thirty-first Conference on Neural Information Processing Systems (NIPS 2017*, 2017.