The University of Osaka
Institutional Knowledge Archive

Title	Robotic Grasping of Unknown Objects Using Similarity Matching
Author(s)	Chen, Hao
Citation	大阪大学, 2025, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/103158
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Robotic Grasping of Unknown Objects Using Similarity Matching

HAO CHEN

September 2025

Robotic Grasping of Unknown Objects Using Similarity Matching

A dissertation submitted to
THE GRADUATE SCHOOL OF ENGINEERING SCIENCE
The University of Osaka
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN ENGINEERING

BY
HAO CHEN

September 2025

Abstract

Robotic grasping has been extensively studied for decades as a fundamental component of the automation industry. Early robotic grasping systems were designed for specific objects and predefined tasks in fully known and programmable environments. Such systems relied on rigid motions that were only suitable for repetitive production scenarios. With recent advancements in Vision Technology (VT) and Deep Learning (DL), it has become feasible to develop more dexterous robotic grasping systems capable of handling previously unseen objects and tasks. However, the wide variety of object types, environmental uncertainty, and the inherent noise in visual sensing pose substantial challenges to achieving generalized grasping with high accuracy and robustness.

To address these challenges, most prior work has focused on learning-based methods, aiming to enhance generalizability through optimized neural network architectures or large-scale training datasets. However, our experiments demonstrate that even state-of-the-art grasp regression models struggle to maintain high performance when deployed in unfamiliar scenarios involving novel objects, environments, or detection conditions. This limitation significantly hinders their practical applicability in real-world settings. To overcome this bottleneck, we propose an alternative solution that explores knowledge transfer between similar objects. Specifically, we introduce a novel strategy for generalized object grasping, termed **Similarity Matching**, which leverages prior knowledge from known templates to guide the grasping of previously unseen objects.

In contrast to conventional learning-based methods, our approach achieves precise and reliable grasping for a wide range of unknown objects using only a small existing dataset, eliminating the need for expensive training or labor-intensive data collection. Extensive real-world evaluations confirm the effectiveness of our method in diverse scenarios, including grasping both isolated and cluttered objects, handling static items on a fixed surface, and dynamic items on a moving conveyor. Furthermore, we demonstrate the scalability of our similarity-based strategy to task-oriented grasping, where functional knowledge—such as the affordance of a mug handle—can be transferred across similar instances to support high-level manipulation tasks.

Contents

List of Figures **VII**

List of Tables **XIII**

Abbreviations **XV**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Objectives 2

1.3 Dissertation Outline 3

2 Literature Review **5**

2.1 Novel Object Grasping 5

2.1.1 Learning-based methods 5

2.1.2 Analysis-based methods 6

2.1.3 Similarity-based methods 7

2.2 Dynamic Object Manipulation 7

2.3 Task-Oriented Grasping 8

3 Initial Similarity Matching for Novel Object Pick-and-Place 11

3.1 Introduction 11

3.2 Similarity Prediction 13

3.2.1 System Overview 13

3.2.2 Category-Association Matching (CAM) 14

3.2.3 Point Cloud Registration (PCR) 15

3.2.4 Similarity Quantification 16

3.3 Pick-And-Place Planning 19

3.3.1 Grasp Transfer 19

3.3.2 Stable Placement 20

3.3.3 Motion Planning 22

3.4 Experiments 22

3.4.1 Experimental Setup 23

3.4.2 Similarity Prediction 24

3.4.3 Novel Object Pick-and-Place 25

3.5 Discussion 27

3.6 Conclusions 28

4 Multi-Level Similarity Matching for Single-View Object Grasping 29

4.1 Introduction 29

4.2 Methods 32

4.2.1 Overview 32

4.2.2 Single-View Object Recognition 33

4.2.3 Multi-Level Similarity Matching 34

4.2.4 LLM-Assisted Semantic Matching 35

4.2.5 C-FPFH-Based Geometric Matching 36

4.2.6 SOBB-Guided Dimensional Matching 39

4.2.7 Candidate Model Selection 41

4.2.8 PDM-Based Point Cloud Registration 42

4.2.9 Imitative Grasp Planning 44

4.2.10 Stability-Aware Grasp Fine-tuning 45

4.3 Experiments 47

4.3.1 Experimental Setup 47

4.3.2 Evaluation of Similarity Matching 48

4.3.3 Grasping Isolated Objects 51

4.3.4 Grasping Cluttered Objects 55

4.3.5 Ablation Study 57

4.3.6 Robustness to Environmental Changes 59

4.4 Conclusions 61

5 Application of Similarity-Based Methods in Dynamic Object Manipulation 63

5.1 Introduction 63

5.2 Methods 65

5.2.1 System Overview 65

5.2.2 Global Visual Detection 66

5.2.3 LLM-Assisted Similarity Matching 67

5.2.4 Static Grasp Planning 68

5.2.5 Adaptive Local Detection 70

5.2.6 Dynamic Grasp Planning 72

5.3 Experiments 73

5.3.1 Experimental Setup 73

5.3.2 Dynamic Grasping Experiments 73

5.3.3 Failure Analysis 75

5.4 Conclusions 76

6 Application of Similarity-Based Methods in Task-Oriented Object Grasping 79

6.1 Introduction 79

6.2 Methods 82

6.2.1 Overview 82

6.2.2 Human Instruction Interpretation 83

6.2.3 Functional Part Recognition 85

6.2.4 Task-Oriented Grasp Planning 89

6.3 Experiments 92

6.3.1 Experimental Setup 92

6.3.2 Task-Oriented Grasping of Unseen Objects 92

6.3.3 Ablation Study 99

6.3.4 Generalization to Novel-Category Objects 101

6.4 Conclusions 102

7 Discussion 109

7.1 Contributions 109

7.2 Open Challenges and Future Work 110

References 111

Acknowledgements 123

Publications 125

List of Figures

3.1 Overview of the proposed pick-and-place system. The inputs include one RGB image, one depth image and one existing database. The output is a complete pick-and-place motion trajectory executable by the robot. . . 14

3.2 Example results of CAM using Panoptic FPN and Word2Vec as the approaches of image segmentation and word embedding. In two tests, (a) *bottle* and (b) *book* are detected. Scores shown below represent the category-association matching results between the novel object and the registered models in the database. The word similarity value is calculated in the embedded space. 15

3.3 Invariant relative distance between the object's positions shown in two images when we make the orientation of the camera unchanged. With a certain translation matrix, we can easily concatenate two single point clouds to obtain a more complete point cloud of the target object. . . . 17

3.4 Identification of the constant coefficients μ_c and μ_p. We make use of an out-of-category object and two registered models with similar shapes but belong to different categories to calculate their $\hat{Q}$ values, which are assumed as the same. 18

3.5 Preplanned grasps on a known model and error in PCR and grasp transfer. (Left) A model is preplanned with a series of robust grasps. (Middle) In PCR, the point cloud of the registered model (red) is not perfectly aligned with the partially-observed point cloud of the target object (green). (Right) Due to the error of the transformation matrix, the calculated position of the target object (yellow) deviates from the real position (white). 20

3.6 Inference of the in-hand posture. We rotate the in-hand object with a small angle θ in two inverse directions by the grasp center. When the original posture is upright (a), the distance between the lowest point of the object and the contact surface does not differ much before and after rotations. When the original posture is right-leaning (b) or left-leaning (c), the distance will differ much in one of the rotations. 21

3.7 A dozen of novel objects used for validation experiments. For each object we find an evaluated most similar model from the database and imitate its preplanned robust grasps to achieve pick-and-place tasks. 23

3.8 Example results of similarity prediction based on CAM and PCR. The red block represents the results of semantic identification by Panoptic FPN. The green block shows the most similar models found with the highest Q values. 24

3.9 Realization of pick-and-place motions based on the results of similarity prediction. The grasp postures of the bottle (left) and the toothbrush (right) are generated from the preplanned grasps on the bottle model and the screwdriver model in Fig. 3.8. 26

4.1 An example of grasping a novel object using our three-step method. First, visual features of the target object obtained in the real world are used for similarity matching with existing database models. Then, grasp planning and fine-tuning are performed in simulation based on the matching results. Finally, the optimized grasp is executed in the real world to achieve the task. 31

4.2 Overview of our proposed approach. The system inputs (shown in blue boxes) include a single-view RGB-D image and an existing object model database. A background image without target objects is taken beforehand. The output (shown in the orange box) is an optimized grasping action for execution. 33

4.3 Three-step generation of C-FPFH descriptors for object point clouds. Due to sensing noise, point features may occasionally vary across surfaces with similar geometry; however, such variations are acceptable during geometric matching, facilitated by our discrete sampling and consecutive clustering methods. 37

4.4 Test results of using C-FPFH-based geometric matching to identify similar models from single-view object point clouds. All candidate models are originally from the YCB dataset, renamed with simplified category names and index numbers for cases where multiple models exist within a single category. 40

4.5 Comparison of different types of bounding boxes applied to a single-view partial point cloud obtained from a table-top object. 41

4.6 (a)-(c): Three steps to perform PDM-based point cloud registration between partial (yellow) and complete (blue) point clouds. (d): Comparison between a traditional registration method and our proposed method. . . 43

4.7 Two-stage grasp fine-tuning, including grasp position and grasp center adjustments to optimize the final grasp quality. 45

4.8 The experimental objects used for similarity matching. The first and second rows display the non-occluded and occluded scenes, respectively. The numbers in the figure correspond to the scene IDs in Table 4.1 and Table 4.2. The third row presents the original 3D model of each object. 48

4.9 Grasping various novel objects using our similarity-based method. The top left corner of each figure displays the single-view object point cloud, the matched candidate model, and the grasp planning with fine-tuning process. The object IDs shown in the bottom-left correspond to those in Table 4.3. 53

4.10 Comparison of different grasp planning methods based on object point clouds. The blue grasps represent all the generated grasp candidates, while the red grasps indicate the final executed grasps that are first recognized as IK-solvable and collision-free from the candidate list. 53

4.11 Objects in clutter used for grasping experiments. Each clutter set contains 3 or 5 objects, and is tested in 5 different arrangements. 57

4.12 Comparison of our method with grasping benchmarks for objects in clutter. In the outputs of GraspNet and HGGD, red grasps indicate high-quality grasps, while blue grasps indicate low-quality ones. Our method generates high-quality grasps for all target objects by rendering similar reference models. 58

4.13 Evaluation of grasp quality using a graph paper. After a pick-and-place motion, the stability of the executed grasp is quantified by measuring the offset between a marker on the object and an initial point on the paper. 59

4.14 Application of our method to achieve grasping tasks in three different scenarios without the original models of the environmental objects. . . . 60

4.15 Comparison of the performance robustness of different methods. 60

4.16 Failures of learning-based methods in on-shelf and hand-over tasks. . . . 61

5.1 Grasping moving cluttered cans by our method. Static and dynamic planning are executed at the end of global and local detection respectively. 65

5.2 Workflow of our proposed system. All visual detections and grasp planning are performed online using only a single in-hand camera. 66

5.3 Fast and accurate grasp planning for cluttered objects based on *collision area* (blue $\cup$ red) and *overlap area* (blue $\cap$ red). 69

5.4 Multi-metric evaluation of grasp sequences in cluttered scenes. 70

5.5 (Left) Speed estimation of moving cluttered objects based on adaptive local visual detection and real-time object tracking. (Right) Evaluation of the calculation results in two conveyor speed modes (5.5 cm/s and 11 cm/s). In each trial, we change either the object arrangement or object type. 72

5.6 Experimental objects in 3 patterns, 5 clutters, and 10 arrangements. . . 74

6.1 Demonstration of our method applied to a *pouring* task with an unseen mug. 81

6.2 Proposed TOG system overview. Blue/Orange indicate Inputs/Outputs. 82

6.3 LLM-guided object-part-task ontology with online and offline components. 84

6.4 Sensitivity of model-based part segmentation to viewpoint changes. . . . 86

6.5 Template-assisted object part recognition using a three-step strategy. . . 87

6.6 Two distance metrics for evaluating similarity between point clusters. . . 88

6.7 Local-to-global point cloud registration through rotation optimization. . 90

6.8 Adjustment of suboptimal grasps via localized bounding box estimation. 91

6.9 Objects used in TOG experiments with predefined ontological knowledge. 93

6.10 Recognition of functional object parts using our geometric analysis method. 94

6.11 Recognition of functional object parts using our geometric analysis method. 95

6.12 Failures of GraspGPT in language-guided grasp selection. Selected grasps are highlighted in green, while the ground truth labels are shown in top-right corner. 97

6.13 Comparison of task-oriented grasp planning between GraspNet and our proposed method. 100

6.14 Performance discrepancy between the two baselines and our full method. 101

6.15 Generalizing TOG to Novel Instances via Ontological Knowledge Extension.102

6.16 Framework for reusing LLMs in iterative prompt optimization. 104

List of Tables

3.1 Experimental results of in-category object pick-and-place. 25

3.2 Experimental results of out-of-category object pick-and-place. 26

4.1 Evaluation Results of Similarity Matching Baselines in Non-Occluded Scenes 50

4.2 Evaluation Results of Similarity Matching Baselines in Occluded Scenes 51

4.3 Experimental Results of Grasping Isolated Objects 52

4.4 Experimental Results of Grasping Cluttered Objects 56

4.5 Ablation Studies On Each Component of Our Method 59

5.1 Experimental Results for Grasping Various Types of Moving Clutter . . 75

6.1 Evaluation of PRA via a Baseline Study Against a VLM-Based Approach 94

6.2 Evaluation of GPA via a Baseline Study Against a Learning-Based Approach 97

6.3 Evaluation of GSR via a Baseline Study Against a Grasp Detection Approach 99

6.4 Ablation Study on Proposed Methods Using Water-Filled Mugs 101

Abbreviations

AABB Axis-Aligned Bounding Box

C-FPFH Clustered Fast Point Feature Histogram

CNN Convolutional Neural Network

CPU Central Processing Unit

CVFH Compact Viewpoint Feature Histogram

DL Deep Learning

DoF Degrees of freedom

FPN Feature Pyramid Networks

FOV Field Of View

GPU Graphics Processing Unit

ICP Iterative Closest Point

IK Inverse Kinematics

LLM Large Language Model

LSTM Long Short-Term Memory

SOBB Semi-Oriented Bounding Box

PCA Principal Component Analysis

PCL Point Cloud Library

PDM Plane Detection and Matching

RANSAC Random Sample Consensus

RGB-D Red, Green, Blue, and Depth

RL Reinforcement Learning

RNN Recurrent Neural Network

RRT Rapidly-exploring Random Tree

SAC-IA Sample Consensus Initial Alignment

SOTA State-Of-The-Art

TOG Task-Oriented Grasping

TSDF Truncated Signed Distance Field

UR5e Universal Robots 5 e-series

VLM Vision-Language Model

VT Vision Technology

Chapter 1

Introduction

1.1 Background and Motivation

In the logistics and retail industry, thousands of objects are handled every day—whether moving along production lines in warehouses or arranged on shelves for merchandising in supermarkets. Due to social issues such as population decline, there is a growing demand for robotic automation in these environments. However, the ability to manipulate the diverse objects—differing in category, shape, and size—remains a significant challenge. Traditional robotic systems designed for object manipulation typically require complete prior knowledge of target objects, such as accurate 3D models representing full geometry, to ensure reliable operation. In practice, preparing and maintaining precise models for a constantly updating inventory of objects is both costly and infeasible. This creates an urgent need for highly dexterous robotic systems capable of handling a broad range of unknown objects without relying on prior object-specific knowledge.

To address this need, vision-based approaches have been widely explored in robotic manipulation [1]. Rather than acquiring complete object models, these methods leverage visual features captured by RGB-D cameras to plan feasible grasp poses based on partial observations. In most cases, Deep Learning (DL) techniques are employed [2] to encode visual inputs into neural networks, which then predict high-confidence grasp configurations through a regression process. Advances in network architecture design [3] and large-scale training data [4] have led to notable success in improving grasp reliability. However, learning-based methods inherently suffer from the black-box nature of neural models and often struggle to generalize across varying scenarios. This limitation becomes a key bottleneck in achieving robust, scalable robotic grasping across diverse objects and tasks.

This dissertation seeks to overcome this bottleneck through a novel strategy that extends beyond traditional learning-based frameworks, termed **Similarity Matching**. Rather than focusing on refining neural networks or optimizing learning policies, we explore the intrinsic similarities among objects, leveraging prior knowledge of object templates to guide the manipulation of previously unseen targets. This approach offers a crucial perspective on using a small amount of data to effectively handle a wide variety of objects. However, the success of transferring knowledge from known templates to unknown targets depends on the quality of the matching process, making the development of an optimal matching framework a central focus of our research.

The work presented in this dissertation traces the evolution of our methodology—from score-based to multi-level matching, from multi-view to single-view observation, and from grasping isolated, static objects to handling cluttered, dynamic ones. As our method evolves, it demonstrates the capability to tackle increasingly complex and challenging scenarios. Extensive evaluations have been conducted to validate the superior efficiency, accuracy, and robustness of the proposed approach in grasping diverse, previously unseen objects under varying conditions.

1.2 Objectives

The general objectives of this dissertation are as follows:

1. Using the proposed similarity matching approach to achieve generalized object grasping capable of handling a wide variety of unknown objects with high success rates. In particular, we emphasize the method's robust performance in real-world

tasks, where significant uncertainties and variations are present.

2. Extending the similarity matching framework from grasping single, static objects to handling cluttered, dynamic environments, and further from basic grasping to task-oriented manipulation. To fully demonstrate the method's scalability, we enhance the original framework to ensure its applicability across a wide range of real-world scenarios.

1.3 Dissertation Outline

This dissertation is organized as follows.

In Chapter 2, we present a comprehensive literature review covering novel object grasping, dynamic object manipulation, and task-oriented grasping.

In Chapter 3, we introduce our initial similarity matching approach for novel object pick-and-place tasks. The proposed method employs a similarity evaluation function that integrates both semantic and geometric features of the observed object to identify a similar database model with pre-existing grasp knowledge. Based on the matched model, a knowledge transfer process is then carried out to enable pick-and-place planning for the novel object. However, this initial approach has several limitations: (1) it requires multi-view observation to extract sufficient object features; (2) the similarity scoring function exhibits instability when encountering objects that fall outside the database knowledge; and (3) it is not capable of handling cluttered object scenes.

In Chapter 4, we propose an improved multi-level framework for high-precision similarity matching. This method adopts a three-stage strategy—matching, planning, and fine-tuning—that enables accurate grasping of diverse, previously unseen objects under single-view observation. By combining a compensatory multi-level matching stage with stability-aware grasp fine-tuning, the approach demonstrates strong robustness and generalization, even for objects entirely novel to the existing knowledge base. Moreover, it effectively handles cluttered scenes and adapts to unfamiliar environments, consistently outperforming existing benchmark methods.

In Chapter 5, we extend the similarity-based grasping approach to more challenging scenarios involving the grasping of moving objects in dense clutter. To address this, we introduce a global-to-local detection and static-to-dynamic planning framework. This framework resolves multiple sources of uncertainty arising from clutter and motion by enabling knowledge transfer not only between known and unknown objects, but also across different visual detection states and grasp planning conditions.

In Chapter 6, the similarity-based framework is further extended to task-oriented grasping, which considers object part affordances beyond basic grasping. We introduce an object-part-task ontology supported by Large Language Models (LLMs) to translate human's task instruction into robot's grasp selection. Through an optimized matching process, a small set of model templates with predefined part segmentation and grasp knowledge can accurately guide task-oriented grasping of previously unseen objects at functionally relevant regions.

Finally, in Chapter 7, we summarize the key achievements and limitations of the proposed methods, and discuss open challenges and potential directions for future work.

Chapter 2

Literature Review

2.1 Novel Object Grasping

Research on robotic grasping has a long and evolving history [5], progressing from analytical approaches [6, 7] to data-driven techniques [8, 9, 10]. In this literature review, we focus specifically on recent works addressing the grasping of unknown objects using vision-based technologies. These approaches can be broadly categorized into three main types: learning-based methods, analysis-based methods, and the most relevant to our work—similarity-based methods.

2.1.1 Learning-based methods

In the 2010s, major advances in Convolutional Neural Networks (CNNs) enabled robotic vision to achieve unprecedented performance in novel object grasping, spawning outstanding works such as Dex-Net [11], GG-CNN [12], and GPD [13]. They leverage a large number of depth images or point clouds with labeled grasp poses to train CNN models capable of predicting high-quality grasp poses for unseen objects. DGGN [14] first uses RGB-D inputs to train a two-stage network regressing grasps from reconstructed 3D scenes. PointNetGPD [15] enhances the performance of GPD by integrating the architecture of PointNet [16] into an end-to-end grasp evaluation network. QT-Opt

[17] represents an attempt at leveraging Reinforcement Learning (RL) for grasp generalization, achieving high grasp success rates on unseen objects through large-scale self-supervised training. In the 2020s, a broader range of methods beyond traditional CNNs has been explored. VGN [18] and GIGA [19] employ Truncated Signed Distance Field (TSDF) representations to efficiently learn grasp detection in cluttered scenes. 3DSGrasp [20] and SCARP [21] perform shape completion on partial point clouds to enhance the performance of single-view grasping. GraspNet [22] and Grasp-Anything [23] construct large-scale grasp datasets and use them to train high-performance models for general object grasping. HGGD [24] provides new insights into generating dense grasps in real-time by utilizing both global and local features of objects in clutter. While these methods perform well in their specific tasks, they share several common limitations: 1) high training cost; 2) high sensitivity to sensing noise and environmental changes; and 3) low error traceability due to complex learning architectures. In contrast, our proposed method is training-free, robust to varying conditions, and fully error-traceable through a simple framework.

2.1.2 Analysis-based methods

A few recent studies try to generate high-quality grasps directly from object point clouds by geometric analysis. Adjigble et al. [25] leverage *zero-moment shift features* [26] to evaluate the local geometric similarity between object surfaces and gripper surfaces, enabling the selection of grasp positions with the highest probability of success. Wu et al. [27] detect *hidden superquadrics* [28] from partial object point clouds to generate and filter reliable grasp candidates through a multi-metric evaluation. Wang et al. [29] propose *visible point-cloud* to efficiently exclude unsafe grasps and determine the optimal grasp pose from a partial view. These methods show the possibility of generalized grasping without model training. However, they require high-precision visual features to achieve good results, whereas our proposed method remains effective even with sparse and noisy visual inputs.

2.1.3 Similarity-based methods

The Columbia Grasp Database [30] first introduced similarity matching in novel object grasping. They leverage similar database models with precomputed grasps to achieve imitative grasping of previously unseen objects. However, their similarity computation requires prior 3D scanning of the target object, leading to inefficient task completion. Herzog et al. [31] develop a template-based grasp planning algorithm that generalizes demonstrated grasps to novel objects with similar local geometry. However, their performance depends heavily on the number and type of grasp templates. Two recent studies [32, 33] utilize ontological classification and scoring functions to identify similar objects with grasping knowledge for guiding the grasping of novel objects. Although these methods perform inconsistently due to their reliance on existing knowledge, their efforts highlight that the potential of using similarity matching has been underestimated, inspiring us to further explore this direction.

2.2 Dynamic Object Manipulation

Dynamic manipulation of moving objects has recently attracted attention in the field of robotic grasping. GG-CNN [34] is a well-known generative grasping network that is capable of handling dynamic scenes where object positions are changed after each grasp attempt. However, they need the objects to remain stationary during the grasp execution. Marturi et al. [35] achieve adaptive grasping for various types of moving objects by developing a local planner for object tracking and a global planner for grasp switching. However, they require prior observation of the target object from multiple viewpoints to obtain its complete surface geometry. Two recent works [36, 37] use Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM), respectively, to predict the future locations of moving objects for precise dynamic grasping. However, they require the target object to be pre-trained with robust grasps, whereas our method can handle novel objects without any prior knowledge. In addition, a few studies [38, 39] incorporate RL algorithms to achieve moving object grasping with a single camera.

However, their methods have only been evaluated on single objects, not in cluttered scenes. In contrast, our proposed method is capable of handling moving objects in dense clutter.

2.3 Task-Oriented Grasping

High-level manipulation tasks require robots not only to grasp objects successfully, but also to grasp at appropriate positions to ensure both safety and functionality. To this end, Task-Oriented Grasping (TOG) has been extensively studied, evolving from analytical approaches [40, 41] to vision-based approaches [42, 43, 44], and more recently to vision-language-based approaches [45, 46, 47]. Here, we highlight a few recent notable works. CAGE [48] is a representative affordance-aware grasping system considering both object and task constraints. GCNGrasp [44] constructs a knowledge graph to train a Graph Convolutional Network, enabling the generalization of task-oriented grasps from predefined instances to novel concepts. OS-TOG [49] and Robo-ABC [50] propose matching frameworks that utilize database objects with labeled affordances to guide the grasping of novel objects. While conceptually similar to our approach, their methods are limited to 2D image matching, which restricts their ability to transfer 6-DoF grasping knowledge. Although all of these approaches effectively achieve generalized TOG, they do not incorporate language models and are therefore limited to simplistic task labels such as *handover* or *cut*.

To address more complex human instructions, VLMs have increasingly been adopted for contextual understanding. LERF-TOGO [51] integrates CLIP embeddings [52] with DINO features [53] to train a multi-scale VLM capable of generating grasps for specific object parts based on language queries. GraspSplats [54] introduces a Gaussian-based feature representation for real-time object motion tracking and enables dynamic language-guided manipulation. FoundationGrasp [55] leverages open-ended knowledge from foundation models (both LLMs and VLMs) to learn generalizable TOG skills. While these approaches successfully associate contextual language inputs with spatial motion outputs, they struggle to achieve high task success rates due to the complexity inherent in integrating vision and language within cross-domain learning frameworks. We cope with this issue by processing visual and verbal inputs separately and connecting them through a predefined ontology.

In recent years, the emergence of high-performance LLMs such as *ChatGPT* and *Claude* has significantly advanced TOG strategies. For instance, GraspGPT [56] leverages LLMs to generate text descriptions for both objects and tasks, which are then encoded with object point clouds and grasp poses to select high-quality task-oriented grasps through a transformer decoder. ATLA [57] utilizes LLMs to generate rich semantic knowledge, accelerating tool learning in diverse manipulation tasks. While these approaches commonly use LLMs to generate instructive text to enhance the learning process, we find that directly regressing grasp poses from text embeddings can lead to unstable performance due to the diversity of LLM descriptions and the uncertainty in real-world observations. Therefore, we propose a sequential strategy that first queries the target object parts using the strong interpretability of LLMs, and then focuses on object geometry for grasp planning. This separation effectively mitigates the complexity of handling linguistic and visual features simultaneously.

Chapter 3

Initial Similarity Matching for Novel Object Pick-and-Place

This thesis chapter originally appeared in the literature as

H. Chen, T. Kiyokawa, W. Wan, and K. Harada, "Category-Association Based Similarity Matching for Novel Object Pick-and-Place Task," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2961–2968, 2022.

3.1 Introduction

A complete vision-based robotic manipulation system contains the modules of object detection, grasp selection and motion planning. Although previous work has studied extensively on each of the modules [58], the accumulated error occurring in sensing, planning and controlling processes has made the whole system difficult to achieve high precision. A few studies have focused on using deep learning to make a highly precise pick-and-place system come true [59, 60, 17, 61]. However, such of learning-based methods are inevitably costly and time-consuming for training. They also have a critical limitation of precision decrease when the conditions of manipulation tasks change, like using different types of cameras or robots.

To break through such limitations, we propose a training-free method using similarity matching between a known database and unknown targets. Given a novel object under uncertainty, we use only one depth sensor equipped on robot hand to obtain its semantic information and point clouds. Meanwhile, we construct a grasp database which also includes semantic information and point clouds of a few known models. By developing a similarity prediction method, we can identify the specific model with the highest similarity to the novel object from the database. Assumed similar objects share similar robust graspable points, we can first plan a series of robust grasps on a known model, and then transfer them to the real-world target object using point cloud registration. In this way, we can avoid the difficulty of grasp planning in partially-observed situations. Moreover, by applying point cloud registration with known models, reorientation of objects in pick-and-place task is easy to realize as we have no need of estimating the initial posture of the target object at the beginning.

Our work mainly makes three contributions.

Proposal of category-association based similarity prediction combined with point cloud registration. Recent technologies like YOLOv5 [62] and DeepLabV3+ [63] have enabled us to extract semantic information from a single image, making it easy to obtain the category name of a novel object with an RGB camera. By introducing a database consisting of the category information of a few known models, we can evaluate the semantic similarity between the categories of novel objects and known models. In addition, we incorporate point cloud registration to evaluate their shape similarity. By combining the two similarity matching results, we can make a reliable prediction of which known model is most similar to the real-world target object.

Construction of a novel object pick-and-place system based on imitating grasps. We construct a vision-based robotic manipulation system aimed at novel objects using the similarity matching results. Each model in the database is preplanned with a series of robust grasps. We use a robotic gripper that is two-finger parallel type in our system and thus the grasps are all antipodal. To achieve pick-and-place tasks with the parallel-jaw gripper, we imitate the preplanned antipodal grasps on known models to plan robust grasps on target objects, and incorporate the idea of rotational adjustment to realize stable placements. Finally we apply the DD-RRT algorithm [64] to plan a collision-free trajectory for the motions of the robot.

High success rate of novel object pick-and-place based on training-free methods. Among the previous studies on novel object manipulation, few of them have achieved a very high success rate even combined with deep reinforcement learning [17]. But through our method, we have demonstrated by experiments that the idea of similarity matching can lead to an average success rate over 90% with in-category novel objects and around 75% with out-of-category novel objects. It has outperformed many existing learning-based methods, not to say other training-free methods.

3.2 Similarity Prediction

3.2.1 System Overview

The overview of our system is shown in Fig. 3.1. The inputs are one RGB image and one depth image including the novel object. Meanwhile there is an existing database consisting of a few known models with the information of categories, point clouds and preplanned robust grasps. We develop the following method to evaluate the similarity between the known models and target objects.

With the RGB image, we apply object segmentation and word embedding as the techniques of category-association matching to get an evaluated score of semantic similarity. With the depth image, we use PCL (Point Cloud Library) [65] to extract interested point clouds and then use SAC-IA (Sample Consensus Initial Alignment) [66] with ICP (Iterative Closest Point) [67] as the approach of point cloud registration to get another evaluated score of shape similarity. Combining the two scores, we can identify the specific model with the highest similarity to the target object from the existing database. With this model, we apply a surface segmentation method [68] to preplan a series of robust grasps on it. With these grasps, we use the transformation matrix

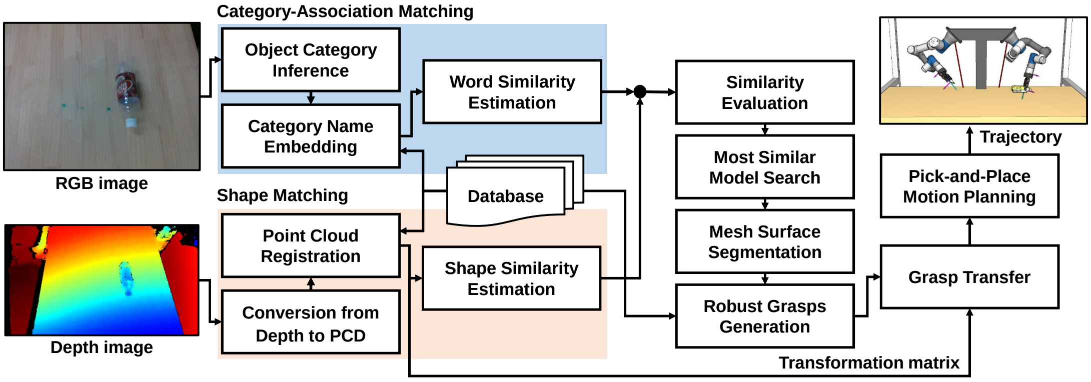

Category-Association Matching
RGB image
Object Category Inference
Category Name Embedding
Word Similarity Estimation
Shape Matching
Database
Point Cloud Registration
Conversion from Depth to PCD
Shape Similarity Estimation
Depth image
Similarity Evaluation
Most Similar Model Search
Mesh Surface Segmentation
Robust Grasps Generation
Transformation matrix
Trajectory
Pick-and-Place Motion Planning
Grasp Transfer

Figure 3.1: Overview of the proposed pick-and-place system. The inputs include one RGB image, one depth image and one existing database. The output is a complete pick-and-place motion trajectory executable by the robot.

obtained from point cloud registration to transfer them from the model to the target object, and develop an optimization algorithm to minimize the transfer error. Meanwhile, we develop a method of rotational adjustment to infer the in-hand posture of the target object and optimize the placement. Finally we apply the DD-RRT algorithm [69] to plan a collision-free trajectory for the pick-and-place motion. In the case that the planning fails in solving IK (Inverse Kinematics), the system will automatically select another grasp from the preplanned grasps and repeat the planning process until success.

3.2.2 Category-Association Matching (CAM)

Every object has a category which it belongs to, like *apple* belongs to *Fruit*, *cat* belongs to *Animal*. Category is an index used to classify things with similar features, it can give us the association of two objects. For instance, when the word *bottle* is given, we will easily associate it with a word like *water*, but will not relate to a word like *book*. In this case, we consider the two words *bottle* and *water* as high similarity, *bottle* and *book* as low similarity. With this definition, we can give a similarity prediction of category-association between two objects.

In our method, we first apply an object segmentation technique called Panoptic FPN (Feature Pyramid Networks) [70] derived from Detectron2 library[1] to extract semantic

---

[1] https://github.com/facebookresearch/detectron2

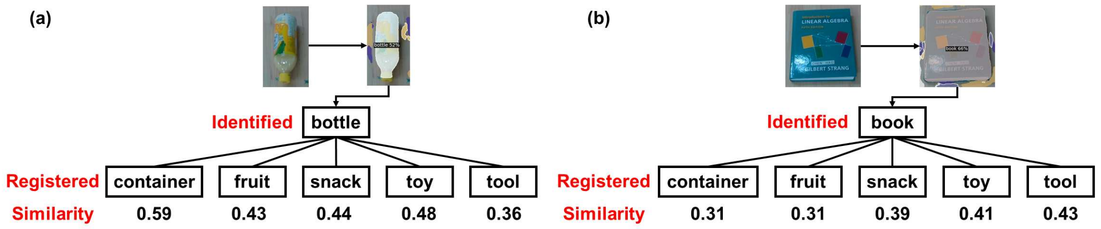


Figure 3.2: Example results of CAM using Panoptic FPN and Word2Vec as the approaches of image segmentation and word embedding. In two tests, (a) *bottle* and (b) *book* are detected. Scores shown below represent the category-association matching results between the novel object and the registered models in the database. The word similarity value is calculated in the embedded space.

information from an RGB image. From the extraction result we can identify the category of the target object and then match it with the registered categories in the database using a word embedding technique called Word2Vec [71]. The algorithm of Word2Vec will output a score between 0 and 1 representing the degree of category-association between the target object and registered models. As shown in Fig. 3.2, we test two novel objects to obtain the scores. One is *bottle*, an in-category object, showing a high score of 0.59 with category *container*, low scores with other categories. Another is *book*, an out-of-category object, showing low scores with all registered categories.

We can get two conclusions from the test results. First is, if the novel object belongs to one of the registered categories, we can easily distinguish it from the other categories by the results of CAM. Second is, if the novel object does not belong to any registered category, the results of CAM are not remarkable to classify the object and thus we need another matching method for further similarity evaluation.

3.2.3 Point Cloud Registration (PCR)

It is obvious that CAM can only help us identify the category of the target object, but is not sufficient to find a precise model with the highest similarity. As objects can be much different even if they belong to the same category, like *bottle* has the type of round-bottom and square-bottom, we still need a method to evaluate the shape similarity. Thus we incorporate PCR in our method.

Before implementing PCR, we have to extract the point cloud of the target object from the complex background. In our case, the target object is in a single state and placed on a fixed desk. The initial position of camera is unchanged before every motion. Based on these assumptions, we can simply use the algorithms of plane segmentation and outlier filter from PCL and set the threshold of coordinates to extract our interested point cloud. In this way, the point cloud of the target object is easy to obtain.

However, another important problem in processing the point cloud is that we can only get a partial view of the target object with only one camera. The incomplete point cloud may lead to unreliability of registration results. To solve this problem, we obtain two point clouds of the target object from different viewpoints by moving the in-hand camera. In this process, we only change the position of camera, but do not change its orientation. As shown in the upper part of Fig. 3.3, when the camera is set with the same orientation at two positions, wherever the object is, the relative distance of its position shown in two images is invariant. Based on this theory, we can concatenate the two point clouds obtained from different positions using a simple translation matrix. An example of concatenation is shown in the lower part of Fig. 3.3. Obviously the concatenated point cloud of a bottle is more complete than the single one, thus making the results of PCR more reliable.

The algorithm of PCR we use is SAC-IA [66] combined with ICP [67]. SAC-IA acts as the role of coarse registration while ICP acts as the role of fine registration. Like CAM, PCR also output a score representing the shape similarity. However, the difference is this score represents the mean of squared distances between two point clouds, which means good fitness will inversely lead to a low score.

3.2.4 Similarity Quantification

Both CAM and PCR are indispensable in our method of similarity prediction. Only CAM is not sufficient to find a precise model. Only PCR is not reliable enough due to the partially-observed situations. With the two scores obtained from the two matchings,

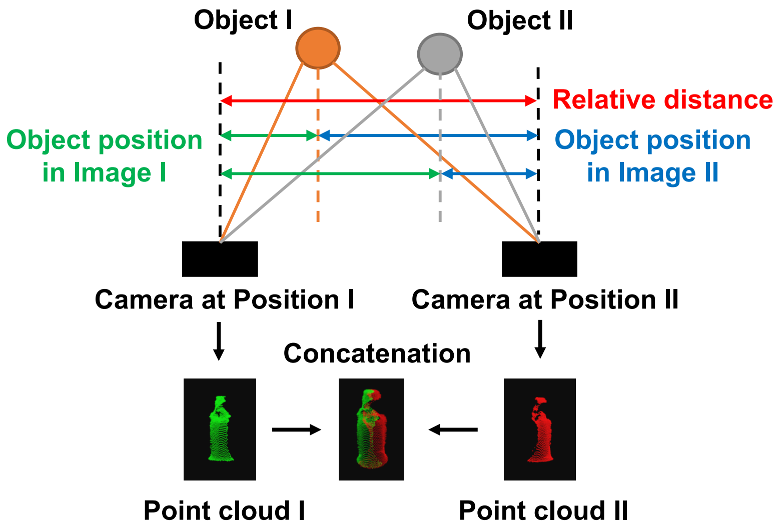


Figure 3.3: Invariant relative distance between the object's positions shown in two images when we make the orientation of the camera unchanged. With a certain translation matrix, we can easily concatenate two single point clouds to obtain a more complete point cloud of the target object.

we quantify the similarity with a simple function shown as below:

$$\hat{Q}(X,Y) = \mu_c \cdot C(X,Y) + \mu_p \cdot P(X,Y), \tag{3.1}$$

where $\hat{Q}(X,Y)$ is the evaluated similarity between the novel object X and the known model Y. $C(X,Y)$ represents the semantic similarity between X and Y related to the result of CAM. $P(X,Y)$ represents the shape similarity between X and Y related to the result of PCR. μ_c and μ_p are constant coefficients that determine the weights of semantic similarity and shape similarity.

As mentioned in Section 3.2.2, the score of CAM is a value between 0 and 1. If we directly use it as $C(X,Y)$, we also have to normalize the score of PCR to a value between 0 and 1 to be used as $P(X,Y)$. Through trials we found the scores of PCR are values within the range of 0.0001 to 0.1. Considering the equal distribution between 0 and 1, we introduce logarithm in the calculation of $P(X,Y)$. Also considering low score

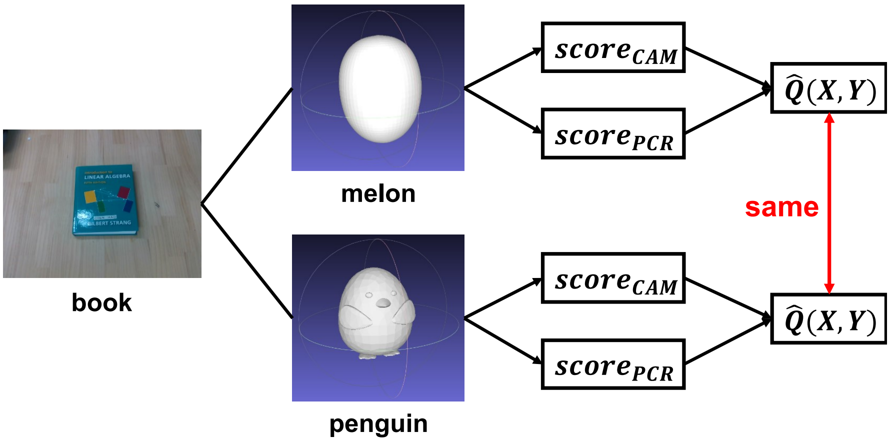


Figure 3.4: Identification of the constant coefficients μ_c and μ_p. We make use of an out-of-category object and two registered models with similar shapes but belong to different categories to calculate their $\hat{Q}$ values, which are assumed as the same.

represents high similarity, we identify the two parameters with the following equations:

$$C(X,Y) = score_{CAM}, \tag{3.2}$$

$$P(X,Y) = \frac{\log 0.1 - \log(score_{PCR})}{\log 0.1 - \log 0.0001}. \tag{3.3}$$

Another essential problem is the setting of μ_c and μ_p. In Section 3.2.2, we have tested that the out-of-category object *book* will output similar scores of CAM when matching with the registered categories in the database, which means *book* has similar semantic similarity with all the registered categories. In this case, if we find two models in different categories but with similar shapes from the database, they are supposed to have both similar semantic similarity and shape similarity with the object *book*, thus the final calculated $\hat{Q}(X,Y)$ should be almost the same. Based on this theory, we select the model *melon* from the category *fruit* and the model *penguin* from the category *toy* to do similarity matching with the object *book*, as shown in Fig. 3.4, and identify the constant coefficients μ_c and μ_p with the following equations:

$$\hat{Q}(melon, book) = \hat{Q}(penguin, book), \tag{3.4}$$

$$\mu_c + \mu_p = 1. \tag{3.5}$$

The result can be easily obtained as μ_c=0.52, μ_p=0.48. Then we can find the most similar model by ranking all the $\hat{Q}(X,Y)$:

$$Y^* = \text{argmax}(\hat{Q}(X,Y)), \tag{3.6}$$

where Y^* represents the specific model obtaining the highest value of $\hat{Q}(X,Y)$.

3.3 Pick-And-Place Planning

3.3.1 Grasp Transfer

During the process of PCR, besides the score of similarity we also get a transformation matrix between the model and the real-world object. With this matrix we can transfer the preplanned robust grasps on known models to the similar grasps on novel objects. Thus we first apply an approach of grasp planning based on the superimposed segmentation of object meshes [68] to preplan a series of robust grasps, and then apply the transformation matrix from PCR to transfer them from registered models to real-world objects to realize imitation learning of robust grasps.

However, the real condition is that the transferred grasps are subject to error due to the inaccuracy of PCR. As shown in Fig. 3.5, the incomplete point cloud of the target object (green) is not perfectly aligned with the point cloud of its similar model (red) in position and orientation due to the partial observation. Thus the transformation matrix obtained from PCR is inevitably subject to error. With this imprecise matrix, the calculated position of the novel object (yellow) deviates from its real position (white). In this case the transferred grasp has a possibility of failing if the error is large.

To reduce the error, we first obtain point clouds of the target object from several viewpoints and apply PCR to get a set of transformation matrices. We extract the rotation and translation information from the matrices and calculate the mean results of position and orientation of the grasps. Among them we remove results that obviously deviate from other results (orientation error $> 90°$) and obtain the final transformation

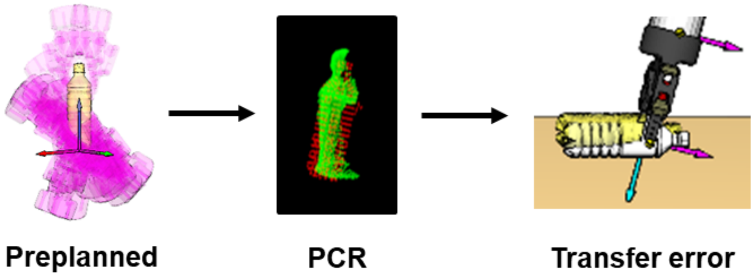


Figure 3.5: Preplanned grasps on a known model and error in PCR and grasp transfer. (Left) A model is preplanned with a series of robust grasps. (Middle) In PCR, the point cloud of the registered model (red) is not perfectly aligned with the partially-observed point cloud of the target object (green). (Right) Due to the error of the transformation matrix, the calculated position of the target object (yellow) deviates from the real position (white).

matrix T.

To improve the precision further, we develop an algorithm to minimize the error. With the matrix T, we compare its error to a threshold δ. If the error is small enough, the algorithm ends, otherwise we move the camera to a new viewpoint to do calculation again and update the final transformation matrix to be $T^{'}$. Then we check the error of the new matrix. If the error increases, we discard this result and move the camera to another new viewpoint. If the error decreases, we compare it to the threshold δ again. If the error is small enough, the algorithm ends, otherwise we reserve this result and move the camera to another new viewpoint. We repeat this process until the error becomes small enough.

In this way, the error of grasp transfer can be controlled well in any case. In our method, the threshold δ is set to be 0.0001 (same as the lower bound of PCR scores).

3.3.2 Stable Placement

Above we have solved the problem of grasping, but a remaining problem is how to place the object stably in desired posture. Unlike general methods, we do not need an extra method to estimate the initial posture of object like [72]. Taking the advantage of a known database, we can simply set the final posture of the target object according to its

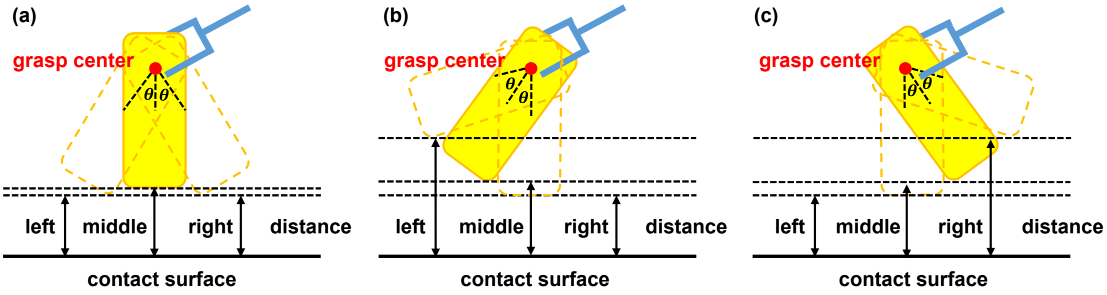


Figure 3.6: Inference of the in-hand posture. We rotate the in-hand object with a small angle θ in two inverse directions by the grasp center. When the original posture is upright (a), the distance between the lowest point of the object and the contact surface does not differ much before and after rotations. When the original posture is right-leaning (b) or left-leaning (c), the distance will differ much in one of the rotations.

similar model. However, due to the inevitable error in grasp transfer, the final placement has the risk of being unstable as the in-hand posture of the target object may not be so good as in ideal condition.

Considering an in-hand model for analysis of stable placements shown in Fig. 3.6, we assume an object to be tall and thin, which is not easy to be placed stably when tilted. To find a stable posture, we rotate the object with a small angle θ in two inverse directions by the grasp center. The rotational plane is selected to be parallel with the two-finger gripper because the object's posture is mostly undetermined in this 2D plane. As seen from the figure, when the object is in an upright posture, the lowest point of the object after rotation in two directions almost share the same distance to the contact surface. But when the object is in a right-leaning or left-leaning posture, the distance becomes much different in one of the rotational directions.

The difference in distance become the key of inferring the in-hand posture of the target object. To measure the distance, we move the object right down to contact the surface in different rotations and obtain the timing of collision by a force sensor. If the time cost before collision is close in different rotations, the in-hand object is supposed to be in an upright posture. If the time cost is much larger in one of the rotations, the in-hand object is supposed to be right-leaning or left-leaning. After knowing the actual in-hand posture, we adjust the object with a small rotation to be placed in a more stable posture. It should be noted that our method of stable placement is based on the following assumptions:

1) The in-hand posture of object does not deviate much from the upright posture;

2) The object hardly slips in horizontal direction when collided with contact surface;

3) The object has a tall and thin shape similar to the object assumed in Fig. 3.6.

In our case, the grasp transfer is controlled to an acceptable error so the first assumption is satisfied. The second assumption depends much on the rotation angle we set. The third assumption seems to be a limitation to our method but actually not a matter when extended to other shapes of objects. The case we consider is a kind of object most likely to be unstable when not in an upright posture. Other kinds of objects with shorter and fatter shape are likely to keep stable in placement even if they are right-leaning or left-leaning.

3.3.3 Motion Planning

After identifying the initial grasp and the final placement, we finally generate a collision-free trajectory of pick-and-place motion using the DD-RRT algorithm [69]. However, due to the uncertainty in the posture of preplanned grasps, the motion planning has a possibility of failing in solving IK (i.e. the calculated position and orientation is unreachable by the robot). In this case, we iterate the algorithm with another grasp until the planning succeeds. The framework of our algorithm is shown in Algorithm 1. To ensure a feasible path can be output, sufficient preplanned robust grasps are required for each model.

3.4 Experiments

To verify our method, we perform real-world experiments of robotic pick-and-place with 12 novel objects, as shown in Fig. 3.7. Among them some are not included in the registered categories (e.g. *toothbrush*), considered as out-of-category novel objects. Others

Algorithm 1: Pick-and-Place Motion Planning

```
Input: A set of preplanned grasps on the known model, $G$
Transformation matrix obtained from PCR, $T$
Final posture of the novel object, $p_f$
Output: Pick-and-place motion trajectory, $M$
Initial posture of the novel object $p_i$ is known by $T$
for each grasp $g \in G$ do
    $M \leftarrow$ DDRRT $\{g, p_i, p_f\}$
    if $M$ is None then
        continue
    else
        break
return $M$
```

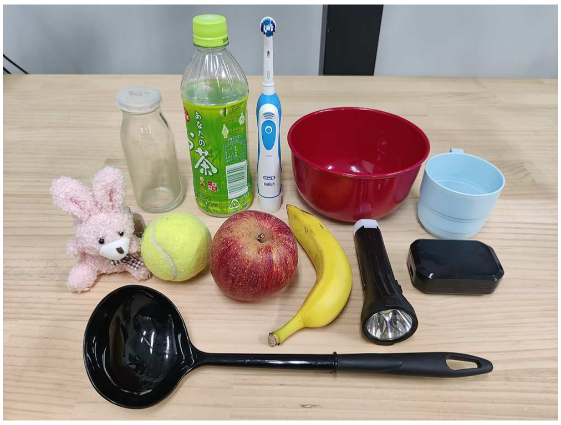


Figure 3.7: A dozen of novel objects used for validation experiments. For each object we find an evaluated most similar model from the database and imitate its preplanned robust grasps to achieve pick-and-place tasks.

like *apple* included in the category *fruit* are considered as in-category novel objects.

3.4.1 Experimental Setup

The configuration of our experiment is one Realsense D435 depth camera, one single-armed UR3e robot, and one Robotiq 2F-85 gripper. The novel object is posed in arbitrary position and orientation on a fixed desk.

Meanwhile, we construct a database composed of 100 models in five different categories: *container, fruit, snack, toy, tool.* The registered information includes the cate-

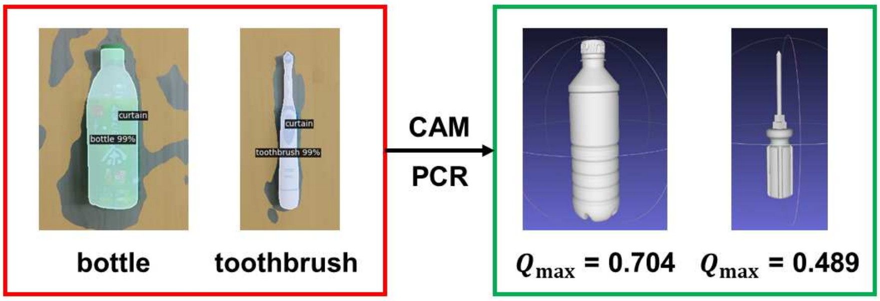


Figure 3.8: Example results of similarity prediction based on CAM and PCR. The red block represents the results of semantic identification by Panoptic FPN. The green block shows the most similar models found with the highest Q values.

gories, point clouds and preplanned antipodal grasps of the models.

3.4.2 Similarity Prediction

We use the depth camera to obtain RGB and depth images including the novel object. With RGB images, we apply image segmentation to extract the category information of novel objects. An example is shown in the red block of Fig. 3.8. The in-category object *bottle* and the out-of-category object *toothbrush* are both correctly identified. With the category names obtained, we input them into CAM to obtain the score of semantic similarity between the novel objects and known models in the database. With depth images, we extract the point clouds of novel objects from the complex background by PCL and input them into PCR to obtain the score of shape similarity. Combining the two scores, we calculate the Q value by Equation (3.1) for each matching and find the most similar model with the highest Q value. As shown in the green block of Fig. 3.8, a *bottle* model and a *screwdriver* model are found with the highest similarity to the object *bottle* and *toothbrush* respectively.

In our experiments, the in-category objects all succeed in finding a similar model both in semantic level and geometric level from the database. The out-of-category objects behave differently in similarity prediction. Although Fig. 3.8 shows a positive result of similarity prediction with the out-of-category object *toothbrush*, it is not always the case. Due to the limitation of image segmentation methods and partially-observed

Table 3.1: Experimental results of in-category object pick-and-place.

Identification result	Bottle	Bottle	Cup	Bowl	Apple	Banana	Teddy bear	Sports ball
Total experiments	21	23	20	19	23	20	22	22
Successful times	19	20	18	18	19	20	22	18
Success rate (%)	90.5	87.0	90.0	94.7	82.6	**100**	**100**	81.8

situations, the results of CAM and PCR are not always reliable. For instance, the out-of-category object *black box* used in our experiment is wrongly identified as *cell phone*, which is not the truth. And the final matching result comes to be a *candy* model in the category *snack*, which is non-similar (a negative result of similarity prediction). We also experiment on such of cases to see to which extent the success rate will differ.

3.4.3 Novel Object Pick-and-Place

With the reference models identified by similarity prediction, we preplan over 100 robust grasps on each of them and transfer these grasps from the models to real-world objects with the transformation matrix obtained from PCR. We incorporate the DD-RRT algorithm to plan an IK-solvable collision-free trajectory to reorient the target object on the basis of the transferred grasps, and adjust small rotations to achieve stable placements.

For each object, we experiment about 20 times of pick-and-place motion with arbitrary initial position and orientation. Our goal is to successfully grasp the target and place it at a specified position stably (no movement occurs after placement). The final posture is determined to be the same as the posture of the reference model. Either the failure of grasping or placing will be counted as failure cases. Fig. 3.9 shows two successful cases of pick-and-place with the in-category object *bottle* and the out-of-category object *toothbrush*. From the total experimental results, the in-category objects show a high average success rate of 90.6%, while the out-of-category objects also perform not badly with an average success rate of 75.9%.

Some interesting facts are found from the experimental results. The two bottles (one plastic bottle and one glass bottle) both show a high success rate of manipulation

Table 3.2: Experimental results of out-of-category object pick-and-place.

Identification result	Toothbrush	Spoon	Flashlight	Cell phone
Total experiments	21	19	23	20
Successful times	14	18	19	12
Success rate (%)	66.7	**94.7**	82.6	60

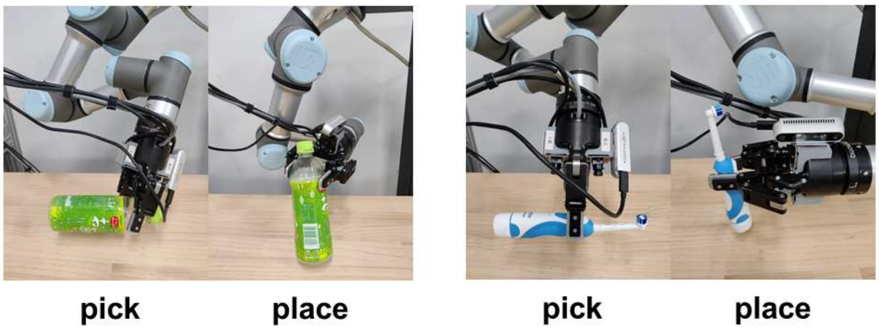


Figure 3.9: Realization of pick-and-place motions based on the results of similarity prediction. The grasp postures of the bottle (left) and the toothbrush (right) are generated from the preplanned grasps on the bottle model and the screwdriver model in Fig. 3.8.

in our experiment. Transparent objects are difficult to recognize in depth sensors, thus are difficult to plan robust grasps based on their point clouds. But through our method incorporating semantic identification, they can find similar models in the category *bottle* to imitate their robust grasps, thus performing much better than traditional methods. The *apple* and the *sports ball* both show a relatively low success rate among in-category objects. This is due to their round shapes. Although we have minimized the error in grasp transfer, it is inevitable that the contact points on the real-world objects are not completely consistent with the contact points on the reference models. In this case, slipping is more likely to occur when grasping objects with round shapes.

Moreover, we find the out-of-category objects differ much in success rate of manipulation. The *spoon* achieves the highest success rate due to its similarity matching result to be a *spanner*, which shares similar contact points. The flashlight takes the second place for the same reason. However, the other two objects, *toothbrush* and *black box* perform much worse. The shape of the *toothbrush* determines it to be difficult to place stably in a standing posture. Although it can be grasped successfully in most cases, it still fail many times when executing the placement. The *black box* is wrongly identified as a *cell phone* and its similarity matching result is a *candy*, which is actually non-similar. An interesting phenomenon is even if in this case, the success rate does not become very low. The reason can be analyzed as follows:

When the identified category name does not belong to any registered category in the database, the scores of CAM are often very low. In this case, the final matching result strongly depends on the scores of PCR. As the score of PCR represents the mean of squared distances between two point clouds, a model of smaller size is likely to get lower score (which will lead to larger Q value, referred to Equation (3.2)) when matched with a partially-observed novel object. Thus the reference model found by similarity prediction is likely to be smaller than the real-world object. As PCR will align two point clouds together, the preplanned robust grasps on a smaller model have a high possibility of being placed within the graspable area of a larger object after transfer. In this case, even if the selected model is non-similar to the real-world object, the transferred grasps still have a fairly high success rate of manipulation. This is considered to be a significant advantage of our method in dealing with completely novel objects.

3.5 Discussion

Through experiments we verify our proposed method to be able to achieve a high success rate of novel object manipulation. There are two remarkable advantages of our method compared with other related works. One is the generalization, the techniques used in our system: image segmentation (Panoptic FPN), word embedding (Word2Vec), and point cloud registration (SAC-IA with ICP) are all replaceable. Moreover, our method can deal with both in-category objects and out-of-category objects. Although the similarity prediction are not correct at all time, its precision can be raised by enlarging the database to increase the possibility of a novel object finding a similar registered model. Another is the simplicity, we only use one depth camera and one robot hand to achieve the whole pick-and-place task. Our system is easy to construct and all the hardware can be substituted with other types.

3.6 Conclusions

In this study, we propose a new perspective of novel object manipulation planning by imitating existing robust grasps. We combine category-association matching with point cloud registration to evaluate the semantic and shape similarity between novel objects and known models. We transfer the preplanned robust grasps from registered models to real-world objects and develop an optimization algorithm to minimize the transfer error. We also incorporate the idea of rotational adjustment to infer the actual posture of in-hand objects and achieve stable placements.

Chapter 4

Multi-Level Similarity Matching for Single-View Object Grasping

This thesis chapter originally appeared in the literature as

H. Chen, T. Kiyokawa, Z. Hu, W. Wan, and K. Harada, "A Multi-Level Similarity Approach for Single-View Object Grasping: Matching, Planning, and Fine-Tuning," *IEEE Transactions on Robotics*, 2025. (Accepted)

4.1 Introduction

Both industrial and service robots are required to deal with a wide range of objects with diverse categories, shapes and arrangements. In traditional robotic systems, detailed properties of each object have to be known and then specific actions can be designed for each manipulation task. However, this process becomes increasingly labor-intensive as the number of object types grows, while remaining ineffective for previously unseen objects. Consequently, there is a pressing need for highly dexterous robotic systems capable of handling a wide variety of novel objects without prior information.

In the last decade, the booming development of vision technology and deep learning has led to many outstanding works in the field of novel object grasping [73]. Representative studies among them use depth images [11, 34], RGB-D images [14, 74] or point clouds [13, 75] as input representations to train neural networks for grasp detection and evaluation on unseen objects. While achieving notable performance, they have limitations in the grasping direction (e.g. only top-down grasping) or rely on high-precision visual features for accurate grasp inference. For improvement, more recent studies employ scene representations [18, 19] and large-scale datasets [22, 23] to provide efficient and adequate training for more generalized grasping systems. However, they still suffer from the disadvantages of high training costs and high sensitivity to sensing noise and environmental changes. Therefore, we recognize that a new perspective beyond traditional learning frameworks is needed to achieve a higher standard of general object grasping, leading us to explore the idea of similarity matching [30, 32].

A recent enlightening work [33] introduces a score-based similarity evaluation framework that assesses object similarity from both semantic and geometric aspects. This method effectively transfers grasping knowledge from similar known objects to enable efficient grasping of unknown target objects. However, it requires multi-view observations and exhibits instability as the variety of target objects increases, primarily due to the difficulty in balancing the quantification of semantic and geometric similarity. Our study significantly advances this work by tackling the challenge of single-view object grasping and developing a more efficient similarity matching approach that fully leverages available visual features. We introduce a novel multi-level matching framework that independently evaluates object similarity across semantics, geometry, and dimensions, avoiding the balancing issues of using a composite scoring function. Notably, we propose the C-FPFH (Clustered Fast Point Feature Histogram) descriptor for geometric matching, which, to the best of our knowledge, is the first geometric descriptor capable of accurately evaluating the similarity between partial and complete point clouds of non-identical objects, demonstrating exceptional effectiveness in handling occlusions. In addition to accurate matching, we incorporate a stability-aware fine-tuning process to optimize the quality of imitative grasps generated from similar references, providing an auxiliary guarantee for achieving robust grasping.

Through extensive real-world experiments with a diverse range of novel objects in

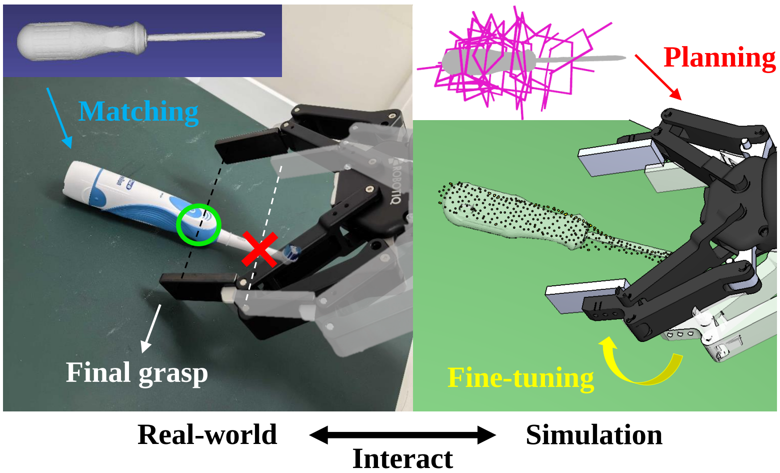


Figure 4.1: An example of grasping a novel object using our three-step method. First, visual features of the target object obtained in the real world are used for similarity matching with existing database models. Then, grasp planning and fine-tuning are performed in simulation based on the matching results. Finally, the optimized grasp is executed in the real world to achieve the task.

both isolated and cluttered scenes, we show that our method, using a small database of fewer than 100 object models, significantly outperforms state-of-the-art (SOTA) benchmarks across key metrics including accuracy, efficiency, and generalization. Fig. 4.1 illustrates an example of grasping a novel object (a toothbrush) using our proposed method. Through visual detection and similarity matching, a screwdriver model with grasping knowledge is identified from an existing database to plan imitative grasps for the toothbrush. A subsequent fine-tuning process is then applied to optimize the grasp quality by positional adjustment. The optimized grasp is finally executed to complete the task. The entire system operates through a seamless interaction between the real world and the simulation.

Our main contributions can be summarized as:

- We propose a multi-level similarity matching approach that integrates semantic, geometric, and dimensional features to efficiently identify potential similar candidates from an existing database for the unknown target object.
- In geometric matching, we introduce the C-FPFH descriptor, a novel feature descriptor designed to accurately assess similarity between partial and complete point

clouds.

- We develop several new methods to enhance the accuracy of similarity matching and grasp planning by exploiting and improving existing techniques such as Large Lanuage Models (LLMs), Oriented Bounding Boxes (OBBs), and plane detection in point clouds.

- We implement a two-stage fine-tuning process after generating imitative grasps to optimize grasp stability based on the local features of observable contact points.

4.2 Methods

4.2.1 Overview

The core idea of our approach is to leverage visual features from single-view observations to identify similar references from an existing database of known object models to guide the grasping of unknown target objects. The main difficulty lies in achieving accurate similarity matching and robust grasp planning in the presence of large visual uncertainty.

Fig. 4.2 illustrates an overview of our methodology. The first step involves extracting object features from single-view RGB-D inputs using segmentation models, including category information and 3D point clouds. The second step applies a multi-level similarity matching approach with an object model database consisting of the following components:

1) *Category names*, used for semantic matching with the detected object category through LLM assistance;

2) *C-FPFH descriptors*, used for geometric matching with the object 3D point cloud via feature comparison;

3) *Bounding extents*, used for dimensional matching with the target object by utilizing a novel type of OBB.

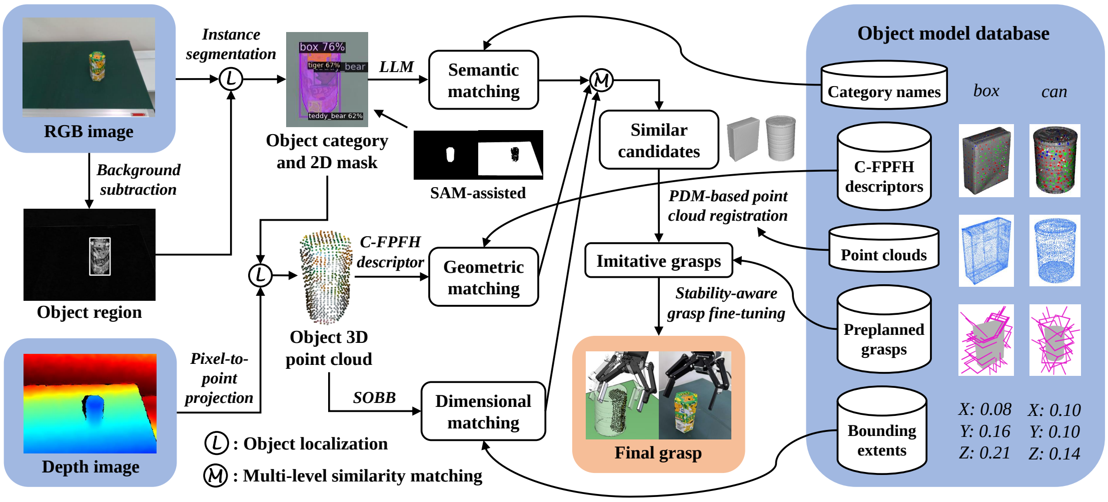


Figure 4.2: Overview of our proposed approach. The system inputs (shown in blue boxes) include a single-view RGB-D image and an existing object model database. A background image without target objects is taken beforehand. The output (shown in the orange box) is an optimized grasping action for execution.

Based on the three levels of matching, we filter a list of similar candidates from the database that are most likely to exhibit high similarity to the target object. For these candidates, we further rank their similarity by leveraging another database component: 4) *Point clouds*, to perform point cloud registration with the object point cloud using a plane-detection-based approach. Based on the registration results, we begin with the most similar candidate and transfer its existing grasping knowledge from the database: 5) *Preplanned grasps*, to generate imitative grasps for the target object.

In the final step, all generated grasps undergo a two-stage fine-tuning process to optimize stability, considering the local features of observable contact points.

4.2.2 Single-View Object Recognition

Our goal is to achieve single-view object grasping under the uncertainty of sparse and noisy visual inputs. To achieve this, we use a consumer-grade 3D camera mounted on the robot end effector for object detection. We first capture a background image without the target object, and then use it as a reference for background subtraction by performing gray-scale differencing with the image containing the object to identify the region where the object is located. Based on this approximate localization, we apply instance segmentation to the RGB input using a SOTA pre-trained model [76] and extract only the results related to the target object, including the object category and its 2D mask.

To remove redundant detection results such as the patterns on the object surface, we also examine the inclusion relationships among different results and retain only the one with the largest coverage within the object region. In cases where the target object fails to be detected (confirmed by comparing the object region with all detection results) due to irregular observation angles or occlusions, we integrate a more powerful segmentation model, SAM [77], to acquire the object mask without category information. Based on this mask, we extract the pixels containing the target object from the depth input and project them into 3D space to obtain the object point cloud.

It should be noted that the object region identified by background subtraction cannot replace the 2D mask acquired by instance segmentation since it lacks pixel-level accuracy and is highly susceptible to lighting variations. The object category and 3D point cloud obtained during the visual detection process are key features for achieving similarity matching.

4.2.3 Multi-Level Similarity Matching

Existing learning-based grasp planning approaches struggle to maintain their optimal performance under varying sensing conditions. To address this issue, we avoid directly regressing grasps from visual inputs. Instead, we utilize existing similar objects with reliable grasping knowledge to guide the grasping of novel objects, which is robust to noise and environmental changes through an appropriate matching framework.

Based on this consideration, we propose a multi-level similarity matching approach that leverages the visual features of the target object to identify reference object models from an existing database, selects candidates from semantic, geometric, and dimensional perspectives, respectively, and finally synthesizes the results to determine an optimal selection of similar candidates. Compared with score-based methods [33], the multi-level matching framework exhibits greater robustness in similarity evaluation by separately assessing potential candidates from each perspective, thus avoiding influences across different perspectives. The implementation details of each matching are illustrated in the following sections.

4.2.4 LLM-Assisted Semantic Matching

The object category obtained in visual detection can be used to identify similar candidates from the database at the semantic level. A common approach for evaluating semantic similarity between two categories is to use word embedding models such as Word2Vec [71] and GloVe [78] to obtain scores representing their cosine similarity (e.g., *bottle* has a high score with *box*, but a low score with *car*). However, this method has several significant drawbacks that can mislead the results of similarity matching: 1) The scores only reflect word similarity in linguistics, not relevance in robotic grasping; 2) The scores vary significantly across different pairs of similar categories; 3) Different meanings of a polysemous word cannot be distinguished. Fortunately, with advancements in LLMs, we address these issues by integrating the GPT-4o model [79] for similarity evaluation, which provides comprehensive knowledge of object similarity specific to robotic grasping and a better understanding of user intent, as implemented below:

Prompt: Which objects in [*bottle, box, cup, mug, apple, hammer*] are likely to be similar to a {*soda_can*} in terms of robotic grasping? Please only answer the category names.

Answer: Bottle, cup, mug.

In the first square bracket, we input all category names contained in the database, simplified to single or compound words that indicate object identities without descriptive terms (e.g., for objects in the YCB dataset [80], 051_large_clamp → clamp, 053_mini_soccer_ball → soccer_ball). In the second curly bracket, we input the detected object category without any simplification, as the descriptive terms in the detection results, e.g., *mouse_(computer_equipment)*, help to better clarify the target object. From the GPT answer, we can easily select candidate models with corresponding category names that are potentially similar to the target object from a semantic perspective. In case that the object fails to be detected and no category information is available, we skip semantic matching and only consider similarity from other perspectives.

4.2.5 C-FPFH-Based Geometric Matching

The object point cloud obtained in visual detection can be used to represent the geometric properties of the target object, enabling the identification of similar candidate models at the geometric level. However, incomplete point clouds from single-view observations introduce significant uncertainty due to large unseen regions and sensing noise, making it extremely difficult to accurately assess the similarity between an observed partial point cloud of an unknown object and a complete point cloud of a database model. To address this issue, rather than exploring the global geometric similarity between partial and complete point clouds, we find it more effective to extract their local geometric features and leverage feature correspondences to represent their similarity. To achieve this, we adopt a point cloud feature extractor, the FPFH descriptor [81], which describes the local geometry of each point in a point cloud using a 33-dimensional vector showcasing the distribution of neighboring normal orientations. It has the advantage of rotational invariance and is suitable for our task where the camera observation pose is uncertain. Utilizing its principle, we propose a novel point cloud geometric descriptor, the C-FPFH descriptor, which aggregates and classifies local geometric features to distill the essential information in a point cloud, enabling accurate similarity evaluation regardless of point cloud completeness.

As shown in Fig. 4.3, generating a C-FPFH descriptor for a point cloud involves three steps: 1) Uniformly sample points using a voxel grid filter of appropriate size (1.5 cm in our task) and exclude edge points where the estimated normals probably deviate from the truth due to the uncertainty of unseen regions. 2) For each sampled point, compute its FPFH descriptor with neighboring points within an appropriate local area

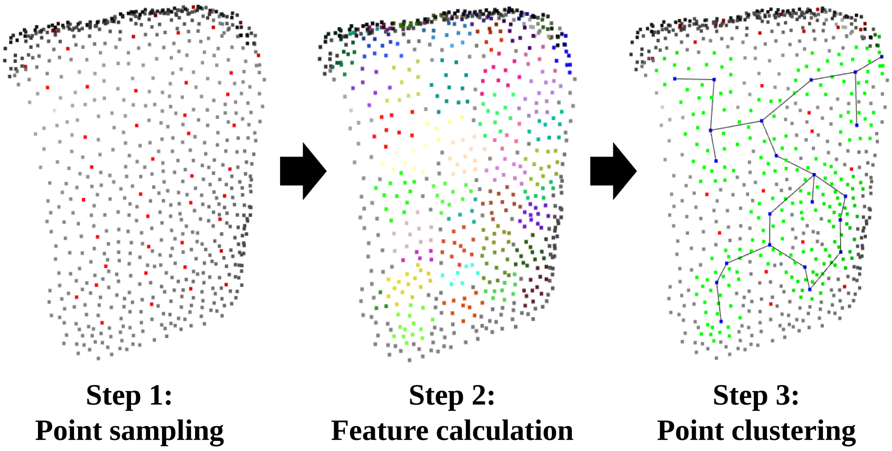


Figure 4.3: Three-step generation of C-FPFH descriptors for object point clouds. Due to sensing noise, point features may occasionally vary across surfaces with similar geometry; however, such variations are acceptable during geometric matching, facilitated by our discrete sampling and consecutive clustering methods.

(a sphere with a radius of 1 cm in our task) to obtain a 33-dimensional vector describing its local geometry. To minimize the effect of sensing noise, instead of using all the values in the vector, we extract only the index numbers of the two most dominant of the 33 vector components as a feature pair to represent the main characteristics of a sampled point. We employ two components rather than one to account for errors caused by noise when the two largest values are close. We aggregate the feature calculation results of all sampled points as: $\{(f_{11}, f_{12}) : n_1, (f_{21}, f_{22}) : n_2, ...\}$, where $(f_{k1}, f_{k2}) : n_k$ is an unordered feature pair with its number of occurrences in the point cloud. Based on this aggregation result, we develop the first metric for similarity evaluation between a partial point cloud p and a complete point cloud c, called QS (quantitative similarity), which is calculated as:

$$\mathrm{QS} = \frac{\sum_{i=1}^{m} \min({}^{p}n_i, {}^{c}n_i)}{\sum_{i=1}^{m} {}^{p}n_i} \tag{4.1}$$

where m is the number of types of feature pairs in p, ${}^{p}n_i$ and ${}^{c}n_i$ are the number of times the same feature pair (f_{i1}, f_{i2}) occurs in p and c, respectively (${}^{p}n_i$ is always greater than 0, while ${}^{c}n_i$ can be 0). A large QS indicates that c contains most of the features in p, in which case it is highly probable that they are similarly shaped objects. However, this metric has limitations as it only evaluates the overlap rate of identical features without considering their spatial distribution. As a result, two dissimilar point clouds containing a large number of discrete identical features may be evaluated as highly similar. Therefore, an additional metric that captures the similarity of feature distribution is essential to complement the evaluation.

3) Considering that the surface features in p are not as complete as those in c, the comparison of feature distribution needs to be performed locally rather than globally. To achieve this, we incorporate the idea of the CVFH descriptor [82] to cluster all sampled points as follows: I) Take an unclustered sampled point as a seed, search its surrounding area (a sphere with a radius slightly larger than the sampling voxel size, 2 cm in our task) to identify other sampled points with the same feature pair and similar normal orientations (differing by no more than 20 degrees in our task); II) If such points are found, connect them as consecutive points and take them as new seeds, continuing the search until no more consecutive points are found; III) Repeat I and II. In this way, each sampled point can be assigned to a unique cluster. All clusters are confined to a single surface without crossing the edges, ensuring that the feature distribution is evaluated in a localized manner. For each cluster with more than two points, we apply principal component analysis (PCA) to obtain three normalized singular values as a representation of its spatial distribution. We aggregate the distribution analysis results for clusters with the same feature pair as: $\{(f_{k1}, f_{k2}) : d_{k1} = [\sigma_{11}, \sigma_{12}, \sigma_{13}]_k, d_{k2} = [\sigma_{21}, \sigma_{22}, \sigma_{23}]_k, ...\}$. For p, we only focus on the distribution of the main features (the cluster containing the highest number of consecutive points with the most frequently occurring feature pair) and obtain its PCA result as ${}^{p}d_s = [\sigma_1, \sigma_2, \sigma_3]_s$, where s denotes that (f_{s1}, f_{s2}) is the main feature pair of p. We ignore unimportant clusters and feature pairs for the presence of sensing noise. For c, we search among the clusters of (f_{s1}, f_{s2}) to identify any PCA result that is close to ${}^{p}d_s$. When such a result exists, we assume that c contains the main point cluster of p. Based on this principle, we develop the second metric for similarity evaluation, called DS (distributional similarity), which is calculated as:

$$\mathrm{DS} = \min_{1 \leq j \leq l} \|{}^{p}d_s - {}^{c}d_{sj}\| \tag{4.2}$$

where l is the number of clusters with feature pair (f_{s1}, f_{s2}) in c. A small DS indicates that c has a region very similar to the main part of p.

The aggregation results of feature calculation and distribution analysis form the C-FPFH descriptor. To save computation time in similarity matching, we pre-generate C-FPFH descriptors for all complete point clouds in the database. This allows us to compute only the descriptor for the partial point cloud of the target object during real-time processing. By setting appropriate thresholds for QS and DS, we can determine the range of similar candidate models from a geometric perspective. Empirically, we assume that when QS > 0.9 and DS < 0.1, p has high geometric similarity to c and the model is selected as a candidate. These two thresholds can be adjusted to increase or decrease the number of candidates; however, setting them too high or too low can negatively impact the matching results.

Fig. 4.4 showcases several test results of using C-FPFH-based geometric matching to identify similar candidate models from single-view object point clouds. The results indicate that most of the selected models demonstrate overall similarity to the target objects. However, some candidates display only partial similarity (marked with red circles in Fig. 4.4) due to the inclusion relationship between p and c, revealing a potential limitation of using local features for similarity evaluation between partial and complete point clouds.

4.2.6 SOBB-Guided Dimensional Matching

To address the limitation of partial similarity in geometric matching, we introduce a third perspective for similarity evaluation: dimensional matching, which assesses the size similarity between the target object and database models by leveraging their dimensional features. For size evaluation of object models or point clouds, 3D bounding boxes such as the axis-aligned bounding box (AABB) and the oriented bounding box (OBB) are commonly used to represent dimensions including length, width, and height through the three bounding extents. However, in our task, neither AABB nor OBB can accurately represent the dimensions of the target object from single-view partial point clouds due to the uncertainty of object poses and large unseen regions, as shown in Fig. 4.5. To address this issue, we develop a new type of bounding box, called semi-oriented bounding

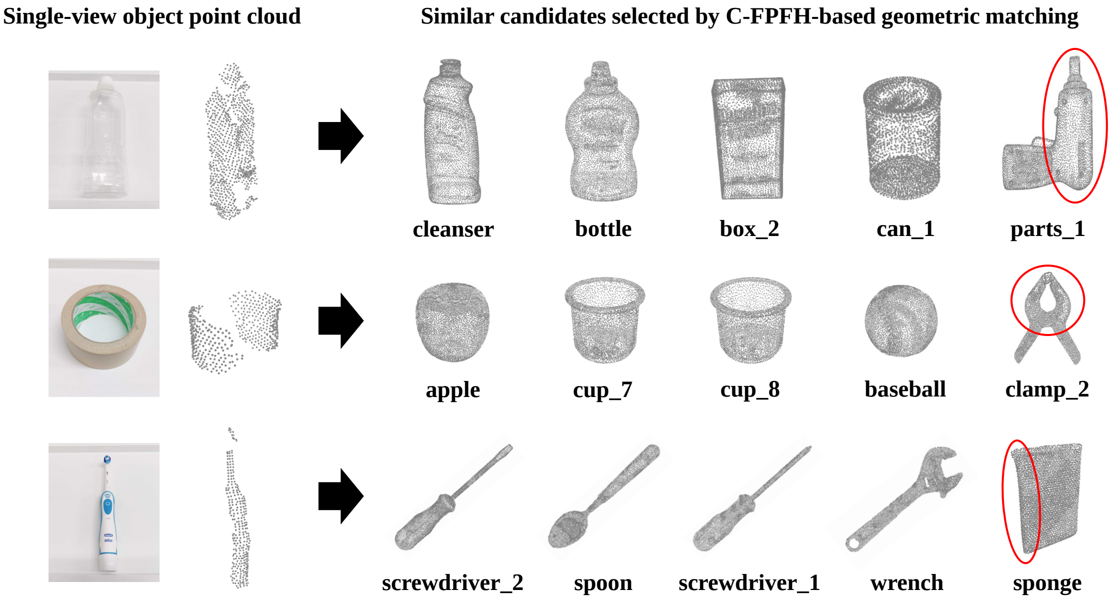


Figure 4.4: Test results of using C-FPFH-based geometric matching to identify similar models from single-view object point clouds. All candidate models are originally from the YCB dataset, renamed with simplified category names and index numbers for cases where multiple models exist within a single category.

box (SOBB), which fixes one direction of the OBB by aligning it with the normal vector of the plane on which the target object is placed.

The generation of the SOBB for an object point cloud involves the following steps: 1) In the robot coordinate system, obtain the normal vector $\vec{n}$ of the plane where the object is placed (for table-top objects in our task, $\vec{n} = \langle 0, 0, 1 \rangle$); 2) Project all points onto a plane perpendicular to $\vec{n}$ and passing through the origin; 3) Downsample the projected points using a grid filter to achieve a uniform density; 4) Apply PCA to the downsampled points to extract the two principal components, $\vec{u}$ and $\vec{v}$, which represent the 2D distribution of the points within the plane; 5) Generate a new coordinate system using normalized $\vec{n}$, $\vec{u}$, and $\vec{v}$ as the orthogonal basis vectors, and compute the AABB of the point cloud in this new coordinate system. By transforming this AABB back to the original coordinate system, we obtain the target SOBB, which accurately represents the dimensions of the target object.

For evaluation of size similarity, we pre-compute the bounding extents (x_i, y_i, z_i) for all database models by aligning their original poses with the axis orientations and

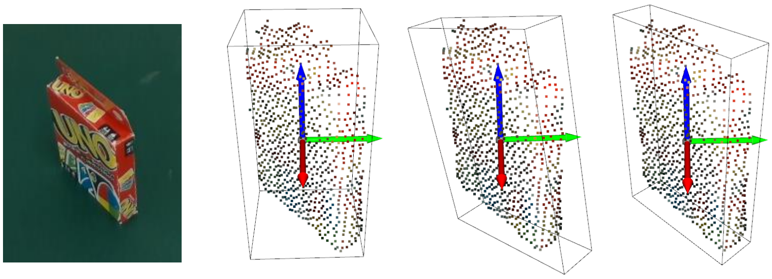


Table-top object **AABB** **OBB** **SOBB**

Figure 4.5: Comparison of different types of bounding boxes applied to a single-view partial point cloud obtained from a table-top object.

obtaining their corresponding AABBs. During real-time processing, we only need to compute the SOBB extents (x, y, z) of the object point cloud and use its dimensional differences with the database models to represent their size similarity (SS) as follows:

$$\mathrm{SS} = \|\mathrm{sort}(x, y, z) - \mathrm{sort}(x_i, y_i, z_i)_{1 \leq i \leq h}\| \tag{4.3}$$

where h is the number of database models, sort() is a function that arranges values in descending order. All calculations are performed in meters. Empirically, we assume that when $\mathrm{SS} < 0.1$, the model is similar in size to the target object and can be selected as a candidate at the dimensional level. This threshold is also adjustable but should remain within an appropriate range, similar to QS and DS.

4.2.7 Candidate Model Selection

The above implementations indicate that each individual level of similarity matching has its limitations. For example, semantic and dimensional matching only provide a coarse identification of potential similar candidates without investigating into detailed object features, whereas geometric matching leverages object features thoroughly but is prone to partial similarity results. Therefore, we propose that the most effective approach for candidate model selection is to use multi-level similarity matching, following the principle that a model identified as similar across more perspectives is more likely to resemble the target object. Based on this principle, the final candidate models are selected in the following order:

(i) Models similar in all three perspectives;

(ii) Models similar in two perspectives (if no models meet the criteria of i);

(iii) Models similar in one perspective (if no models meet the criteria of i and ii);

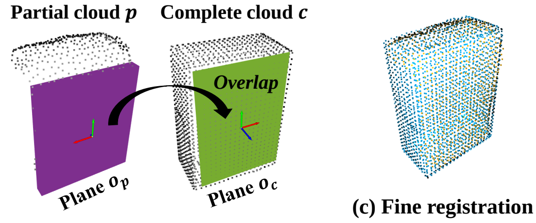
(iv) Non-similar models (if no models meet the criteria of i, ii, and iii).

This approach ensures that all perspectives are considered equally, thereby avoiding bias or balancing issues.

4.2.8 PDM-Based Point Cloud Registration

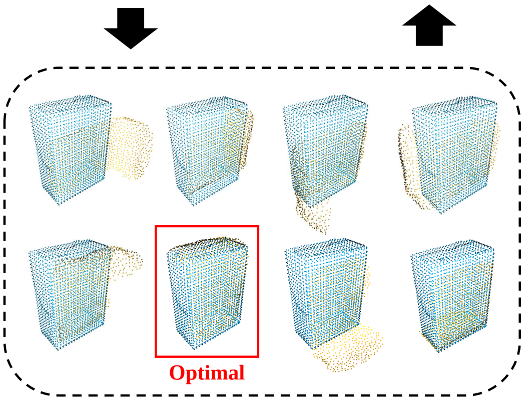
After selecting candidate models through similarity matching, two issues remain to be solved: 1) Which candidate model should be prioritized as the reference for object grasping? 2) How to transfer the grasping knowledge from a similar database model to the unknown target object? To address them, we perform point cloud registration between the partial point cloud p of the target object and the complete point clouds c of all candidate models to obtain their fitness scores and transformation matrices, which can be used to determine the model priority and transfer grasping knowledge, respectively.

Traditional point cloud registration methods employ algorithms such as RANSAC [83] and ICP [84] to achieve coarse-to-fine registration between point clouds of **identical objects**. However, our task copes with point clouds of **similar objects**, where the inherent randomness of the RANSAC algorithm can lead to suboptimal initial alignments and unreliable registration results (see Fig. 4.6d). To solve this issue, we adopt a plane detection method [85] to improve the coarse registration process. As shown in Fig. 4.6a-c, we first detect the largest plane o_p in p and the plane o_c in c with the area closest to o_p. Then, we align p with c by overlapping the two coordinate systems located at the centers of o_p and o_c. However, this overlapping does not necessarily result in an optimal alignment between p and c due to the uncertainty of the axis orientation.

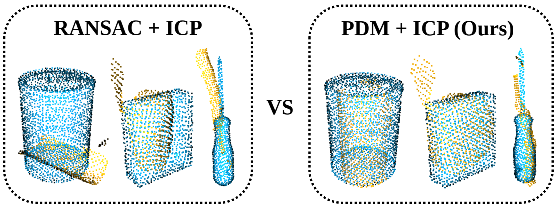


(a) Plane detection and matching

(c) Fine registration using ICP



(b) Coarse registration through rotation attempts



(d) Comparison of different registration methods

Figure 4.6: (a)-(c): Three steps to perform PDM-based point cloud registration between partial (yellow) and complete (blue) point clouds. (d): Comparison between a traditional registration method and our proposed method.

To address this, in the next step, we apply various rotations around the plane center to identify the optimal rotation that minimizes the distance between p and c, and use it as the coarse registration result. Based on this result, we finally employ the ICP algorithm for fine registration. The advantage of using plane detection and matching (PDM) is that it ensures a large overlap area between the two registered point clouds, regardless of their similarity and completeness. Additionally, the results are consistent across different trials, eliminating randomness. A performance comparison of RANSAC + ICP versus PDM + ICP is shown in Fig. 4.6d. In most cases, PDM + ICP achieves better results. However, in scenarios where no plane is detected in p due to the complex object shapes or significant sensing noise, PDM becomes inapplicable and we switch the coarse registration method to RANSAC.

4.2.9 Imitative Grasp Planning

Each point cloud registration result includes a fitness score representing the overlap rate between the two registered point clouds and a transformation matrix defining their relative pose. Based on the fitness score, we rank the candidate models from highest to lowest priority and use them sequentially during grasp planning. Based on the transformation matrix, we transfer the grasping knowledge from candidate models to the target object through the following steps:

1) We preplan hundreds of robust grasps for each database model using a mesh segmentation approach [68]; 2) For the candidate model with the highest priority, we apply the transformation matrix to its preplanned grasps to generate imitative grasps on the target object; 3) These grasps are then evaluated in a simulation environment[1], where we compute IK solutions and perform collision detection to exclude infeasible grasps; 4) For the remaining grasps, we let the robot reach each grasp pose in the simulation and perform further filtering as follows: Assuming that a cube model filling the gripper closure region is $\mathbb{C}$, the gripper model is $\mathbb{G}$, and the object point cloud is $\mathbb{P}$, a grasp is not on the object when $\mathbb{C} \cap \mathbb{P} = \emptyset$, and a grasp collides with the object when $\mathbb{G} \cap \mathbb{P} \neq \emptyset$. Such infeasible grasps occur due to the inherent differences between the similar model and the real object, and are excluded during planning; 5) If no feasible grasp is obtained from the first candidate model, we proceed to the next model in the priority order and repeat Steps 2-4 until a valid grasp is identified.

It should be noted that although we have original models of the environmental objects in our simulations (e.g., the platform on which the object is placed), our grasp planning does not rely on them. Instead, we can utilize the visual information in the background image to reconstruct the surrounding obstacles. This capability is validated in the final part of our experiments.

---

[1] https://github.com/wanweiwei07/wrs

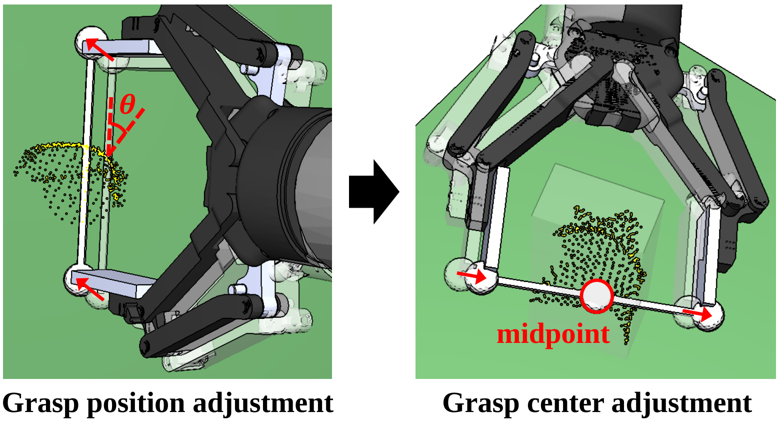


Figure 4.7: Two-stage grasp fine-tuning, including grasp position and grasp center adjustments to optimize the final grasp quality.

4.2.10 Stability-Aware Grasp Fine-tuning

Considering the inherent differences between similar models and real objects, imitative grasps have the potential to be unstable in real-world tasks. To evaluate and enhance their stability, we focus on the local features of contact points within the observable region of the object point cloud to implement a two-stage grasp fine-tuning process, as shown in Fig. 4.7. In the first stage, we create a stick model connecting the two ends of the gripper, denoted as $\mathbb{S}$, and use a ray-hit algorithm to detect its intersections with the object point cloud $\mathbb{P}$. If no intersection is detected, the grasp is considered to be located in the unseen regions of the object and is retained as a potential grasp, as its stability cannot be assessed. If only a single intersection point is detected, it is identified as one of the two contact points between the gripper and the object. In cases where multiple intersection points are detected, the two outermost points are regarded as the two contact points.

For each contact point, we identify its nearest neighboring point p_0 in $\mathbb{P}$ and compute the acute angle θ between the normal at p_0 and $\mathbb{S}$. Based on this angle, we classify all feasible grasps other than potential grasps into three types:

1. When $\theta < 20^\circ$, the grasp is considered stable enough to be executed directly without fine-tuning;

2. When $\theta > 40^\circ$, the grasp is too unstable for fine-tuning and is therefore discarded;
3. When $\theta \in [20^\circ, 40^\circ]$, the grasp is not sufficiently stable but can be improved through fine-tuning for better quality.

Below, we illustrate the fine-tuning process for the case where only one contact point is detected. The two-contact-point case differs only in that θ needs to be calculated for both contact points and the criteria for grasp classification become: 1) $\forall\, \theta < 20^\circ$; 2) $\exists\, \theta > 40^\circ$; 3) All other cases. Additionally, when two contact points are detected, one of them is randomly selected as the reference point for fine-tuning. After fine-tuning, grasp stability is re-evaluated using the updated contact points.

In the case of $\theta \in [20^\circ, 40^\circ]$, we search for the k_1-nearest neighbors of p_0 as $\{p_1, p_2, ..., p_{k_1}\}$ ordered from nearest to farthest, and then calculate θ for each neighbor starting from p_1 until a point satisfying $\theta < 20^\circ$ is found. This point is marked as p_0^*. To prevent grasping an uneven region, we further inspect the surrounding of p_0^* by searching for its k_2-nearest neighbors as $\left\{p_1^*, p_2^*, ..., p_{k_2}^*\right\}$. We calculate the angle between the normal of p_0^* and the normal of each of its neighbor as θ^*. When $\forall\, \theta^* < 10^\circ$, the region around p_0^* is considered flat and suitable for grasping. In this case, we translate the original grasp pose along the vector $\overrightarrow{p_0 p_0^*}$ to adjust the grasp position without applying any rotation. Otherwise, we continue querying the next neighboring point of p_0 until an eligible p_0^* is found. In our task, k_1 should be large to capture sufficient neighborhood information, while k_2 should be small to focus on local geometric features. Empirically, we set $k_1 = 100$ and $k_2 = 5$ as appropriate values.

In addition to adjusting the grasp position, we recognize that the grasp stability is also influenced by the location of the grasp center. When the distances between the object and the two finger ends differ significantly, one finger end may contact the object first, potentially causing unpredictable shifting or rotating motion during the grasping process. Therefore, in the second stage, we reapply the ray-hit algorithm to detect the intersections between the updated $\mathbb{S}$ after the first-stage adjustment and the SOBB of $\mathbb{P}$. We then obtain the midpoint of the two intersection points, denoted as p_c^*, and translate the grasp pose along the vector $\overrightarrow{p_c p_c^*}$ to refine the grasp center, where p_c

denotes the original grasp center after the first-stage adjustment. In the case where two contact points are detected, their midpoint can be directly used as p_c^*. This grasp center refinement is also applied in the case of $\theta < 20°$.

At either fine-tuning stage, if an adjusted grasp fails to solve IK or results in a collision, we discard it and query the next grasp candidate. The retained potential grasps are only used when all evaluable grasps have been assessed as unstable.

4.3 Experiments

4.3.1 Experimental Setup

To validate our proposed method, we conduct several experiments in similarity matching and novel object grasping using a UR5e robot arm equipped with a Robotiq 2F-140 adaptive gripper and a hand-mounted RealSense D435 depth camera. To verify the generalizability of our approach under low data volume conditions, we generate a database containing no more than 100 object models derived from the YCB dataset [80]. For each model, we preplan about 200 antipodal grasps within the gripper's width range and pre-compute the C-FPFH descriptors along with the bounding extents required for matching.

The experimental objects are placed on a fixed platform within the robot's reachable workspace, and are observed by the hand camera from a diagonal downward viewpoint. All computations are performed on a computer equipped with a Ryzen 7 5800H CPU and a GeForce RTX 3060 GPU.

To clearly showcase performance, we compare our method with several baselines, including SOTA benchmarks and the previous similarity approach. For all learning-based methods, we use their pre-trained models without additional fine-tuning.

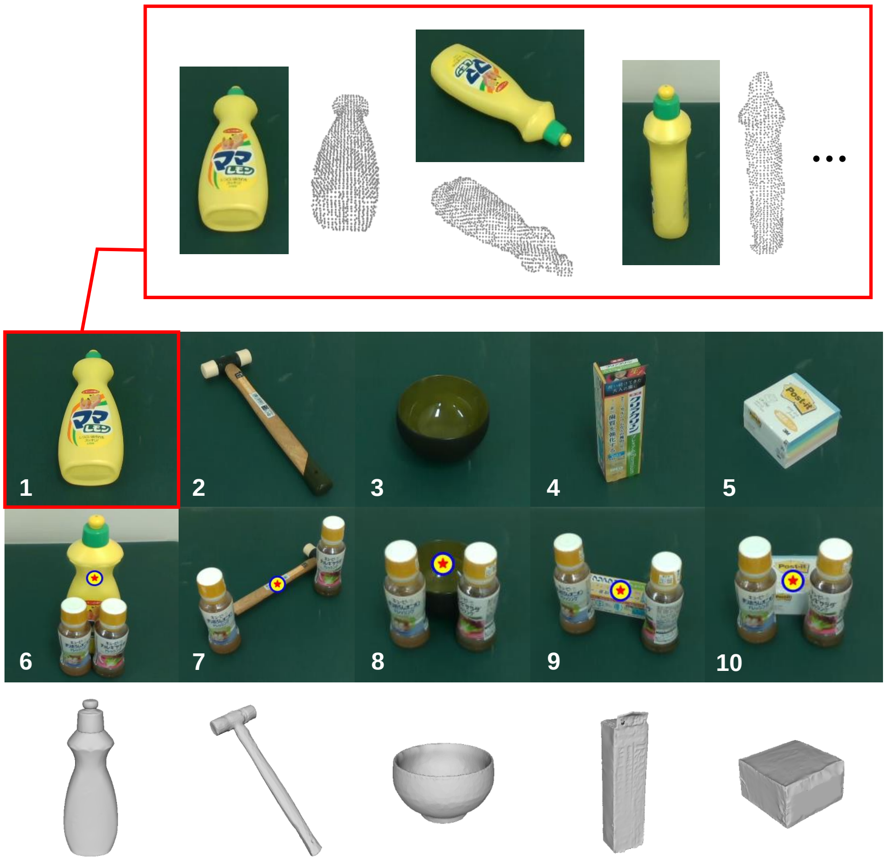


Figure 4.8: The experimental objects used for similarity matching. The first and second rows display the non-occluded and occluded scenes, respectively. The numbers in the figure correspond to the scene IDs in Table 4.1 and Table 4.2. The third row presents the original 3D model of each object.

4.3.2 Evaluation of Similarity Matching

To verify the effectiveness of our proposed similarity matching approach, we carry out comparative experiments using the following baselines: 1) **w/o MM** (Without Multi-level Matching); 2) **w/o SM** (Without Semantic Matching); 3) **w/o GM** (Without Geometric Matching); 4) **w/o DM** (Without Dimensional Matching); 5) **Full Matching**. For the first baseline, we directly apply point cloud registration to rank all database models from most to least similar without performing multi-level matching. In the second through fourth baselines, we perform incomplete matching by excluding one of the three levels, followed by the point cloud registration process to rank candidates. The last baseline uses the complete method.

To accurately assess the performance of similarity evaluation across different baselines, we utilize 5 different types of experimental objects with their original 3D models, as shown in Fig. 4.8, and incorporate them into the database. During the matching process, if the original model of the target object is identified within the highest priority list of candidate models, we consider it an accurate match. Based on this principle, we develop the following two evaluation metrics for the matching results: 1) **Matching Accuracy (MA)**, which is defined as:

$$\mathrm{MA}_x = \frac{\text{Number of accurate matches}}{\text{Total number of matching attempts}}$$

Here, x indicates that a match is considered accurate if the original model appears in the top x candidates. We need this x to be adaptable to different detection conditions and to allow for matching errors when the database contains existing models similar to the newly added objects. 2) **Matching Time (MT)**, which records the total time taken for the matching process, measured in seconds (s), from the start of similarity matching to the completion of point cloud registration.

We first test in non-occluded scenes, where the objects are placed on a stationary platform without surrounding obstacles. For each object, we obtain its visual features from 5 different observation angles and use these features to perform each baseline, generating the corresponding matching results. We set $x = 2$ as a strict criterion in this case to clearly distinguish the performance of different baselines. From the experimental results shown in Table 4.1, the comparison between **w/o MM** and **Full Matching** highlights the advantage of using multi-level similarity matching for achieving significantly higher matching accuracy while reducing computation time. The reason for this discrepancy is that without multi-level matching, we need to perform point cloud registration with all database models, the results of which are susceptible to the uncertainty of partial observation, resulting in low accuracy and efficiency. Besides, the absence of either semantic or dimensional levels in matching leads to a noticeable performance drop compared to the full matching method, emphasizing their importance in effectively narrowing down the range of similar candidates. On the other hand, however, the baseline **w/o GM** obtains very similar results to **Full Matching**, failing to demonstrate

Table 4.1: Evaluation Results of Similarity Matching Baselines in Non-Occluded Scenes

Scene ID	1		2		3		4		5		Average	
	MA_2	MT	MA_2	MT	MA_2	MT	MA_2	MT	MA_2	MT	MA_2	MT
w/o MM	3/5	3.62s	3/5	3.40s	4/5	3.28s	2/5	3.14s	1/5	2.86s	13/25 (52%)	3.26s
w/o SM	3/5	0.45s	3/5	0.44s	3/5	0.65s	4/5	0.69s	2/5	0.76s	15/25 (60%)	0.60s
w/o GM	5/5	0.20s	4/5	0.35s	4/5	0.42s	5/5	0.32s	3/5	0.28s	21/25 (84%)	**0.31s**
w/o DM	4/5	0.75s	3/5	0.33s	4/5	0.77s	3/5	0.33s	2/5	0.27s	17/25 (64%)	0.49s
Full Matching	5/5	0.30s	4/5	0.35s	4/5	0.58s	5/5	0.30s	4/5	0.21s	**22/25 (88%)**	0.35s

the effectiveness of our geometric matching method. This occurs because the database contains the categories of all experimental objects, making semantic and dimensional matching alone sufficient for achieving accurate results.

Therefore, for further validation, we test in occluded scenes where the objects are partially occluded by two small bottles acting as obstacles. During visual detection, we manually place a marker in the image (see Fig. 4.8) to extract only the recognition result of the target object. Similar to the non-occluded scenes, we perform 5 matching attempts for each object and baseline using various observation angles and occlusion conditions (in principle, occluding no more than half of the object to ensure matchability). The difference is that in this case we set $x = 5$ to allow for larger matching errors due to the increased visual uncertainty. From the experimental results shown in Table 4.2, we observe that the accuracy of **w/o MM** becomes worse due to the sparser visual information, whereas the performance of **Full Matching** remains relatively stable. More importantly, the advantage of using C-FPFH-based geometric matching is clearly demonstrated by a significant performance drop in the baseline **w/o GM**. The reason for this is that in occluded scenes, the reliability of both semantic and dimensional matching declines due to incomplete object recognition, making the matching accuracy heavily dependent on the results of geometric matching. We even find that in some cases, dimensional matching negatively impacts the matching results, whereas semantic matching still helps to achieve accurate results when the occluded object is correctly recognized. Additionally, the full matching method achieves the lowest computation time by narrowing down similar candidates to a smaller range, while other baselines yield

Table 4.2: Evaluation Results of Similarity Matching Baselines in Occluded Scenes

Scene ID	6		7		8		9		10		Average	
	MA_5	MT	MA_5	MT	MA_5	MT	MA_5	MT	MA_5	MT	MA_5	MT
w/o MM	2/5	3.29s	1/5	2.92s	1/5	2.58s	2/5	2.83s	1/5	2.06s	7/25 (28%)	2.74s
w/o SM	3/5	1.21s	2/5	0.97s	3/5	0.72s	3/5	1.14s	4/5	0.72s	15/25 (60%)	0.95s
w/o GM	3/5	1.67s	2/5	0.26s	2/5	0.45s	3/5	0.83s	2/5	0.63s	12/25 (48%)	0.77s
w/o DM	4/5	0.48s	4/5	0.39s	4/5	0.49s	3/5	0.52s	4/5	0.43s	**19/25 (76%)**	0.46s
Full Matching	4/5	0.41s	3/5	0.21s	3/5	0.38s	4/5	0.34s	4/5	0.36s	18/25 (72%)	**0.34s**

more candidates when a certain matching level is absent, leading to longer computation times.

By combining the results from both scenes, our complete similarity matching method demonstrates optimal performance in terms of accuracy and efficiency under varying detection conditions and across different object types.

4.3.3 Grasping Isolated Objects

To verify the performance of novel object grasping using our method, we conduct grasping experiments in two scenarios: isolated and cluttered scenes. We first focus on single-object grasping by comparing our proposed method with two learning-based benchmarks, **PointNetGPD** [15] and **3DSGrasp** [20], as well as the **previous similarity approach** [33]. All these approaches address the grasping of isolated novel objects by leveraging object point clouds. The key differences are that, PointNetGPD directly predicts grasps from single-view point clouds; 3DSGrasp performs shape completion to reconstruct unseen regions of the point cloud and plans grasps based on the refined cloud; the previous similarity approach utilizes multi-view point clouds for similarity matching, followed by grasp knowledge transfer from similar references to the unknown target. For PointNetGPD and 3DSGrasp, we rank the grasp candidates based on their quality scores and proceed sequentially from the highest-scoring grasp until a feasible one is found. For both the previous and our similarity approaches, we process the grasp candidates in no specific order and continue the computation until a feasible grasp is

Table 4.3: Experimental Results of Grasping Isolated Objects

Object ID		1	2	3	4	5	6	7	8	9	10	Average	APT
PointNet-GPD	GSR	7/8	6/8	4/6	2/4	-	3/6	8/9	4/7	3/4	2/6	67%	$\approx$ 6s
	PSR	8/10	8/10	6/10	4/10	0/10	6/10	9/10	7/10	4/10	6/10	58%	
3DSGrasp	GSR	7/10	3/10	5/7	-	2/2	2/4	8/8	0/6	3/4	6/10	59%	$\approx$ 8s
	PSR	10/10	10/10	7/10	0/10	2/10	4/10	8/10	6/10	4/10	10/10	61%	
Previous similarity approach	GSR	8/10	7/10	4/7	10/10	7/8	10/10	9/10	5/8	5/6	4/10	78%	$\approx$ 5s
	PSR	10/10	10/10	7/10	10/10	8/10	10/10	10/10	8/10	6/10	10/10	89%	
Our method	GSR	10/10	10/10	8/9	9/10	9/9	10/10	10/10	8/10	10/10	10/10	**96%**	**$\approx$ 2s**
	PSR	10/10	10/10	9/10	10/10	9/10	10/10	10/10	10/10	10/10	10/10	**98%**	

identified.

We use three evaluation metrics to assess the performance of each method: 1) **Grasp Success Rate (GSR)**; 2) **Plan Success Rate (PSR)**; 3) **Average Planning Time (APT)**. Considering three possible outcomes in a grasping task: I) Grasp planning fails (no grasp output); II) Grasp planning succeeds, but the output grasp fails to catch or lift the object; III) Grasp planning succeeds, and the output grasp successfully catches and lifts the object, we define GSR and PSR as follows:

$$\mathrm{GSR} = \frac{\mathrm{Num(III)}}{\mathrm{Num(II+III)}},\ \mathrm{PSR} = \frac{\mathrm{Num(II+III)}}{\mathrm{Num(I+II+III)}}$$

where Num() denotes the total number of outcomes contained in the bracket. APT is the approximate average of the duration from the start of similarity matching to the completion of grasp planning, recorded only when the grasp planning is successful. These three metrics represent the accuracy, generalizability, and efficiency of each implemented method, respectively.

For the experimental setup, we select 10 previously unseen objects varying in category, shape and size (see Fig. 4.9), and place them in arbitrary poses on a stationary platform within the camera's field of view and the robot's reachable workspace. During each trial, we apply our object recognition method to obtain the single-view point cloud of the target object and execute the corresponding grasp planning baseline based on

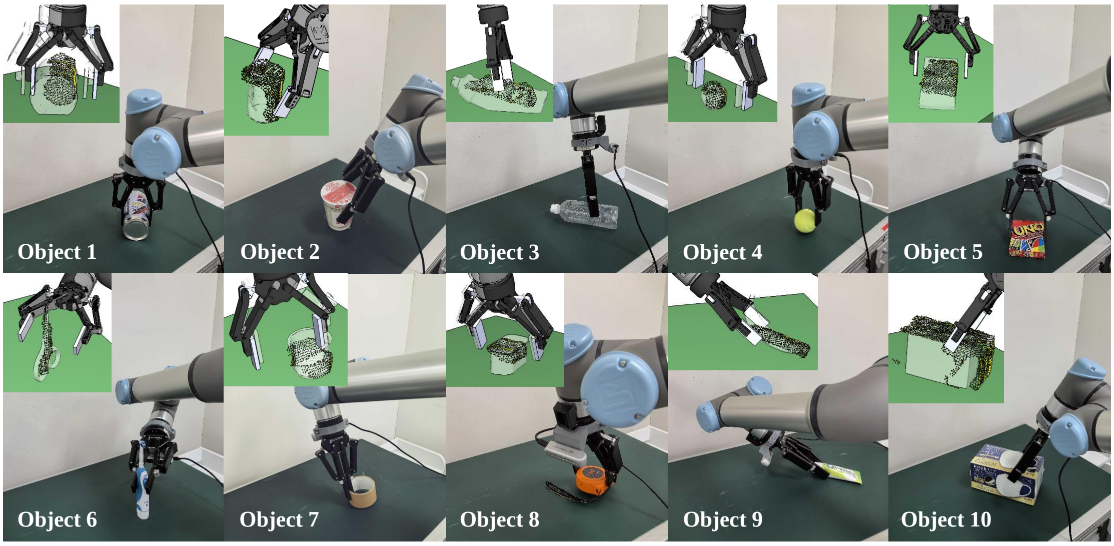


Figure 4.9: Grasping various novel objects using our similarity-based method. The top left corner of each figure displays the single-view object point cloud, the matched candidate model, and the grasp planning with fine-tuning process. The object IDs shown in the bottom-left correspond to those in Table 4.3.

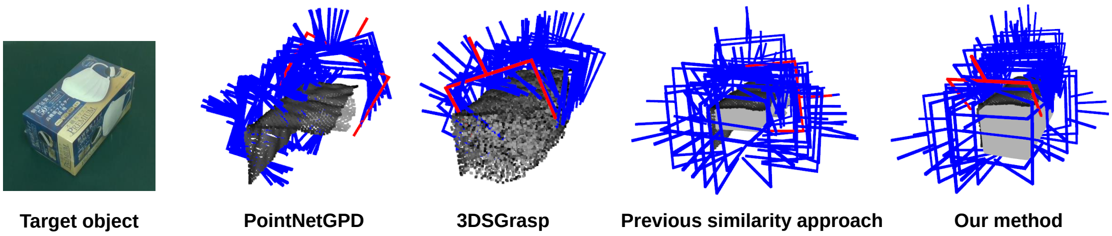

Target object
PointNetGPD
3DSGrasp
Previous similarity approach
Our method

Figure 4.10: Comparison of different grasp planning methods based on object point clouds. The blue grasps represent all the generated grasp candidates, while the red grasps indicate the final executed grasps that are first recognized as IK-solvable and collision-free from the candidate list.

this point cloud. For each object and method, we perform 10 planning and grasping attempts, considering a grasp successful if the object is steadily caught and lifted to a specified height (approximately 20 cm). As can be seen from the results shown in Table 4.3, our proposed method significantly outperforms all other methods in terms of both success rate and computational efficiency, demonstrating its superior overall performance.

For the learning-based methods, we observe that both PointNetGPD and 3DSGrasp achieve very low PSRs for specific objects, such as Objects 4 and 5. Although their network outputs provide reasonable grasp poses, most of these poses are inexecutable (IK unsolved or collisions detected) due to improper grasp positions and orientations, which is likely to be an inherent limitation of their training models. The previous similarity approach can successfully generate executable grasps in most cases; however, its GSR remains unsatisfactory, primarily due to the gap between multi-view and single-view detection conditions, the limitations of score-based matching, and the absence of subsequent fine-tuning.

An example visualization of the grasp planning results using different methods is shown in Fig. 4.10. The grasp candidates generated by PointNetGPD are mostly unstable, such as grasping the object edge from an improper angle. 3DSGrasp can refine the partial point cloud to approximate the actual object shape; however, its errors in point refinement can still lead to suboptimal grasps that lack sufficient stability. This limitation is specifically noticeable for objects with uneven and smooth surfaces, such as Objects 2 and 8. The previous similarity approach can generate more reasonable and diverse grasps by utilizing a similar reference model; however, an inappropriate matching result derived from scoring functions may still lead to infeasible grasps, such as colliding with the unseen regions of the target object. Additionally, its grasp stability cannot be guaranteed due to the lack of a fine-tuning process. In contrast, our method achieves optimal grasp planning by leveraging an accurately matched reference model and improving the grasp quality through a two-stage fine-tuning approach.

Regarding APT, the learning-based methods require additional time for grasp sampling or point cloud completion, leading to longer computation durations. The previous similarity approach requires matching with all models in the database, making it also time-intensive. In contrast, our method pre-sorts candidate models using a multi-level similarity matching approach, cutting computational time by more than half compared to other methods.

4.3.4 Grasping Cluttered Objects

For further validation of object grasping performance, we conduct experiments in cluttered scenes, where multiple objects are randomly arranged to create more challenging grasping scenarios. We compare our approach with two SOTA methods: a widely known large-scale benchmark for general object grasping, **GraspNet** [22], and a recent advanced work on grasping objects in clutter, **HGGD** [24]. Both methods detect grasps directly from the scene RGB-D input, making it difficult to distinguish which grasp corresponds to which object. Therefore, for these two methods, we develop the following steps to achieve the task of clearing cluttered objects: 1) Capture the RGB-D image of the scene and apply the grasp detection method; 2) Rank all detected grasps based on their evaluated quality scores; 3) Compute IK iteratively until the first executable grasp in the ranked list is found; 4) Execute the identified grasp to remove the corresponding object from the scene, return the robot to its initial pose, and repeat the above process. In contrast, our method completes detection and computation at once by simultaneously performing similarity matching and grasp planning for all objects in the clutter. We achieve object recognition similarly to the single-object scene, but with the difference that all recognition results (excluding redundant ones) within the clutter region are extracted and processed in separate threads. During grasp planning, the following principles are applied: I) When planning grasps for one object, all other objects are treated as obstacles; II) Once a feasible grasp is generated for an object, it is removed from the obstacle list to facilitate planning for the remaining objects. We also record the grasp generation sequence and execute the planned grasps in the corresponding order.

To assess task performance, we utilize two evaluation metrics: 1) **Grasp Success Rate (GSR)**; 2) **Declutter Rate (DR)**. GSR is computed similarly to the single-object scene, but differs in that a single re-planning attempt is allowed after a grasp failure, in which case the number of executed grasps may exceed the total number of objects in the clutter. DR evaluates the task completion rate and is calculated as:

$$\mathrm{DR} = \frac{\text{Number of successfully removed objects}}{\text{Total number of objects to be removed}}$$

Table 4.4: Experimental Results of Grasping Cluttered Objects

Clutter ID		3 objects			Average	5 objects			Average
		1	2	3		4	5	6	
GraspNet	GSR	7/13	8/13	8/12	23/38 (60.5%)	12/16	8/11	-	20/27 (74.1%)
	DR	7/15	8/15	8/15	23/45 (51.1%)	12/15	8/15	0/15	20/45 (44.4%)
HGGD	GSR	9/17	6/10	10/12	25/39 (64.1%)	13/17	9/13	2/2	24/32 (75.0%)
	DR	9/15	6/15	10/15	25/45 (55.6%)	13/15	9/15	2/15	24/45 (53.3%)
Our method	GSR	15/16	14/16	14/14	**43/46 (93.5%)**	15/17	14/16	13/14	**42/47 (89.4%)**
	DR	15/15	14/15	14/15	**43/45 (95.6%)**	15/15	14/15	13/15	**42/45 (93.3%)**

For the experimental setup, we leverage two types of clutter configurations as shown in Fig. 4.11: Clutters 1-3 consist of 3 objects selected from the single-object grasping experiment, while Clutters 4-6 comprise 5 objects with a broader variety of types. We perform 5 attempts for each clutter set under different object arrangements, each featuring varying degrees of occlusion. To avoid intermediate collisions unrelated to the grasp pose, we design a pregrasp pose by retracting 8 cm from the grasp pose and a leaving pose by moving 20 cm vertically upwards from the grasp pose. The robot follows this sequence of movements: initial pose → pregrasp pose → grasp pose (gripper closed) → leaving pose → placement pose (gripper open) → initial pose. A task stops midway when there is no executable grasp output or when two consecutive grasp failures occur. From the experimental results shown in Table 4.4, we observe that GraspNet and HGGD exhibit similar performance in both GSR and DR due to their respective advantages of large-scale training and well-designed learning frameworks. However, they struggle to handle specific objects, such as the small and thin items in Clutter 6, where we observe that very few viable grasps can be generated. Such failures can be attributed to the inherent limitations of learning-based approaches when the sensing condition and operating environment differ significantly from their training setup.

Additionally, as shown in Fig. 4.12, GraspNet fails to generate high-quality grasps for occluded objects positioned at the back. Meanwhile, HGGD can generate a few good grasps for the objects behind, but they are too sparse to ensure an executable grasp, and the front objects are ignored in this case. In contrast, our method efficiently generates robust grasps for both non-occluded and occluded objects. Even when the

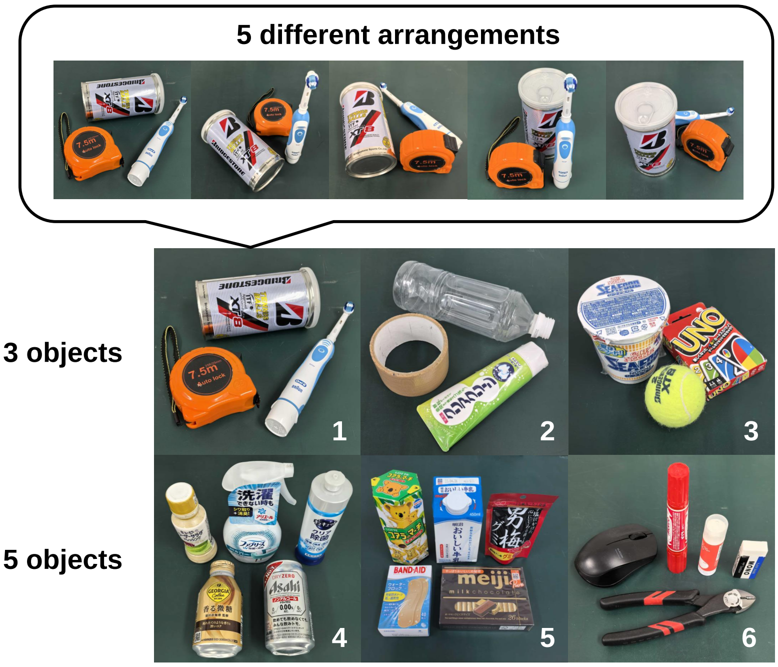


Figure 4.11: Objects in clutter used for grasping experiments. Each clutter set contains 3 or 5 objects, and is tested in 5 different arrangements.

matching results are suboptimal due to sparse and noisy visual inputs, the subsequent fine-tuning process in our method ensures high final grasp quality. Therefore, failures in grasp planning and execution are rarely observed with our approach, demonstrating its superior performance across all types of clutter sets and object arrangements.

4.3.5 Ablation Study

To validate the effectiveness of the main components in our approach, we conduct ablation studies using the same baselines from the similarity matching experiment, including **w/o MM**, **w/o SM**, **w/o GM**, and **w/o DM**. These baselines exclude one or all three levels in similarity matching, relying primarily on the processes of point cloud registration and grasp fine-tuning to generate final grasps. In addition, we introduce an extra baseline, **w/o GF** (Without Grasp Fine-tuning), which applies the complete matching method but excludes the two-stage fine-tuning process during grasp planning.

We reuse the objects from the single-object grasping experiment to conduct tri-

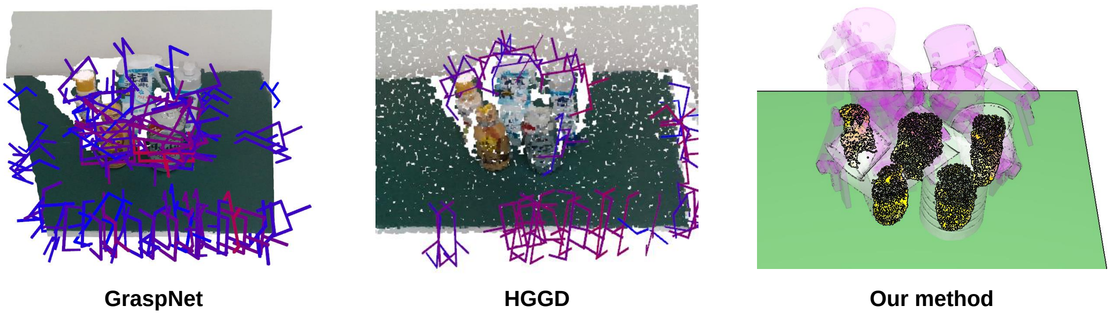


Figure 4.12: Comparison of our method with grasping benchmarks for objects in clutter. In the outputs of GraspNet and HGGD, red grasps indicate high-quality grasps, while blue grasps indicate low-quality ones. Our method generates high-quality grasps for all target objects by rendering similar reference models.

als. To better distinguish the performance of different baselines, in addition to **Grasp Success Rate (GSR)**, we introduce a new evaluation metric called **Average Offset (AO)**. This metric is measured using a graph paper with 5 mm grid units. At the initial stage, we mark a specific grid point on the paper and a corresponding point on the edge of the target object, aligning them perfectly (see Fig. 4.13). After planning a grasp, the robot executes the grasp, lifts the object, returns to the initial pose, and places the object back in its original position. During this process, the object may remain stationary or shift slightly within the gripper, depending on the stability of the grasp. Once the object is placed back, we locate the marked point on the object, identify the nearest grid point, and calculate its distance from the original marked grid point as the offset value. A smaller offset value indicates higher grasp quality. The use of graph paper offers a significant advantage, allowing efficient distance measurement based on the grid size without requiring complex procedures, as high precision is unnecessary. The offset is disregarded in cases of grasp failure. For each object and baseline, we perform 5 attempts and compute the average of the resulting offsets (AO) in both non-occluded and occluded scenes. In the occluded scenes, similar to the similarity matching experiment, two small bottles are used as obstacles during visual detection but are removed before grasp execution.

As shown in the experimental results in Table 4.5, our full method consistently delivers optimal performance across both scenes, highlighting the effectiveness of all

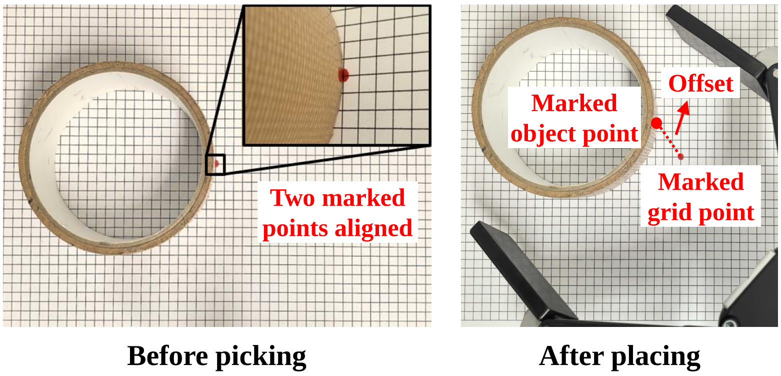


Figure 4.13: Evaluation of grasp quality using a graph paper. After a pick-and-place motion, the stability of the executed grasp is quantified by measuring the offset between a marker on the object and an initial point on the paper.

Table 4.5: Ablation Studies On Each Component of Our Method

	Non-occluded scene		Occluded scene	
	GSR ↑	AO ↓	GSR ↑	AO ↓
w/o MM	74%	11.4 mm	54%	25.3 mm
w/o SM	82%	10.5 mm	74%	17.0 mm
w/o GM	90%	**6.1 mm**	66%	21.4 mm
w/o DM	78%	9.4 mm	72%	15.5 mm
w/o GF	72%	15.2 mm	56%	24.1 mm
Full method	**92%**	7.1 mm	**84%**	**11.1 mm**

components in our approach. The baseline **w/o GM** achieves comparable results in the non-occluded scene but suffers a noticeable performance drop in the occluded scene, which aligns with the findings from the similarity matching experiment and confirms that our proposed C-FPFH descriptor is specifically effective in handling occlusions. Furthermore, the importance of multi-level matching and grasp fine-tuning is surprisingly close, as evidenced by the results of **w/o MM** and **w/o GF**. This underscores that both accurate matching and an effective fine-tuning process are critical for achieving high-quality grasps.

4.3.6 Robustness to Environmental Changes

In the final experiment, we validate the robustness of our proposed method across varying environments by performing grasping tasks in three different scenarios: **table-top**,

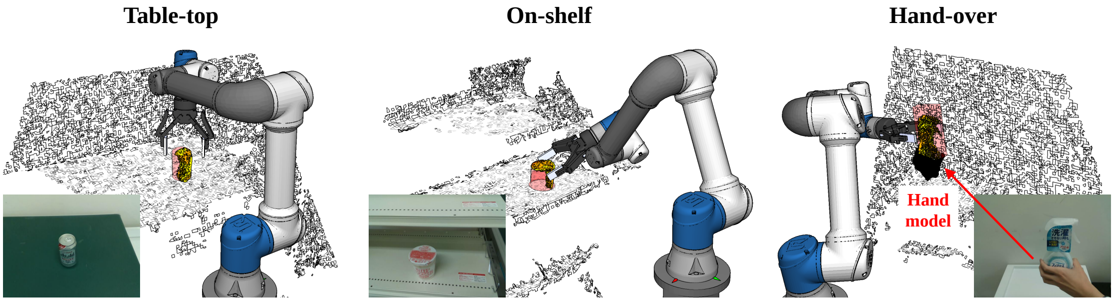


Figure 4.14: Application of our method to achieve grasping tasks in three different scenarios without the original models of the environmental objects.

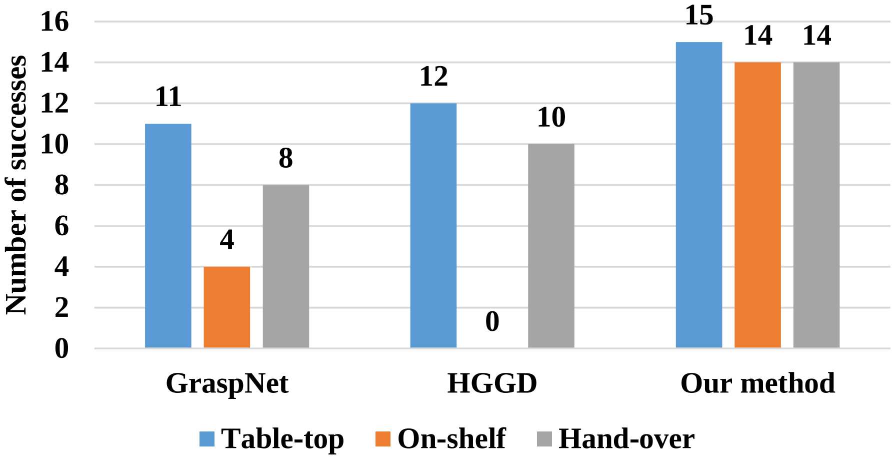


Figure 4.15: Comparison of the performance robustness of different methods.

on-shelf, and **hand-over**, as shown in Fig. 4.14. Unlike the previous experiments, we assume that the original models of environmental objects such as the platform are not available, requiring all environmental features to be acquired through visual detection. To handle this, we utilize the RGB-D information captured from the background image to reconstruct a scene model based on the point cloud features. This reconstructed model is then used for collision detection during grasp planning, which demonstrates that our method is independent of pre-existing environmental object models. In the hand-over task, we further recover a hand model based on the recognition of the human hand to prevent grasps that may collide with the hand.

We compare the performance of our method with two learning-based benchmarks, GraspNet and HGGD, using the 3 objects shown in Fig. 4.14, for which all the methods can generate feasible grasps to ensure comparability. For each object and method, we

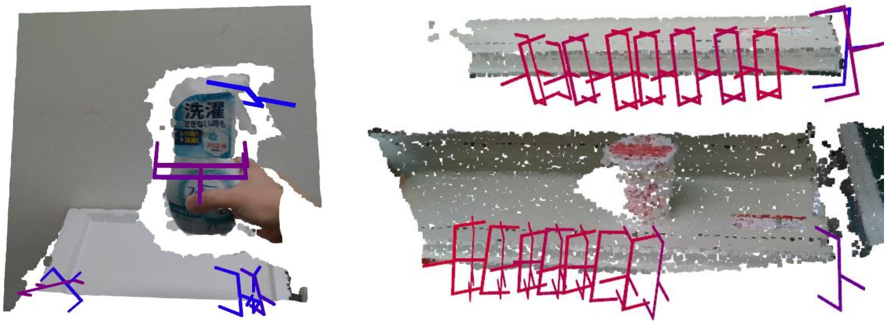


Figure 4.16: Failures of learning-based methods in on-shelf and hand-over tasks.

perform 5 grasping attempts and record the number of successful attempts. The results, summarized in Fig. 4.15, clearly demonstrate that our method consistently achieves robust and high performance across all scenarios, whereas both GraspNet and HGGD exhibit significant variability in different environments, highlighting their sensitivity to environmental changes. As illustrated by the failure cases in Fig. 4.16, the learning-based methods may produce very few grasp candidates for the target object when it is positioned close to the camera during a hand-over task, and tend to detect high-quality grasps on environmental objects rather than the target object during an on-shelf task. These observations underscore the effectiveness of our method in addressing the key limitations of learning-based approaches.

4.4 Conclusions

In this study, we present a novel framework for single-view object grasping by introducing a multi-level similarity matching approach that accurately identifies similar reference models from an existing database to guide the grasping of unknown target objects. The matching process simultaneously evaluates object similarity from the aspects of semantics, geometry, and dimensions to optimize the selection of potential candidate models. Notably, we introduce the C-FPFH descriptor, a novel geometric descriptor, which efficiently evaluates the similarity between partial point clouds from observed objects and complete point clouds from database models. This descriptor demonstrates exceptional effectiveness in handling occlusions. Additionally, we integrate LLM to assist with semantic matching, propose the SOBB for accurate dimensional matching, develop a PDM-based point cloud registration method to achieve imitative grasp planning, and incorporate a two-stage grasp fine-tuning process to optimize the final grasp quality.

Chapter 5

Application of Similarity-Based Methods in Dynamic Object Manipulation

This thesis chapter originally appeared in the literature as
H. Chen, T. Kiyokawa, W. Wan, and K. Harada, "Adaptive Grasping of Moving Objects in Dense Clutter via Global-to-Local Detection and Static-to-Dynamic Planning," *in Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, 2025.

5.1 Introduction

In logistic warehouses, a wide variety of daily items are transported on conveyor belts every day, and human workers are required to pick out target items from an unorganized clutter and pack them into delivery boxes. To achieve robotic automation for this task, an interesting issue is posed: grasping unknown objects in clutter moving on a conveyor.

Existing studies on robotic grasping for unknown objects predict grasps directly from partial point clouds [15, 86] or perform shape completion before planning [87, 21]. However, they have limited generalization to object types outside of their training dataset due to their data dependency. Moreover, these methods mainly focus on single or static target objects, whereas in real-world scenarios, objects are often in cluttered or dynamic states where multiple uncertainties coexist.

Therefore in this study, we aim at developing a grasping strategy that can cope with novel object types and grasping scenarios where the target objects are moving in dense clutter. The key to realize such a grasping task is the proposed global-to-local detection and static-to-dynamic planning. In the global-to-local detection, we first capture the object features used for grasp planning from a global image, then assess their movement states from a local viewpoint. During grasp planning, we first perform static planning to generate robust grasp poses, followed by dynamic planning to enable real-time grasping of the moving objects. The combination of global-to-local detection and static-to-dynamic planning provides an effective method for grasping moving cluttered objects that are outside of existing datasets, since it addresses multiple real-world uncertainties step-by-step rather than all at once as in traditional learning-based methods. The experimental results demonstrate the exceptional effectiveness of our method in grasping novel objects under high uncertainty.

Our main contributions can be summarized as:

- We propose a global-to-local detection and static-to-dynamic planning approach for grasping moving objects in dense clutter on a conveyor belt.
- We develop an adaptive object tracking method that accurately estimates the movement speed and the instantaneous position of the cluttered objects.
- We develop a recurrent algorithm to achieve continuous dynamic grasping for the moving objects.

Fig. 5.1 showcases the general flow of our system. We utilize the flexibility of an in-hand camera to switch between global and local observation poses, and leverage the results of static grasp planning to achieve real-time dynamic object grasping.

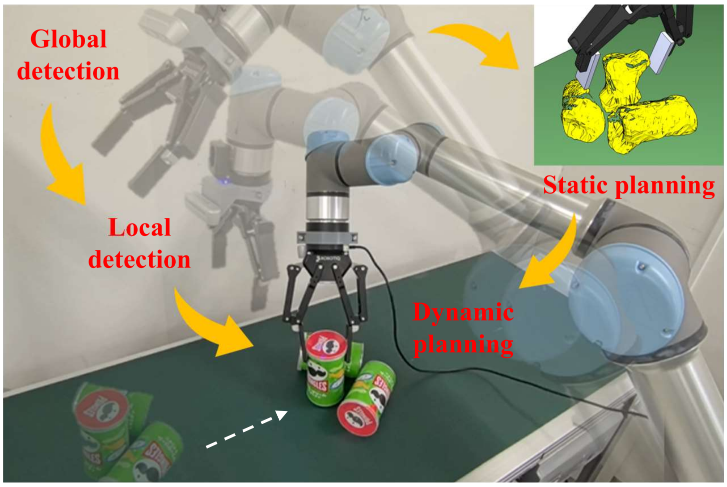


Figure 5.1: Grasping moving cluttered cans by our method. Static and dynamic planning are executed at the end of global and local detection respectively.

5.2 Methods

5.2.1 System Overview

Our goal is to grasp unknown objects from a dense clutter moving on a conveyor belt. Fig. 5.2 shows the complete system workflow. Similar to an industrial setting, the cluttered objects are initially positioned at the start of the conveyor belt, with the robot located some distance away. To achieve the grasping task, we first let the robot move to a global observation pose where the in-hand camera can observe the entire clutter from a diagonal downward view. In this pose, we use the camera to capture an instant RGB-D image and perform instance segmentation on the RGB input to obtain object categories, and combine the results with the depth input to obtain object point clouds. Based on these two types of information, we implement a similarity matching method similar to [33] to obtain a similar reference model for each object in the clutter from an existing database. During this process, we utilize LLMs to greatly improve the matching efficiency. Based on the reference models, we perform static grasp planning to simultaneously generate robust grasp poses for all objects in the clutter at a fixed position in front of the robot. The grasp sequence for different objects is also determined by the planning results.

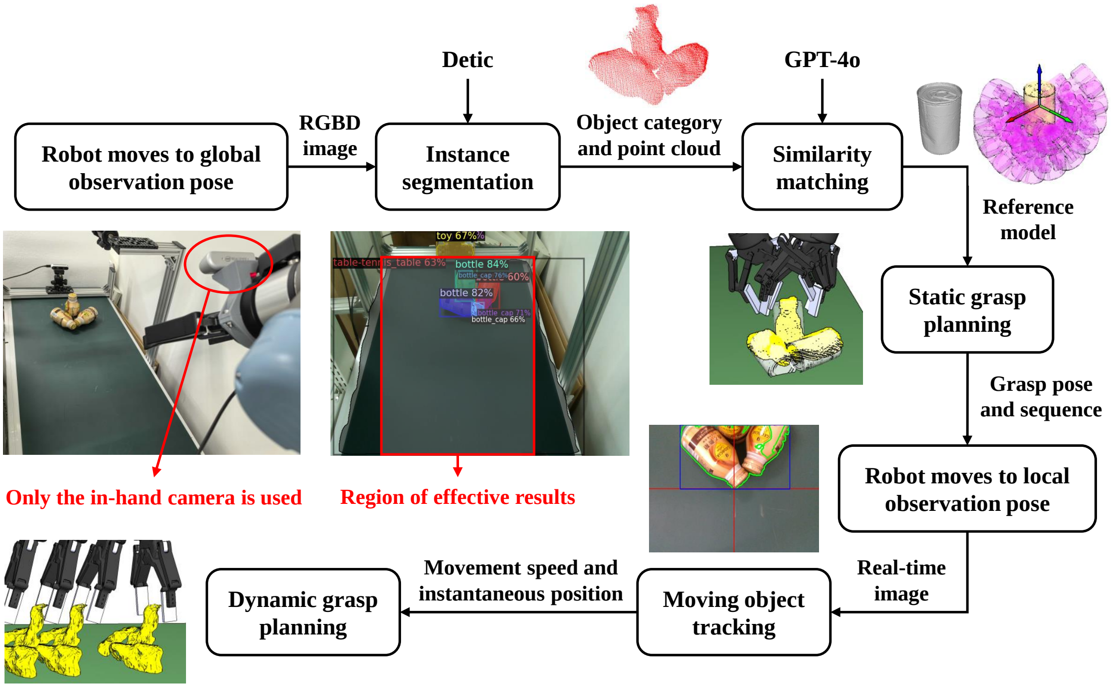


Figure 5.2: Workflow of our proposed system. All visual detections and grasp planning are performed online using only a single in-hand camera.

During static grasp planning, an adaptive local observation pose with a top-down view is also generated based on object locations and arrangements. When this pose is determined, we immediately let the robot move there and start capturing real-time images to assess the movement state of the clutter. We incorporate a moving object tracking method to obtain the movement speed and instantaneous position of the clutter based on two key time points. Based on these two results, we finally develop a recurrent algorithm to achieve dynamic grasping for moving cluttered objects.

5.2.2 Global Visual Detection

We use a single in-hand camera for visual detection for two reasons: 1) For cluttered objects, it is difficult to identify object correspondences in different viewpoints when using multiple cameras; 2) We need a diagonal downward view to get more surface information about the objects, and a top-down view to accurately track their movement, in which case a flexible in-hand camera becomes a better choice.

During global detection, the camera obtains an instant RGB-D image containing the cluttered objects and feeds it into a SOTA instance segmentation model of Detic [76]. To exclude the segmentation results of background objects, we define a region based on the position of the conveyor belt in the image and extract only the results within that region. For redundant results such as *bottle_cap* in *bottle*, we ignore them by calculating the containment relationship between detected bounding boxes. From the filtered segmentation results, we can obtain the category name and 3D point cloud of each object in the clutter. Based on them, we can perform similarity matching to find reference models from an existing database to guide the grasping of unknown objects, as was done in [33]. However, their method requires matching with all database models, which is computationally long and not suitable for the task of grasping moving objects.

5.2.3 LLM-Assisted Similarity Matching

Considering that most database models are irrelevant to the target object, we improve matching efficiency by using LLMs (GPT-4o [79] in our task) to pre-screen potential candidates based on object categories, passing only these candidates for further matching. The implementation is as follows:

Prompt: Which objects in the {*YCB dataset*} are likely to be similar to {*can*} in terms of robotic grasping? Please list the 5 most likely object names with their indices.

Answer: ... 005_tomato_soup_can, 006_mustard_bottle, 010_potted_meat_can, 021_bleach_cleanser, 025_mug ...

In the prompt, the first bracket can be filled in with an existing model database that is known to LLMs and can be used for similarity matching, and the second bracket can be filled in with the obtained category name of each object in the clutter. From the answer, we can extract only the key information about the selected candidates by recognizing their indices. For these candidates, we further perform point cloud registration between their point clouds and the obtained point cloud of each object in the clutter using the RANSAC [83] and ICP [84] algorithms. Due to the pre-screening of model candidates, registration can be finished in a short time.

5.2.4 Static Grasp Planning

Each model in the database is pre-planned with over one hundred antipodal grasps using a mesh surface segmentation approach [68]. These grasps are robust through an accurate analysis of the complete mesh model. For grasp planning of an unknown object, we transfer the pre-planned grasps from a similar database model to the unknown object based on the transformation matrix obtained from point cloud registration, similar to what was done in [33]. The grasp planning process is performed in a robot simulation environment[1].

In our task, the objects are moving and we need to quickly generate grasps for real-time dynamic grasping. For this purpose, we first assume that the cluttered objects are right in front of the robot and perform static grasp planning at this fixed position p_0. Then during dynamic grasping, we directly query the planned static grasp poses to quickly generate real-time grasps. To ensure the efficiency and accuracy of grasp planning, we define two types of areas called *collision area* and *overlap area* based on the axis-aligned bounding box (AABB) of each object in the clutter, as shown in Fig. 5.3. In each grasp planning, there is one target object and several surrounding objects. For the target object, we extend its AABB in the opposite direction of the moving direction to generate a blue area, and for the surrounding objects, we extend their AABBs in both the positive and negative directions of the moving direction to generate a red area. The concatenation and intersection of the blue area and the red area indicate *collision area* and *overlap area*, respectively.

To avoid collisions between the gripper and all moving objects before and after grasping, the pregrasp pose defined by backing off the grasp pose for a small distance cannot be located within *collision area.* In addition, at the grasp pose, the target object cannot collide with the gripper fingers and should be located within the gripper's closure area. Based on these rules, we exclude all infeasible grasps included in the grasps

---

[1] https://github.com/wanweiwei07/wrs

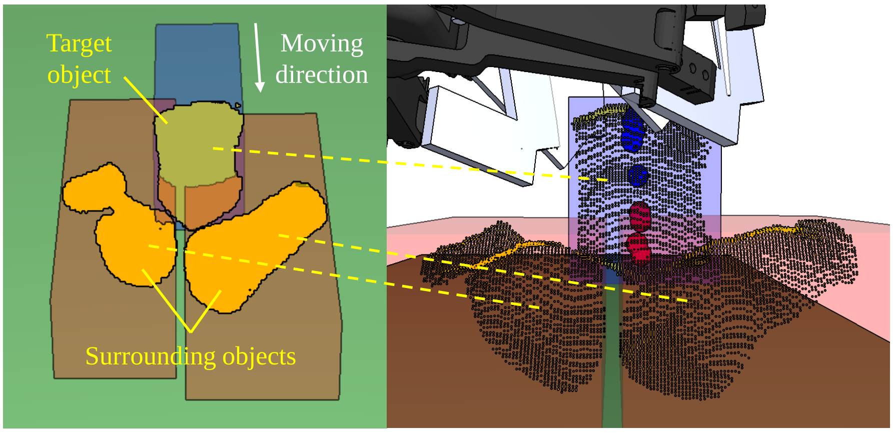


Figure 5.3: Fast and accurate grasp planning for cluttered objects based on *collision area* (blue ∪ red) and *overlap area* (blue ∩ red).

transferred from similar models. To further accelerate the planning process, we prioritize the grasps whose centers are outside *overlap area*. As shown in Fig. 5.3 right, the grasp poses with blue centers are more likely to be collision-free than those with red centers in cluttered scenes.

Another important issue is the determination of the grasp sequence. While the grasp planning for all objects in the clutter is performed simultaneously, they need to be grasped in a certain order. As shown in Fig. 5.4, we consider several factors for this including: 1) the confidence score during instance segmentation S_c, as it reflects the object occlusion rate and the reliability of the similarity matching results; 2) the fitness score during point cloud registration S_f, as it determines the quality of the grasps transferred from similar models; and 3) the speed of the grasp generation, as faster grasp planning represents lower grasping difficulty. Both S_c and S_f can be directly obtained from the corresponding algorithms, while the speed of the grasp generation needs to be calculated over a time period T ($T = 5s$ in our task). We take the total number of IK-solvable and collision-free potential grasps generated within this time period $|G_p|$ minus the number of unstable grasps $|G_u|$ as the evaluation metric. The grasp stability is evaluated based on the relative angle θ between the gripper opening direction and the estimated normal direction of the contact point. θ is taken as an acute angle, as shown in Fig. 5.4 right. When $\theta > 30°$ at any contact point, we consider the grasp unstable and categorize it into G_u. When contact points are within the invisible area causing θ

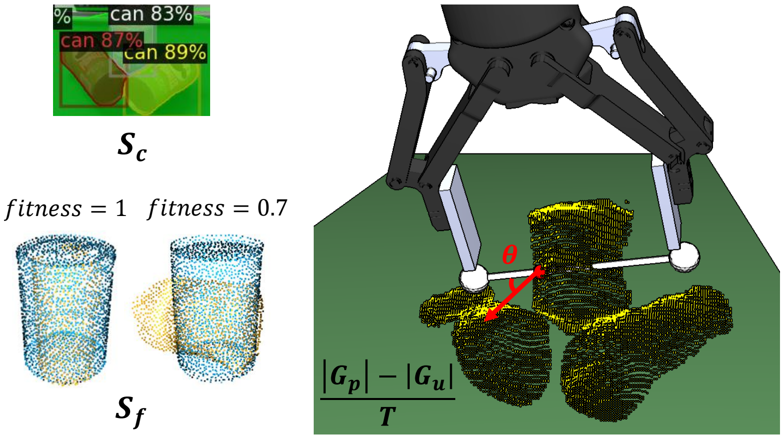


Figure 5.4: Multi-metric evaluation of grasp sequences in cluttered scenes.

to be unavailable, we categorize such grasps as potential grasps and use them only when none of the grasps in G_p are evaluated as stable. Finally, we develop a multi-metric function for determining the grasp sequence as follows:

$$P = S_c * S_f * \frac{|G_p| - |G_u|}{T} \tag{1}$$

where P denotes the grasp priority of each object in the clutter. Each time after planning grasps for the object with the highest priority, we remove that object from the grasp planning of the other objects and recalculate the grasp priority for the remaining objects.

5.2.5 Adaptive Local Detection

In a dynamic grasping task, we need to obtain the movement state of moving objects in addition to appropriate grasp poses. For this purpose, we move the in-hand camera to an overhead position that can observe the objects from a top-down view perpendicular to the moving direction. We use a frame-difference based object tracking algorithm[2] to obtain the 2D bounding box for cluttered objects in real-time images. By utilizing the fact that the camera's field of view (FOV) is known, we can accurately estimate the clutter's movement speed by capturing two key time points: 1) t_1, when the clutter just enters the camera view; 2) t_2, when the clutter reaches the horizontal centerline (HC)

---

[2] https://learnopencv.com/object-tracking-using-opencv-cpp-python

of the camera view. The movement speed can be obtained by dividing the distance the clutter traveled between t_1 and t_2 by the time difference. However, since the FOV of a typical camera is only known in vertical and horizontal directions, the travel distance is only available when the clutter enters from the vertical centerline (VC) of the camera view, as shown in Fig. 5.5 left. Therefore, we adaptively set the camera position so that the frontmost point of the clutter is aligned with VC, and the camera height is maintained at a certain distance (30 cm in our task) above the highest point of the clutter. This can be easily achieved since the point clouds of all objects in the clutter have been obtained during global visual detection.

Two other factors that significantly affect the accuracy of speed estimation are: 1) the first emerging point, which represents the part of an object that first appears at a certain position in the camera view, as shown in Fig. 5.5 left. When this point changes between t_1 and t_2, the real travel distance is an unknown value between the lengths of the green and red lines. However, we can approximate it by obtaining the depth of the first emerging point at t_1 and t_2 (denoted as d_1 and d_2, respectively) and averaging their values; 2) the detection latency, which is an unavoidable error occurring when the moving clutter has traveled a short distance beyond the target line before being detected. This error becomes significant at higher movement speeds and can be compensated by accounting for the number of pixels traversed, denoted as Δh_1 and Δh_2 for t_1 and t_2, respectively. We can then derive the following equation for accurate speed estimation:

$$\widetilde{v} = \frac{(d_1 + d_2)/2 * \tan(0.5 + (\Delta h_2 - \Delta h_1)/h)\alpha}{t_2 - t_1} \tag{2}$$

where α is the camera's FOV in the moving direction and h is the vertical resolution of the camera view. We validate the efficiency of this calculation method by testing in two conveyor speed modes, each with 10 trials using different types or arrangements of cluttered objects, as shown in Fig. 5.5 right. The average test results for the two speed modes are 5.45 ± 0.10 (cm/s) and 11.05 ± 0.23 (cm/s), both of which are very close to the ground truth with minimal fluctuation.

In addition, we can also obtain the instantaneous position p_2 of the clutter at t_2

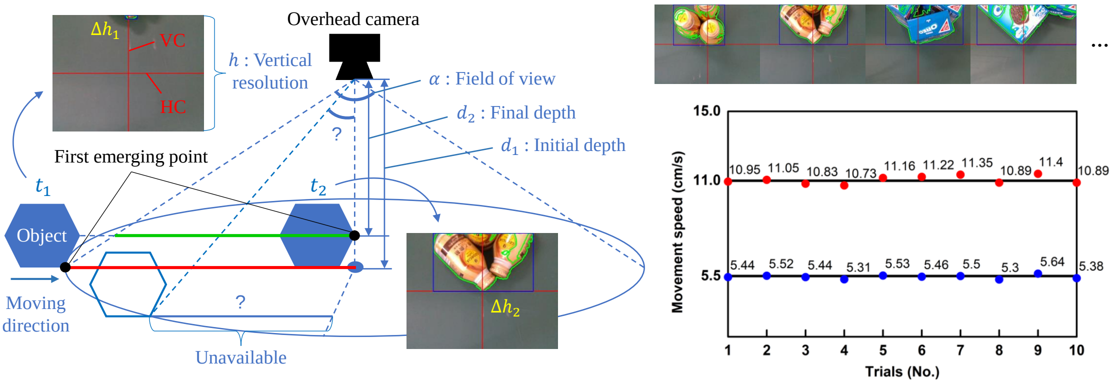


Figure 5.5: (Left) Speed estimation of moving cluttered objects based on adaptive local visual detection and real-time object tracking. (Right) Evaluation of the calculation results in two conveyor speed modes (5.5 cm/s and 11 cm/s). In each trial, we change either the object arrangement or object type.

since the camera position is known, and perform dynamic grasp planning based on $\widetilde{v}$ and p_2.

5.2.6 Dynamic Grasp Planning

Using the results of static grasp planning and local visual detection, we develop a recurrent algorithm to achieve dynamic grasping for moving objects in dense clutter (see Algorithm 2). In the first step of grasping the first-priority object, we set a time interval of $\Delta t = 1$s and predict the future position of the clutter at each subsequent Δt after t_2 based on $\widetilde{v}$ and p_2. Taking the future state t_3 as an example, we translate the static grasps to the predicted position p_3 and check their feasibility. When there is more than one feasible grasps at p_3, we select the closest grasp and let the robot move to its pregrasp pose. During this process, we record both the time used for grasp planning and the time used for the robot to move from the local observation pose to the pregrasp pose as t_p and t_m, respectively. Meanwhile, we assume the time taken by the robot to approach the grasp pose from the pregrasp pose to be $t_a = 0.5$s. If $t_p + t_m + t_a < \Delta t$, it means that the robot has enough time to complete the grasping motion before the clutter reaches p_3. In this case, we let the robot wait for a small period of time (the time difference) and then execute the grasp. Otherwise, we proceed to the next future state and repeat the process from planning to moving until the time condition is satisfied.

In the subsequent steps of grasping lower-priority objects, we delay t_2 by the amount of time taken to grasp the previous objects and repeat the same process as in the first step. Each step is done in a separate thread to prevent conflicts.

5.3 Experiments

5.3.1 Experimental Setup

We carry out grasping experiments using a UR5e robot arm equipped with a Robotiq 2F-140 adaptive gripper, an in-hand RealSense D435 depth camera and a standard-type conveyor belt. The existing database we use for similarity matching is the YCB dataset [80], excluding objects without mesh models or with distorted models. All computations are performed on a PC equipped with a Ryzen 7 5800H CPU and a GeForce RTX 3060 GPU. The Detic model and GPT-4o model are pre-loaded to reduce task processing time.

5.3.2 Dynamic Grasping Experiments

We select various types of novel objects and make them into dense clutter for grasping experiments. In order to verify the effectiveness and generalizability of our method, we categorize the clutters into three patterns (see Fig. 5.6): (P1) three identical objects; (P2) three different objects; (P3) five different objects. For each pattern, we create five different clutter sets (C1-C5). For each clutter set, we generate ten different object arrangements (A1-A10). Thus, we conduct experiments on a total of $3 \times 5 \times 10 = 150$ clutter scenarios.

We test in two conveyor speed modes (5.5 cm/s and 11.0 cm/s) and evaluate the results by two metrics: **Success Rate (SR)** and **Execution Rate (ER)**, calculated as SR = Number of Successful Grasps / Number of Grasps Performed, ER = Number of Grasps Performed / Number of Objects Targeted. Due to time constraints in the dynamic grasping task, we set the number of objects targeted to two in the 5.5 cm/s

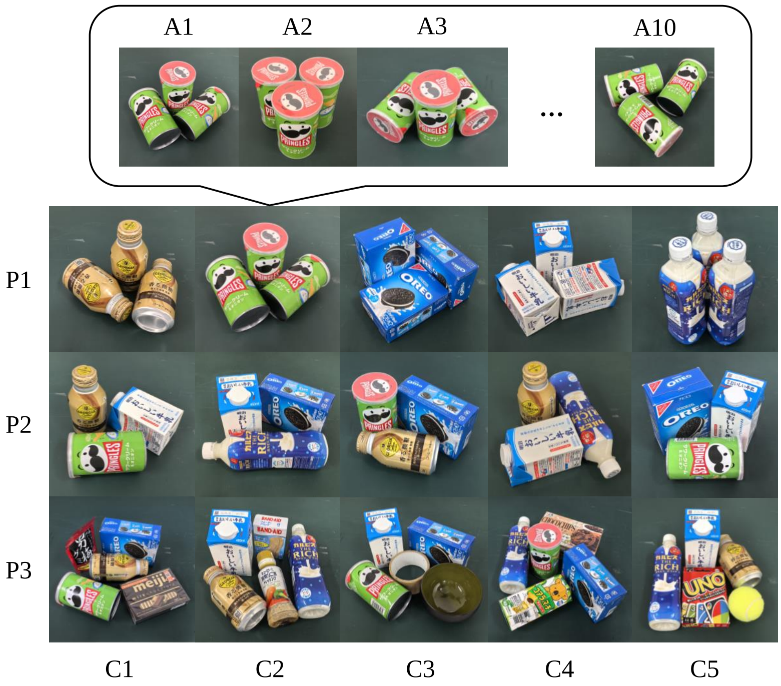


Figure 5.6: Experimental objects in 3 patterns, 5 clutters, and 10 arrangements.

speed mode and one in the 11.0 cm/s speed mode. The objects to be grasped are determined autonomously by the planning process. In general, SR and ER can be used to represent the accuracy and efficiency of our method, respectively.

A complete task cycle is: Place the cluttered objects at the start of the conveyor belt → Run the conveyor, start visual detection and grasp planning → Grasp the first object and place it in a nearby box → Grasp and place the second object (only in the 5.5 cm/s scenario) → Stop the conveyor. An object is considered successfully grasped if it is placed in the box without being dropped during the process.

Table 5.1 shows the experimental results. Overall, our method can handle moving clutter with different object numbers, types and arrangements with both high success rate and execution rate. In both speed modes, P1 and P2 achieve close results, indicating that our method performs stably with multiple object types coexisting. In the 5.5 cm/s speed mode, most execution failures occur in grasping the second object (the clutter has reached the end of the conveyor belt before a grasp is output), leading to lower ER compared to the 11.0 cm/s speed mode where only one object is targeted.

Table 5.1: Experimental Results for Grasping Various Types of Moving Clutter

Speed mode		Conveyor speed = 5.5 cm/s						Conveyor speed = 11.0 cm/s					
Clutter set		C1	C2	C3	C4	C5	Average	C1	C2	C3	C4	C5	Average
P1	SR	76%	88%	94%	82%	80%	84%	67%	80%	80%	78%	70%	**75%**
	ER	85%	80%	95%	85%	75%	84%	90%	100%	100%	90%	100%	**96%**
P2	SR	88%	89%	82%	78%	93%	**86%**	67%	70%	78%	75%	63%	71%
	ER	85%	90%	85%	90%	75%	**85%**	90%	100%	100%	80%	90%	92%
P3	SR	77%	73%	79%	75%	88%	78%	60%	63%	57%	75%	67%	64%
	ER	65%	75%	70%	60%	85%	71%	100%	80%	70%	80%	90%	84%

However, in contrast, we observe more grasping failures under higher conveyor speed due to its smaller allowance for positional error during grasping. When the number of objects increases such as in P3, the processing time also increases and more surrounding obstacles lead to more collisions, resulting in lower SR and ER compared to scenes with fewer objects. In addition, we find that objects with regular shapes are more likely to be selected for priority grasping, e.g., the green can and the blue box in Fig. 5.6. The reason is that they achieve better results in point cloud registration and find stable grasps more quickly due to their regularly shaped point clouds.

5.3.3 Failure Analysis

For both grasp failures and execution failures, we summarize the reasons and possible solutions as follows:

1) **Large occlusion rate.** When the cluttered objects are arranged in such a way that a large part is invisible to the camera, the extracted object point clouds are too sparse for stable similarity matching and grasp planning results. A possible solution is to add more viewpoints or observe more times to obtain more complete appearance of the objects.

2) **Too dense arrangement.** When the objects are closely packed and there are too few graspable areas, the planning process takes a long time and the clutter may

have reached the end of the conveyor belt before a feasible grasping motion is output. A possible solution is to combine other skills, such as pushing objects, to create more space for grasping.

3) **Intermediate collision.** When the first object is grasped and moved to the placement position, other objects in the clutter are still moving, and collisions between the robot and the moving objects may occur during this process. A possible solution is to develop a real-time motion planning method to avoid collisions in such dynamic scenes.

5.4 Conclusions

In this study, we present a novel framework using global-to-local detection and static-to-dynamic planning to achieve grasping of moving objects in dense clutter. Especially, we use an improved similarity matching method to efficiently plan grasp poses and sequences for cluttered objects, and propose an adaptive object tracking method to accurately estimate the movement speed and the instantaneous position of the moving clutter. Based on these results, we develop a recurrent algorithm to achieve continuous dynamic grasping.

The effectiveness and generalizability of our method is verified through real-world experiments. However, additional methods can be combined to address the limitations of the current approach in future work.

Algorithm 2: Dynamic grasp planning for cluttered objects

```
Input: Grasp poses and sequences from static planning, estimated movement speed
  and instantaneous position
  Step 1: Grasp the first-priority object $o_1$
  Initialize $n = 3$, $t_c = 0\text{s}$, $t_a = 0.5\text{s}$ and $\Delta t = 1\text{s}$
  while True do
    $t_n = t_{n-1} + \Delta t$ (Initially, $t_3 = t_2 + 1\text{s}$)
    Predict the clutter's future position $p_n = p_{n-1} + \tilde{v}\Delta t$
    for each static grasp pose $g \in o_1$ do
      Translate the grasp from $p_0$ to $p_n$
      Compute IK solutions and check collisions
      if $g$ is IK-solvable and collision-free then
        Save $g$ into $G_f$
    if $G_f \neq \emptyset$ then
      Find the optimal grasp pose $g^* \in G_f$ with the minimum distance to the local
      observation pose
      Let the robot move to the pregrasp pose of $g^*$
      Record the planning time $t_p$ and motion time $t_m$
      Calculate the cumulative time $t_c = t_c + t_p + t_m$
      if $t_c + t_a < (n-2)\Delta t$ then
        Wait for $(n-2)\Delta t - (t_c + t_a)$ seconds
        Let the robot move to $g^*$ and execute the grasp
        break
    $n = n + 1$
  Step 2: Grasp the second-priority object $o_2$
  Record the entire duration of grasping the first object $t_d$
  In another thread, delay $t_2$ by $t_d$ with all other parameters unchanged and repeat
  the same process as in Step 1
  Step 3 ($o_3$), Step 4 ($o_4$), ... : Same as Step 2 =0
```

Chapter 6

Application of Similarity-Based Methods in Task-Oriented Object Grasping

This thesis chapter originally appeared in the literature as
H. Chen, T. Kiyokawa, W. Wan, and K. Harada, "Generalizable task-oriented object grasping through LLM-guided ontology and similarity-based planning," *Robotics and Autonomous Systems.* (Under Review)

6.1 Introduction

Robotic grasping is advancing beyond conventional pick-and-place operations toward more human-like behaviors, where understanding *part affordance* [88, 89] is crucial for achieving task-oriented grasping (TOG). For instance, a robot is expected to identify and grasp the *handle* of a cup in a *pouring* task. In real-world scenarios, however, human instructions are often intuitive and semantically rich, extending far beyond simple action labels like *pouring*. As a result, the accurate interpretation of user intensions becomes essential for both human-guided manipulation and human-robot collaboration tasks.

Moreover, the reliable recognition of object functional parts from partial observation while ensuring the generation of high-quality grasps remains a significant challenge, leaving this research problem largely unsolved.

More than a decade ago, the concept of *semantic grasping* was introduced [40, 41] to enable high-level manipulation considering task constraints, rather than focusing solely on basic grasping. However, at the time, even with the aid of vision and tactile sensors, analytical approaches struggled to generate high-quality grasps without human supervision and showed limited generalization. With the rapid progress in deep learning and image segmentation, more advanced strategies based on visual recognition have emerged [42, 43, 44], demonstrating notable performance in planning task-oriented grasps across a wide variety of objects. Nevertheless, these vision-based methods typically represent target tasks using predefined cues such as *grasp from the handle* or simplistic labels like *handover* or *cut*, lacking the ability to translate contextual human instructions into specific robotic actions. This interaction challenge has been widely addressed in recent years with the advent of vision-language models (VLMs) [45, 46, 47]. By leveraging high-dimensional language embeddings and attention-based learning frameworks, robots are now increasingly capable of responding to natural human commands. However, several limitations persist in current research: 1) low grasping DoF [45]; 2) reliance on complete mesh models or point clouds [46]; and 3) significant performance degradation when generalizing to unseen objects [47].

To overcome these limitations, we propose a novel strategy for TOG that integrates an object-part-task ontology guided by large language models (LLMs) with a grasp detection method based on similarity matching with known templates. This approach enables accurate interpretation of intuitive human instructions and robust 6-DoF grasp generation by leveraging references from similar pre-existing instances. An example application of our proposed method is shown in Fig. 6.1. Given a previously unseen object and a natural human instruction, we first process the semantic information by associating the human instruction with an existing ontology and identifying the functional part of the target object in the given task. Then, we utilize the geometric information obtained from single-view observation and employ an analytical method to recognize the

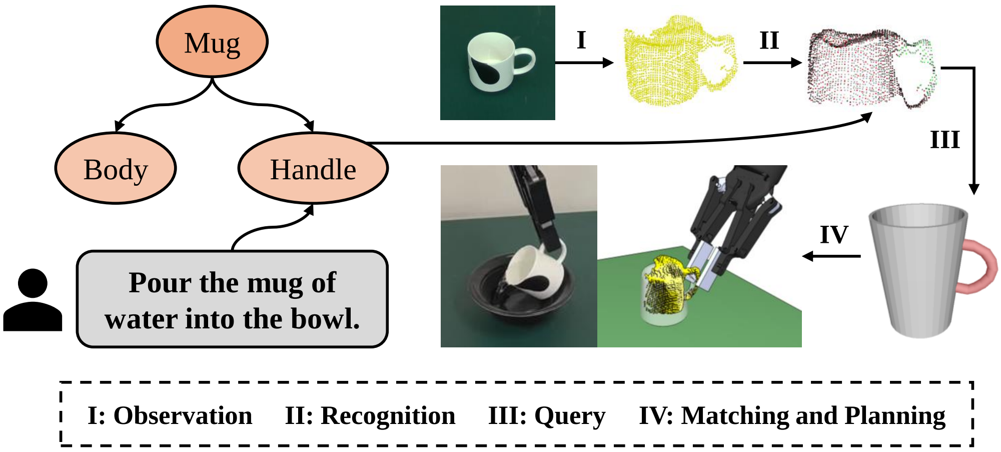

Mug
Body
Handle
Pour the mug of water into the bowl.
I
II
III
IV
I: Observation
II: Recognition
III: Query
IV: Matching and Planning

Figure 6.1: Demonstration of our method applied to a *pouring* task with an unseen mug.

corresponding functional part, represented as a point cluster. Based on this recognition result, we query a similar model template containing prior segmentation and grasping knowledge as a reference. Finally, through an optimized matching and planning process, a high-quality 6-DoF grasp pose is generated and executed to successfully accomplish the task. The effectiveness of our method is validated by extensive real-world experiments, notably demonstrating strong generalization to novel-category objects via the scalable integration of LLM-guided ontology and similarity-based planning.

Our main contributions can be summarized as:

- We propose an object-part-task ontology guided by LLMs that efficiently and accurately maps intuitive human instructions to the corresponding functional parts of target objects.

- We introduce a geometry-based part recognition method that is robust to viewpoint variations by leveraging similar model templates as references for functional part matching and imitative grasp planning.

- We optimize the quality of generated grasps from similar references through a combination of local-to-global point cloud registration and stability-aware positional adjustment.

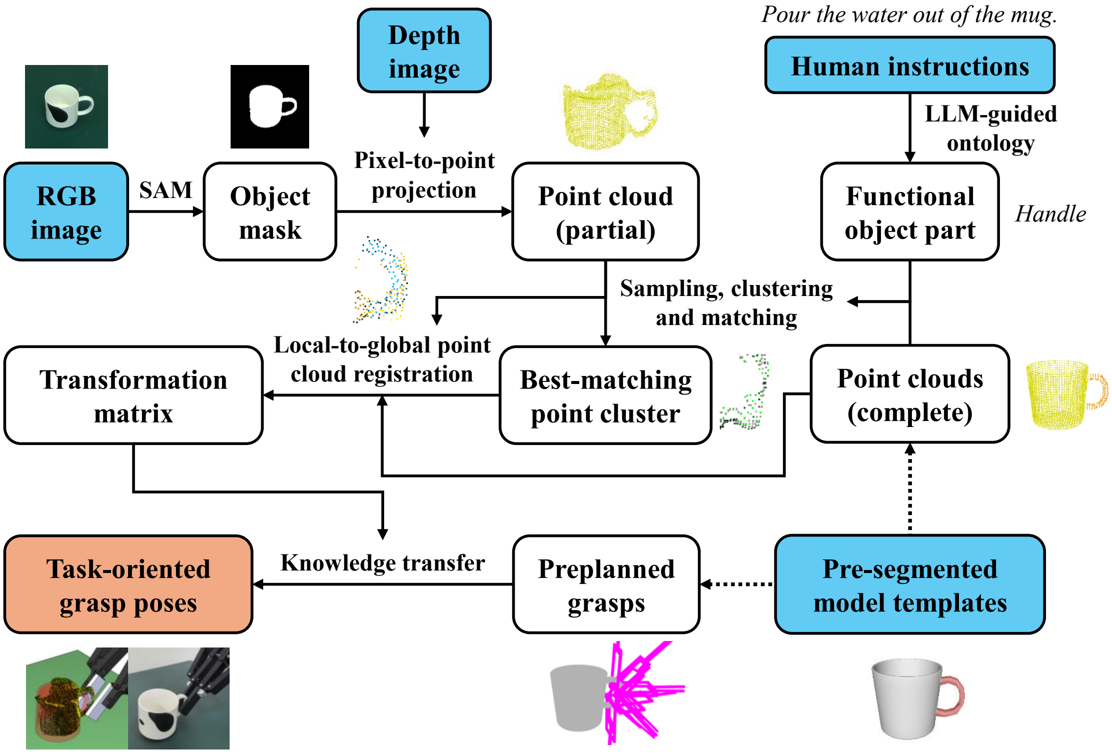

Pour the water out of the mug.
Human instructions
LLM-guided ontology
Functional object part
Handle
Depth image
RGB image
SAM
Object mask
Pixel-to-point projection
Point cloud (partial)
Sampling, clustering and matching
Local-to-global point cloud registration
Transformation matrix
Best-matching point cluster
Point clouds (complete)
Task-oriented grasp poses
Knowledge transfer
Preplanned grasps
Pre-segmented model templates

Figure 6.2: Proposed TOG system overview. Blue/Orange indicate Inputs/Outputs.

6.2 Methods

6.2.1 Overview

Our goal is to enable a robot to execute task-oriented grasps based on intuitive human instructions. Given an instruction, the robot uses an RGB-D camera to detect the target object, identify the relevant functional part, and plan feasible grasps that fulfill the intended task. The overall process consists of three core components: *Human Instruction Interpretation*, *Functional Part Recognition*, and *Task-Oriented Grasp Planning*. A high-level overview of our TOG system is illustrated in Fig. 6.2.

On the left side, we process RGB-D images captured by a wrist-mounted camera through mask extraction and pixel-to-point projection, resulting in a partial point cloud of the target object. On the right side, we process human instructions using an LLM-guided ontology to identify the relevant functional part, which is used to guide the subsequent matching and planning stages. Meanwhile, we prepare a database of model templates, each segmented into multiple functional parts according to the defined ontology. These templates contain both complete point clouds and preplanned grasping knowledge.

Based on the identified functional part and the segmented template point clouds, we apply a sampling-clustering-matching method to recognize the corresponding part (a point cluster) within the observed object point cloud. We then perform local-to-global point cloud registration using both the full template point cloud and the selected point cluster to align the observed object with the best-matching model template, yielding a transformation matrix. This matrix is used to transfer preplanned grasping knowledge from the model template to the observed object. Finally, a dedicated optimization process generates robust task-oriented grasps tailored to the given instructions and object geometry.

6.2.2 Human Instruction Interpretation

In human-guided manipulation and human-robot collaboration tasks, accurately interpreting human instructions is essential for generating appropriate robot actions. However, the rich semantic knowledge embedded in contextual instructions is difficult to extract without a powerful inference model. Recent advances in LLMs have made this level of understanding feasible. In the context of TOG, a key challenge is determining *which part of the object the robot should grasp*—the central problem addressed in this work. To associate human instructions with relevant object parts, we propose an LLM-guided object-part-task ontology.

As shown in Fig. 6.3, the proposed ontology consists of two components: offline and online. The offline ontology predefines object classes with multi-layer part hierarchies. For instance, a *mug* is divided into *handle* and *body*, while the *body* can be further subdivided into *inside* and *outside*. Different object parts correspond to different grasping strategies, which are flexibly selected according to task requirements. The online ontology handles user-provided instructions, which are not previously known. To establish connections between task instructions and relevant functional parts, we leverage

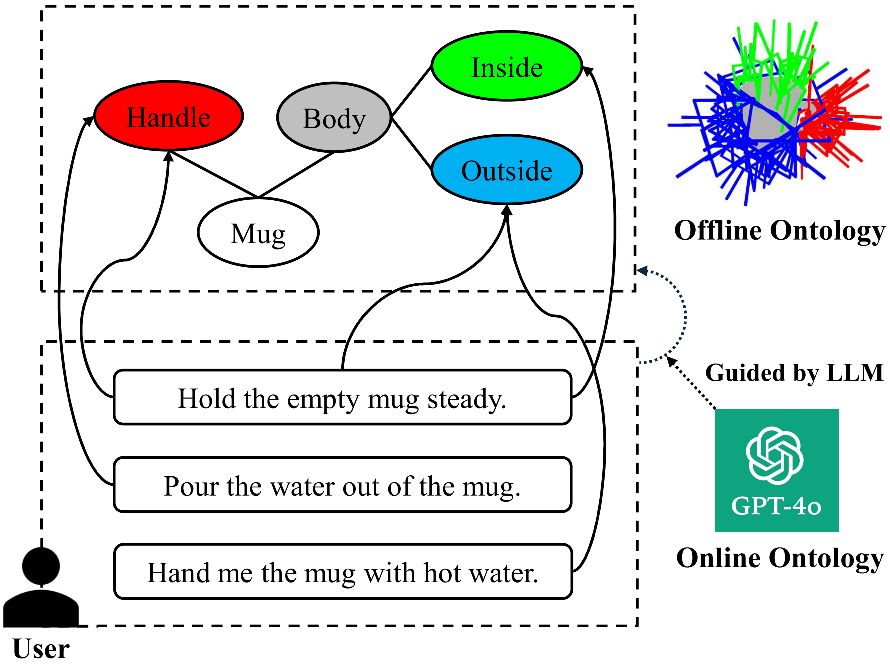

Handle
Body
Inside
Outside
Mug
Offline Ontology
Guided by LLM
GPT-4o
Online Ontology
Hold the empty mug steady.
Pour the water out of the mug.
Hand me the mug with hot water.
User

Figure 6.3: LLM-guided object-part-task ontology with online and offline components.

the latest LLM, GPT-4o [79], to bridge the online and offline components. The prompt used for this association is originally structured as follows:

> *Given the following ontology ... A robot is given the following commands ... Question: Which part(s) of the object should the robot grasp?*

Despite the strong interpretability of LLMs, we observe that their answers can vary across trials, lacking accuracy and consistency. To address this, we first adopt the prompt optimization method from [90], which uses LLMs themselves as prompt optimizers. Implementation details are provided in Appendix A. While the optimized prompts incorporating task constraints improve answer correctness, they still fall short of human-level reliability. Through a large number of trials, we identify two prompt design principles that significantly enhance LLM performance: 1) Step-by-step reasoning; 2) Using an answer template. In practice, we append a step-by-step reasoning template to the end of each prompt, following this structure:

> *The command is ... Step 1: Identify the type of task ... Step 2: Apply task constraints ... Analyzing the object parts ... Best choice for the robot ...* ***Conclusion:***

The robot should grasp ...

Using this structured prompt, we observe that the LLM's response accuracy approaches 100% across varying tasks. The functional part information is easily extracted from the **Conclusion** line of the answer.

6.2.3 Functional Part Recognition

To map the semantic identification of functional object parts into geometric space for subsequent planning, we need to extract part-level features from visual inputs. A common approach is to train a vision model for object part segmentation. However, we observe that even state-of-the-art segmentation models [91] suffer from viewpoint sensitivity, where segmentation results become inconsistent under varying viewpoints. As shown in Fig. 6.4, components such as the *mug handle* and the *bottle cap* are not reliably identified when the objects are placed in different poses. To address this limitation, we propose a new strategy based on template-assisted geometric analysis.

Given the RGB input containing the target object, we first apply a powerful class-agnostic segmentation method that is robust to viewpoint variations, SAM [77], to obtain an object mask indicating the object's planar location. Using the corresponding depth input, we perform pixel-to-point projection to map the 2D mask into a 3D point cloud, capturing the object's spatial geometry. However, due to partial observation, the resulting object point cloud often contains large unseen areas, making it difficult to perform reliable part segmentation with incomplete geometric features. To address this, we incorporate pre-segmented model templates associated with the predefined ontology and use them as references to guide part segmentation. This template-guided process consists of three main steps: sampling, clustering and matching, as illustrated in Fig. 6.5. We begin by converting all model templates from mesh representations to point clouds using a voxel grid filter with an appropriate leaf size (5 mm in our task). Then, in the sampling step, we apply the same voxel filter to downsample the observed object point cloud, ensuring uniform point density. Next, we take each sampled point as a seed and search for its k-nearest neighbors to form a local point cluster. The number of

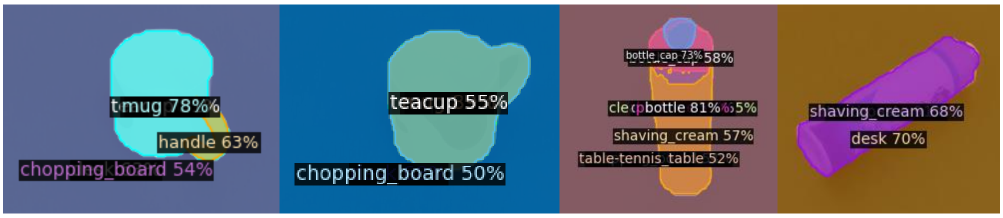


Figure 6.4: Sensitivity of model-based part segmentation to viewpoint changes.

neighbors k is determined using the following equation:

$$k = \frac{N(o_{all})N(m_{part})}{N(m_{all})} \tag{6.1}$$

where $N(\cdot)$ denotes the number of points in a point cloud. o_{all} and m_{all} represent the entire point clouds of the observed object and the model template, respectively. m_{part} corresponds to the point cloud of the functional part within the template. Finally, the sampled point clusters are matched against the pre-segmented model templates to identify the best-matching cluster, the one that most closely resembles the target functional part.

However, the matching process is non-trivial. As illustrated in Fig. 6.5, point clouds captured by consumer-grade cameras often suffer from low precision due to sensing noise. To ensure the accuracy, efficiency, and stability of the matching process under such conditions, we introduce a multi-metric similarity evaluation method that considers both local and global geometric similarity. Letting a sampled point cluster be denoted as o_{part}, we first apply Principal Component Analysis (PCA) to evaluate the similarity in point distribution between o_{part} and m_{part}, using the following metric:

$$d_{pca} = \left\| \frac{\sigma_o}{|\sigma_o|} - \frac{\sigma_m}{|\sigma_m|} \right\|_2 \tag{6.2}$$

where d_{pca} represents the PCA-based distributional difference between o_{part} and m_{part}. σ_o and σ_m denote the 3-dimensional PCA singular value vectors of o_{part} and m_{part}, respectively, normalized by their Euclidean norms. This evaluation allows for a quantitative comparison between point distributions. However, PCA alone is insufficient to fully capture shape similarity. An intuitive example is that a cube and a sphere may

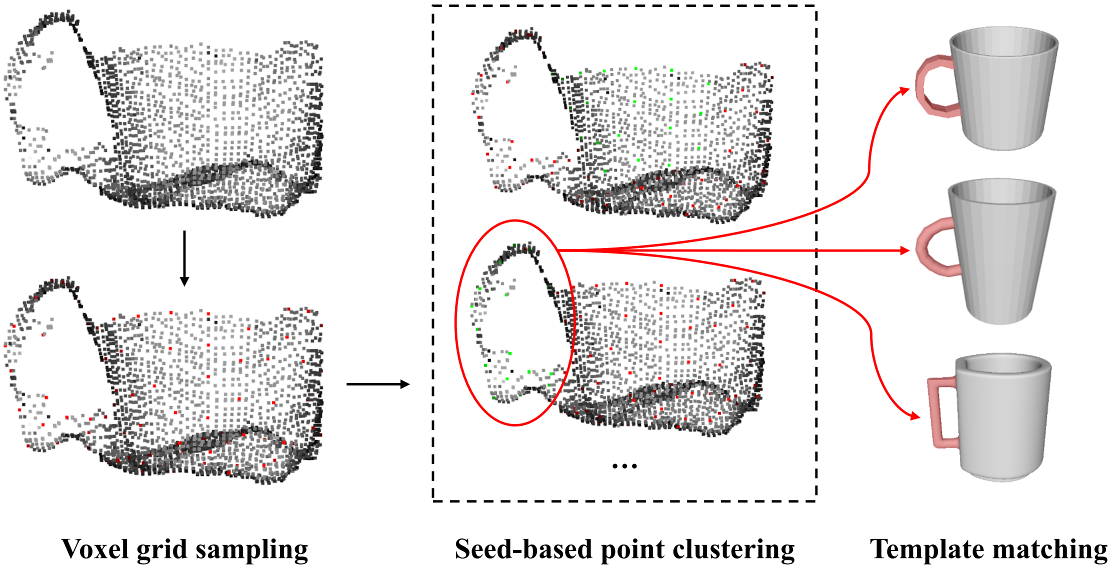


Figure 6.5: Template-assisted object part recognition using a three-step strategy.

yield similar PCA results due to geometric symmetry, despite having fundamentally different shapes.

To address this issue, we introduce a second evaluation metric: Point-to-Point Distance (PPD). As illustrated on the left side of Fig. 6.6, we use the center point (for m_{part}) and the seed point (for o_{part}) to calculate the PPD. The center point is identified by first computing the center of the bounding box of m_{part}, and then finding its nearest neighbor within m_{part}. Based on this point, we calculate the standard deviation of its distances to all other points in m_{part}, denoted as $D(s_m)$. For the seed point sampled in o_{part}, we similarly compute the standard deviation of its distances to all other points in the local cluster, denoted as $D(s_o)$. The PPD is then evaluated as:

$$d_{ppd} = \left| \frac{D(s_o)}{\max(|s_o - \overline{s_o}|)} - \frac{D(s_m)}{\max(|s_m - \overline{s_m}|)} \right| \tag{6.3}$$

where $\max(\cdot)$ denotes the maximum deviation from the mean occurring during the point distance calculation. The resulting value, d_{ppd}, reflects the difference in point dispersion between m_{part} and o_{part}, offering an additional cue for distinguishing functional parts with different spatial shapes. However, this metric is non-directional and does not capture the full 3D geometric structure, making it unsuitable for use in isolation.

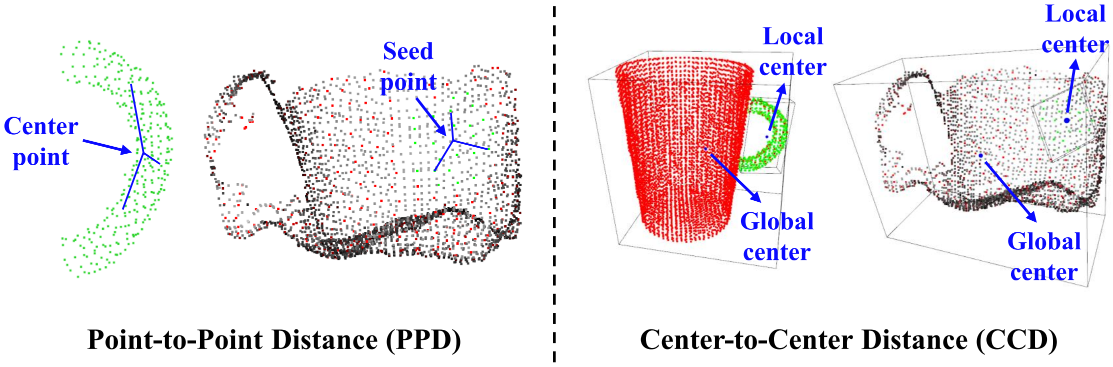


Figure 6.6: Two distance metrics for evaluating similarity between point clusters.

While the combination of d_{pca} and d_{ppd} performs well for evaluating similarity between point clusters, they focus primarily on local geometry and can become less reliable when visual features are sparse or noisy. To address this limitation, we incorporate a third metric: Center-to-Center Distance (CCD). As illustrated on the right side of Fig. 6.6, we obtain the bounding box centers of both the entire point clouds m_{all}, o_{all} and the part point clouds m_{part}, o_{part}, denoted as global center and local center, respectively. The CCD is then evaluated as:

$$d_{ccd} = \left| \frac{c_o}{b_o} - \frac{c_m}{b_m} \right| \tag{6.4}$$

where c_o and c_m represent the distances between the local and global centers within the observed object and the model template, respectively. b_o and b_m denote the maximum possible distances, defined as half the diagonal length of the bounding boxes of o_{all} and m_{all}, respectively. The resulting value, d_{ccd}, reflects the difference in the relative global positions of m_{part} and o_{part} within their respective entire objects, thereby enhancing functional part recognition especially when local visual features are sparse or noisy.

Combining the three metrics, we define the final evaluation function as:

$$d = d_{pca} + d_{ppd} + d_{ccd} \tag{6.5}$$

where d represents the overall difference between a template functional part and a sampled point cluster within the observed object. Smaller values of d indicate higher similarity. Note that in Equations 6.2-6.4, all metrics have been normalized using appropriate boundary values, allowing them to be combined directly through addition. The final output of the matching process is the point cluster o^*_{part} that achieves the lowest average d across matches with m_{part} from all available templates.

6.2.4 Task-Oriented Grasp Planning

After recognizing the functional part represented by o^*_{part}, the next step is to plan robust grasps within its region to achieve the goal of TOG. However, due to the sparse and noisy point features in o^*_{part}, directly generating reliable grasps from the point cloud is extremely challenging. To overcome this, we leverage the matched model template again to transfer grasping knowledge from known objects to unknown targets. For each model template, we use a mesh-segmentation-based approach [68] to preplan over 50 grasps on each subdivided part based on the offline ontology. For a given task, only the preplanned grasps associated with the relevant functional part are utilized. To transfer this grasping knowledge from the template to the object, we first need to obtain the transformation matrix between the point clouds o_{all} and m_{all} to determine their relative pose. However, the incomplete point cloud from partial observation poses a challenge for accurate alignment with the complete template point cloud using traditional registration methods like ICP [84]. To address this, we propose a novel strategy called local-to-global registration, which aligns the point clouds step by step, from the functional part to the entire object, thereby enhancing alignment accuracy.

As illustrated in Fig. 6.7, we begin by matching o^*_{part} with m_{part} using a combined registration algorithm, RANSAC + ICP, implemented in Open3D[1]. This process is repeated iteratively until a sufficient number of point correspondences (over half of the points in o^*_{part}) are identified, indicating that the two point clouds are well-aligned. Next, we apply the resulting transformation matrix, denoted as T_{loc}, to initially align o_{all} with m_{all}. At this stage, although the functional parts are aligned, the entire objects may still show some misalignment due to rotational errors in the local registration.

---

[1] http://open3d.org/docs/release/tutorial/pipelines

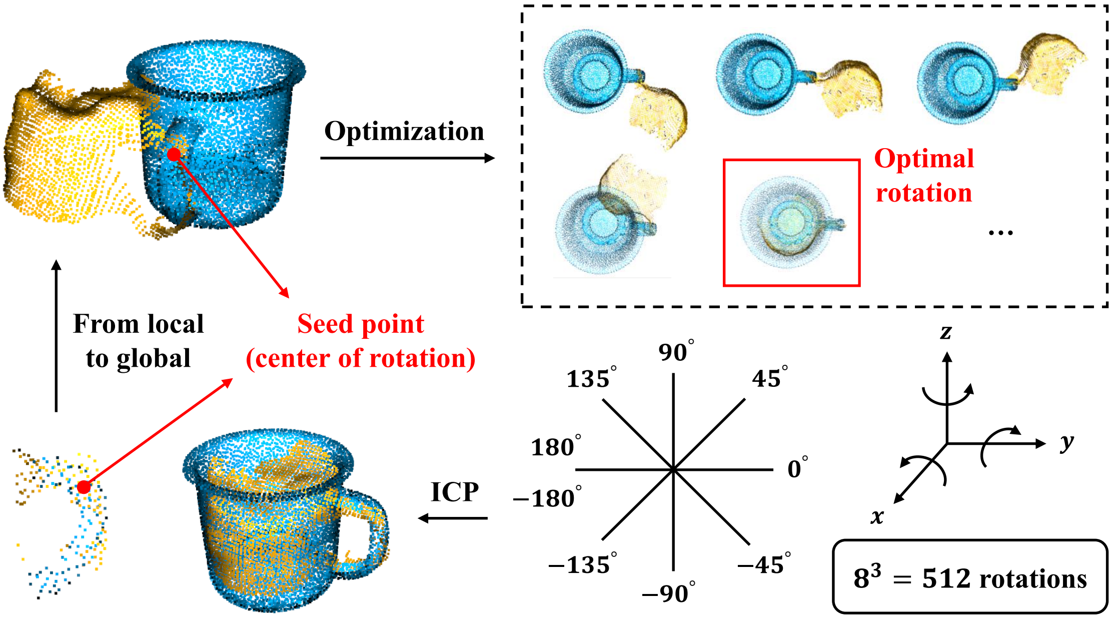


Figure 6.7: Local-to-global point cloud registration through rotation optimization.

Directly applying ICP at this point often fails to improve the result. To overcome this, we introduce an optimization method that rotates o_{all} around the seed point of o^*_{part} to find an optimal rotation, denoted as T_{opt}, which minimizes the distance between o_{all} and m_{all}. The rotation space is discretized by sampling angles from $-180°$ to $180°$ in $45°$ increments along each of the three axes, resulting in $8^3 = 512$ total rotation attempts. This optimization process does not take long with parallel computing. Once the optimal rotation is identified, we apply the ICP algorithm to further refine the alignment between o_{all} and m_{all}, ensuring precise registration. Assuming the resulting transformation matrix from ICP is T_{icp}, the final transformation from o_{all} to m_{all} is then calculated as $T = T_{loc}T_{opt}T_{icp}$.

While this local-to-global registration is performed between the target object and all relevant model templates, we use only the best-matching template, which achieves the highest fitness score in the final ICP process, as the reference for grasp planning. Assuming the set of preplanned grasps on the best-matching template is G_m, and the camera pose relative to the robot base (the origin of the world coordinate system) is T_0, we can generate a corresponding set of imitative grasps on the target object as $G_o = T_0T^{-1}G_m$. To ensure executable grasps for real-world tasks, we perform IK computation

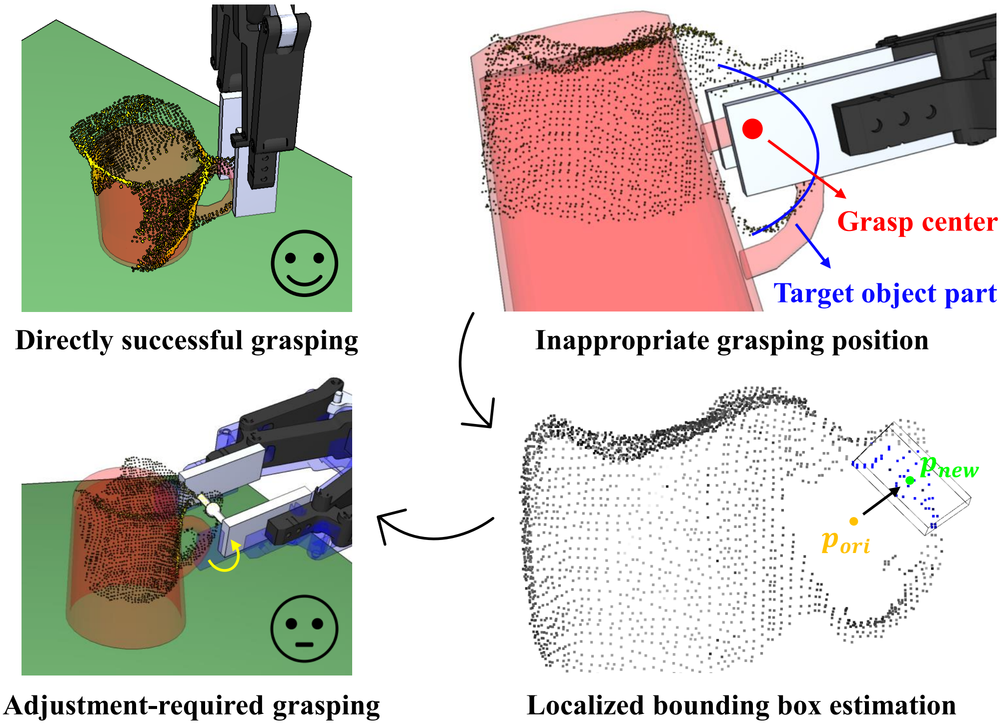

Directly successful grasping
Inappropriate grasping position
Grasp center
Target object part
p_{new}
p_{ori}
Adjustment-required grasping
Localized bounding box estimation

Figure 6.8: Adjustment of suboptimal grasps via localized bounding box estimation.

and collision detection within a simulation environment[2], filtering out infeasible grasps that are either unreachable or result in collisions. For selecting reliable grasps positioned on the functional part, we prioritize those located within the region of o^*_{part} by generating a cube collision model M_{cube} within the gripper closure area and a point cloud collision model M_{pcd} corresponding to o^*_{part}. When $M_{cube} \cap M_{pcd} \neq \emptyset$, the grasp is considered correctly positioned and selected for execution.

However, we observe that the transferred imitative grasps are not always highly stable due to inherent differences between the model template and the target object. As shown in Fig. 6.8, when the two point clouds align perfectly, the generated grasps are typically executable without failure. However, when small deviations occur, misalignment between the grasp center and the target object part may introduce potential risks, such as: 1) Collision between the grasp pose and other parts of the object; 2) Suboptimal contact between the gripper and the object, which may reduce grasp stability. To detect such conditions, we generate a stick collision model M_{stick} connecting the two ends of

---

[2] https://github.com/wanweiwei07/wrs

the gripper. When $M_{stick} \cap M_{pcd} = \emptyset$, the grasp center is considered misaligned with the target object part and requires adjustment. Assuming the original grasp center is p_{ori}, we search for its nearest neighbor point in o^*_{part}, denoted as p_{neg}. We then search for the k-nearest neighbors of p_{neg} and estimate a localized bounding box (LBB) based on these points. Here, the value of k is approximately set to half the number of points in o^*_{part}. The center of the LBB is adopted as the new grasp center, denoted as p_{new}. Finally, the original grasp pose is translated along the vector $\overrightarrow{p_{ori}p_{new}}$ to let the grasp center align with the target object part, thereby improving stability.

6.3 Experiments

6.3.1 Experimental Setup

To validate the effectiveness of our proposed method, we conduct the following TOG experiments: 1) a benchmark study on grasping various types of unseen objects under different human instructions; 2) an ablation study to evaluate the local-to-global registration and the stability-aware grasp adjustment; and 3) an investigation into the generalization to novel-category objects that are not part of the existing ontology. All experiments are performed in the real world, using a UR5e robot arm equipped with a Robotiq 2F-140 adaptive gripper and a wrist-mounted RealSense D435 RGB-D camera. The experimental objects are placed on a fixed platform within the robot's reachable workspace, and are observed by the wrist camera from a diagonal downward view. All computations are carried out on a computer equipped with a Ryzen 7 5800H CPU and a GeForce RTX 3060 GPU.

6.3.2 Task-Oriented Grasping of Unseen Objects

For the first TOG experiment, we select three types of divisible objects with functional parts: *mug*, *bottle*, and *scissor*. Each category includes three previously unseen items with varying shapes and sizes, and their predefined ontological knowledge is stored in

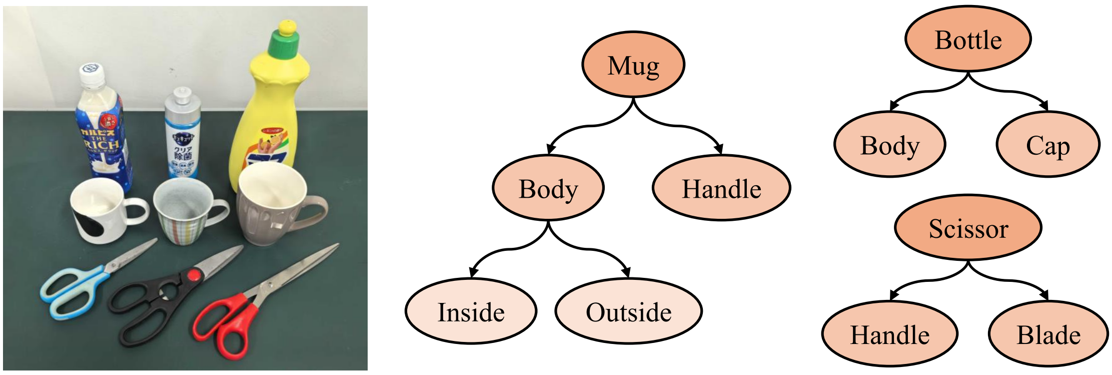


Figure 6.9: Objects used in TOG experiments with predefined ontological knowledge.

a database, as shown in Fig. 6.9. Meanwhile, we incorporate a set of model templates extracted from open-source libraries such as GrabCAD into the database. Each template is pre-segmented according to the predefined ontology, and is accompanied by downsampled point clouds and preplanned grasping knowledge for each functional part.

Evaluation of Part Recognition Accuracy

We begin by evaluating our geometry-based part recognition method through a comparison with a state-of-the-art VLM-based approach, VLPart [91]. We select three salient functional parts: *mug handle*, *bottle cap*, and *scissor handle*, as target regions to facilitate a clear assessment of recognition performance. Each object is placed in 5 randomized poses with their functional parts visible to the camera. For the VLPart trials, we capture RGB images and apply a segmentation model based on Cascaded Swin Transformers, pre-trained on a combination of datasets including LVIS, PACO, and others. A trial is considered successful if the functional part is accurately recognized, with both correct labels and mask regions. For our method, we capture object point clouds and apply the proposed three-step strategy based on template-assisted geometric analysis. A recognition is considered accurate if the best-matching point cluster is predominantly located on the functional part, defined as more than two-thirds of the points falling within the correct region. The final part recognition accuracy (PRA) is calculated as:

$$\text{PRA} = \frac{\text{Number of correctly recognized parts}}{\text{Number of total recognition attempts}}$$

Object Class	Mug	Bottle	Scissor	Average PRA
Functional part	Handle	Cap	Handle	
VLPart	10/15	11/15	5/15	26/45 (57.8%)
Our method	12/15	14/15	15/15	**41/45 (91.1%)**

Table 6.1: Evaluation of PRA via a Baseline Study Against a VLM-Based Approach

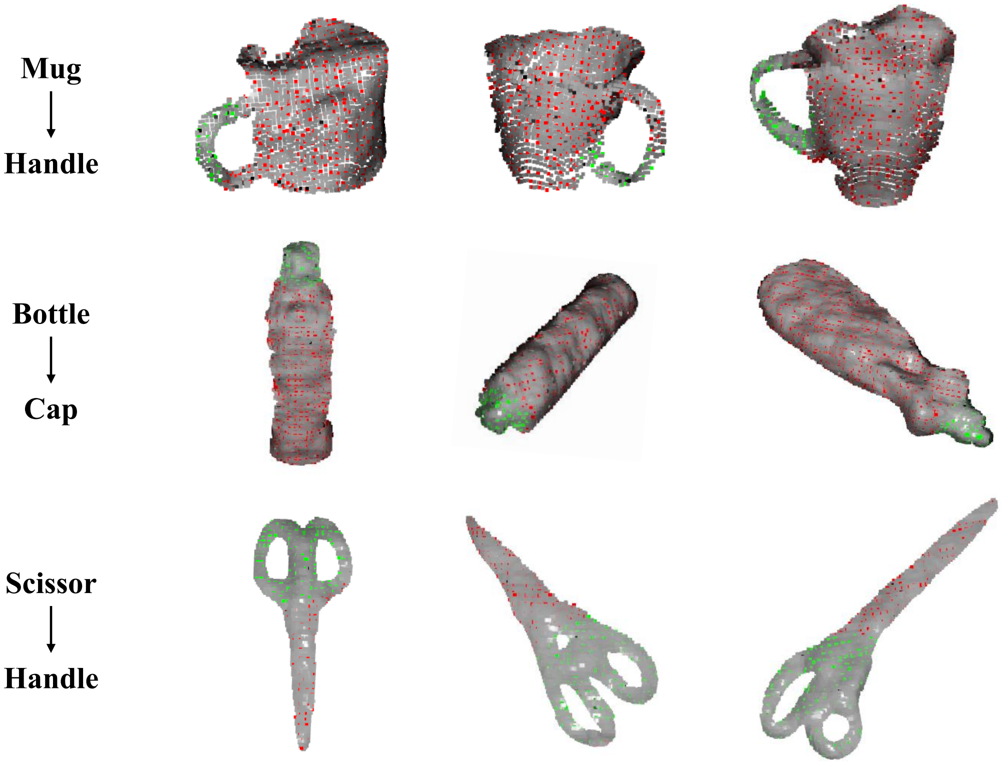


Figure 6.10: Recognition of functional object parts using our geometric analysis method.

The experimental results are presented in Table 6.1. Our method significantly outperforms VLPart in recognizing various functional parts across different object categories. Notably, in the case of the *scissor handle*, VLPart exhibits frequent failures with the red scissor featuring an irregular handle, whereas our method maintains consistent performance. This performance gap is primarily attributed to the viewpoint sensitivity of model-based approaches, as discussed in Section 6.2.3. In contrast, our method mitigates this issue by leveraging 3D geometric features, which remain relatively invariant under changes in viewpoint. Fig. 6.10 illustrates the robustness of our method in recognizing various functional object parts from different perspectives.

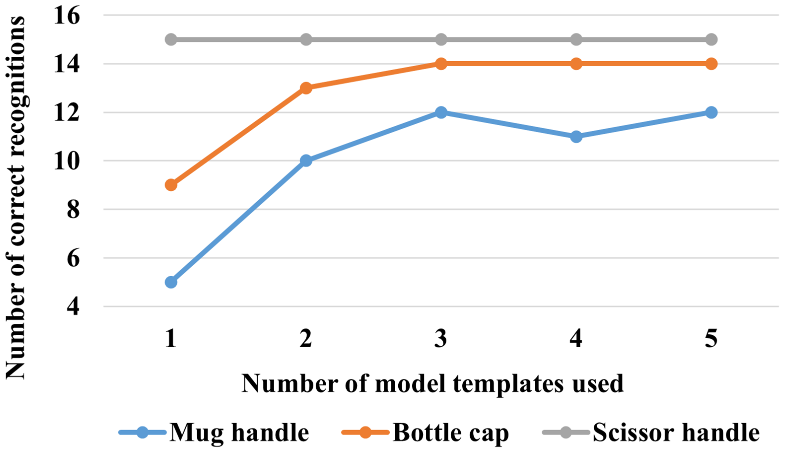


Figure 6.11: Recognition of functional object parts using our geometric analysis method.

Additionally, we investigate the impact of model template quantity on part recognition performance. For each object class, we prepare 5 model templates that vary in shape and size. During evaluation, we progressively increase the number of templates used for similarity matching from a single template up to all five. As in the previous experiment, we conduct 5 trials per object under each condition and record the number of correct recognitions. As shown in Fig. 6.11, both the *mug cap* and *bottle cap* reach optimal performance when the number of templates increases to 3, with no further improvement observed beyond that point. This suggests that using multiple templates helps average out matching errors and enhance recognition accuracy; however, excessive template inclusion yields no additional benefit. On the other hand, object parts with distinctive geometric features, such as the *scissor handle*, demonstrate high recognition accuracy even with a minimal number of templates. Based on these findings, we standardize the use of 3 model templates per object class in our TOG framework.

Evaluation of Grasp Selection Accuracy

Next, we evaluate our grasp selection method, which leverages LLM-guided ontological reasoning, against GraspGPT [56], a state-of-the-art learning-based approach that employs neural network inference. This experiment focuses specifically on selecting appropriate grasping positions based on human instructions without considering grasp stability. Due to GraspGPT's inability to generate grasps in real time, we use object point clouds obtained from the recognition experiments and apply our similarity-based planning method to pre-generate a set of feasible grasp candidates. For both GraspGPT and our method, the inputs consist of object point clouds, pre-generated grasp candidates, and a natural language instruction. The output is a single optimal grasp selected either through neural inference (GraspGPT) or ontological reasoning (our method). In our method, the selected grasp is the first candidate located within the region of the identified functional part. To evaluate performance, we design the following task instructions for the experiments, each targeting a specific functional part of the selected objects:

1) *Pour the water out of the mug.* (Mug → Handle)

2) *Hold the coffee-filled mug steady.* (Mug → Body → Outside)

3) *Shake the bottle before I drink it.* (Bottle → Body)

4) *Open the bottle for me.* (Bottle → Cap)

5) *Cut the paper with the scissors.* (Scissor → Handle)

6) *Hand the scissors to me.* (Scissor → Blade)

Instructions 1, 3, and 5 correspond to robot manipulation tasks, while Instructions 2, 4, and 6 involve human-robot interaction. Each instruction is tested on 5 object point clouds captured from different viewpoints during the recognition experiments. A selection is considered accurate if the chosen grasp lies within the region of the target functional part. The final grasp selection accuracy (GSA) is calculated as:

$$\text{GSA} = \frac{\text{Number of correctly selected grasps}}{\text{Number of total selection attempts}}$$

The experimental results are presented in Table 6.2. Our method significantly outperforms GraspGPT in selecting appropriate grasps based on intuitive human instructions. Notably, we observe substantial performance differences for Instructions 4

Object Class	Mug		Bottle		Scissor		Average GSA
Instruction No.	1	2	3	4	5	6	
GraspGPT	10/15	9/15	14/15	3/15	12/15	5/15	53/90 (58.9%)
Our method	11/15	14/15	15/15	11/15	15/15	13/15	**79/90 (87.8%)**

Table 6.2: Evaluation of GPA via a Baseline Study Against a Learning-Based Approach

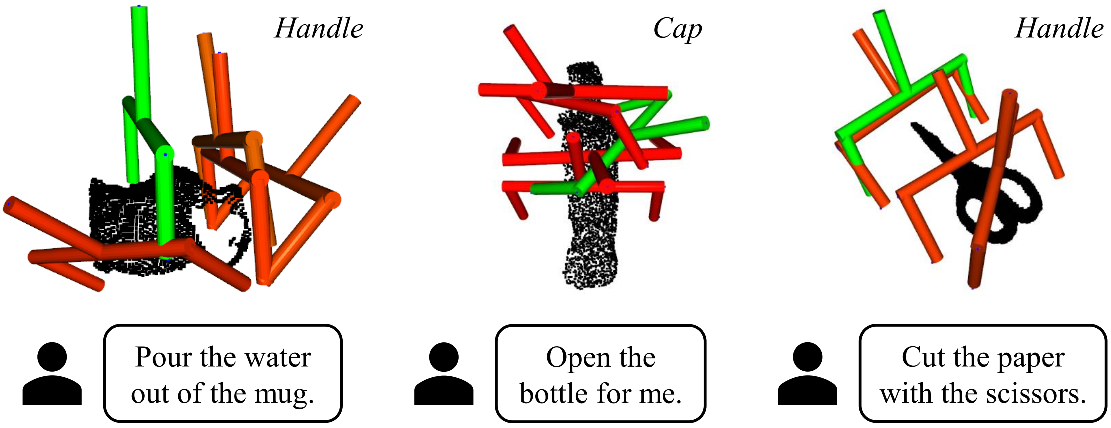

Handle
Cap
Handle
Pour the water out of the mug.
Open the bottle for me.
Cut the paper with the scissors.

Figure 6.12: Failures of GraspGPT in language-guided grasp selection. Selected grasps are highlighted in green, while the ground truth labels are shown in top-right corner.

and 6. We hypothesize that GraspGPT struggles in these cases because the *bottle* category is absent from its training dataset, and the *cap*, being a small and distinctive part of the *bottle*, is likely overlooked during grasp selection. Additionally, the *scissor* in Instruction 6 requires careful handling in a handover task, whereas GraspGPT does not incorporate constraints such as *the robot should handle dangerous parts instead of the human*. Example failure cases are shown in Fig. 6.12. In contrast, our method fully leverages the interpretability of LLMs through prompt optimization and ontological reasoning, enabling highly reliable grasp position determination. The sequential strategy from language interpretation to part recognition, then to grasp selection, also demonstrates better stability compared to methods that encode all features into the same embedding space.

Evaluation of Grasp Success Rate

Finally, we evaluate the reliability of part-aware grasp generation by comparing our method with GraspNet [22], a grasp detection benchmark trained on large-scale datasets. For GraspNet, we capture RGB-D images of the scene and use its pre-trained baseline model to generate grasp candidates along with associated quality scores. Since GraspNet is not capable of interpreting human instructions or recognizing functional object parts, this experiment focuses solely on grasp generation performance, assuming the target object parts are known. Given a specific object part, we apply our template-assisted method to identify the best-matching point cluster o^*_{part} within the observed point cloud. For both GraspNet and our method, we select only grasp poses located within the region of o^*_{part} for execution. In terms of grasp selection strategy, GraspNet ranks candidates by their predicted quality scores, while our method selects the first candidate that satisfies the stability-aware adjustment process. A grasp is considered successful if the object is grasped at the designated functional part and lifted without being dropped. Failure cases include the absence of an executable grasp, an incorrect grasp location, or the object dropping during lifting. Each functional part of each object is subjected to 5 grasp attempts, with the object placed in different orientations. In cases of part recognition failure, the trial is discarded and repeated with a new object pose. The final grasp success rate (GSR) is calculated as:

$$\text{GSR} = \frac{\text{Number of successful grasps}}{\text{Number of total grasping attempts}}$$

The experimental results are presented in Table 6.3. Our method significantly outperforms GraspNet in grasping specific parts across various objects. We observe that GraspNet struggles to generate grasp candidates on small object parts, such as the *bottle cap*, due to its tendency to favor larger surface areas that are more likely to ensure grasp robustness, as shown in Fig. 6.13. For thin-shaped objects like the *scissor*, GraspNet fails to generate feasible grasps on either part in nearly all trials, highlighting a key limitation of conventional grasp detection approaches when applied to TOG. In contrast, our method consistently achieves high GSR across diverse object

Object Class	Mug		Bottle		Scissor		Average PSR
Target part	Handle	Body (Outside)	Cap	Body	Handle	Blade	
GraspNet	3/15	10/15	3/15	12/15	0/15	2/15	30/90 (33.3%)
Our method	12/15	15/15	12/15	14/15	13/15	10/15	**76/90 (84.4%)**

Table 6.3: Evaluation of GSR via a Baseline Study Against a Grasp Detection Approach

parts by leveraging similarity-based grasp planning combined with a localized positional adjustment process to optimize grasp performance.

6.3.3 Ablation Study

In the second TOG experiment, we further verify our local-to-global point cloud registration and stability-aware grasp adjustment through two baseline comparisons: *direct registration* and *no grasp adjustment*. In the *direct registration* baseline, we employ a conventional point cloud registration pipeline using RANSAC followed by ICP to align the partial object point cloud with the complete template. In the *no grasp adjustment* baseline, we execute the first generated grasp that is both IK-solvable and collision-free without considering the position of its grasp center. To clearly illustrate differences in grasp quality, we reuse the mugs from previous experiments, fill them with water, and designate their handles as the grasping targets. Each mug undergoes five grasp planning attempts under each baseline condition. Two types of failure are considered: 1) *unsuccessful planning*, defined as either the absence of a generated grasp or a grasp targeting an incorrect object part, and 2) *unstable grasping*, identified by water spilling from the mug during execution. For performance evaluation, we define the following two metrics:

$$\text{PR (Planning Rate)} = \frac{\text{Number of successfully planned grasps}}{\text{Number of total grasp planning attempts}}$$

$$\text{SR (Stabilization Rate)} = \frac{\text{Number of stable grasps}}{\text{Number of executed grasps}}$$

In the case of *unsuccessful planning*, the trial is terminated without executing any grasp and excluded from the calculation of SR. The experimental results are presented

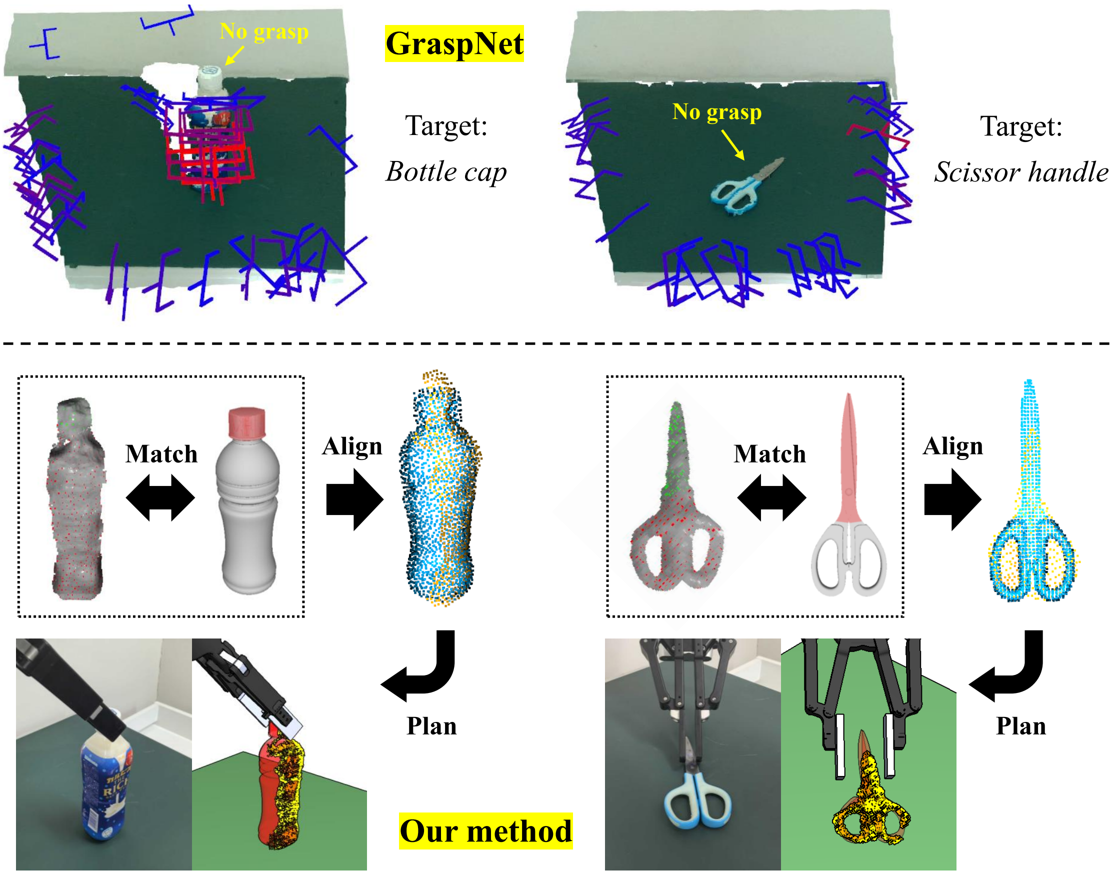


Figure 6.13: Comparison of task-oriented grasp planning between GraspNet and our proposed method.

in Table 6.4. Our full method significantly outperforms both baseline approaches across both evaluation metrics. The notable performance gap between the *direct registration* baseline and our full method highlights the effectiveness of the proposed local-to-global point cloud registration in achieving precise matching between observed objects and model templates, which is an essential prerequisite for the reliable transfer of grasping knowledge and robust grasp generation. Additionally, the substantial improvement in SR observed between the *no grasp adjustment* baseline and our full method underscores the importance of the stability-aware grasp adjustment in enhancing the quality of the executed grasps. A visual comparison showing the superiority of our full method is provided in Fig. 6.14.

Indicator	PR ↑	SR ↑
Direct registration	33% (5/15)	60% (3/5)
No grasp adjustment	80% (12/15)	50% (6/12)
Full method	**87% (13/15)**	**77% (10/13)**

Table 6.4: Ablation Study on Proposed Methods Using Water-Filled Mugs

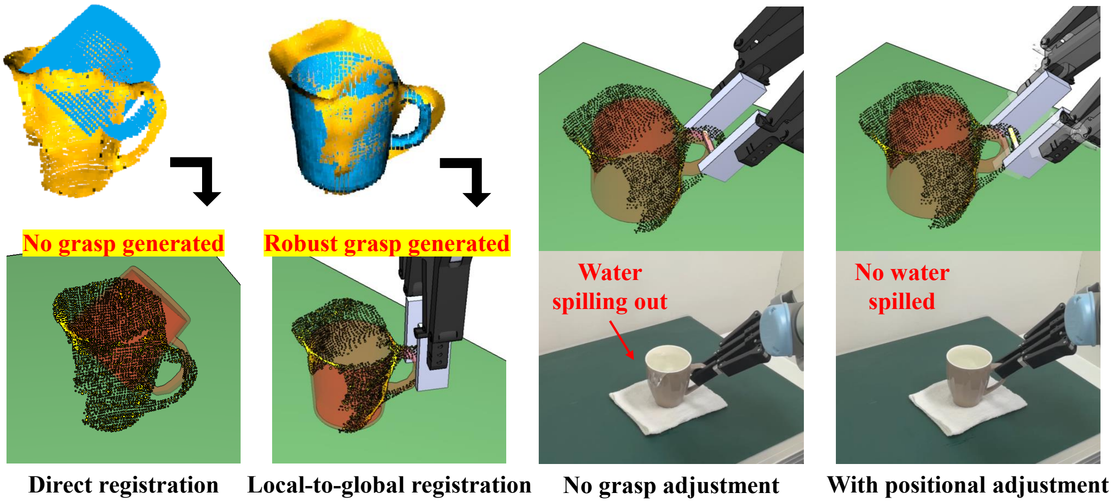


Figure 6.14: Performance discrepancy between the two baselines and our full method.

6.3.4 Generalization to Novel-Category Objects

Finally, in addition to delivering high performance on objects included in the ontology, our method demonstrates strong generalizability to novel-category objects, extending beyond the existing knowledge base. This is accomplished by leveraging the scalable inferential capabilities of LLMs, which can be activated by appending the following instruction to the prompt:

If the target object is not listed in the ontology, find its closest object in the ontology and use its part information.

The answer template is also modified accordingly, as described in Appendix B. Through this prompt refinement, a novel target object can be mapped to a similar known instance within the ontology, allowing the reuse of both ontological knowledge and corresponding model templates for subsequent recognition, matching, and plan-

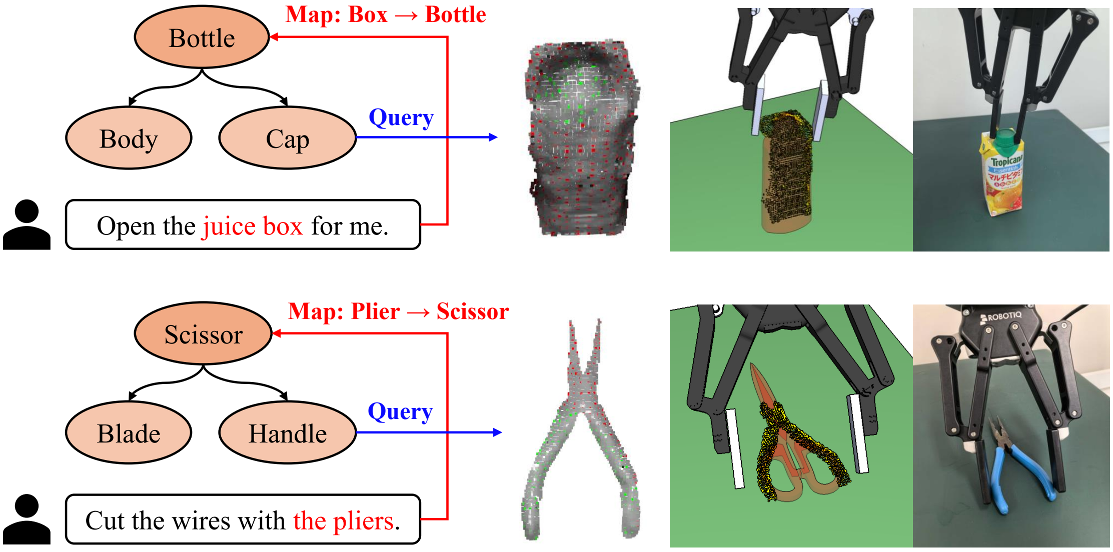


Figure 6.15: Generalizing TOG to Novel Instances via Ontological Knowledge Extension.

ning. Fig. 6.15 illustrates two representative examples involving a *juice box* and a pair of *pliers*. Despite lacking prior knowledge of these objects, the system successfully maps them to the *bottle* and *scissor* categories in the ontology, respectively. Their subclasses and associated templates are then employed to guide functional part recognition and similarity-based grasp planning. From this perspective, our approach exhibits a tree-like growth pattern: each newly introduced instance contributes to the exponential expansion of existing knowledge. Compared to conventional model-based methods that encode diverse object features into a single network, our framework offers superior reliability and robustness in handling novel objects through structural and scalable ontological reasoning.

6.4 Conclusions

In this paper, we present a novel strategy for task-oriented grasping of unseen objects. Our approach introduces an object-part-task ontology consisting of online and offline components, associated through guidance from LLMs. By leveraging optimized user prompts, the LLM accurately interprets intuitive human instructions and infers corresponding functional object parts based on ontological knowledge. Additionally, we propose a novel part recognition method that utilizes pre-existing model templates to identify target object parts as point clusters. Based on this representation, we employ a local-to-global point cloud registration framework, followed by a stability-aware grasp adjustment process, to transfer grasping knowledge from the best-matching model template to the unseen target object, enabling robust task-oriented grasp generation. We validate our method through extensive real-world experiments on a variety of unseen objects, demonstrating significant improvements over state-of-the-art approaches. Furthermore, we show the method's strong generalization to novel object categories by leveraging the scalability of LLM-guided ontological reasoning.

Despite its strengths, the proposed method has several limitations: 1) it cannot handle completely novel objects with no semantically or geometrically similar references in the ontology; 2) its matching accuracy degrades when the viewpoint provides insufficient features of the target part; and 3) it does not currently account for constraints imposed by subsequent manipulation tasks beyond grasp execution. Addressing these limitations will be the focus of our future work.

Appendix A: LLM Prompt Optimization

To optimize LLM prompts for accurate instruction interpretation, we adopt the framework proposed in [90], which consists of three components: 1) *Executor*, the LLM that processes the given prompt and generates a corresponding response; 2) *Evaluator*, a human supervisor who assesses the LLM's output and determines the next step—either terminating the optimization if the result is satisfactory, or providing feedback for improvement if not; and 3) *Improver*, where the LLM is reused to refine the prompt based on human feedback. The improved prompt is then fed back into the *Evaluator*, initiating a new iteration of the process (see Fig. 6.16). The key advantage of this framework is that humans only need to act as advisors, offering suggestions rather than revising the prompts themselves. The LLM (essentially a more efficient "engineer"), through repeated iterations and human supervision, ultimately generates an optimal prompt. Below, we illustrate an example of prompt optimization for instructing a TOG task:

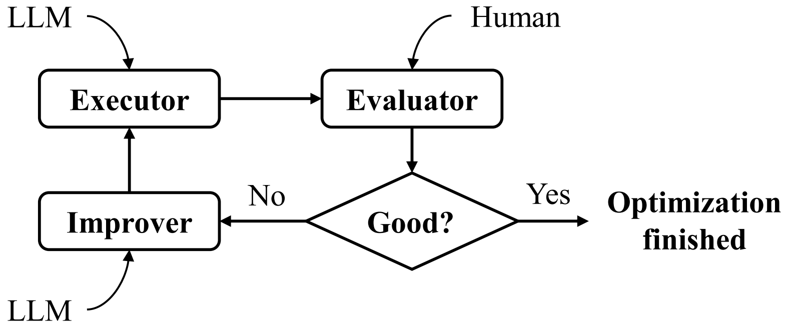


Figure 6.16: Framework for reusing LLMs in iterative prompt optimization.

Create Agent 1

User: *Given the following ontology:*

Mug → Handle, Mug → Body → Inside, Mug → Body → Outside

Scissor → Handle, Scissor → Blade

Considering these two commands given to a robot:

1. "Hand me the mug with hot water in it."

2. "Pick up the scissor and pass it to me.

Question: Which part(s) of each object should the robot grasp? Please only answer the part names.

Agent: *1. Mug → Handle 2. Scissor → Handle*

(Ground truth) 1. Mug → Body → Outside 2. Scissor → Blade

User: *Both the answers are wrong because in a human-robot interaction task, we want the robot to handle dangerous parts instead of the human. Meanwhile, the robot and human should take different parts to ensure sufficient operating space. How can we improve the prompt to obtain correct answers?*

Agent: *You can improve the question by explicitly stating the criteria for grasping ... (Unimportant statements)*

Revised Prompt: "Given the following ontology:

Mug → Handle, Mug → Body → Inside, Mug → Body → Outside

Scissor → Handle, Scissor → Blade
Consider these two commands given to a robot:
1. Hand me the mug with hot water in it
2. Pick up the scissor and pass it to me
Which part(s) of the objects should the robot grasp, ensuring that (a) the robot handles dangerous parts instead of the human, and (b) the robot and human grasp different parts for sufficient operating space?"
This revision ensures that the intended grasping strategy is clear, leading to correct answers ... (Unimportant statements)

Create Agent 2

User: *(The revised prompt from Agent 1)*

Agent: *1. Mug → Handle 2. Scissor → Blade*
(Ground truth) 1. Mug → Body → Outside 2. Scissor → Blade

User: *The first answer is still wrong because in a human-robot interaction task, we want the robot to handle dangerous parts instead of the human. Meanwhile, the robot and human should take different parts to ensure sufficient operating space. How can we improve the prompt to obtain correct answers?*

Agent: *To ensure correct answers, we should explicitly state the constraints more clearly ... (Unimportant statements)*
Revised Prompt: "Given the following ontology:
Mug → Handle, Mug → Body → Inside, Mug → Body → Outside
Scissor → Handle, Scissor → Blade
A robot is given the following commands:
1. Hand me the mug with hot water in it
2. Pick up the scissor and pass it to me

Constraints:

1. The robot should grasp parts that are either difficult to manipulate or potentially dangerous, while the human should grasp the safer, easier-to-handle parts.

2. The robot and human should each grasp a different part of the object to ensure enough operating space.

Question: Which part(s) of each object should the robot grasp? Answer with only the part names."

This version explicitly defines the constraints, making it clear that ... (Unimportant statements)

Create Agent 3

User: *(The revised prompt from Agent 2)*

Agent: *1. Mug → Body → Outside 2. Scissor → Blade*

(Ground truth) 1. Mug → Body → Outside 2. Scissor → Blade

Prompt optimization completed

Appendix B: Modified Answer Template

To fully leverage the scalability of LLMs for novel-category objects, we design the following step-by-step reasoning process as the answer template:

The given command is "Empty the bowl into the sink."

Step 1: Identify the Type of Task

The robot is instructed to pick up the bowl and pour its contents into the sink.

There is no human involved in handling the bowl.

Therefore, this is a manipulation task.

Step 2: Find the Closest Object in the Ontology

The bowl is not in the ontology. Let's look for the most structurally similar object:

The bowl, like the mug, is a container with a concave inside and round outer body.

The key difference: a mug typically has a handle, whereas a bowl usually does not.

Despite this, the bowl's structure (body with inside and outside) closely aligns with the mug's body parts.

So, we map: Bowl ≈ Mug (without handle)

Relevant parts:

Body → inside, Body → outside

Step 3: Apply Task Constraints

The robot must grasp the bowl in a way that allows it to tilt and pour the contents into the sink.

This requires a secure and controlled grip—likely from the outside of the body to allow for rotation.

Analyzing the Object Parts

Body (inside): Contains the contents—not suitable for gripping.

Body (outside): Provides surface area for a stable grip and control when tilting.

Best Choice for the Robot

The robot should grasp the body (outside) of the bowl to securely hold, tilt, and empty its contents into the sink.

Conclusion*: The Robot Should Grasp the Body (Outside)*

This enables the robot to effectively manipulate the bowl for pouring while maintaining control and avoiding contact with the contents.

Chapter 7

Discussion

7.1 Contributions

In this dissertation, we present **Similarity Matching**, a novel strategy for grasping unknown objects by leveraging existing knowledge of similar known objects. In Chapter 3, we propose a two-factor scoring function that integrates semantic and geometric similarity between the target object and existing database models. By computing similarity scores from Category-Association Matching (CAM) and Point Cloud Registration (PCR), the database model most likely to resemble the target object is identified as the reference for grasp planning. This initial matching approach, although having limitations such as requiring multi-view observations, highlights the potential of similarity-based strategies in in novel object grasping.

As an improvement, in Chapter 4, we propose a multi-level similarity matching approach that separately evaluates object similarity from semantic, geometric, and dimensional perspectives. The compensatory nature of multi-level matching significantly enhances the accuracy and robustness of similar model identification. Notably, we introduce a novel geometric descriptor, the C-FPFH descriptor, which is capable of reliably assessing similarity between a partially observed object and a complete database model. Additionally, the final grasp planning result is optimized through an improved point cloud registration method based on plane detection and a stability-aware grasp fine-tuning process based on local contact geometry. Extensive real-world experiments show that our similarity-based approach achieves superior performance in grasping a wide range of novel objects, both in isolated and cluttered scenes.

As a further extension, we apply our similarity-based method to dynamic object manipulation and task-oriented object grasping, as presented in Chapter 5 and Chapter 6, respectively. For dynamic manipulation of moving objects in dense clutter, we introduce a global-to-local detection and static-to-dynamic planning framework that addresses a challenging grasping task by transferring knowledge across multiple object states. For task-oriented grasping of previously unseen objects, we incorporate an object-part-task ontology guided by LLMs that associates human instructions with functional part selection, and develop a template-based part recognition method for affordance-aware grasp generation. These application cases demonstrate the strong generalizability of our similarity-based approach across diverse object configurations and task conditions.

7.2 Open Challenges and Future Work

While our current approach focuses on robotic grasping, real-world tasks often demand a broader range of actions. To this end, future work can explore the following directions:

1. Extending similarity-based methods to high-level manipulation tasks beyond grasping. More complex scenarios, such as assembly tasks, are promising applications where similarity matching can be effectively applied.
2. Integrating force feedback for soft object manipulation. Beyond rigid objects, future work should consider the grasping of soft and deformable items by incorporating tactile and force feedback.
3. Incorporating more versatile actions such as pushing. In non-prehensile contexts, introducing additional skills like pushing can help overcome more complex manipulation challenges.

References

[1] Y. Cong, R. Chen, B. Ma, H. Liu, D. Hou, and C. Yang, “A comprehensive study of 3-d vision-based robot manipulation,” *IEEE Transactions on Cybernetics*, vol. 53, no. 3, pp. 1682–1698, 2023.

[2] M. A. Rashed, R. N. Farhan, and W. M. Jasim, “Robotic grasping based on deep learning: A survey,” in *2023 Second International Conference on Advanced Computer Applications (ACA)*, 2023, pp. 1–7.

[3] Q. Jiao, W. Hu, G. Hao, J. Shao, and Q. Jiao, “A digital twin of intelligent robotic grasping based on single-loop-optimized differentiable architecture search and sim-real collaborative learning,” *Journal of Intelligent Manufacturing*, 2024.

[4] S. Deng, M. Yan, S. Wei, H. Ma, Y. Yang, J. Chen, Z. Zhang, T. Yang, X. Zhang, H. Cui, Z. Zhang, and H. Wang, “Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data,” *arXiv preprint arXiv:2505.03233*, 2025.

[5] H. Zhang, J. Tang, S. Sun, and X. Lan, “Robotic grasping from classical to modern: A survey,” *arXiv preprint arXiv:2202.03631*, 2022.

[6] K. Shimoga, “Robot grasp synthesis algorithms: A survey,” *The International Journal of Robotics Research*, vol. 15, no. 3, pp. 230–266, 1996.

[7] A. Bicchi and V. Kumar, “Robotic grasping and contact: a review,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics*

and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 1, 2000, pp. 348–353.

[8] J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis—a survey,” *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[9] Z. Xie, X. Liang, and C. Roberto, “Learning-based robotic grasping: A review,” *Frontiers in Robotics and AI*, vol. 10, 2023.

[10] T. Li, Y. Yan, C. Yu, J. An, Y. Wang, and G. Chen, “A comprehensive review of robot intelligent grasping based on tactile perception,” *Robotics and Computer-Integrated Manufacturing*, vol. 90, p. 102792, 2024.

[11] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, “Learning ambidextrous robot grasping policies,” *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[12] D. Morrison, P. Corke, and J. Leitner, “Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach,” in *Robotics: Science and Systems (RSS)*, 2018.

[13] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, “Grasp pose detection in point clouds,” *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[14] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, “Learning 6-dof grasping interaction via deep geometry-aware 3d representations,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3766–3773.

[15] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, “Pointnetgpd: Detecting grasp configurations from point sets,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3629–3635.

[16] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2017, pp. 77–85.

[17] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, “Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation,” in *The Conference on Robot Learning (CoRL)*, 2018.

[18] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, “Volumetric grasping network: Real-time 6 dof grasp detection in clutter,” in *The Conference on Robot Learning (CoRL)*, 2020.

[19] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, “Synergies between affordance and geometry: 6-dof grasp detection via implicit representations,” in *Robotics: Science and Systems (RSS)*, 2021.

[20] S. S. Mohammadi, N. F. Duarte, D. Dimou, Y. Wang, M. Taiana, P. Morerio, A. Dehban, P. Moreno, A. Bernardino, A. Del Bue, and J. Santos-Victor, “3ds-grasp: 3d shape-completion for robotic grasp,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3815–3822.

[21] B. Sen, A. Agarwal, G. Singh, B. B., S. Sridhar, and M. Krishna, “Scarp: 3d shape completion in arbitrary poses for improved grasping,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3838–3845.

[22] H.-S. Fang, C. Wang, M. Gou, and C. Lu, “Graspnet-1billion: A large-scale benchmark for general object grasping,” in *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020, pp. 11 444–11 453.

[23] A. D. Vuong, M. N. Vu, H. Le, B. Huang, H. T. T. Binh, T. Vo, A. Kugi, and A. Nguyen, “Grasp-anything: Large-scale grasp dataset from foundation models,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 030–14 037.

[24] S. Chen, W. Tang, P. Xie, W. Yang, and G. Wang, “Efficient heatmap-guided 6-dof grasp detection in cluttered scenes,” *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4895–4902, 2023.

[25] M. Adjigble, N. Marturi, V. Ortenzi, V. Rajasekaran, P. Corke, and R. Stolkin, “Model-free and learning-free grasping by local contact moment matching,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2933–2940.

[26] U. Clarenz, M. Rumpf, and A. Telea, “Robust feature detection and local classification for surfaces based on moment analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 5, pp. 516–524, 2004.

[27] Y. Wu, W. Liu, Z. Liu, and G. S. Chirikjian, “Learning-free grasping of unknown objects using hidden superquadrics,” in *Robotics: Science and Systems (RSS)*, 2023.

[28] W. Liu, Y. Wu, S. Ruan, and G. S. Chirikjian, “Robust and accurate superquadric recovery: a probabilistic approach,” in *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022, pp. 2666–2675.

[29] X. Wang, S. Nisar, and F. Matsuno, “Robust grasp detection with incomplete point cloud and complex background,” *Advanced Robotics*, vol. 35, no. 10, pp. 619–634, 2021.

[30] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, “The columbia grasp database,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 1710–1716.

[31] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal, “Template-based learning of grasp selection,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2379–2384.

[32] A. Mitrevski, P. G. Plöger, and G. Lakemeyer, “Ontology-assisted generalisation of robot action execution knowledge,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6763–6770.

[33] H. Chen, T. Kiyokawa, W. Wan, and K. Harada, “Category-association based similarity matching for novel object pick-and-place task,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2961–2968, 2022.

[34] D. Morrison, P. Corke, and J. Leitner, “Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach,” in *Robotics: Science and Systems (RSS)*, 2018.

[35] N. Marturi, M. Kopicki, A. Rastegarpanah, V. Rajasekaran, M. Adjigble, R. Stolkin, A. Leonardis, and Y. Bekiroglu, “Dynamic grasp and trajectory planning for moving objects,” *Autonomous Robots*, vol. 43, pp. 1241–1256, 2018.

[36] I. Akinola, J. Xu, S. Song, and P. K. Allen, “Dynamic grasping with reachability and motion awareness,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9422–9429.

[37] C.-C. Wong, M.-Y. Chien, R.-J. Chen, H. Aoyama, and K.-Y. Wong, “Moving object prediction and grasping system of robot manipulator,” *IEEE Access*, vol. 10, pp. 20 159–20 172, 2022.

[38] P. Chen and W. Lu, “Deep reinforcement learning based moving object grasping,” *Information Sciences*, vol. 565, pp. 62–76, 2021.

[39] B. Huang, J. Yu, and S. Jain, “EARL: Eye-on-hand reinforcement learner for dynamic grasping with active pose estimation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[40] H. Dang and P. K. Allen, “Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 1311–1317.

[41] D. Song, K. Huebner, V. Kyrki, and D. Kragic, “Learning task constraints for robot grasping using graphical models,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 1579–1585.

[42] R. Detry, J. Papon, and L. Matthies, “Task-oriented grasping with semantic and geometric scene understanding,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3266–3273.

[43] F.-J. Chu, R. Xu, and P. A. Vela, “Learning affordance segmentation for real-world robotic manipulation via synthetic images,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1140–1147, 2019.

[44] A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta, “Same object, different grasps: Data and semantic knowledge for task-oriented grasping,” in *The Conference on Robot Learning (CoRL)*, 2020.

[45] C. Tang, D. Huang, L. Meng, W. Liu, and H. Zhang, “Task-oriented grasp prediction with visual-language inputs,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 4881–4888.

[46] Y.-L. Wei, J.-J. Jiang, C. Xing, X.-T. Tan, X.-M. Wu, H. Li, M. Cutkosky, and W.-S. Zheng, “Grasp as you say: Language-guided dexterous grasp generation,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, 2024, pp. 46 881–46 907.

[47] J. Zhang, W. Xu, Z. Yu, P. Xie, T. Tang, and C. Lu, “Dextog: Learning task-oriented dexterous grasp with language condition,” *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 995–1002, 2025.

[48] W. Liu, A. Daruna, and S. Chernova, “Cage: Context-aware grasping engine,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2550–2556.

[49] V. Holomjova, A. J. Starkey, B. Yun, and P. Meißner, “One-shot learning for task-oriented grasping,” *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8232–8238, 2023.

[50] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu, “Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation,” in *The European Conference on Computer Vision (ECCV)*, 2025, pp. 222–239.

[51] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, “Language embedded radiance fields for zero-shot task-oriented grasping,” in *The Conference on Robot Learning (CoRL)*, 2023.

[52] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *The International Conference on Machine Learning (ICML)*, 2021.

[53] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9630–9640.

[54] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang, “Graspsplats: Efficient manipulation with 3d feature splatting,” in *The Conference on Robot Learning (CoRL)*, 2024.

[55] C. Tang, D. Huang, W. Dong, R. Xu, and H. Zhang, “Foundationgrasp: Generalizable task-oriented grasping with foundation models,” *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 12 418–12 435, 2025.

[56] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, “Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping,” *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7551–7558, 2023.

[57] A. Z. Ren, B. Govil, T.-Y. Yang, K. Narasimhan, and A. Majumdar, “Leveraging language for accelerated learning of tool manipulation,” in *The Conference on Robot Learning (CoRL)*, 2022.

[58] G. Du, K. Wang, S. Lian, and K. Zhao, “Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review,” *Artificial Intelligence Review*, vol. 54, 2021.

[59] M. Gualtieri, A. T. Pas, and R. W. Platt, “Pick and place without geometric object models,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7433–7440.

[60] L. Berscheid, P. Meissner, and T. Kröger, "Self-supervised learning for precise pick-and-place without object model," *IEEE Robotics and Automation Letters*, vol. 5, pp. 4828–4835, 2020.

[61] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *The International Symposium on Robotics Research (ISRR)*, 2019.

[62] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[63] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[64] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 995–1001.

[65] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[66] R. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 3212–3217.

[67] K.-L. Low, "Linear least-squares optimization for point-to-plane icp surface registration," in *Technical Report*, 2004.

[68] W. Wan, K. Harada, and F. Kanehiro, "Planning grasps with suction cups and parallel grippers using superimposed segmentation of object meshes," *IEEE Transactions on Robotics*, vol. 37, pp. 166–184, 2021.

[69] A. Yershova, L. Jaillet, T. Siméon, and S. LaValle, "Dynamic-domain rrts: Efficient exploration by controlling the sampling domain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 3856–3861.

[70] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, “Panoptic segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9404–9413.

[71] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *The International Conference on Learning Representations (ICLR)*, 2013.

[72] R. Newbury, K. He, A. Cosgun, and T. Drummond, “Learning to place objects onto flat surfaces in upright orientations,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 4377–4384, 2021.

[73] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, D. Fox, and A. Cosgun, “Deep learning approaches to grasp synthesis: A review,” *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3994–4015, 2023.

[74] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morena, P. Qu Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez, “Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching,” *The International Journal of Robotics Research*, vol. 41, pp. 690 – 705, 2019.

[75] A. Mousavian, C. Eppner, and D. Fox, “6-dof graspnet: Variational grasp generation for object manipulation,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[76] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, “Detecting twenty-thousand classes using image-level supervision,” in *The European Conference on Computer Vision (ECCV)*, 2022, pp. 350–368.

[77] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment any-

thing," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.

[78] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, vol. 14, 2014, pp. 1532–1543.

[79] OpenAI, "Gpt-4: Openai's generative pre-trained transformer, version 4," 2024. [Online]. Available: https://openai.com/research/gpt-4

[80] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[81] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 3212–3217.

[82] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "Cad-model recognition and 6dof pose estimation using 3d cues," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2011, pp. 585–592.

[83] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[84] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, 1991, pp. 2724–2729.

[85] A. M. C. Araújo and M. M. Oliveira, "A robust statistics approach for plane detection in unorganized point clouds," *Pattern Recognition*, vol. 100, p. 107115, 2020.

[86] A. Alliegro, M. Rudorfer, F. Frattin, A. Leonardis, and T. Tommasi, "End-to-end learning to grasp via sampling from object point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9865–9872, 2022.

[87] H. Fang, H.-S. Fang, S. Xu, and C. Lu, “Transcg: A large-scale real-world dataset for transparent object depth completion and a grasping baseline,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7383–7390, 2022.

[88] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, “3d affordancenet: A benchmark for visual object affordance understanding,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[89] Y. Li, N. Zhao, J. Xiao, C. Feng, X. Wang, and T.-s. Chua, “Laso: Language-guided affordance segmentation on 3d object,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14 251–14 260.

[90] C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen, “Large language models as optimizers,” in *The International Conference on Learning Representations (ICLR)*, 2024.

[91] P. Sun, S. Chen, C. Zhu, F. Xiao, P. Luo, S. Xie, and Z. Yan, “Going denser with open-vocabulary part segmentation,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 15 407–15 419.

Acknowledgements

The 5 years, from 2020 to 2025, from master to phd, have totally changed my life into a different career. The time goes so fast before I can keep all the happiness and sadness, successes and failures into my mind. It is a really tough journey to get this degree, both physically and mentally, something only those who have experienced can truly know. My biggest feeling is, an individual has limited energy and ability in sticking to what he wants to achieve. However, the people behind him will always be the infinite source of power, pushing him forwards until reaching that goal.

First of all, I would like to express my greatest appreciation to Prof. Harada, who gave me the chance to study in OU, to become a researcher from the very beginning when I knew nothing about research. He supports me not only in research guidance, but also in many other aspects, without which I might not be able to maintain my life here. I really feel lucky to become his student, and I will remember this favor and return it as much as possible in the future. Also, I want to appreciate Prof. Kiyokawa, who gave me much help when I was new as a master student, and Prof. Wan, who gave me many useful suggestions when I met problems in doing research. Additionally, I'd like to thank all my workmates, including those who are still in lab, and those who have graduated, for all the help you have ever given me.

Last but not least, I deeply appreciate the support from my girlfriend, Siyuan Shen, who always stands on my side, whenever it's a good day or a bad day. Without her, I will never become what I am like today. And I want to say thank you to all my friends and family, who are in China or around the world, for your invariant support on me even when you are thousands of miles away.

I don't have very big dreams. But for me, PhD is just a start. I will continue to work hard from now on, not only for myself, but also for everyone who loves me and whom I love.

Publications

Journal Papers:

1. **H. Chen**, T. Kiyokawa, Z. Hu, W. Wan, and K. Harada, “A Multi-Level Similarity Approach for Single-View Object Grasping: Matching, Planning, and Fine-Tuning,” *IEEE Transactions on Robotics*, 2025. (Accepted)
2. **H. Chen**, T. Kiyokawa, Z. Hu, W. Wan, and K. Harada, “Novel Object Grasping Using an Object Ontology,” *Journal of the Robotics Society of Japan*, vol. 41, no. 7, pp. 643-646, 2023.
3. **H. Chen**, T. Kiyokawa, W. Wan and K. Harada, "Category-Association Based Similarity Matching for Novel Object Pick-and-Place Task," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2961-2968, 2022.
4. **H. Chen**, T. Kiyokawa, W. Wan, and K. Harada, “Generalizable task-oriented object grasping through LLM-guided ontology and similarity-based planning,” *Robotics and Autonomous Systems.* (Under Review)

International Conference Papers (with peer-review):

1. **H. Chen**, T. Kiyokawa, W. Wan, and K. Harada, “Adaptive Grasping of Moving Objects in Dense Clutter via Global-to-Local Detection and Static-to-Dynamic Planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, 2025.
2. **H. Chen**, T. Kiyokawa, W. Wan and K. Harada, "Category-Association Based Similarity Matching for Novel Object Pick-and-Place Task," in *IEEE International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA, 2022. (Transferred from RA-L)

Local Conference papers (without peer-review):

1. **H. Chen**, T. Kiyokawa, W. Wan and K. Harada, "Vision-Based Dynamic Grasping of Moving Objects in Cluttered Scenes", 第25回計測自動制御学会システムインテグレーション部門講演会, 盛岡, 2024年12月.
2. **H. Chen**, T. Kiyokawa, W. Wan and K. Harada, "Single-view dexterous object grasping under high uncertainty using multi-level similarity matching", *The 42th Annual Conference of the Robotics Society of Japan*, Osaka, 2024.
3. 余昌志, 陳浩, 清川拓哉, 万偉偉, 原田研介, “局所形状の類似性評価に基づくオクルージョンのある未知物体の把持計画,” 第24回計測自動制御学会システムインテグレーション部門講演会, 新潟, 2023年12月.
4. **H. Chen**, T. Kiyokawa, W. Wan and K. Harada, "Similarity-based fast grasp planning of unknown objects on a conveyor belt", *The 41th Annual Conference of the Robotics Society of Japan*, Sendai, 2023.
5. 陳浩, 清川拓哉, 胡正涛, 万偉偉, 原田研介, “把持物体のオントロジーを用いた未知物体の把持,” 第40回日本ロボット学会学術講演会, 東京, 2022年09月.
6. **H. Chen**, T. Kiyokawa, W. Wan, and K. Harada, “Vision-Based Novel Object Grasping based on Object-Association Matching with Grasp Database,” *The 39th Annual Conference of the Robotics Society of Japan*, Online, 2021.

Awards:

- IEEE Robotics and Automation Society Japan Joint Chapter, IEEE Robotics and Automation Society Japan Joint Chapter Young Award (ICRA2022), May 2022.
- The 39th Annual Conference of the Robotics Society of Japan (RSJ2021), International Session Best Presentation Award Finalist, September 2021.