Generalizing Grasping via Shape Matching and Learning

WENZHENG ZHANG

Supervisor: Professor Fabio Ramos

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

Jan 2026

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning. I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Wenzheng Zhang

Jan 2026

Abstract

Robotic grasping has traditionally relied on analytic models grounded in contact mechanics and exact object geometry. Although principled, these methods often falter in real-world settings due to sensing inaccuracies, noise, and incomplete object models. To overcome such limitations, recent work has shifted toward data-driven approaches that leverage rich sensory inputs, such as RGB-D images and point clouds, enabling robots to learn grasp strategies from large-scale datasets. Despite their adaptability, purely learning-based methods frequently struggle to generalize to unseen objects, diverse grippers, and novel environments. This motivates the development of hybrid approaches that integrate analytic reasoning with data-driven models. However, most existing hybrids remain weakly integrated, using analytic methods only as auxiliary evaluators or post-processing steps.

This thesis advances a synergistic hybrid framework for robust and generalizable grasp synthesis under partial observation. First, we formulate grasp synthesis as rigid shape matching between object and gripper point clouds, exploiting the gripper's Signed Distance Field (SDF) for efficient collision avoidance. To accelerate optimization, we use a parallelized Stochastic Gradient Descent Iterative Closest Point (SGD-ICP) algorithm, which achieves significant speedups over conventional approaches. Building on this, we introduce a Stein Variational Gradient Descent (SVGD)–based extension that generates diverse grasp distributions, reducing sensitivity to initialization and allowing incorporation of prior knowledge and task constraints. Finally, we propose a novel hybrid framework that integrates Energy-Based Models (EBMs) with analytical ICP gradients within the SVGD pipeline, combining learned priors with geometric optimization to produce robust grasps directly from partial point clouds.

Through ablation studies, we show how dataset curation critically shapes the learned energy landscape, with targeted datasets outperforming larger but less structured ones. Extensive simulation experiments confirm that combining analytic and data-driven methods within the proposed framework improves grasp quality, generalization, and robustness, achieving more consistent outcomes than either approach in isolation.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Fabio Ramos, for his invaluable guidance, encouragement, and unwavering support throughout the course of my PhD. His insights and advice have been instrumental in shaping both my research and my development as a researcher. I am also grateful to Fahira Maken and Tin Lai for their constructive feedback and helpful discussions, which have significantly strengthened this work.

I wish to thank my colleagues and lab mates in the School of Computer Science at the University of Sydney, especially those under Prof. Ramos' supervision, for providing a stimulating and supportive research environment. Our many discussions, brainstorming sessions, and collaborations greatly enriched my research experience.

This research would not have been possible without the financial support of the Faculty of Engineering Research Scholarship, provided by the Faculty of Engineering. I gratefully acknowledge this support, as well as the research facilities and resources made available by the University.

I am indebted to my friends and peers for their constant encouragement, companionship, and for making this journey more enjoyable.

Finally, and most importantly, I would like to thank my family for their unconditional love, patience, and encouragement. To my parents, who have always believed in me, your support has been my greatest source of strength. This thesis is dedicated to you.

During the preparation of this thesis, I wrote the content and used generative AI tools (GPT-4 and GPT-5) to assist with sentence structure, grammar checking, and readability. All content generated or modified with AI assistance was carefully reviewed to correct possible errors, inaccuracies, or bias. I take full responsibility for the submitted thesis, confirm that the work is my own, and affirm that the use of generative AI complied with University guidelines and policies.

CONTENTS

Declaration ii

Abstract iii

Acknowledgements iv

List of Figures ix

List of Tables xv

List of Algorithms xvii

Nomenclature xviii

Authorship Attribution xx

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Problem Statement 3

1.3 Contributions 4

1.4 Outline 4

Chapter 2 Background 6

2.1 Robotic Manipulation 6

2.1.1 In-hand manipulation 7

2.1.2 Trajectory planning 9

2.2 Grasping 10

2.2.1 Types of Grippers and End-Effectors 11

2.2.2 Analytical Methods 13

2.2.3 Learning-Based Methods 16

2.2.4 Hybrid Methods 18

2.2.5 Grasp Quality Metrics 18

2.2.6 Object Pose Detection . . . 21
2.2.7 Collision Avoidance . . . 23
2.3 Stochastic Gradient Descent . . . 24
2.4 Stein Variational Gradient Descent . . . 26
2.5 Iterative Closest Point . . . 28
2.5.1 Rotation Representations . . . 30
2.5.2 Stochastic Gradient Descent ICP . . . 31
2.5.3 Stein ICP . . . 32
2.6 Neural Network . . . 33
2.6.1 Network Structure . . . 33
2.6.2 Encoders and Autoencoders . . . 36
2.6.3 Energy-Based Models . . . 37
2.7 Summary . . . 39

Chapter 3 Grasping as Rigid Shape Matching 40
3.1 Introduction . . . 40
3.2 Related Work . . . 41
3.2.1 Analytic Approaches . . . 41
3.2.2 Data-Driven Approaches . . . 42
3.2.3 Grasp Quality Metrics . . . 42
3.3 Methodology . . . 43
3.3.1 Inputs and Pre-processing . . . 44
3.3.2 Cost Functions . . . 44
3.3.3 Gradients of the Grasp Cost Function . . . 45
3.3.4 Collision Checking . . . 46
3.3.5 Parallel Implementation . . . 47
3.4 Experiments . . . 48
3.4.1 Simulation . . . 48
3.4.2 Real Experiment . . . 53
3.5 Summary and Discussion . . . 53

Chapter 4 Grasping with Annealed Stein ICP 56
4.1 Introduction . . . 56
4.2 Related Work . . . 56

4.3 Methodology . . . 58
4.3.1 Stein ICP . . . 58
4.3.2 Cost Functions . . . 59
4.3.3 Collision Checking . . . 60
4.3.4 Algorithm . . . 60
4.4 Experiments . . . 62
4.4.1 Ablation Study . . . 62
4.4.2 Simulation Result . . . 63
4.4.3 Real Experiment . . . 69
4.5 Summary and Discussion . . . 72

Chapter 5 Stein Energy-Based Grasp Synthesis 73
5.1 Introduction . . . 73
5.2 Related Work . . . 74
5.3 Methodology . . . 75
5.3.1 Data Generation . . . 76
5.3.2 Network Architecture . . . 76
5.3.3 Network Training . . . 78
5.3.4 SVGD Optimization . . . 79
5.4 Ablation . . . 83
5.4.1 Data Processing . . . 83
5.4.2 Hybrid Model . . . 85
5.5 Experiments . . . 89
5.5.1 Simulation . . . 89
5.5.2 Real Experiment . . . 93
5.6 Summary and Discussion . . . 95

Chapter 6 Conclusions and Future Work 96
6.1 Summary of Contributions and Related Chapters . . . 96
6.2 Future Work . . . 97
6.2.1 Task-Oriented Grasp Initialization and Evaluation Metrics . . . 97
6.2.2 Data Collection and Processing for Learning . . . 97
6.2.3 Sensor Feedback Control for Grasping . . . 98

Bibliography **99**

List of Figures

1.1 (a) Hero of Alexandria's design of singing birds (Łuniewicz and Jagiełło, 2024). (b) Al-Jazari's automaton musicians (Kennedy and Stanton, 1924). (c) Leonardo da Vinci's design for mechanical knight (Pedretti and Rosheim, 2006). 1

1.2 (a) The Unimate robot (Marsh, 2022). (b) Honda's ASIMO humanoid robot (Honda Motor Co., Ltd., 2001). (c) Boston Dynamics' Atlas robot (Boston Dynamics, 2024). 2

2.1 Illustration of in-hand manipulation (Ma and Dollar, 2011). 7

2.2 Illustration of a process to achieve a precise grasp while minimizing object displacement (Tian et al., 2024). 8

2.3 Illustration of forward kinematic and inverse kinematic of a robotic arm (The MathWorks Inc., 2023). 8

2.4 Illustration of using waypoints to avoid collision during grasping. 10

2.5 Types of traditional grippers: (a) vacuum gripper (suction cups); (b) parallel pneumatic gripper (pinching action); (c) hydraulic gripper (collet type); (d) servo-electric gripper (pinching action). Types of soft grippers: (e) granular jamming-based soft gripper; (f) gripper with soft fingers driven by positive pneumatic pressure; (g) soft gripper with cable-driven fingers (Subramaniam et al., 2020). 11

2.6 Image of a Franka Hand (Franka Emika GmbH, 2016–). 12

2.7 Illustration of the finger control of a Franka Hand (Franka Emika GmbH, 2016–). 12

2.8 Image of a Barrett Hand (Barrett Technology, 1990–). 13

2.9 Range of motion of the Barrett Hand: horizontal spread $384\ \mathrm{mm}$, vertical spread $334\ \mathrm{mm}$, depth $187\ \mathrm{mm}$ (Barrett Technology, 1990–). 13

2.10 Image of Kinova grippers (Kinova Robotics Inc., 2006–). 13

2.11 Examples of form closure in planar grasps. The gray shapes represent objects, and the brown disks represent fingertip contact points. First-order form closure, where the minimum number

of contacts fully constrains the object against any small movement. Higher-order form closure, where additional or strategically placed contacts provide increased constraint and stability. No form closure—insufficient or poorly positioned contacts allow the object to move freely. Adapted from (Prattichizzo and Trinkle, 2008). 14

2.12 Force closure grasp of a sphere (Zaidi et al., 2017). 14

2.13 Illustrations of different types of data used for grasping (Gou et al., 2021). 16

2.14 Pipeline of the 3DSGrasp grasping strategy (Mohammadi et al., 2023). 17

2.15 Illustration of a force closure grasp (left) and non force closure grasp (right) (Christopoulos, 2010). 19

2.16 Illustration of grasp failure caused by the moment generated by the object's center of mass (CoM) (Generated by GPT OpenAI et al. (2023)). 19

2.17 Illustration of Eye-to-hand (a) and Eye-in-hand (b) (Lin et al., 2022). 21

2.18 Illustration of a SDF, the blue points are inside the object. (Park et al., 2019). 23

2.19 An illustration of how the gradient descent algorithm finds the minimum of a function (Goodfellow et al., 2016). 24

2.20 Illustration of the SVGD algorithm (Zhang and Curtis, 2020). Starting from an initial set of particles approximating the target distribution (a), SVGD iteratively updates these particles by simultaneously attracting them towards regions of high probability and repelling them from each other to prevent collapse to a single mode, as shown in (b) and (c). This process enables efficient exploration of complex, multimodal distributions. 26

2.21 Illustration of the ICP matching process (Wan et al., 2019). 28

2.22 Illustration of two common metrics for ICP: point-to-point (top) and point-to-plane (bottom) (Będkowski and Masłowski, 2012). 29

2.23 Illustration of a basic neural network unit consisting of input, hidden, and output layers connected by weighted edges (Roth, 2016). 33

2.24 Illustration of a point cloud encoder–decoder architecture with a latent space representation (Mousavian et al., 2019). The encoder Q maps the input point cloud X—an unordered set of 3D points capturing object geometry—into a compact latent vector z in a continuous latent space. This latent representation captures the essential geometric features of the object while being invariant to point order, enabling efficient processing for downstream

tasks such as recognition, segmentation, and grasp synthesis. The decoder P reconstructs the point cloud from z. 36

2.25 Energy landscapes in the (X, Y) space learned by two neural networks modeling the function $Y = X^2 - \frac{1}{2}$ (LeCun and Huang, 2005). The left plot shows a quadratic energy surface (smooth, convex paraboloid), while the right plot illustrates a non-quadratic, saturated energy surface with flat regions. In both cases, the color represents the magnitude of the energy, with warmer colors (yellow–pink) indicating higher energy (less plausible states) and cooler colors (green) indicating lower energy (more plausible states). Blue dots indicate points sampled from the target function. 37

3.1 a) An illustration of uniformly distributed 48 initial θ_0. b) Point cloud $\mathcal{C}$, a combination of table's and object's point clouds. c) 3 different configurations used for KG3 gripper. d) Gripper contact surface's point clouds $\mathcal{S}$ of three configurations. 43

3.2 A visualization of grasp simulations for objects from the KIT database with the Franka hand. For each set, the one on the left is the plot of grasp generated, and the one on the right is the corresponding simulation result. 49

3.3 A grasp simulation for Google Scanned Objects. 50

3.4 A visualization of the best grasp pose and its simulation result for KG3. the plot on the left is the best grasp poses for 10 trials with 48 initializations after 50 iterations. The middle and right ones are two of the ten simulation results. 51

3.5 (a) Both grasp poses are successful, but a configuration with a smaller opening between fingers would provide a better grasp. (b) The case where occlusion is too large leading to failure due to collisions. It demonstrates a limitation of our algorithm in dealing with significant occlusions. 52

3.6 An illustration of real grasp with KG3 gripper. Object on the bottom right shows our algorithm having trouble with flat object with large occlusion. 54

4.1 An illustration of the optimization process. Green, blue and black point clouds are the initial poses of three preshapes of the KG3 gripper. Blue plots show the optimization process to find the grasp pose of a singe preshape. 57

4.2 Preshapes used for simulation in this chapter. On the left, we have two preshapes of Barrett Hand. On the right, we selected two preshapes from the ten used for Franka Hand. The full point cloud generates SDF, and the partial point cloud is used to optimize. 57

4.3 (a) Initializations sampled from a mixture of Gaussian to provide some prior knowledge of the object. (b) Initializations sampled from the Fibonacci sequence and projected onto a quarter of the sphere, facing the direction of the robot arm. 62

4.4 Plots of success rate and computation time with an increasing number of initializations for different sampling methods: (a) Gaussian and (b) Fibonacci. Sampling from a mixture of Gaussians leads to a higher success rate and faster computation time with fewer initializations as it provides some prior knowledge of objects. 62

4.5 Best grasp poses with Franka Hand for 50 trials. On the left is the pose generated by our algorithm, and on the right is the pose generated by AnyGrasp. This visualization highlights a key advantage of our gradient-based optimization: it inherently optimizes for suitable gripper preshapes that align the fingers with the object's surface geometry to maximize contact area. In contrast, AnyGrasp appears to generate poses better suited for a fully opened, neutral gripper state. 65

4.6 Best grasp poses with Barrett Hand for 50 trials. 66

4.7 The best grasp poses identified across 30 trials for SGD-ICP (top) and AS-ICP (bottom). The results show that AS-ICP, which incorporates SVGD, produces a more diversified set of poses, especially in rotational space. This aligns with the core objective of using SVGD to enhance exploration. 68

4.8 The picture on the left illustrates fifty grasp poses generated with the same point cloud. The photo on the right is the actual grasp for one of the five grasps in Table. 4.4. Our algorithm can grasp objects with large occlusions and noisy point clouds. 70

4.9 (1) Hand Helping Tool (2)-(6) Grasp poses for Hand Helping Tool with five different shapes. 71

5.1 A brief summary of our algorithm. Both the object and gripper point clouds are fed into separate PCD Encoders to produce 64-dimensional features. These features are then concatenated and scored by the EBM to output a single energy value. The system uses SVGD to iteratively update the transformation parameters θ by leveraging gradients from the EBM and the cost function of ICP. The best pose is selected with minimum energy and matching error. 77

5.2 Influence of kernel bandwidth on SVGD optimization performance for translation (top) and rotation (bottom). The median heuristic (labeled "None") was found to be inadequate for the translation component. The performance within a fixed bandwidth range was relatively stable. We selected $\sigma = 3$ for translation as it performed reliably across our experimental objects, while retaining the median heuristic for rotation to minimize manual hyperparameter. 81

5.3 Energy plots for the EBM trained under different data processing schemes for a top-down grasp. In these plots, blue indicates low energy (favorable grasps), and red indicates high energy (unfavorable grasps). Panels (a) and (e) show results using the entire dataset. Panels (b) and (f) display results when training with only positive examples, paired with an equal number of negative examples sampled from the dataset. Panels (c) and (g) illustrate the outcome when negative examples are generated by uniform sampling around each positive instance. Panels (d) and (h) demonstrate the best performance, achieved by balancing the number of positive examples within each group (defined by object orientation and elevation). The top row corresponds to training without the gripper point cloud, whereas the bottom row includes the gripper point cloud, resulting in significantly improved performance. 83

5.4 Plots of average success rate after lifting and shaking. The results demonstrate a clear trend of increasing performance as the model improves. Our method outperforms individual analytical and data-driven methods, other hybrid approaches, and baseline methods. 85

5.5 Kernel density estimates (KDEs) of success rates for all evaluated methods. The KDEs transform discrete success rate measurements into smooth, continuous probability distributions. The vertical axis shows the estimated probability density, where a higher density at a given success rate indicates that more experimental trials resulted in that performance level. Set 1 compares various hybrid model variants, including ablations of different architectures and training data. Set 2 compares key baselines and representative methods, including analytical, learning-based, and hybrid approaches. Our hybrid model's performance improves as the model becomes better, achieving higher and sharper peaks near a high success rate, indicating improved reliability and overall performance compared to baseline methods in Set 2. 86

5.6 Comparison of grasp diversity across different methods. Translation diversity (x-axis, up to 98th percentile) and rotation diversity (y-axis) are shown for all attempted (blue) and successful (red) grasps, each are computed separately. AnyGrasp produces identical poses, while GPD yields high translation but low rotation diversity. AS-ICP results are concentrated at low diversity, and EBM at high rotation diversity. 88

5.7 Comparison of grasp diversity for various hybrid models. The EBM trained on the initial dataset closely follows the distribution of ICP and exhibits high translation diversity. In contrast, the model trained on the second dataset shows lower translation diversity but substantially higher rotational diversity. Hybrid models overall tend to mirror the distribution of the learned EBM. 88

5.8 Examples of grasp robustness under dynamic testing. Left: A successful grasp that withstands vigorous shaking. Right: A grasp that succeeds in lifting but fails when shaken, with the object falling from the gripper (trajectory shown in red). The green arrows indicate the shaking trajectory applied in both tests. 90

5.9 Simulation results comparing our approach with baseline methods. Our method achieved an average success rate of 60.9%, outperforming AnyGrasp (31.1%), GPD (48.4%), and AS-ICP (56.6%). We also achieved higher object-based minimum and maximum success rates, as indicated by the error bars. 91

5.10 Examples of objects and their corresponding point clouds from different viewpoints, which can differ significantly. 91

5.11 Examples of objects used in simulation that our methods achieves less then 50% success rate (left) and more than 70% (right). There is no obvious visual distinction between object categories with less than 50% success rate and those above 70%, suggesting that occlusion and viewpoint-specific visibility are key factors. 92

5.12 Average success rate of five grasps for ten objects, where our method achieves an average success rate of 72 percent. 93

5.13 Illustration of the real experiment. For each object, the left panel shows the camera's view, the middle panel presents the scanned point cloud with the grasp pose predicted by our method, and the right panel depicts the KG3 gripper executing the actual grasp. 94

List of Tables

2.1 Common Grasp Quality Metrics. 20

2.2 Common Evaluation Metrics for Robotic Grasping Models. 20

2.3 Common Representations for 3D Rotations. 30

3.1 Computation time (seconds) and success rate for Franka, KG3, and Barrett hands. 50

3.2 Average success rate for grasping 10 objects. SplitPSO: success rate reported by Kiatos et al. (2021). Ours (a): success rate for best grasp poses of 20 trials, best grasp pose is the one with the least matching error from each trial's converged grasp poses. Ours (b): success rate of all converged grasp poses for a single trial. 52

3.3 Using Barrett hand, SplitPSO achieves 4.25 seconds on 10 objects; the iterative PPO-JPO method requires 3.263 seconds on 12 objects; the combination of MDISF and GTO averages 4.47 seconds for 10 objects, and ours achieves 0.71 seconds on 10 objects. Using a Parallel gripper, the ISF method accomplishes 2.33 seconds on 9 objects and our method achieves 1.56 seconds on 10 objects. 52

3.4 Computation time and success rate for KG3 real experiment. 55

4.1 Parameters used for Alg. 4.1. The initialization count (6) sets the number of top-down starting poses. 64

4.2 Simulation results comparing AnyGrasp and our method using the Franka Hand in the Isaac Gym simulator. Success rates (rate) and computation times (seconds) are shown. 64

4.3 Simulation result comparison between SplitPSO and our method. The simulation uses Barrett gripper and is carried out in Isaac Gym simulator. 67

4.4 Number of successes for five grasps, where the input for each grasp corresponds to a different orientation of the object. 69

5.1 Comparison of methods by group-level mean, standard deviation, and fraction of groups exceeding selected success thresholds. The arrows indicate whether higher (↑) or lower (↓)

values are preferred. Thresholds (0.1, 0.5, 0.9) denote the fraction of groups whose success rate is above 10%, 50%, and 90%, respectively, providing a sense of performance across low, moderate, and high success regimes. 86

5.2 Computation time breakdown for the hybrid method compared to the original AS-ICP. 92

List of Algorithms

3.1 Parallel Shape Matching 47

4.1 Parallel Shape Matching with AS-ICP 61

5.1 Stein Energy-Based Grasp 82

Nomenclature

Abbreviations

BI	Bayesian Inference
CPU	Central Processing Unit
dof	degree of freedom
EBM	Energy Based Model
GP	Gaussian Process
GPU	Graphics Processing Unit
i.i.d.	independent and identically distributed
ICP	Iterative Closest Point
KDE	Kernel Density Estimate
ReLU	Rectified Linear Unit
s.t.	such that
SDF	Signed Distance Field
SGD	Stochastic Gradient Descent
SVGD	Stein Variation Gradient Descent
VI	Variational Inference

Notation

a	a scalar
$\mathbf{a}$	a vector
$\mathbf{A}$	a matrix
$\mathcal{A}$	a space
A	a functional or a measure

Matrix operations

$[\mathbf{A}]_{ij}$	the element in the i'th row and j'th column of $\mathbf{A}$
$\det(\mathbf{A})$	determinant
$\mathbf{A}^\intercal$	transpose
$\mathbf{A}^{-1}$	inverse

Basic operations

$\operatorname*{arg\,max}_{x \in \mathcal{X}}$	the argument of the maximum, i.e. $\arg\max_{x \in \mathcal{X}} f(x) := \{ x \in \mathcal{X} \mid \forall x' \in \mathcal{X}\ f(x') \leq x \}$
$\operatorname*{arg\,min}_{x \in \mathcal{X}}$	the argument of the minimum, i.e. $\arg\min_{x \in \mathcal{X}} f(x) := \{ x \in \mathcal{X} \mid \forall x' \in \mathcal{X}\ f(x') \geq x \}$
$\lvert a \rvert$	the absolute value of $a \in \mathbb{R}$
$\lVert f \rVert_{\mathcal{F}}$	the norm of $f \in \mathcal{F}$
$\sup$	Supremum (least upper bound) over the function class $\mathcal{F}$

Special symbols

$\mathbb{E}$	expected value
$\mathcal{D}_n$	a dataset containing n entries
$k(\cdot, \cdot)$	a positive definite kernel function
$p(\cdot)$	the probability of an event involving random variables
$p(X \mid Y)$	the conditional probability of X given Y

Authorship Attribution

The contributions presented in this thesis have been published in the following conferences and journals, which form the core chapters of this thesis. The author's attribution includes, but is not limited to, the motivation, conceptualization, formalization, derivation, theorization, experimentation, and communication of the following academic articles.

Chapter 4: Grasping with Annealed Stein ICP

W. Zhang, F. A. Maken, T. Lai, and F. Ramos. 2024. Grasping by parallel shape matching. In *2024 Australasian Conference on Robotics and Automation (ACRA)*, pages xxx–xxx. Australasian Conference on Robotics and Automation, Brisbane, Australia. [Online; accessed 2026-01-11]

I developed the method, designed the experimental details, implemented the idea as code, conducted all the experiments, analyzed the data, formulated the theoretical proofs, and wrote the drafts of the paper. It was nominated for the Best Paper Award.

In addition to the authorship attribution statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Wenzheng Zhang

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

CHAPTER 1

Introduction

ROBOTICS, the idea of creating machines that imitate living beings, dates back thousands of years. Early inventors across various cultures built mechanical devices that could perform repetitive or lifelike actions without human control. Modern robotics, however, began to take shape in the mid-20th century with the advent of programmable industrial robots, revolutionizing manufacturing processes.

Today, robotics encompasses a broad range of systems from simple automated arms to complex humanoid robots, capable of performing diverse tasks in dynamic and unstructured environments. Among these capabilities, robotic grasping — the ability to perceive, plan, and execute object manipulations — remains a critical and challenging area of research, essential for enabling robots to interact effectively with the physical world.

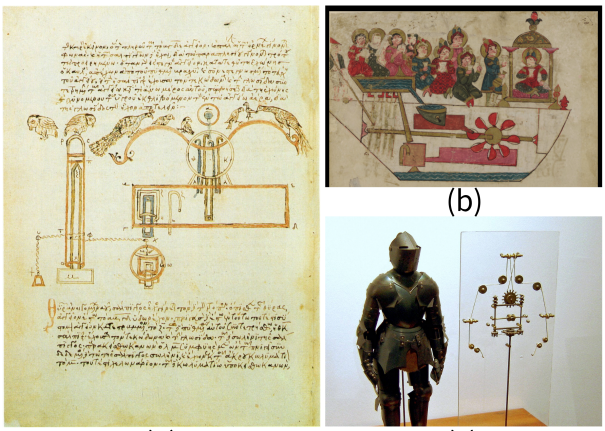

FIGURE 1.1. (a) Hero of Alexandria's design of singing birds (Łuniewicz and Jagiełło, 2024). (b) Al-Jazari's automaton musicians (Kennedy and Stanton, 1924). (c) Leonardo da Vinci's design for mechanical knight (Pedretti and Rosheim, 2006).

1.1 Motivation

Hero of Alexandria in ancient Greece designed singing birds (Figure 1.1(a)). During the Islamic Golden Age, engineers like Al-Jazari developed intricate programmable machines such as automaton musicians (Figure 1.1(b)), showcasing early robotic concepts using gears and levers. Later, in medieval and Renaissance Europe, figures like Leonardo da Vinci sketched designs for mechanical knights capable of basic movements (Figure 1.1(c)), reflecting humanity's long-standing fascination with automating complex tasks.

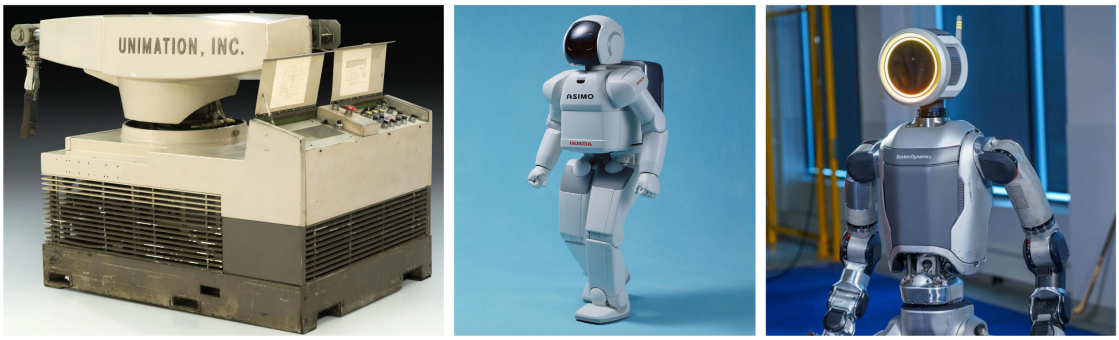

FIGURE 1.2. (a) The Unimate robot (Marsh, 2022). (b) Honda's ASIMO humanoid robot (Honda Motor Co., Ltd., 2001). (c) Boston Dynamics' Atlas robot (Boston Dynamics, 2024).

The evolution of modern robotics began in the mid-1900s with the creation of robotic arms designed for industrial applications. The Unimate robot (Figure 1.2(a)), developed in the 1960s by George Devol and Joseph Engelberger, is widely recognized as the first programmable industrial robot. Deployed at General Motors, the Unimate automated tasks such as welding and handling hot metal parts, marking a transformative shift in manufacturing by increasing productivity and improving workplace safety. This early success laid the groundwork for the widespread adoption of robotics in factories and paved the way for today's sophisticated automated manufacturing systems.

Building on these industrial foundations, modern robotics has advanced to develop humanoid robots capable of operating in human-centric environments. Honda's ASIMO (Figure 1.2(b)) was among the first humanoids to showcase advanced bipedal locomotion and manipulation abilities, while Boston Dynamics' Atlas (Figure 1.2(c)) is notable for its dynamic balance and mobility. These robots target applications ranging from healthcare and personal assistance to disaster response, where human-like interaction and manipulation are critical.

The field of robotic manipulation has been further shaped by the rise of data-driven methods. In research settings, approaches that learn from demonstrations (Wang et al., 2024), simulations (Li and Cappelleri, 2024), or real datasets (Nakahara and Calandra, 2025) have shown promise for enabling adaptive behavior in complex environments. Advances in computer vision, primarily driven by deep learning, have significantly enhanced robots' ability to perceive and interpret unstructured scenes (Blumenkamp et al., 2023). However, translating these advances into robust, reliable, and generalizable physical interactions—such as grasping a previously unseen object—remains a fundamental challenge.

Successful robotic grasping requires the tight integration of accurate perception, robust planning, and precise execution to handle diverse objects under uncertainty and partial observation. Analytical and learning-based approaches each excel at distinct aspects of this problem. While the benefits of combining them are clear, current hybridization often remain superficial—for instance, employing analytical models merely for post-processing or candidate selection rather than guiding the learning or optimization process (Wu et al., 2022). This highlights the need for deeply integrated systems that leverage complementary strengths. Consequently, the core challenge of grasp synthesis under partial observation demands a more synergistic integration of methods, which forms the central focus of this thesis.

1.2 Problem Statement

Analytical and learning-based approaches each excel at distinct aspects of this problem. Analytical (or model-based) methods derive grasp strategies from principles of physics and geometry (Ferrari and Canny, 1992), offering robustness guarantees and sample efficiency when accurate object models are available. However, their reliance on precise models makes them brittle to the uncertainties and partial observability inherent in real-world settings. Conversely, learning-based (or data-driven) methods circumvent the need for explicit modeling by learning grasp strategies directly from data (Lenz et al., 2015). This allows them to handle perceptual noise and complex object shapes effectively, but often at the cost of requiring vast datasets and providing limited robustness guarantees (Fang et al., 2023). This dichotomy creates a natural complementarity: one paradigm provides generalizable, principled reasoning, while the other offers adaptability to real-world complexity.

Yet, most current hybrid approaches remain weakly integrated. A common pattern is to apply analytic methods only as a final post-processing to optimize grasps proposed by a learning-based network (Fang et al., 2023), or to use analytic metrics solely for evaluation (Mahler et al., 2016). This sequential, "black-box" concatenation fails to leverage the core strengths of each paradigm during the generative process itself. Since analytic methods are highly sensitive to initial conditions, simply applying them at the end of a pipeline does not guide or improve the learning-based generation for robust grasps. A more profound, synergistic integration—where analytic principles actively constrain and inform the data-driven generation loop—is therefore necessary to fully harness the complementary benefits of both approaches.

This raises a fundamental question:

Can we develop a method that deeply integrates analytic and learning-based approaches, harnessing the strengths of both while mitigating their individual limitations?

This thesis seeks to explore such synergistic hybrid frameworks to advance robust and generalizable robotic grasp synthesis.

1.3 Contributions

Grasping as Rigid Shape Matching. We formulated grasp synthesis as rigid shape matching using point clouds of both the object and gripper. This allowed us to utilize the gripper's Signed Distance Field (SDF) for efficient collision avoidance, reducing dependence on precise object models. Additionally, we used a parallelized Stochastic Gradient Descent-based ICP (SGD-ICP) algorithm, significantly accelerating optimization compared to existing approaches.

Grasping with Annealed Stein ICP. We proposed integrating Stein Variational Gradient Descent (SVGD) with our SGD-ICP optimization to generate a diverse pose distribution. This hybrid approach reduced sensitivity to initializations around the object and enabled incorporating prior knowledge and task-specific constraints. Crucially, it generated a diverse distribution of grasp poses rather than repetitive outcomes typically observed in conventional optimization.

Stein Energy-Based Grasp Synthesis. We developed a novel hybrid framework combining learning-based (Energy Based Models, EBMs) and analytical (Iterative Closest Point, ICP) approaches within the SVGD optimization pipeline, specifically tailored for robust grasp synthesis from partial point clouds. We provided a detailed analysis of how dataset quality and selection impact the learned energy function, highlighting that curated datasets often yield superior performance compared to larger but less focused datasets. Extensive simulation results and ablation studies demonstrated that blending optimization methods with data-driven models significantly enhances generalization and overall grasp quality.

1.4 Outline

This thesis is organized as follows. The necessary theoretical background to understand the main contributions is presented in chapter 2. This chapter introduces fundamental concepts in robotic manipulation,

providing background on analytic, learning-based, and hybrid approaches for grasp synthesis. Additionally, it formally presents key techniques including SGD, SVGD, ICP, and EBMs.

The core contributions of this thesis are detailed in the following three chapters: chapter 3, chapter 4, and chapter 5, each following a consistent structure. Each chapter begins with a motivating introduction, followed by a review of related work, a detailed presentation of the proposed methodology, and a series of experiments validating the approach. Specifically, chapter 3 formulates grasp synthesis as a rigid shape matching problem optimized via stochastic gradient descent. Next, chapter 4 extends this framework by integrating SVGD, laying the foundation for a gradient-based hybrid method and demonstrating its efficacy in grasp synthesis from partial observations. Finally, chapter 5 further incorporates gradients from a trained EBM into the SVGD framework to enhance grasp optimization.

The thesis concludes in chapter 6 with a summary of the contributions and suggestions for future research directions.

CHAPTER 2

Background

This chapter reviews the foundational concepts underpinning the methods proposed in this thesis. Our work addresses the robotic grasp synthesis problem considering both full object models and partial observations. While many existing approaches fall into either analytical or data-driven categories, hybrid methods remain relatively weakly integrated. In contrast, this thesis proposes a more integrated approach by combining gradients from both analytical and learning-based methods within the Stein Variational Gradient Descent (SVGD) framework.

We begin by reviewing the definition and key research areas in robotic manipulation, with particular emphasis on in-hand manipulation and trajectory planning (section 2.1). Next, we focus on the core subject of this thesis: robotic grasping (section 2.2). This includes an overview of different types of grippers, as well as analytical, learning-based, and hybrid grasp synthesis methods. Relevant concepts such as evaluation metrics, sensor calibration, and collision avoidance are also introduced.

Following this, we explore optimization techniques starting with stochastic gradient descent (section 2.3), which serves as a fundamental tool for grasp pose synthesis. We then present SVGD (section 2.4) as a powerful framework to integrate gradients from both analytical and data-driven approaches, enabling robust and efficient grasp optimization. Specifically, we delve into the analytical perspective using the ICP algorithm (section 2.5), and the learning perspective via EBMs (section 2.6). Finally, this chapter concludes with a summary linking the background material to the subsequent contributions of this thesis.

2.1 Robotic Manipulation

Defining manipulation precisely can be challenging. One widely accepted definition, as stated in the Annual Review of Control, Robotics, and Autonomous Systems (Mason, 2018), is:

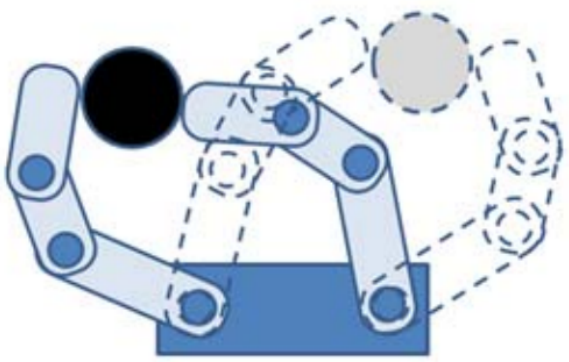

FIGURE 2.1. Illustration of in-hand manipulation (Ma and Dollar, 2011).

Manipulation refers to an agent's control of its environment through selective contact.

Here, we provide a brief overview of robotic manipulation focusing on topics most relevant to this thesis: in-hand manipulation and trajectory planning. We do not delve deeply into these topics, assuming their successful operation in both simulation and real experiments. The following chapters report the time required for grasp pose computation. This focus on optimization time—rather than end-to-end execution time—is consistent with standard evaluation protocols in the grasp synthesis literature, where algorithmic efficiency is compared under controlled conditions. For practical deployment, the total cycle time including perception, motion, and physical execution is critical; however, as robot actuation times were constant across our comparisons, the reported computation times are the primary difference in the overall system. A complete system-level evaluation presents a valuable direction for future work. For more information, readers are referred to Mason (2001) and Mason (2018). We will discuss grasping in more detail in Section 2.2.

2.1.1 In-hand manipulation

In-hand manipulation is a significant research area within robotic manipulation, closely related to dexterity — the ability to change an object's position and orientation within the hand (Bicchi, 2000), as illustrated in Figure 2.1. It is the fine, sensor-guided adjustments made after initial contact. Human dexterity relies on such tactile feedback to secure an object, a capability researchers aim to replicate, as shown in work using tactile sensors to learn feedback control models from an initial pose (Figure 2.2) (Tian

FIGURE 2.2. Illustration of a process to achieve a precise grasp while minimizing object displacement (Tian et al., 2024).

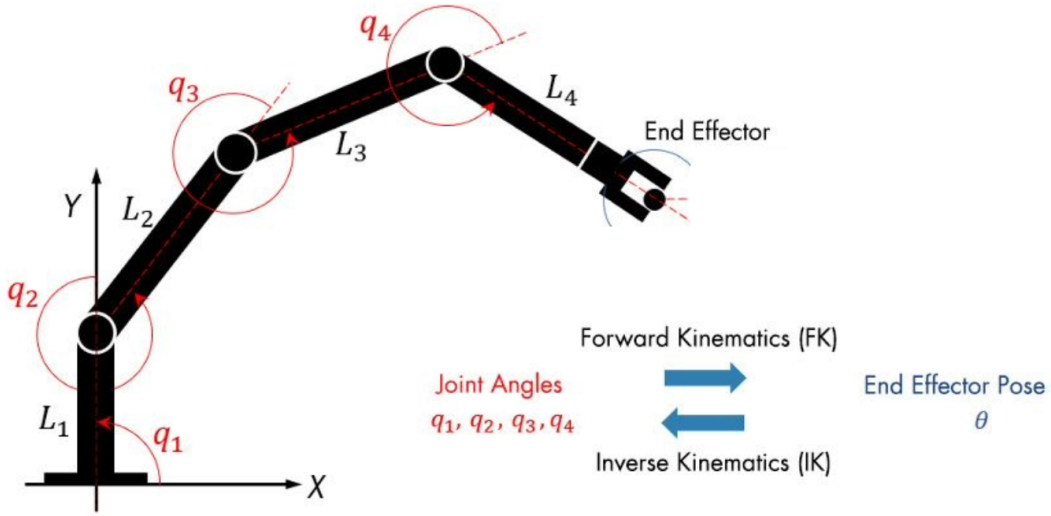

FIGURE 2.3. Illustration of forward kinematic and inverse kinematic of a robotic arm (The MathWorks Inc., 2023).

et al., 2024). While this post-contact control is crucial for robust grasping, this thesis focuses on the logically prior and distinct challenge of grasp synthesis: computing a stable initial grasp pose from partial observations. Consequently, and in alignment with a significant body of foundational grasping literature, we adopt a standard simplification: the gripper executes a predefined, open-loop closing action upon reaching the target pose. This approach isolates the synthesis problem for study but introduces a known limitation, as it does not adjust for object displacement during closure, which can lead to grasp failure. The choice of this model clarifies the boundary of our contribution; we advance methods for finding grasp poses, assuming a common, simplified execution. Readers interested in the subsequent challenges of in-hand manipulation are referred to Ma and Dollar (2011) and Pfanne (2022).

2.1.2 Trajectory planning

Trajectory planning is a broad research area covering many robotic platforms, including mobile robots, UAVs, and robotic arms. Here, we focus on the scenario where a gripper mounted on a robotic arm needs to reach a specified pose. The simplest approach is to provide the desired grasp pose to the robot and use inverse kinematics (IK) to reach it.

As illustrated in Figure 2.3, given joint inputs $\mathbf{q}$ for a robotic arm, forward kinematics maps these to an end-effector pose in task space:

$$\boldsymbol{\theta} = f(\mathbf{q}) \tag{2.1}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the joint configuration vector, containing the angular or linear positions of the robot's n joints, and $\boldsymbol{\theta}$ denotes the position and orientation of the end-effector.

Inverse kinematics aims to find joint values $\mathbf{q}$ that achieve a desired pose $\boldsymbol{\theta}_d$. Numerically, IK is often solved iteratively using the Jacobian:

$$\Delta\boldsymbol{\theta} = \mathbf{J}(\mathbf{q})\,\Delta\mathbf{q} \tag{2.2}$$

where $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}_d - f(\mathbf{q})$ and $\mathbf{J}(\mathbf{q}) = \frac{\partial f}{\partial \mathbf{q}}$. The joint values are updated iteratively as follows:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \alpha\,\mathbf{J}^{\dagger}(\mathbf{q}_k)\,(\boldsymbol{\theta}_d - f(\mathbf{q}_k)) \tag{2.3}$$

where $\mathbf{J}^{\dagger}(\mathbf{q}_k) = \mathbf{J}^T(\mathbf{q}_k)\left(\mathbf{J}(\mathbf{q}_k)\mathbf{J}^T(\mathbf{q}_k)\right)^{-1}$ is the Jacobian pseudo-inverse and $\alpha \in (0, 1]$ is a step size parameter.

In this thesis, our method generates the desired pose $\boldsymbol{\theta}_d$ and uses the built-in IK solver from the simulator or robot arm to reach the goal. However, simply commanding a single final pose can lead to two common failure modes. First, the IK solver may fail to find a collision-free trajectory due to the long planning horizon. Second, the gripper may reach the target position faster than it achieves the desired orientation, causing a collision with the object.

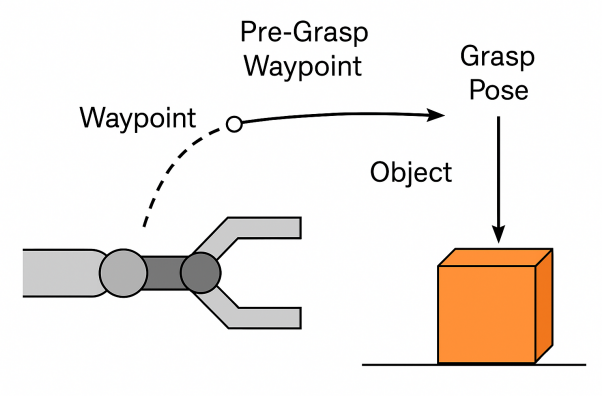

FIGURE 2.4. Illustration of using waypoints to avoid collision during grasping.

Advanced trajectory planning methods, such as those in Ekrem and Aksoy (2023), can handle collision avoidance effectively. Since this thesis focuses on grasp synthesis, we rely on the built-in IK solver but provide a series of pre-defined waypoints that are known to avoid collisions, as shown in Figure 2.4. For more details on trajectory planning, readers are referred to Choset et al. (2005).

2.2 Grasping

Grasping is a fundamental aspect of robotic manipulation, serving as a prerequisite for in-hand manipulation and trajectory planning discussed in the previous Section. A secure grasp is essential before reorienting an object in-hand, as it ensures stability during manipulation and prevents accidental dropping, which is particularly critical in pick-and-place tasks. This concept has been recognized in the context of grasp gaits, where maintaining stability during finger reconfigurations is necessary to avoid losing control of the object (Leveroni and Salisbury, 1996). In many learning-based studies, ground-truth grasp poses are assumed to be provided as part of the dataset; however, developing a reliable method to generate grasp poses remains challenging due to the complexity of gripper designs and the diversity of objects. This thesis focuses on grasp pose optimization using both analytical and learning-based methods. Accordingly, this Section introduces various types of grippers, grasp quality metrics, optimization techniques, and learning-based approaches. For a comprehensive overview of grasping, readers are referred to bg (2013).

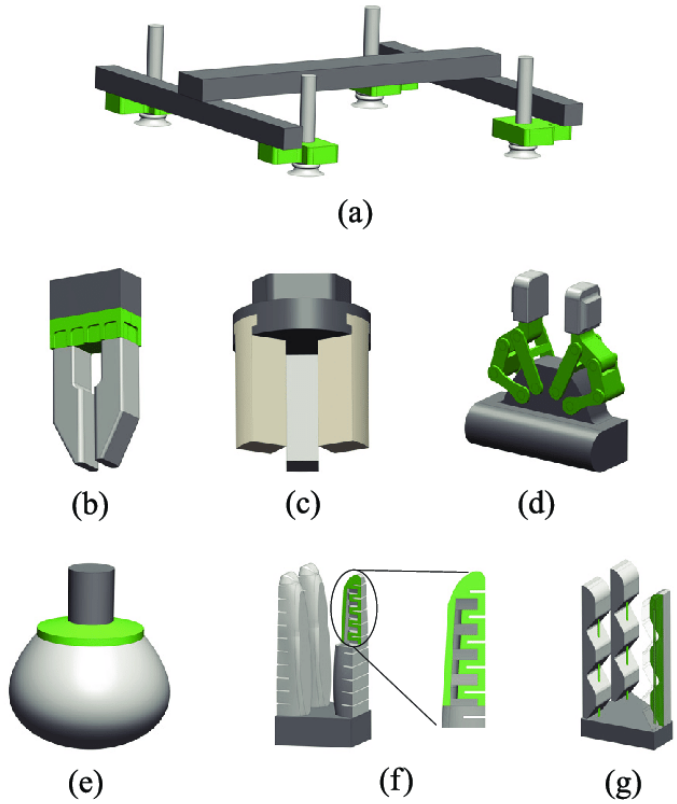

FIGURE 2.5. Types of traditional grippers: (a) vacuum gripper (suction cups); (b) parallel pneumatic gripper (pinching action); (c) hydraulic gripper (collet type); (d) servo-electric gripper (pinching action). Types of soft grippers: (e) granular jamming-based soft gripper; (f) gripper with soft fingers driven by positive pneumatic pressure; (g) soft gripper with cable-driven fingers (Subramaniam et al., 2020).

2.2.1 Types of Grippers and End-Effectors

Grippers can be categorized into mechanical grippers, vacuum grippers, and soft grippers, as illustrated in Figure 2.5. Within each category, various standard and customized grippers exist, each exhibiting different kinematics and requiring distinct control strategies. This diversity complicates the development of a unified grasp synthesis method. This thesis utilizes three different grippers: Franka, Barrett, and KG3.

Franka Hand

The Franka Hand (Figure 2.6) (Franka Emika GmbH, 2016–) is a widely used gripper in research. It is a parallel gripper with two fingers that open and close along the same axis, as shown in Figure 2.7. The gripper has a maximum opening width of approximately 80 mm and can handle payloads up to 3 kg.

FIGURE 2.6. Image of a Franka Hand (Franka Emika GmbH, 2016–).

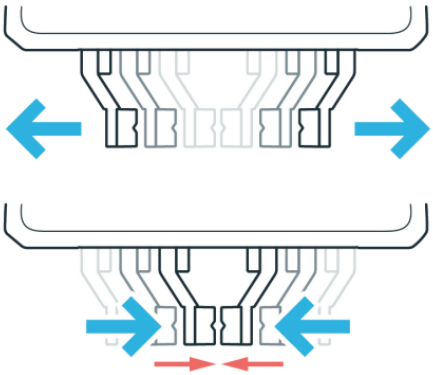

FIGURE 2.7. Illustration of the finger control of a Franka Hand (Franka Emika GmbH, 2016–).

Barrett Hand

The Barrett Hand (Figure 2.8) (Barrett Technology, 1990–) is another commonly used gripper, featuring three fingers with four degrees of freedom. Its range of motion is depicted in Figure 2.9. Some models are equipped with tactile sensors on the fingertips to enhance grasp feedback and control.

KG3 Gripper

The KG3 gripper (Figure 2.10, right) is developed by Kinova. It features three fingers, with two designed to operate in parallel, resembling a two-fingered gripper (Figure 2.10, left) (Kinova Robotics Inc., 2006–).

FIGURE 2.8. Image of a Barrett Hand (Barrett Technology, 1990–).

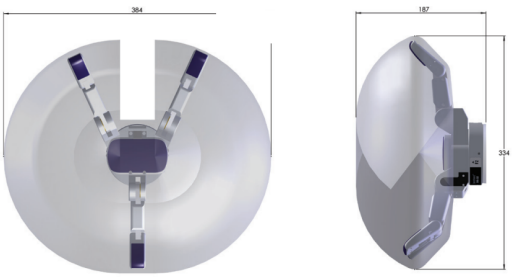

FIGURE 2.9. Range of motion of the Barrett Hand: horizontal spread 384 mm, vertical spread 334 mm, depth 187 mm (Barrett Technology, 1990–).

FIGURE 2.10. Image of Kinova grippers (Kinova Robotics Inc., 2006–).

2.2.2 Analytical Methods

Early work on grasp synthesis focused on theoretical analyses of what constitutes a good grasp, often involving the analysis of the wrench, which comprises a linear component (force) and an angular component (moment) acting at a point (Murray et al., 1994). The objective is to identify grasps that maintain object stability in the presence of disturbances. Two critical grasp properties are form closure (Brost, 1991) and force closure (Nguyen, 1987). A set of contacts achieves form closure if, for every possible

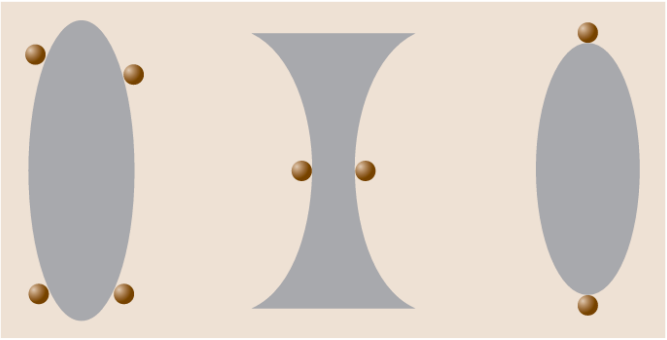

FIGURE 2.11. Examples of form closure in planar grasps. The gray shapes represent objects, and the brown disks represent fingertip contact points. **Left:** First-order form closure, where the minimum number of contacts fully constrains the object against any small movement. **Middle:** Higher-order form closure, where additional or strategically placed contacts provide increased constraint and stability. **Right:** No form closure—insufficient or poorly positioned contacts allow the object to move freely. Adapted from (Prattichizzo and Trinkle, 2008).

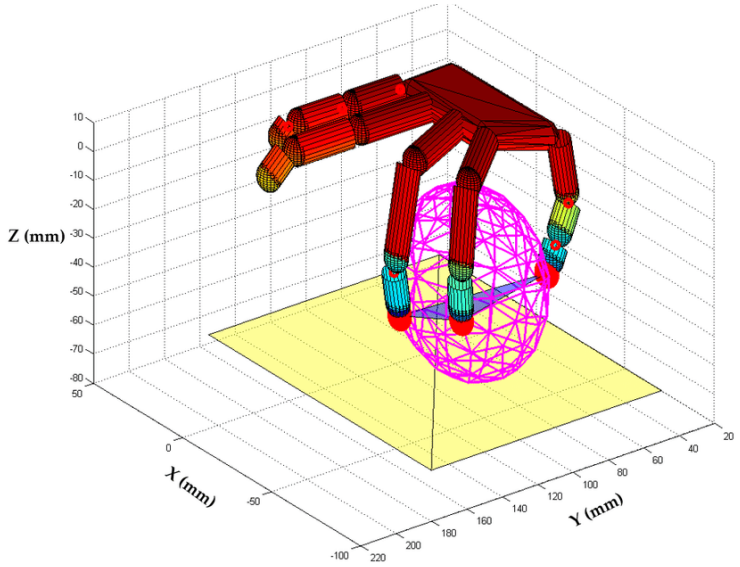

FIGURE 2.12. Force closure grasp of a sphere (Zaidi et al., 2017).

instantaneous motion of the object, at least one contact applies a unilateral constraint preventing that motion, as illustrated in Figure 2.11. Force closure is achieved if the contacts can resist any arbitrary external force and moment applied to the object through admissible contact forces within friction cones, as shown in Figure 2.12.

By the late 1990s, research shifted from pure analysis to optimization. One of the earliest and most influential works is by Ferrari and Canny (1992), who introduced the concept of the grasp wrench space (GWS)—the set of all wrenches a robotic hand can apply to an object through its contact points,

considering friction constraints. Their approach involves modeling contact points and friction cones to determine the set of possible wrenches a grasp can exert, computing the convex hull of these wrenches to form the GWS, and evaluating grasp quality based on the distance from the origin to the closest facet of this convex hull.

The emergence of simulation and planning tools such as GraspIt! (Miller and Allen, 2004) in the early 2000s marked a significant breakthrough, providing a platform for large-scale grasp data testing and collection. GraspIt! also introduced the concept of an eigengrasp space to reduce the dimensionality of the hand's joint space, facilitating more efficient grasp planning (Ciocarlie et al., 2007). Concurrently, efforts were made to relax assumptions on object shapes, extending from simple polygons (Han et al., 2000) to general shapes (Roa and Suárez, 2009).

Despite the high precision of grasping with a full object's model, analytical methods have many limitations:

- **Dependence on Accurate Object Models**: Most analytical methods require precise geometric and physical models of objects. In real-world scenarios, obtaining such detailed models is challenging due to sensor noise, occlusions, and the diversity of object shapes and materials.
- **Simplified Assumptions**: To make the problem tractable, analytical methods often incorporate simplifying assumptions, such as point contacts, rigid bodies, and known friction coefficients. These assumptions may not hold in real-world situations, where contacts are distributed, objects may deform, and friction properties are uncertain.
- **Computational Complexity**: Calculating optimal grasps analytically can be computationally intensive, especially for complex objects and multi-fingered hands. The high dimensionality of the configuration space and the need to evaluate numerous potential contact points make real-time planning difficult.

Consequently, analytical methods are sensitive to uncertainty and struggle in scenarios where object models are unavailable. Coupled with long computation times, they are less suited for real-world grasping tasks. For a detailed survey on analytical methods, readers are referred to Zhang et al. (2022).

Analytical metrics are summarized in Section 2.2.5, while optimization techniques are presented in Sections 2.3 and 2.4.

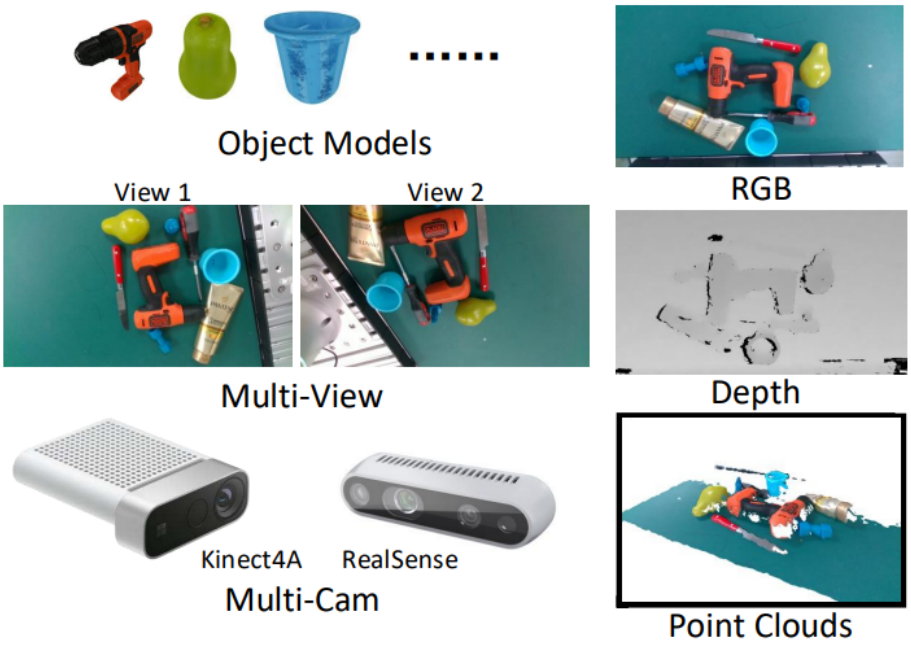

FIGURE 2.13. Illustrations of different types of data used for grasping (Gou et al., 2021).

2.2.3 Learning-Based Methods

Before introducing learning-based methods, it is important to mention two major types of visual sensing used for grasping: RGB cameras and depth sensors. RGB cameras capture color images, aiding in object recognition and scene understanding. Depth sensors provide 3D information about the environment, enabling the creation of point clouds for object localization and shape estimation. Different types of data used for grasping are shown in Figure 2.13.

Recent advancements in computational power have enabled the integration of machine learning into grasp synthesis. A notable application is addressing the challenge of grasping with partial observations. Point cloud completion techniques have been introduced to reconstruct a full point cloud of an object from partial data. For instance, Mohammadi et al. (2023) proposed 3DSGrasp, which utilizes a transformer-based encoder-decoder architecture to complete the point cloud, subsequently used to generate grasp poses, as illustrated in Figure 2.14.

One of the pioneering works in this domain is by Saxena et al. (2008), who developed learning algorithms that predict grasp poses directly from visual inputs, eliminating the need for explicit object models. Learning-based grasp synthesis approaches can be broadly categorized into two types: sampling-based and end-to-end methods.

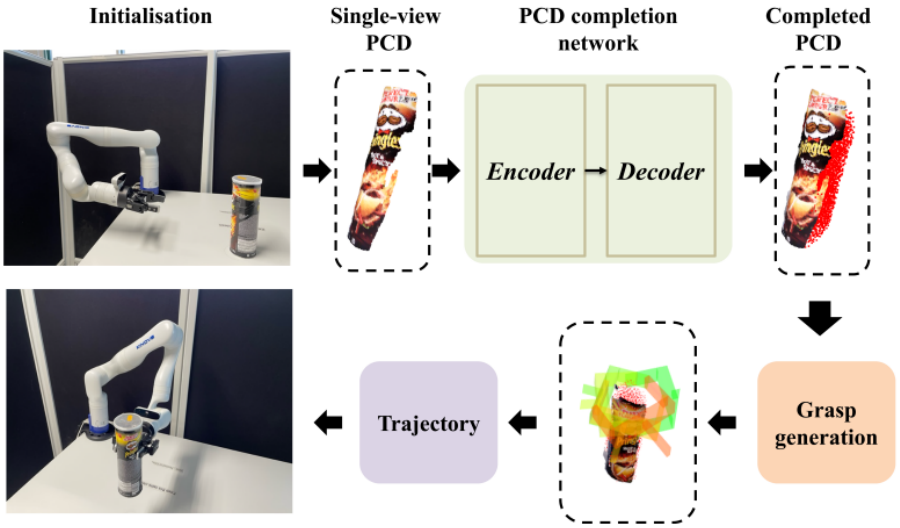

FIGURE 2.14. Pipeline of the 3DSGrasp grasping strategy (Mohammadi et al., 2023).

- **Sampling-Based Methods**: These methods generate a set of grasp candidates using strategies such as Euclidean space sampling or latent space sampling, either randomly or uniformly. Each candidate is then evaluated using a model trained on a large dataset comprising pairs of object inputs and corresponding grasp poses.
- **End-to-End Methods**: These approaches train a network to predict grasp poses directly from inputs like RGB images or point clouds, bypassing the intermediate step of candidate generation.

Despite promising results, learning-based methods face several limitations:

- **Generalization**: Due to the nature of the training process, learning-based methods often struggle to generalize to objects that differ significantly from the training data or to environments that vary from the data collection settings. Transferring trained models between different grippers also poses challenges.
- **Data Generation and Annotation**: Training these networks requires large amounts of labeled data. Generating new data for varying requirements can be time-consuming, even with simulation assistance. Differences as minor as a defect in the gripper can necessitate new data collection.
- **Partial Observation**: Similar to analytical approaches, occlusions present significant challenges to trained models. Missing information exacerbates generalization issues.

Currently, no standardized architecture exists for these models. For a comprehensive survey of various network structures, readers are referred to Zhang et al. (2022) and Newbury et al. (2023). A detailed discussion of network architectures and EBMs is provided in Section 2.6.

2.2.4 Hybrid Methods

To leverage the complementary advantages of analytical and data-driven grasp synthesis, recent research has explored hybrid approaches that integrate both methodologies. However, these approaches often remains weakly integrated. Common hybrid strategies include utilizing analytical grasp quality metrics as cost functions during the training of data-driven models (Mahler et al., 2017), or as evaluation criteria for selecting among grasp candidates proposed by learning-based methods (ten Pas et al., 2017). Additionally, optimization techniques are frequently applied as a post-processing step to refine the initial grasp poses generated by data-driven approaches, ensuring stability and collision avoidance (Sundermeyer et al., 2021).

Some research has even attempted to develop selection frameworks that dynamically choose between analytical and learned methods based on the input characteristics or task requirements (Komoda et al., 2024). Despite these advancements, existing hybrid systems typically do not fully exploit the deeper synergies between analytical and learning-based methodologies.

In this thesis, we propose a novel hybrid grasp synthesis framework that deeply integrates analytical and data-driven approaches by simultaneously leveraging gradients derived from both methodologies to optimize grasp poses.

2.2.5 Grasp Quality Metrics

An essential component common to both analytical and learning-based grasp synthesis methods is the use of grasp quality metrics, which play a crucial role in both training and evaluation phases. Among these metrics, force closure is fundamental. As illustrated in Figure 2.15, a grasp achieves force closure if it can counteract arbitrary external forces and torques exerted on the object. Practically, this means that the resultant grasp forces must lie within the friction cones at each contact point (represented by the dashed lines in the figure).

Another important grasp quality metric is the distance to the object's center of mass (CoM). As depicted in Figure 2.16, grasping an object significantly far from its CoM generates a moment due to the leverage

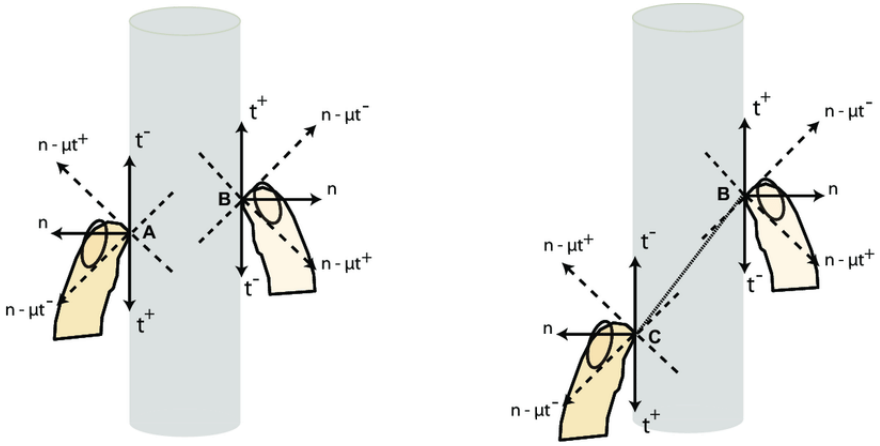

FIGURE 2.15. Illustration of a force closure grasp (left) and non force closure grasp (right) (Christopoulos, 2010).

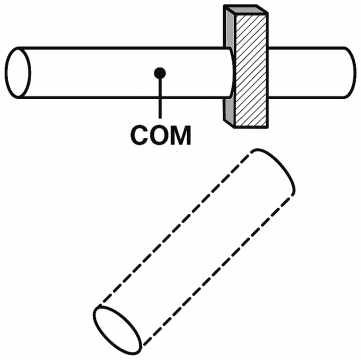

FIGURE 2.16. Illustration of grasp failure caused by the moment generated by the object's center of mass (CoM) (Generated by GPT OpenAI et al. (2023)).

effect. This moment tends to rotate or destabilize the object, potentially leading to grasp failure. Therefore, minimizing the distance between the grasp point and the object's CoM is generally desirable to enhance stability.

A selection of commonly used analytical grasp metrics is presented in Table 2.1. Comprehensive details and additional metrics can be found in Mnyussiwalla et al. (2022). Additionally, Table 2.2 summarizes frequently employed empirical metrics used to evaluate grasping algorithms in practice.

Metric	Description
Force Closure Nguyen (1987)	Determines whether a grasp can resist arbitrary external forces and torques, ensuring object immobilization. A fundamental criterion for grasp stability.
Volume of the Grasp Wrench Space (GWS) Miller and Allen (1999)	Represents the set of all wrenches (force and torque combinations) that a grasp can apply to an object. The volume indicates the grasp's ability to resist disturbances; a larger volume suggests a more robust grasp.
Minimum Singular Value of the Grasp Matrix Li and Sastry (2002)	The smallest singular value of the grasp matrix reflects the grasp's resistance to disturbances in the weakest direction. A higher value indicates better stability and dexterity.
Area of the Grasp Polygon Mirtich and Canny (1994)	Calculated by connecting the contact points of the fingers, forming a polygon. The area reflects the spatial distribution of contact points; a larger area generally correlates with increased grasp stability.
Distance to the Object's Center of Mass (COM) Ding et al. (2001)	Measures the distance between the centroid of the grasp (formed by contact points) and the object's center of mass. Minimizing this distance reduces the likelihood of object rotation or slippage during manipulation.
Minimization of Grasping Forces Daoud et al. (2011)	Evaluates the efficiency of a grasp by assessing the total force required to maintain it. Lower required forces imply energy-efficient and potentially safer grasps, especially important when handling delicate objects.

TABLE 2.1. Common Grasp Quality Metrics.

Metric	Description
Grasp Success Rate	The percentage of successful grasps, calculated as the number of successful grasps divided by the total number of attempted grasps.
Completion/Clearance Rate	The percentage of objects removed from a cluttered environment, computed as the number of objects grasped divided by the total number of objects in the clutter.
Computation Time	The time required to compute grasp hypothesis generation, including the duration for planning and evaluating grasp candidates.
Average Precision (AP)	The percentage of successful grasps for a single object, determined by dividing the number of successful grasps by the total number of attempted grasps for that specific object.

TABLE 2.2. Common Evaluation Metrics for Robotic Grasping Models.

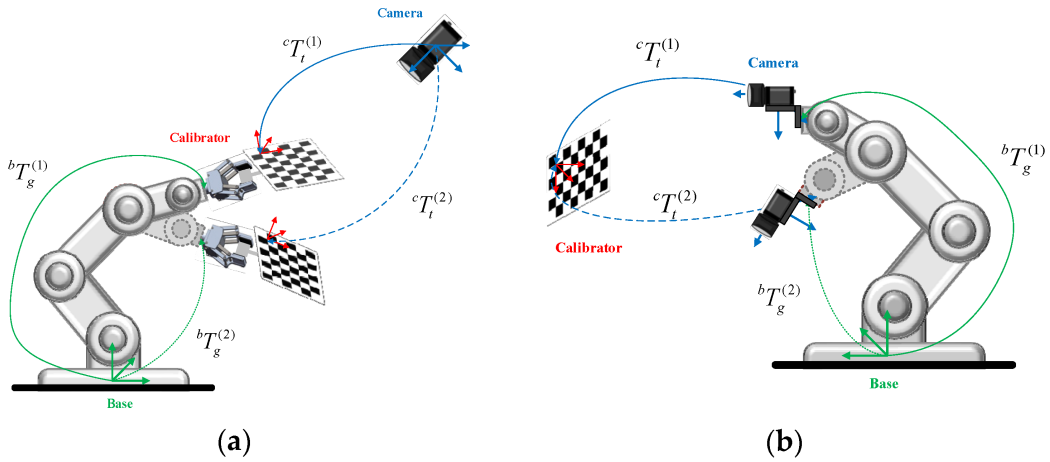

FIGURE 2.17. Illustration of Eye-to-hand (a) and Eye-in-hand (b) (Lin et al., 2022).

2.2.6 Object Pose Detection

Two primary grasping tasks commonly encountered in robotic manipulation are grasping isolated objects and clearing objects from cluttered scenes. Both tasks are inherently object-centric; thus, precise localization of objects within the workspace is critical for successful grasping. Accurate object localization typically relies on visual sensing systems, which can be mounted either on the robot's gripper (eye-in-hand) or at a fixed location independent of the manipulator (eye-to-hand).

In practice, precisely measuring the transformation between the sensor and gripper, or determining the exact location of a fixed sensor, is challenging. Therefore, automated calibration methods are employed to establish accurate spatial relationships and enable reliable object pose estimation.

Eye-in-Hand Calibration

Eye-in-hand calibration addresses scenarios where the sensor (typically a camera) is mounted directly on the robot's gripper or end-effector. This calibration aims to determine the transformation between the sensor coordinate frame and the gripper coordinate frame, known as the hand-eye calibration problem. The calibration can be represented by the following Equation (Tsai and Lenz, 1989):

$$AX = XB, \tag{2.4}$$

where:

- A represents known transformations of the gripper between two distinct robot poses.

- B represents corresponding known transformations of the sensor (camera) between two image poses.
- X is the unknown rigid-body transformation from the gripper frame to the sensor frame.

Common approaches to solving this equation include methods proposed by Tsai and Lenz (1989) and Horaud and Dornaika (Horaud and Dornaika, 1995).

Eye-to-Hand Calibration

Eye-to-hand calibration involves a fixed sensor mounted separately from the robot. Here, the calibration problem seeks to determine the transformation between the fixed sensor frame and the robot's base coordinate frame. This relationship enables the translation of detected object poses from the sensor's coordinate frame into the robot's base frame, and is typically formulated as (Park and Martin, 1994):

$$AX = ZB, \tag{2.5}$$

where:

- A represents known transformations of the robot's gripper relative to the robot base.
- B represents known transformations of a target object with respect to the fixed sensor.
- X is the unknown transformation from the fixed sensor frame to the robot base frame.
- Z is the transformation from the calibration target to the gripper (often assumed to be known and constant).

Solutions to this type of calibration are provided by Park and Martin (1994).

In practice, both calibration methods involve capturing multiple images at various known configurations of the robot's end-effector as illustrated in Figure 2.17. The accuracy of the calibration typically improves with an increased number of diverse poses, as this diversity helps to avoid singularities and minimize estimation errors. Solving the calibration equations involves optimization algorithms that minimize the difference between observed and predicted sensor transformations. Popular methods for solving these equations include nonlinear least squares optimization, singular value decomposition (SVD), and iterative optimization techniques. For simpler calibration setups, the OpenCV library (Itseez, 2015) provides well-established and user-friendly packages, offering convenient implementations of these calibration

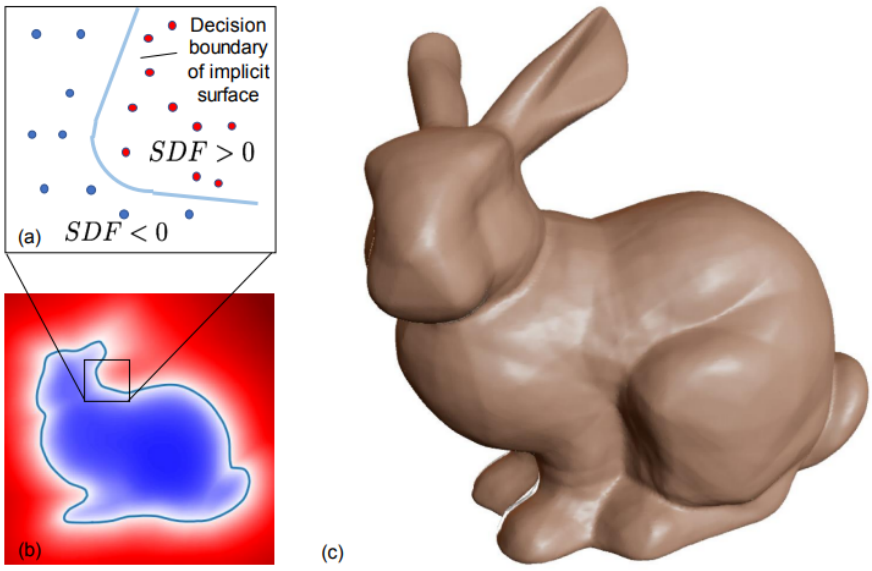

FIGURE 2.18. Illustration of a SDF, the blue points are inside the object. (Park et al., 2019).

algorithms. This significantly reduces the complexity and implementation effort for practical robotic applications.

2.2.7 Collision Avoidance

Collision avoidance is a crucial requirement in robotic grasping to ensure that the gripper interacts with the object only at intended contact points. Different approaches handle this challenge in distinct ways. Analytical methods typically achieve collision avoidance through explicit geometric computations, while sampling-based methods filter out invalid grasp candidates. Although end-to-end learning methods aim to generate collision-free grasp poses directly, they often require fine-tuning or additional constraints to guarantee they are free of collisions.

A widely used technique for collision detection is the *Signed Distance Field (SDF)* (Malladi et al., 1995). An SDF is a scalar field that assigns to each point p in space a value representing the shortest distance $d(p, \partial\Omega)$ to the object's surface $\partial\Omega$, as illustrated in Figure 2.18. The sign of the distance indicates whether the point is inside or outside the object:

$$\text{SDF}(p) = \begin{cases} -d(p, \partial\Omega), & \text{if } p \text{ is inside the object,} \\ d(p, \partial\Omega), & \text{if } p \text{ is outside the object.} \end{cases} \tag{2.6}$$

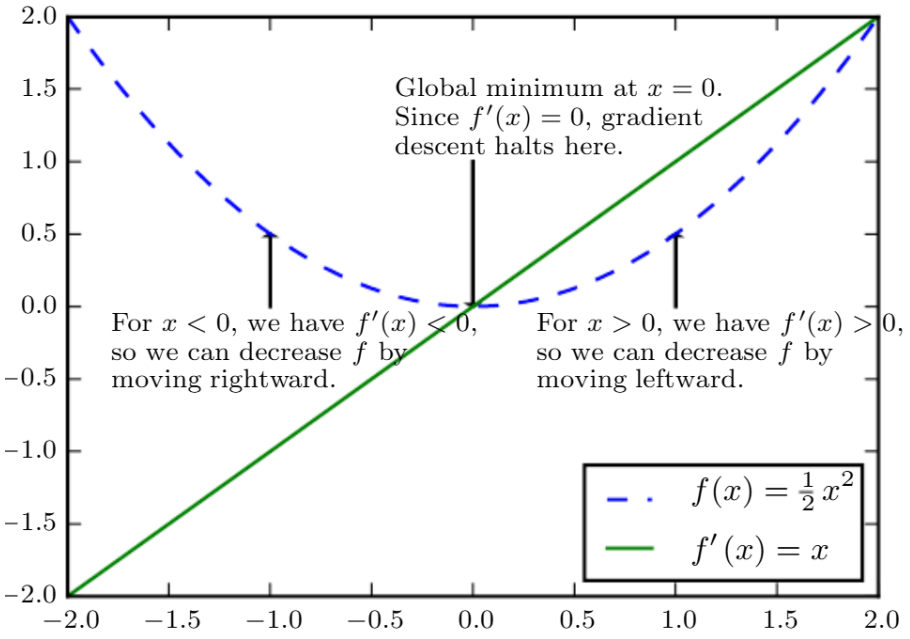

FIGURE 2.19. An illustration of how the gradient descent algorithm finds the minimum of a function (Goodfellow et al., 2016).

It is important to note that this convention can be reversed depending on the definition of the user. In practice, the SDF is often computed from accurate mesh representations of target objects. Tools such as Open3D (Zhou et al., 2018) provide efficient implementations for generating SDFs from meshes. However, the reliance on precise object models limits the applicability of this approach in real-world scenarios, where high-fidelity models may be unavailable or difficult to obtain.

To address this limitation, recent research has explored the use of point cloud completion techniques (discussed in Section 2.2.3) to reconstruct full object models from partial observations. Another promising approach involves training neural networks to approximate the SDF directly from partial observations, bypassing the need for explicit 3D reconstruction (Park et al., 2019). These learning-based methods extend the utility of SDFs in practical grasping applications, especially in unstructured environments.

2.3 Stochastic Gradient Descent

Optimization plays a central role in robotics and machine learning, enabling systems to learn models, plan motions, and refine control strategies. In many applications, we define a performance metric or loss function that quantifies how well a model or policy performs a task. The problem then reduces to finding the parameters that minimize this loss (or equivalently, maximize performance). When the loss

function is differentiable, gradient-based optimization methods are particularly effective because they exploit the function's slope to guide parameter updates. Among these, Gradient Descent (GD) is one of the most widely used algorithms, forming the foundation for many optimization techniques, including its stochastic variant, Stochastic Gradient Descent (SGD).

Given a differentiable function $f(\theta)$, which can be called the objective function or cost function, of parameter θ, the aim is to find θ^* that minimizes $f(\theta)$ (equivalently maximizing $-f(\theta)$). Gradient descent (GD) is an iterative optimization algorithm to determine θ^* as illustrated in Figure 2.19:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta f(\theta_t) \tag{2.7}$$

where θ_t are the parameters at iteration t, η is the learning rate and $\nabla_\theta f(\theta_t)$ is the gradient of the objective function at θ_t.

The gradient provides the steepest direction of increase, and the negative of the gradient gives the direction to find the minimum (Goodfellow et al., 2016). The presence of a local minimum, a point where $f(\theta)$ is lower than all neighboring points, makes the optimization problem difficult, especially for multidimensional input.

Stochastic gradient descent (SGD) (Robbins and Monro, 1951) is a gradient descent algorithm that uses a mini-batch instead of the full data, that uses the first order gradients. The mini-batch of examples is independent identically distributed (iid) and can be used to obtain an unbiased estimate of the gradient. Replacing N, the number of data, with a mini-batch of size $m << N$, we obtain the general SGD update equation:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta_t} f(\theta_t). \tag{2.8}$$

Due to the random sampling of the mini-batch, the gradients of SGD are noisy even when parameter values are close to the optimal solution, which can help escaping local minima.

The learning rate η is an important factor affecting the performance of the optimization process: too large can lead to divergence and too small can lead to slow convergence. Its value may be a constant chosen by trial and error, or decay linearly until iteration τ:

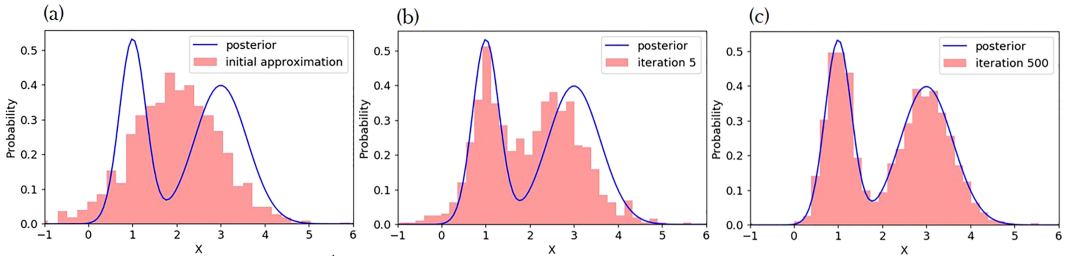

FIGURE 2.20. Illustration of the SVGD algorithm (Zhang and Curtis, 2020). Starting from an initial set of particles approximating the target distribution (a), SVGD iteratively updates these particles by simultaneously attracting them towards regions of high probability and repelling them from each other to prevent collapse to a single mode, as shown in (b) and (c). This process enables efficient exploration of complex, multimodal distributions.

$$\eta_t = (1 - \frac{t}{\tau})\eta_0 + \frac{t}{\tau}\eta_\tau \tag{2.9}$$

The SGD step-size can be set according to adaptive schemes such as ADAM (Kingma and Ba, 2014) and AMSGrad (Reddi et al., 2019). A sufficient condition to guarantee convergence of SGD is that:

$$\sum_{t=1}^{\infty} \eta_t = \infty, \qquad \sum_{t=1}^{\infty} \eta_t^2 < \infty. \tag{2.10}$$

2.4 Stein Variational Gradient Descent

Bayesian and Variational Inference

In Bayesian inference (BI, Bernardo and Smith (1994)), the goal is to estimate a posterior distribution over parameters θ given data $\mathcal{D}$:

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)\, p(\theta)}{p(\mathcal{D})}, \tag{2.11}$$

where $p(\mathcal{D}|\theta)$ is the likelihood, $p(\theta)$ is the prior, and $p(\mathcal{D})$ is the evidence (a normalizing constant).

Directly sampling from $p(\theta|\mathcal{D})$ is often intractable. Variational inference (VI, Jordan et al. (1999)) addresses this by introducing a simpler, parameterized distribution $q(\theta)$ and finding the q that minimizes the Kullback–Leibler (KL, Kullback and Leibler (1951)) divergence to the true posterior:

$$\mathrm{KL}(q \,\|\, p) = \mathbb{E}_{\theta\sim q}\left[\log \frac{q(\theta)}{p(\theta|\mathcal{D})}\right]. \tag{2.12}$$

Classical VI assumes q belongs to a chosen parametric family (e.g., mean-field Gaussians), which can lead to biased approximations if the true posterior lies outside this family.

Stein's Identity and Stein Discrepancy

SVGD (Liu and Wang, 2016) removes the need for a fixed parametric family by representing q non-parametrically using a set of particles $\{\theta^i\}_{i=1}^M$. It relies on Stein's identity (Stein, 1972), which states that for any smooth vector-valued function $\phi(\theta)$ and target density $p(\theta)$:

$$\mathbb{E}_{\theta\sim p}\left[\mathcal{T}_p\phi(\theta)\right] = 0, \quad \mathcal{T}_p\phi(\theta) = \nabla_\theta \log p(\theta)^\top \phi(\theta) + \nabla_\theta \cdot \phi(\theta), \tag{2.13}$$

where $\mathcal{T}_p$ is the Stein operator. This identity holds for all ϕ in a certain function class when $\theta \sim p$.

The Stein discrepancy (Gorham and Mackey, 2015) measures the violation of Stein's identity for a candidate distribution q:

$$\mathbb{D}(q \,\|\, p) = \sup_{\phi\in\mathcal{F}} \mathbb{E}_{\theta\sim q}\left[\mathcal{T}_p\phi(\theta)\right], \tag{2.14}$$

where $\mathcal{F}$ is typically chosen as the unit ball in a reproducing kernel Hilbert space (RKHS). When $\mathbb{D}(q \,\|\, p) = 0$, q matches p.

SVGD can be interpreted as a variational inference method that evolves q along a smooth transport map to reduce $\mathrm{KL}(q \,\|\, p)$. Let $T(\theta) = \theta + \epsilon\phi(\theta)$ be a small perturbation of the identity map. The functional gradient of the KL divergence with respect to ϕ is:

$$\frac{d}{d\epsilon}\mathrm{KL}(q_T \,\|\, p)\bigg|_{\epsilon=0} = -\mathbb{E}_{\theta\sim q}\left[\mathcal{T}_p\phi(\theta)\right]. \tag{2.15}$$

Choosing ϕ to maximize the decrease in KL under the RKHS constraint leads to the optimal update direction:

$$\phi^*(\cdot) = \mathbb{E}_{\theta'\sim q}\left[k(\theta', \cdot)\nabla_{\theta'} \log p(\theta') + \nabla_{\theta'} k(\theta', \cdot)\right], \tag{2.16}$$

where k is a positive-definite kernel.

SVGD Algorithm

With M particles $\{\theta^i\}_{i=1}^M$, SVGD updates each particle as:

$$\theta^i_{k+1} = \theta^i_k + \eta\,\phi(\theta^i_k), \tag{2.17}$$

where η is the step size and ϕ is given by the empirical version of Equation (2.16):

$$\phi(\theta^i) = \frac{1}{M}\sum_{j=1}^{M}\left[k(\theta^j, \theta^i)\nabla_{\theta^j} \log p(\theta^j) + \nabla_{\theta^j} k(\theta^j, \theta^i)\right]. \tag{2.18}$$

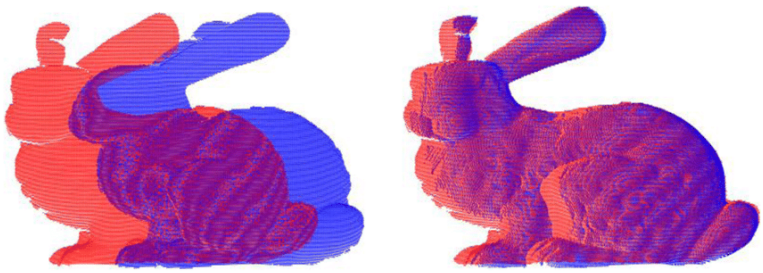

FIGURE 2.21. Illustration of the ICP matching process (Wan et al., 2019).

The kernel term encourages particles to spread out, preserving diversity, while the gradient term pushes particles toward high-probability regions of p. This makes SVGD a non-parametric variational inference algorithm that blends optimization and sampling, capable of approximating complex, multimodal target distributions without assuming a fixed functional form for q. An illustration of SVGD is provided in Figure 2.20.

The most commonly used kernel function $k(\cdot, \cdot)$ (Liu and Wang, 2016) is the Radial Basis Function (RBF), also known as the Gaussian kernel, defined as:

$$k(\theta, \theta') = \exp\left(-\frac{\|\theta - \theta'\|^2}{h}\right), \tag{2.19}$$

where $\|\theta - \theta'\|^2$ is the squared Euclidean distance between two particles, and h is the kernel bandwidth parameter that controls the sensitivity of particle interactions. A smaller h encourages local exploration, whereas a larger h promotes smoother, broader particle interactions. The kernel encourages particles to move collectively towards regions of high probability or low energy. At the same time, it maintains particle diversity, preventing convergence to a single point and thereby facilitating effective exploration of the solution space.

2.5 Iterative Closest Point

Iterative Closest Point (ICP) is a well-established algorithm that determines the rigid transformation, consisting of rotation and translation, required to align a source point cloud with a target point cloud, as illustrated in Figure 2.21. Originally introduced by Besl and McKay (1992), ICP has been applied to areas such as 3D reconstruction, object pose estimation, and robotic grasp planning, where precise geometric

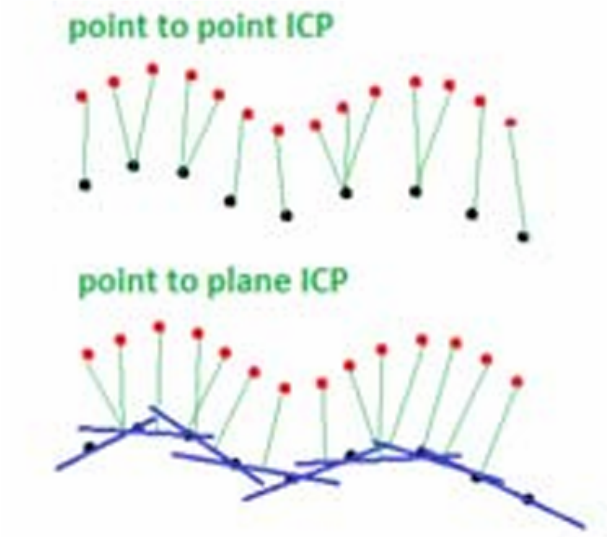

FIGURE 2.22. Illustration of two common metrics for ICP: point-to-point (top) and point-to-plane (bottom) (Będkowski and Masłowski, 2012).

alignment is critical. The algorithm proceeds through the following two matching and minimization steps until convergence is achieved:

(1) *Matching Step:* pairs the transformed source point cloud, $\mathcal{S}' = \{s'_i\}_{i=1}^{N}$, with the reference point cloud $\mathcal{R} = \{r_i\}_{i=1}^{M}$ on the basis of a distance metric, where s_i and $r_i \in R^3$ are N and M points in 3D space. The commonly used point-to-point distance metric finds the pair with the nearest neighbor as follows:

$$\hat{r}_i = \underset{r_j \in \mathcal{R}}{\operatorname{argmin}} \|s'_i - r_j\| \tag{2.20}$$

where $\hat{r}_i$ is the closest point to $s'_i = (Rs_i + t)$, $R \in \mathbb{R}^{3\times3}$ is the rotation matrix, and $t \in \mathbb{R}^{3\times1}$ is the translation vector. Another common metric is the point-to-plane distance, which typically converges faster than point-to-point metrics when the initial alignment is already close to the true transformation. Instead of measuring the Euclidean distance between corresponding points, point-to-plane distance measures the orthogonal projection of the difference vector $(s'_i - \hat{r}_i)$ onto the surface normal $\mathbf{n}_i$ at $\hat{r}_i$:

$$d_{\text{point-to-plane}}(s'_i, \hat{r}_i) = \mathbf{n}_i^{\top}(s'_i - \hat{r}_i), \tag{2.21}$$

where $\mathbf{n}_i \in \mathbb{R}^3$ is the unit normal vector at $\hat{r}_i$. This metric penalizes only the component of the error vector along the surface normal, effectively ignoring tangential discrepancies. As a result, point-to-plane ICP often exhibits improved convergence rates in fine alignment stages, particularly

for smooth surfaces, since it aligns points by reducing their perpendicular deviation from the target surface rather than fully matching their 3D coordinates.

(2) *Minimization Step:* updates transformation parameter θ_k to minimize a loss function defined by the distance between the paired points in the source and reference point clouds. The updated equation for the point-to-point distance metric is defined as follows:

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i}^{N} \|R_k s_i + t_k - \hat{r}_i\|^2 \tag{2.22}$$

where k is the iteration number. Equation (2.22) can be solved in closed-form using either Horn's method (Horn, 1987) or singular value decomposition (SVD) (Arun et al., 1987).

While ICP is effective, it comes with certain limitations. The algorithm is prone to convergence at local minima, particularly if the initial alignment is far from optimal. In scenarios with noise, incomplete data, or significant outliers, correspondences may become unreliable, and the algorithm's performance may degrade. In grasp synthesis applications, ICP can be employed to align a gripper's point cloud from a given grasp pose to object's point cloud. However, there is a need for a good initial pose and an accurate point cloud to achieve reliable performance.

2.5.1 Rotation Representations

Representation	**Parameters**	**Advantages**	**Disadvantages**
Rotation Matrix	9 (3 DOF)	Direct application to vectors; easy to concatenate rotations by matrix multiplication.	Redundant parameters; prone to numerical drift and requires orthonormalization.
Euler Angles	3	Intuitive and easy to visualize; corresponds to roll, pitch, and yaw.	Suffer from gimbal lock; discontinuities in representation.
Axis-Angle	4 (3 for axis + 1 for angle)	Compact representation; suitable for interpolation and understanding rotation about an axis.	Less intuitive than Euler angles; angle-axis ambiguity for zero rotation.
Quaternions	4	Numerically stable; no gimbal lock; efficient interpolation (slerp).	Less intuitive; requires normalization to maintain unit norm.
Rotation Vector	3	Compact and suitable for incremental updates and optimization algorithms.	Needs conversion for practical applications; can be less intuitive.

TABLE 2.3. Common Representations for 3D Rotations.

Rotations in three-dimensional space are elements of the special orthogonal group $\mathrm{SO}(3)$, defined as:

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^\top R = I,\ \det(R) = 1\}. \tag{2.23}$$

Here, R is a rotation matrix that preserves lengths and angles (orthogonality) and maintains orientation (unit determinant). A rotation matrix has nine parameters but only three degrees of freedom, corresponding to the three independent rotational axes in 3D space. While rotation matrices are numerically stable and easy to compose via matrix multiplication, they are overparameterized and not minimal.

A list of common 3D rotation representations is provided in Table 2.3. Euler angles are one such minimal representation, specifying rotations about a sequence of coordinate axes. They are widely used in robotic grasping but suffer from singularities such as gimbal lock, where the loss of one degree of freedom causes ambiguity in rotation representation.

To avoid such issues, this thesis adopts quaternions for rotation representation. Quaternions are four-dimensional unit vectors (q_w, q_x, q_y, q_z) that provide a continuous, singularity-free representation of 3D orientation, with only one unit-norm constraint:

$$\|q\| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} = 1. \tag{2.24}$$

In our parameterization, a pose is represented as:

$$\theta = \{x, y, z, q_w, q_x, q_y, q_z\}, \tag{2.25}$$

where (x, y, z) denotes the 3D translation, and (q_w, q_x, q_y, q_z) is the unit quaternion corresponding to the rotation.

2.5.2 Stochastic Gradient Descent ICP

Stochastic Gradient Descent ICP (SGD-ICP) (Maken et al., 2019, 2022b) solves the optimization problem of ICP (2.22) with stochastic gradient descent (SGD) (Section 2.3). It samples a mini-batch $\mathcal{M}$ of m points from the source cloud, and computes the gradient of Equation (2.22) to update θ in the following equation:

$$\theta_{k+1} = \theta_k - \eta A\bar{g}(\theta_k, \mathcal{M}_k) \tag{2.26}$$

where η is the learning rate, $A \in R^{7\times7}$ acts as a pre-conditioner for the gradients, and $\bar{g}(\theta_k, \mathcal{M}_k)$ are the average gradients which are computed as follows:

$$\bar{g}(\theta_k^{1:3}, \mathcal{M}_k) = \frac{1}{m}\sum_{i}^{m}((R_k s_i + t_k) - \hat{r}_i)\frac{\partial t_k}{\partial \theta_k^{1:3}} \tag{2.27}$$

$$\bar{g}(\theta_k^{4:7}, \mathcal{M}_k) = \frac{1}{m}\sum_{i}^{m}((R_k s_i + t_k) - \hat{r}_i)\frac{\partial R_k}{\partial \theta_k^{4:7}} s_i \tag{2.28}$$

where $\bar{g}(\theta_k^{1:3}, \mathcal{M}_k)$ and $\bar{g}(\theta_k^{4:7}, \mathcal{M}_k)$ are the gradients of the cost function w.r.t translations and rotations respectively.

2.5.3 Stein ICP

Stein ICP (Maken et al., 2022a) uses SGD-ICP gradients in the SVGD framework (Section 2.4). Stein ICP approximates an intractable but differentiable posterior distribution of transformation parameters $p(\theta)$ by constructing a non-parametric variational distribution represented by a set of K particles $\{\theta_j\}_{j=0}^{K}$. The SVGD algorithm (Liu and Wang, 2016) provides the optimal update direction for particles:

$$\phi^*(\cdot) = \mathbb{E}_{\theta\sim q}[\nabla_\theta \log p(\theta) k(\theta, \cdot) + \nabla_\theta k(\theta, \cdot)] \tag{2.29}$$

In practice, the update rule for particles is given by:

$$\theta_{k+1} = \theta_k + \eta\hat{\phi}^*(\theta_k). \tag{2.30}$$

In Stein ICP, the gradients of the log of posterior $\nabla \log p(\theta)$ in Equation (2.29) are replaced by the gradients of the log of the likelihood function ($\bar{g}(\theta_j, \mathcal{M})$) from SGD-ICP((2.27), (2.28)) and the gradient of the log of priors ($\nabla_\theta \log p(\theta_j)$) as follows:

$$\begin{aligned}\hat{\phi}^*(\theta) = \sum_{k=1}^{K}[&- (N\bar{g}(\theta_k, \mathcal{M}) + \nabla_\theta \log p(\theta_k))k(\theta_k, \theta) \\ &+ \nabla_\theta k(\theta_k, \theta)],\end{aligned} \tag{2.31}$$

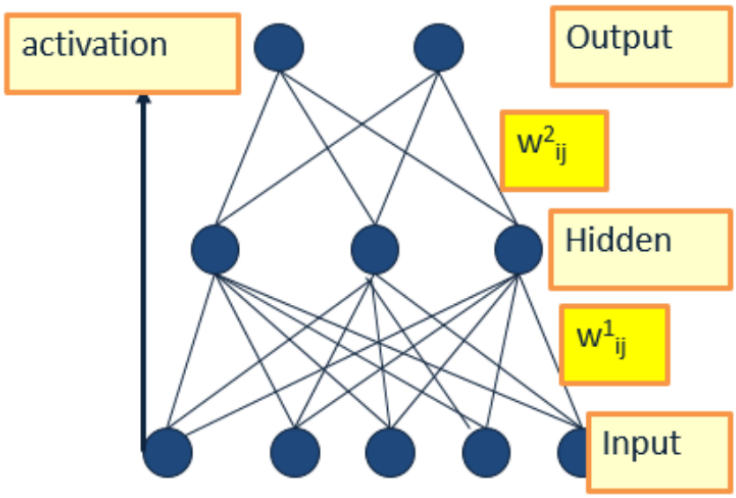

FIGURE 2.23. Illustration of a basic neural network unit consisting of input, hidden, and output layers connected by weighted edges (Roth, 2016).

where $\nabla \log p(\theta)$ is represented by gradients from Gaussian priors for translations and von Mises priors for rotations.

The first gradient term in Equation (2.31), weighted by a kernel function, determines the steepest direction for the log probability. Conversely, the second term represents the gradient of the kernel function, acting as a repulsive force that promotes dispersion among particles and prevents them from converging to local modes of the log probability.

2.6 Neural Network

2.6.1 Network Structure

Neural networks are computational models composed of interconnected layers of units called neurons. These networks identify patterns in data to perform tasks such as classification, prediction, and generation. The neural network workflow typically consists of two main stages: training and inference.

During training, the network learns a mapping from input data to desired outputs using a dataset that may be labeled (supervised learning, LeCun et al. (2015)), partially labeled (semi-supervised learning, Zhu (2005)), or unlabeled (unsupervised or self-supervised learning, LeCun et al. (2015)). It generates predictions from the input data, evaluates them using a task-specific loss or objective function, and iteratively adjusts its internal parameters to minimize this objective. The parameter updates are computed using

backpropagation, an algorithm that applies the chain rule of calculus to efficiently calculate gradients of the loss with respect to each parameter in the network. These gradients are then used by an optimization algorithm, such as gradient descent, to refine the parameters. During inference, the trained network applies the learned mapping to new, unseen data to produce predictions or decisions without further parameter updates.

A multilayer perceptron (MLP) is a classical feedforward neural network composed of multiple layers of neurons, where each neuron in one layer is fully connected to every neuron in the next. Figure 2.23 illustrates a basic MLP unit. The input layer of an MLP receives a vector representation of the data, which may consist of raw features (e.g., sensor readings), processed features extracted through preprocessing or other models, or a combination of both, depending on the application.:

$$\mathbf{x} = [x_1, x_2, \ldots, x_D]^T,$$

where D denotes the number of features.

The input propagates through one or more hidden layers. For the l-th hidden layer containing n_l neurons, the pre-activation input to neuron j is computed as:

$$z_j^{(l)} = \sum_{i=1}^{n_{l-1}} w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)},$$

where:

- $w_{ij}^{(l)}$ is the weight connecting neuron i in layer $l-1$ to neuron j in layer l,
- $a_i^{(l-1)}$ is the activation of neuron i in the previous layer (with $a_i^{(0)} = x_i$ for the input layer),
- $b_j^{(l)}$ is the bias term of neuron j.

The neuron applies a nonlinear activation function σ to produce its output:

$$a_j^{(l)} = \sigma(z_j^{(l)}).$$

Common activation functions include:

- **ReLU (Rectified Linear Unit):**

$$\sigma(z) = \max(0, z),$$

- **Sigmoid:**

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

These nonlinearities enable the network to capture complex, nonlinear relationships in data.

The activations from the final hidden layer feed into the output layer, which produces the network's predictions.

Backpropagation computes how much each network weight w_{ij}, connecting neuron i in one layer to neuron j in the next, contributes to the overall loss E. It does so by repeatedly applying the chain rule from the output layer back toward the input layer. The gradient of the loss with respect to w_{ij} is given by

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial a_j} \cdot \sigma'(z_j) \cdot a_i,$$

where a_i is the activation of neuron i in the previous layer, $z_j = \sum_i w_{ij} a_i + b_j$ is the pre-activation input to neuron j, $\sigma'(z_j)$ is the derivative of the activation function at z_j, and $\frac{\partial E}{\partial a_j}$ is the error signal for neuron j.

For hidden layers, the error signal $\frac{\partial E}{\partial a_j}$ is computed recursively from all neurons l in the next layer that receive connections from neuron j:

$$\frac{\partial E}{\partial a_j} = \sum_{l \in L} \frac{\partial E}{\partial a_l} \cdot \sigma'(z_l) \cdot w_{jl},$$

where L denotes the set of neurons in the next layer connected to j.

Once the gradients are computed, weights are updated using gradient descent:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \cdot \frac{\partial E}{\partial w_{ij}^{(l)}},$$

where η is the learning rate, which controls the size of each update step. Intuitively, backpropagation first computes the error at the output, then propagates this information backward through the network, and finally adjusts each weight slightly in the direction that reduces the overall error.

Other widely used neural network architectures include Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). For a comprehensive overview, readers are referred to LeCun et al. (2015).

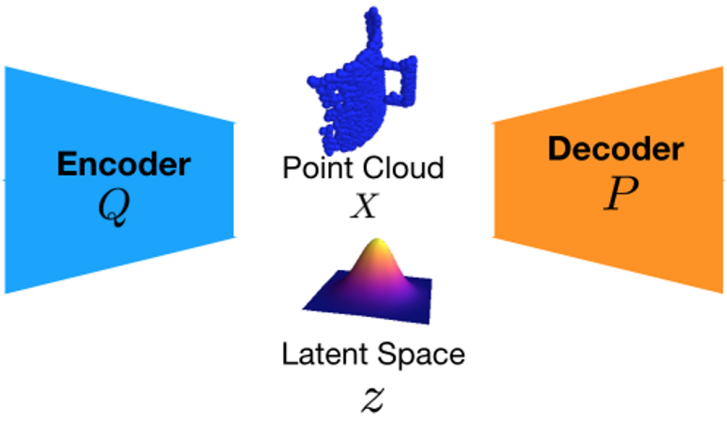

FIGURE 2.24. Illustration of a point cloud encoder–decoder architecture with a latent space representation (Mousavian et al., 2019). The encoder Q maps the input point cloud X—an unordered set of 3D points capturing object geometry—into a compact latent vector z in a continuous latent space. This latent representation captures the essential geometric features of the object while being invariant to point order, enabling efficient processing for downstream tasks such as recognition, segmentation, and grasp synthesis. The decoder P reconstructs the point cloud from z.

2.6.2 Encoders and Autoencoders

An encoder is a neural network module that transforms input data into a compact, informative representation, often called a latent vector or embedding. This encoding captures essential features while reducing dimensionality, enabling efficient processing for subsequent tasks.

An autoencoder is a neural network designed to learn such encodings in an unsupervised manner. It consists of two components:

- **Encoder** f_θ: Maps input $\mathbf{x}$ to latent representation $\mathbf{z}$:

$$\mathbf{z} = f_\theta(\mathbf{x}), \quad \mathbf{z} \in \mathbb{R}^d, \quad d < \dim(\mathbf{x}),$$

  where θ represents encoder parameters.

- **Decoder** g_ϕ: Attempts to reconstruct $\mathbf{x}$ from $\mathbf{z}$:

$$\hat{\mathbf{x}} = g_\phi(\mathbf{z}) = g_\phi(f_\theta(\mathbf{x})),$$

  where ϕ represents decoder parameters.

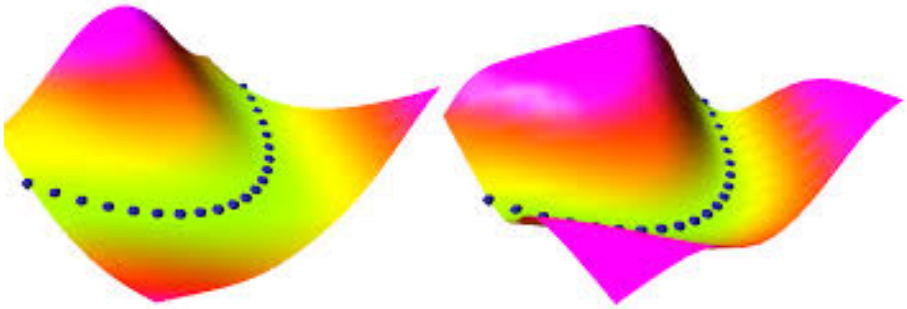

FIGURE 2.25. Energy landscapes in the (X, Y) space learned by two neural networks modeling the function $Y = X^2 - \frac{1}{2}$ (LeCun and Huang, 2005). The left plot shows a quadratic energy surface (smooth, convex paraboloid), while the right plot illustrates a non-quadratic, saturated energy surface with flat regions. In both cases, the color represents the magnitude of the energy, with warmer colors (yellow–pink) indicating higher energy (less plausible states) and cooler colors (green) indicating lower energy (more plausible states). Blue dots indicate points sampled from the target function.

In this thesis, the encoder is used to transform high-dimensional, unordered point cloud data into a structured and compact latent representation. The latent space representation is not only more compact but also facilitates learning by imposing an order-invariant and semantically meaningful structure on otherwise unstructured 3D data. The latent vector captures the core geometric information of the object without storing every single 3D point. This compact form allows the network to compare, manipulate, and reconstruct objects efficiently.

The encoder processes raw 3D coordinates through multiple layers inspired by architectures such as PointNet (Qi et al., 2017) and graph neural networks (Kipf and Welling, 2016), which remain permutation invariant and capture both local and global geometric features. This produces a fixed-length latent embedding summarizing the object's shape and spatial configuration for further processing.

2.6.3 Energy-Based Models

Energy-Based Models (EBMs) form a general framework for representing complex probability distributions by associating each possible input x with a scalar energy value $E_\theta(x)$. The key principle is that lower energy corresponds to more plausible or desirable configurations, while higher energy indicates less likely configurations.

This is formalized by defining the probability density function:

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z_\theta}, \tag{2.32}$$

where $Z_\theta = \int \exp(-E_\theta(x))\, dx$ is the partition function, ensuring the distribution is normalized.

A quadratic energy function, such as $E(x) = \frac{1}{2}\|x - \mu\|^2$, produces a convex, bowl-shaped energy landscape (Figure 2.25, left). Such landscapes are smooth, with a single global minimum, making optimization straightforward. However, in real-world problems—especially in robotic grasping—energy landscapes are rarely this simple. Non-quadratic energy functions (Figure 2.25, right) can have flat regions, sharp ridges, or multiple local minima, reflecting the complexity of real grasp success probabilities under diverse object shapes and poses. These landscapes require more sophisticated optimization methods to avoid local minima and capture the multimodal nature of the problem.

Inference corresponds to finding the configuration x with the lowest energy:

$$\hat{x} = \arg\min_x E_\theta(x). \tag{2.33}$$

In the context of grasping, x could represent a grasp pose (position + orientation), and the EBM is trained so that physically stable, collision-free grasps have lower energy than unstable or infeasible ones.

A common approach for training EBMs is contrastive learning, where the model learns to assign:

- Low energy to positive samples (successful grasps)
- High energy to negative samples (unsuccessful grasps)

This is typically implemented by minimizing the difference in energy between positive and negative examples, often using a margin-based loss:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, m + E_\theta(x_i^+) - E_\theta(x_i^-)\right), \tag{2.34}$$

where x_i^+ and x_i^- are positive and negative samples, respectively, and m is the margin enforcing separation in the energy space.

2.7 Summary

This chapter established the essential theoretical background required to understand and implement the grasp synthesis techniques developed in this thesis. It highlighted the diverse approaches to robotic manipulation and grasping, underscoring the motivation for a hybrid methodology that synergizes analytical rigor with data-driven flexibility. The optimization foundations laid out by stochastic gradient descent and SVGD provide the computational framework for our integrated approach. Together, these topics form a cohesive base that supports the thesis's novel contributions in grasp synthesis under uncertainty and partial observations.

For readers seeking more comprehensive insights on robotic manipulation, we recommend Mason (2001) and Mason (2018), which offer in-depth discussions on manipulation theory and applications. The mathematical foundations of gradient descent are well covered by Goodfellow et al. (2016). For a thorough treatment of Bayesian learning principles, see the book by Bishop (2006). Finally, Dawid and LeCun (2023) presents both the theoretical background and practical considerations for EBMs, including common training loss functions.

CHAPTER 3

Grasping as Rigid Shape Matching

3.1 Introduction

Grasp synthesis has long been a central research challenge in robotics, with two primary methodologies emerging: analytic and data-driven approaches. Analytic methods rely on precise models of both the gripper and the target object to compute optimal grasp positions. These optimal grasps are typically validated via simulations that account for factors such as friction, force closure, and form closure. However, the inherent discrepancies between these mathematical models and the complexities of the real world often lead to performance degradation when transitioning to physical robots. Additionally, optimizing high-dimensional configurations for multi-fingered hands is computationally intensive, which hinders real-time application.

In contrast, data-driven approaches have gained significant attraction due to their higher success rates and faster pose generation times. By learning from large datasets—whether collected from real-world trials or generated in simulation—these methods predict grasp quality without requiring complete knowledge of underlying physics or geometry. Yet, a notable challenge for data-driven techniques is generalization: models trained on specific datasets often struggle when faced with objects that significantly differ from those encountered during training.

With the recent improvements in computational power, research has increasingly shifted towards data-driven methods. Nevertheless, there remains a strong interest in revisiting and enhancing analytic approaches to overcome their limitations. In this chapter, we propose a novel formulation of grasp synthesis that treats the problem as a rigid shape matching task between the gripper's inner surface and the object's surface. Unlike traditional methods that optimize both the palm pose and the finger joint angles, our

approach fixes the gripper shape during optimization. This simplification not only reduces computational complexity but also enables collision checking through the gripper's SDF, thereby reducing the dependency on an accurate object model.

Our contributions are as follows:

(1) We formulate the grasp synthesis problem as a rigid shape matching task using the point clouds of both the object and the gripper.
(2) We leverage the gripper's SDF for efficient collision avoidance, removing the reliance on highly accurate object models.
(3) We implement a parallel Stochastic Gradient Descent-based ICP (SGD-ICP) algorithm to accelerate the optimization process, achieving significant speed improvements over existing methods.

3.2 Related Work

3.2.1 Analytic Approaches

The analytic approach relies on object and gripper models and selects the grasp pose based on various grasp quality metrics (Sahbani et al., 2012). The Contact Point Optimization (CPO) method (Fan et al., 2018b) combines Palm Pose Optimization (PPO) with the objective of generating grasp poses for a three-fingered hand based on a given pose of a parallel gripper. In Kiatos et al. (2021), this dual optimization technique is applied to generate grasp poses for objects in cluttered environments using a three-fingered hand. Iterative Surface Fitting (ISF) (Fan et al., 2018a) has been employed for grasping with customized parallel grippers. The algorithm is generalized to multi-fingered hands for both precision and power grasps in Fan et al. (2019). Additionally, the combination of multidimensional iterative surface fitting (MDISF) and grasp trajectory optimization (GTO) (Fan and Tomizuka, 2019) has been employed to generate optimal grasps using three-fingered hands. Optimizing in high-dimensional spaces for multi-fingered hands can be a time-intensive process. In this chapter, we simplify this by bypassing the need to optimize finger joint angles, resulting in a significant reduction of the time required to find a suitable grasp pose.

3.2.2 Data-Driven Approaches

The data-driven approach relies on large labeled datasets and requires significant training time. However, it achieves higher success rates, particularly with objects similar to those used during the training. In recent studies, both Gao et al. (2022) and Wang et al. (2022) have employed this approach by generating heatmaps of given objects and subsequently sampling grasp poses in the form of grasp rectangles. Gao et al. (2022) proposes a Residual Hourglass Network, achieving a remarkable success rate of 97.8% on the Cornell dataset, while Wang et al. (2022) utilizes a transformer model and achieves a success rate of 97.99% on the same dataset. It is important to note that these methods are limited to parallel grippers and planar grasps. To overcome these limitations, Multilevel Convolutional Neural Networks have been proposed to grasp objects using multifingered hands, utilizing grasp rectangles. This approach achieves a success rate of 95.82% for 16 objects (Yu et al., 2020). However, it should be noted that the success rate for regular-shaped objects is significantly higher compared to objects with irregular shapes.

Most of these existing methods face challenges in terms of generalizing to objects that are not present in the training datasets. In contrast, we eliminate the need for training and demonstrate the enhanced ability to generalize to unfamiliar objects, which can be applied alongside these methods to enhance grasp quality.

3.2.3 Grasp Quality Metrics

Dexterity, equilibrium, stability, and dynamic behavior are fundamental characteristics of grasp synthesis (Shimoga, 1996). These characteristics are formulated as grasp quality metrics, categorized into two main groups: contact points and hand configuration, further divided into 24 different quality indices (Roa and Suárez, 2015). To guide the selection and combination of metrics, a comprehensive evaluation of the variability, correlation, and sensitivity of ten selected metrics has been conducted Rubert et al. (2018). A correlation study involving 16 metrics has been carried out for multi-fingered hands (Mnyussiwalla et al., 2022). Based on this study, the authors selected force closure, area of the grasp polygon, distance to the object's center of mass, and configuration of the finger joints as criteria for in-hand manipulation planning. Our approach utilizes predefined configurations of the gripper as input shown in Figure 3.1(c). These configurations are then optimized to minimize the matching error against the object. Our grasping cost function is comprised of both the distance to the object's centre of mass and the matching error, which we describe in Section 3.3.

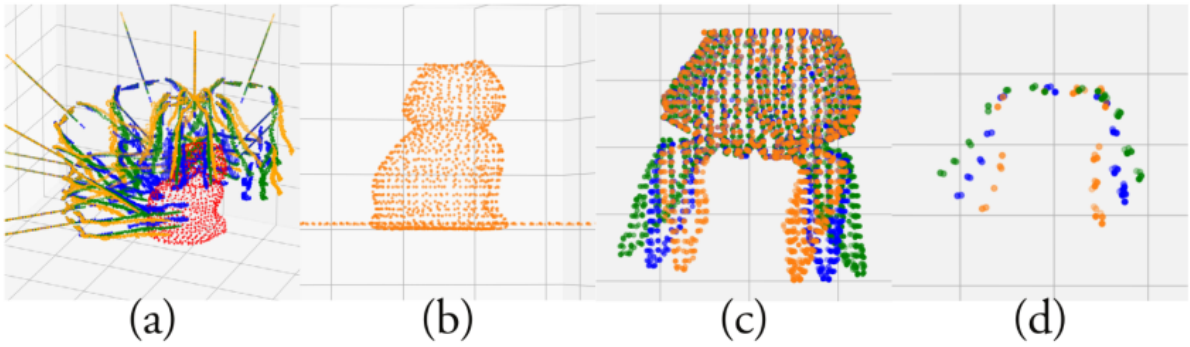

FIGURE 3.1. a) An illustration of uniformly distributed 48 initial θ_0. b) Point cloud $\mathcal{C}$, a combination of table's and object's point clouds. c) 3 different configurations used for KG3 gripper. d) Gripper contact surface's point clouds $\mathcal{S}$ of three configurations.

3.3 Methodology

In this chapter, we investigate grasp synthesis for unknown objects while avoiding the expensive optimization of finger joints. Our algorithm takes as input a gripper's inner surface point cloud $\mathcal{S}$ (Figure 3.1(d)), an object's 3-D point cloud $\mathcal{R}$ for matching, a complete gripper's point cloud $\mathcal{G}$ (Figure 3.1(c)) for generating SDF and object's point cloud with the table's point cloud $\mathcal{C}$ (Figure 3.1(b)) for collision checking. Specifically, we utilize point clouds of different finger configurations as fixed inputs, which remain unchanged during the optimization process. The algorithm's objective is to determine an optimal grasp pose parameterized by $\theta = \{t, q\}$, where $t = \{x, y, z\}$ represents the 3D translation and $q = \{q_w, q_x, q_y, q_z\}$ represents the unit quaternion, with q_w being the scalar and $\{q_x, q_y, q_z\}$ the vector part of the quaternion. The grasp pose is optimized using parallel SGD-ICP with the following constrained loss:

$$\begin{aligned} \min_{\theta_i} \quad & \mathcal{L}(\mathcal{R}, T_\theta(\mathcal{S})) \\ s.t. \quad & \text{dist}(\mathcal{C}, \text{SDF}(T_\theta(\mathcal{G}))) > 0, \end{aligned} \tag{3.1}$$

where $T_\theta(*)$ is the transformed point cloud w.r.t. θ.

To ensure a physically plausible grasp pose, the loss function is constrained by the distance between the object's point cloud and the gripper's SDF. A collision occurs when the point cloud of the gripper is inside the physically infeasible space of the object's point cloud, which corresponds to a negative distance value from the SDF.

The loss function, $\mathcal{L}$, is defined as a weighted combination of two grasp quality metrics $\mathcal{L}_{com}(\mathcal{R}, T(\mathcal{S}))$ and $\mathcal{L}_{ct}(\mathcal{R}, T(\mathcal{S}))$ in the following equation:

$$\mathcal{L} = 0.03\mathcal{L}_{com} + 0.97\mathcal{L}_{ct}, \tag{3.2}$$

These metrics are explained in the next Section 3.3.2. The highly unbalanced weights were chosen for a specific, functional purpose within this initial Stochastic Gradient Descent (SGD) optimization framework. The center-of-mass cost ($\mathcal{L}_{com}$) alone tends to drive all candidate poses toward a single, non-contact point at the object's centroid, causing premature optimization collapse. To counteract this and anchor the optimization toward physically realistic surface interactions from the outset, the contact cost ($\mathcal{L}_{ct}$) is heavily prioritized. The specific ratio of 0.97:0.03 was determined empirically in our simulation environment to achieve this balance. We acknowledge that this ratio is a tunable hyperparameter and its optimal value may vary for different grippers, object sets, or physical environments. This weighted formulation serves as a pragmatic baseline to validate the core shape-matching algorithm. In Chapter 4, this explicit weighting is eliminated by adopting the SVGD framework, whose intrinsic repulsive properties naturally maintain pose diversity without requiring manually balanced costs.

3.3.1 Inputs and Pre-processing

The inputs to our algorithm are point clouds of both the gripper and the object. Unlike other methods mentioned in related work, which involve the expensive optimization of finger joint angles, we utilize point clouds of different finger configurations as fixed inputs, which remain unchanged during the optimization process.

We extract the following geometrical properties from the given input clouds:

- SDF of each gripper configuration for collision checking.
- Approximate centre of mass of the object to minimize the moment while holding the object.

3.3.2 Cost Functions

Unlike traditional ICP problems, where source and reference point clouds belong to the same entity with variations possibly arising from partial observations, point clouds of the gripper and the object are likely to be quite distinct. The proposed algorithm aims to approximate shape matching by maximising the selected grasp quality metrics, rather than achieving a perfect match.

These grasp quality metrics are selected to achieve two main objectives: minimize the matching error to estimate the grasp pose ($\mathcal{L}_{ct}$) and minimize the moment at contact points to ensure stable grasp ($\mathcal{L}_{com}$):

- $\mathcal{L}_{ct}$ minimizes the distance between object's and gripper's point clouds to estimate the grasp pose. Pairing of points with zero distance can be considered as contact points of grasping. Thus, minimizing the least square distance can be considered as maximizing the contact surface. For pairs of points $Pairs = \{s'_i, \hat{r}_i\}_{i=0}^{m}$ from $\mathcal{S}$ and a mini-batch of size m from $\mathcal{R}$, we use the following SGD-ICP point-to-point distance metric:

$$\mathcal{L}_{ct} = \frac{1}{m} \sum_{s'_i, \hat{r}_i \in \text{Pairs}} (\|s'_i - \hat{r}_i\|^2) \tag{3.3}$$

- $\mathcal{L}_{com}$ minimizes the distance between gripper's tool centre point (TCP), which is defined as the center of gripper's contact surface point cloud, and the object's centre of mass (CoM), defined as follows:

$$\mathcal{L}_{com} = \min_{\theta} \|(R \cdot TCP + t) - CoM\|^2. \tag{3.4}$$

3.3.3 Gradients of the Grasp Cost Function

Different to original SGD-ICP (Section 2.5) that uses Euler angle representation for rotations, we use quaternions. Using the loss function $\mathcal{L}$, we get the following average gradients at iteration k:

$$\begin{aligned} \bar{g}(\theta_k^{1:3}, \mathcal{S}_k) = \frac{1}{m} \Big\{ & \sum_{s'_i, \hat{r}_i \in \text{Pairs}} (s'_i - \hat{r}_i) \frac{\partial t_k}{\partial \theta_k^{1:3}} \\ & + ((R_k TCP + t_k) - CoM) \frac{\partial t_k}{\partial \theta_k^{1:3}} \Big\}, \end{aligned} \tag{3.5}$$

for translation components and

$$\begin{aligned} \bar{g}(\theta_k^{4:7}, \mathcal{S}_k) = \frac{1}{m} \Big\{ & \sum_{s'_i, \hat{r}_i \in \text{Pairs}} (s'_i - \hat{r}_i) \frac{\partial R_k}{\partial \theta_k^{4:7}} s_i \\ & + ((R_k TCP + t_k) - CoM) \frac{\partial R_k}{\partial \theta_k^{4:7}} TCP \Big\}, \end{aligned} \tag{3.6}$$

for the rotation parameters.

3.3.4 Collision Checking

The ICP algorithm does not take collision checking into account during the optimization process. However, in the context of grasping tasks involving different shapes of gripper and object, collision avoidance becomes crucial. Simple ICP matching may lead to the gripper being partly inside the object, resulting in a physically infeasible grasp pose. The objective is to efficiently move the gripper out of the object's inner part in case of a collision while simultaneously running SGD-ICP samples, which yield non-collision results in parallel for an efficient grasp pose update.

We perform collision checking using the SDF of the gripper rather than the object's SDF or normal. This design choice is motivated by the asymmetry and occlusion in point cloud: the gripper's geometry is fully known and can be sampled uniformly to create a complete SDF. In contrast, the object's observed point cloud is inevitably incomplete due to several sources of occlusion: the object rests on a support surface may have self-occluding geometry, can be partially blocked by environmental clutter, and is typically observed from a limited sensor viewpoint. These factors lead to significant missing regions, making any object-centric SDF or surface normal estimates derived from the scan unreliable for precise collision evaluation. Therefore, using the gripper's SDF provides a more robust reference frame.

If the gripper tends to grasp the object from physically unreachable parts of the object, i.e. in collision with the object, we will get a negative distance value from SDF in Equation (3.1). Since we use the SDF of the gripper instead of the object which uses points in $\mathcal{C}$ to compute distance, our method finds the closest matching points from $r_{col} \in \mathcal{C}$ to $\mathcal{S}'$, where r_{col} are the points colliding with the gripper. This is in contrast to finding matches from $\mathcal{S}$ to $\mathcal{R}$ as is done in standard ICP. The loss function in Equation (3.2) will be replaced by the distance between r_{col} and their nearest points on the gripper's contact surface s', as given below:

$$\begin{aligned} \mathcal{L}_{col} &= \frac{1}{N_{col}} \sum_{r_{col} \in \mathcal{C}} \min_{s' \in \mathcal{S}} \|r_{col} - s'\|^2, \\ s.t. \quad & \operatorname{dist}(\mathcal{C}, \operatorname{SDF}(T(\mathcal{G}_i))) < 0, \end{aligned} \tag{3.7}$$

where N_{col} is the number of colliding points in object point cloud. In Equation (3.7), we do not account for $\mathcal{L}_{com}$ as the gripper needs to move away from the object's CoM. If the z component of gripper's pose is lower than $5\,\mathrm{mm}$, which could result in a collision with the table, it is moved up by $1\,\mathrm{mm}$ in each subsequent iteration.

3.3.5 Parallel Implementation

Algorithm 3.1: Parallel Shape Matching

Input: Gripper's point cloud $\mathcal{S} = \{s_i\}_{i=1}^{N}$ as the source point cloud, object's point cloud $\mathcal{R} = \{r_i\}_{i=1}^{M}$ as the reference point cloud, complete gripper's point cloud $\mathcal{G}$, multiple initial values of transformation parameters $\Theta_0 = \{\theta_0^j\}_{j=1}^{J}$ for different gripper's configurations, mini-batch size m, step size η = 1, maximum iteration count k_{max}, and object-table point cloud $\mathcal{C}$.

Output: θ that minimizes the loss function

$CoM \leftarrow$ Compute center of mass of $\mathcal{R}$
$SDF \leftarrow$ Compute SDF of the $\mathcal{G}$
while *not converged or* $k \leq k_{max}$ **do**
  for *each* $\theta_k^j \in \Theta_k$ *in parallel* **do**
    $\mathcal{S}_k^j \leftarrow$ Transform the source cloud with θ_k^j
    $\mathcal{M}_k \leftarrow$ Select a mini-batch of size m from the reference cloud
    Pairs $\leftarrow \emptyset$
    for $s_i^j \in \mathcal{S}_k^j$ **do**
      $\hat{r}_i \leftarrow$ closest points in $\mathcal{M}_k$ to s_i^j
      Pairs $\leftarrow$ Pairs $\cup \quad \{s_i^j, \hat{r}_i\}$
    end
    $\mathcal{L}_k^j \leftarrow$ Compute contact loss using (3.2)
    $\bar{g}_k^j \leftarrow$ Compute gradients using (3.5) and (3.6)
    $\rho = \frac{|\mathcal{L}_k^j - \mathcal{L}_{k-1}^j|}{\mathcal{L}_{k-1}^j}$
    if $\mathrm{dist}(\mathcal{C}, \mathrm{SDF}(T_{\theta_k^j}(\mathcal{G}))) < 0$ **then**
      $\bar{g}_{col}^j \leftarrow$ Compute gradients using (2.27) and (2.28)
      $\bar{g}_k^j = \bar{g}_{col}^j \leftarrow$ Replace $\bar{g}_k^j$ with $\bar{g}_{col}^j$
      // check if there is collision between the gripper and the table
      if $min_z(T_{\theta^j}(\mathcal{G})) < 0.005$ **then**
        $\theta_z^j + 0.001$
      end
      // check convergence
      if $\rho \leq 0.01$ **then**
        $\rho = 1$
      end
    end
    if $\rho \geq 0.01$ **then**
      $\theta_{k+1}^j \leftarrow$ Update θ_k^j with $\bar{g}_k^j$ in (2.26)
    end
  end

  k = k +1
end
return $\theta = \mathrm{argmin}_{\theta^j} \mathcal{L}$

A summary of the Parallel Shape Matching based grasp method is provided in Algorithm 3.1. Given $\mathcal{R}$ and $\mathcal{G}$, the SDF of each configuration of the gripper ($\mathcal{G}$) and the center of mass of the object ($\mathcal{R}$) are computed in lines 1 and 2. To enable parallel implementation, we combine the SDF of all configurations into one $\mathcal{SDF} = \{SDF_i\}$ by adding a constant offset ϵ between them, which can be considered as spreading them in space. From line 4 to 27, all operations are performed in parallel for all initializations by combining the point clouds and parameters into multidimensional tensors $\mathcal{S}$, $\mathcal{M}$, $\mathcal{C}$ and θ. In line 5, the source point cloud is transformed with the J transformation parameters. Then, a mini-batch is sampled from the reference cloud in line 6. Next, in lines 7 to 11, the corresponding closest points from the mini-batch are sought and stored in pairs for all points within each transformed source cloud. This is followed by computing the loss, gradients, and the relative error difference in lines 12, 13, and 14 respectively. In line 15, the collision condition is checked. Instead of transforming SDFs of the gripper, we perform an inverse transformation on the object point cloud for each initialization, denoted as $\mathcal{R}' = \{r'_i\}$. These point clouds are then divided into different groups according to the SDF_i they are matching with. Then the same ϵ of the corresponding SDF_i is added to r'. In case of a collision, $\bar{g}_{col}$ is computed in line 16 which then replaces the gradients in line 17. If the gripper is very close to the table surface, it is moved upwards in lines 18 to 20. If the relative error difference indicates convergence, it is set to a non-converged state, in lines 21 to 23, in order to move the gripper out of the object. In line 26, θ is updated with gradient $\bar{g}$ until convergence is achieved.

3.4 Experiments

3.4.1 Simulation

Our study begins with a comprehensive examination of the success rate and efficiency of our proposed method, employing two distinct grippers—namely, the Frank parallel gripper and the Kinova Gripper (KG3) within the Isaac Gym simulator (Makoviychuk et al., 2021). Configurations encompass three variations for KG3 and seven for Franka, strategically initialized to cover half of a semi-sphere to accommodate the robot arm base position. The gripper's point cloud undergoes downsampling with a voxel size of 0.025, while the object's point cloud is initially downsampled to a range of 1100 to 1900 points for collision checking. A subsequent downsampling step of 0.005 is undertaken for matching purposes.

The mini-batch size dynamically adjusts with the iteration count k, equating to the total number of points in the object's point cloud ($N_{\mathcal{R}}$) at the 20th iteration, i.e. $m = \frac{min(k,20)}{20} N_{\mathcal{R}}$. Batch size augmentation

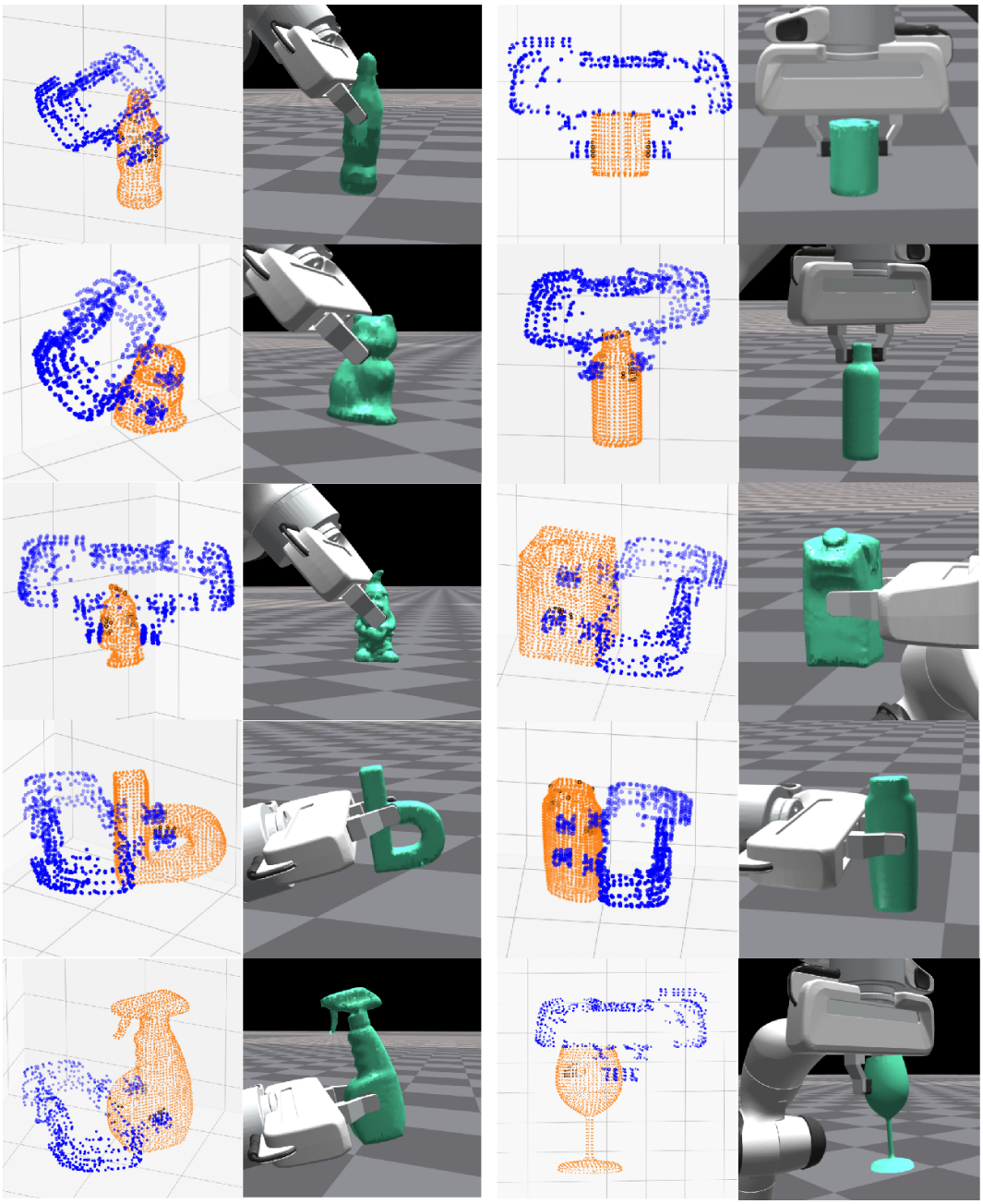

FIGURE 3.2. A visualization of grasp simulations for objects from the KIT database with the Franka hand. For each set, the one on the left is the plot of grasp generated, and the one on the right is the corresponding simulation result.

occurs as the gripper approaches objects, enhancing detail for effective matching. Learning rate, set to 1, and the cost weights remain constant during optimization. SDF computation is executed using Open3D, with a convergence criterion of $\rho <= 0.01$. Initial poses undergo parallel optimization for 50 iterations, with the converged θ exhibiting the minimum cost selected as the final grasp pose.

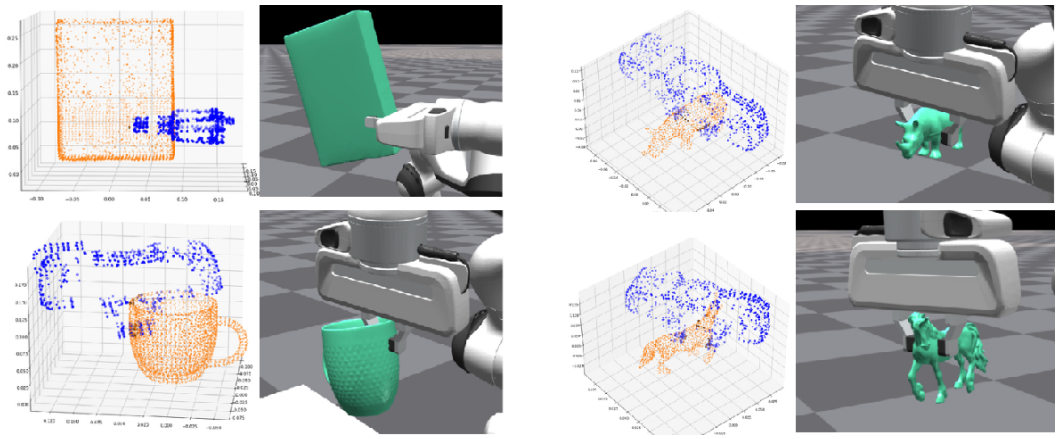

FIGURE 3.3. A grasp simulation for Google Scanned Objects.

Object	Franka		KG3		Barrett	
	Success Rate	**Time (s)**	**Success Rate**	**Time (s)**	**Success Rate**	**Time (s)**
Cat	100%	1.59	100%	1.18	100%	0.89
Bottle	100%	1.46	100%	0.96	100%	0.77
Can	100%	1.50	100%	1.11	100%	0.66
Detergent	100%	1.47	100%	0.94	100%	0.83
Dwarf	100%	1.56	100%	0.97	100%	0.94
LetterP	100%	1.50	100%	1.04	100%	0.86
Milk	100%	1.72	100%	1.10	100%	0.85
Shampoo	50%	1.65	100%	0.96	100%	0.38
Spray	100%	1.46	100%	1.00	100%	0.77
Glass	100%	1.69	100%	1.17	100%	0.19
Average	95.0%	1.56	100%	1.04	100%	0.71

TABLE 3.1. Computation time (seconds) and success rate for Franka, KG3, and Barrett hands.

A successful grasp is defined as the gripper lifting and holding the object for 5 seconds. Computation time excludes the time required for generating the SDF of the gripper. For the Franka arm, seven configurations are explored, each corresponding to different finger separations, resulting in a total of 84 initializations. The gripper's point cloud is sampled from the mesh, and objects are selected from the KIT object models database (Kasper et al., 2012) and Google Scanned Objects (Downs et al., 2022).

For KG3, three configurations, each corresponding to a set of finger joint angles, yield a total of 48 initializations. The KG3 gripper's point cloud is acquired through simulation with a depth camera. Algorithm 3.1 is executed for each grasp, producing converged grasp poses with the least match error selected. A series of 10 grasps is performed for each object, and Table 3.1 presents the success rate

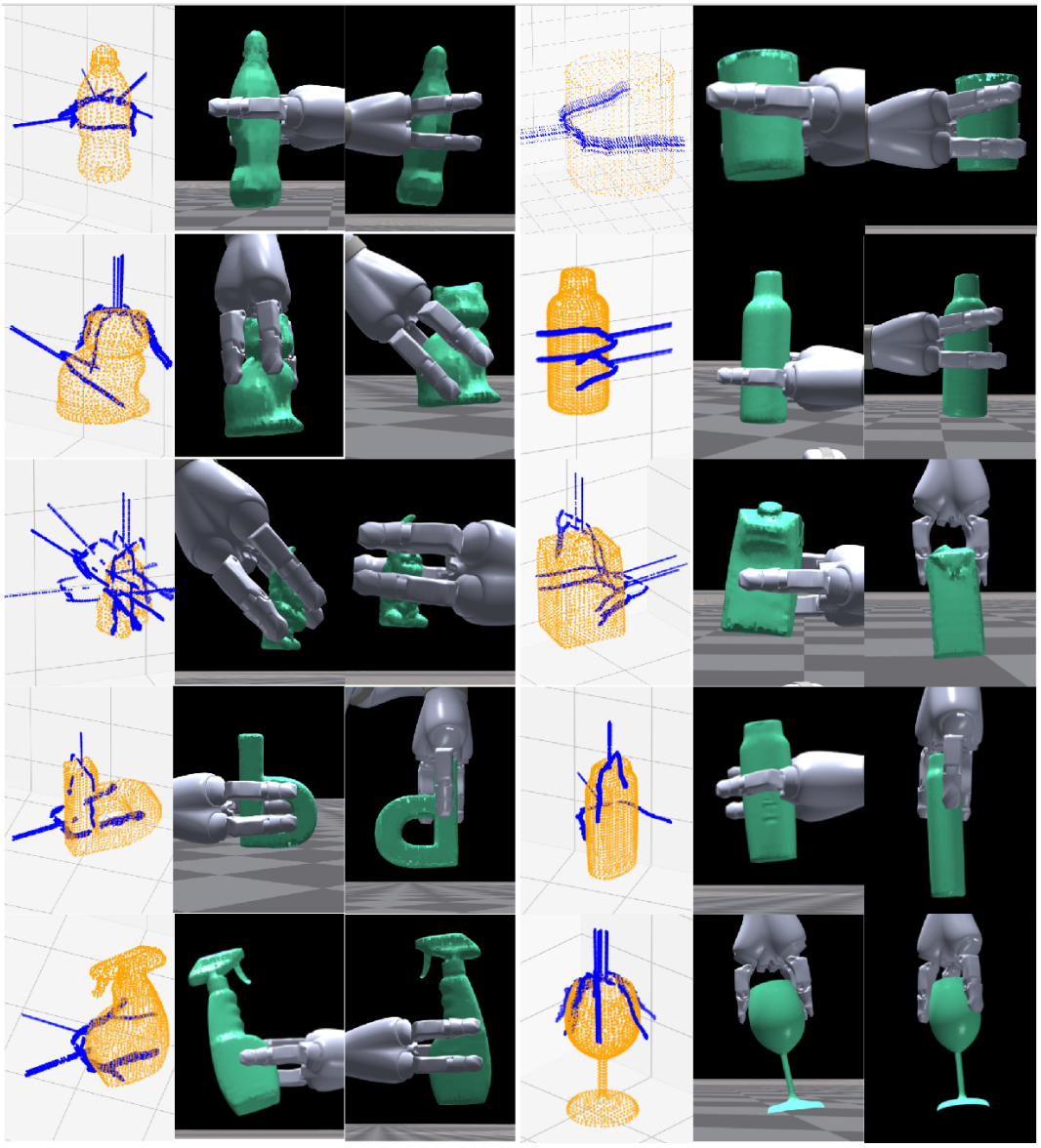

FIGURE 3.4. A visualization of the best grasp pose and its simulation result for KG3. the plot on the left is the best grasp poses for 10 trials with 48 initializations after 50 iterations. The middle and right ones are two of the ten simulation results.

and computation time for the two grippers across 10 different objects. Notably, the Franka hand and KG3 achieve average success rates of 95% and 100%, with corresponding average computation times of 1.56 seconds and 1.04 seconds. The longer computation time for Franka is attributed to the increased configurations.

A series of simulations was conducted using the Barrett Hand to perform a comparative analysis with the SplitPSO algorithm proposed by Kiatos et al. (2021). The proposed algorithm optimizes palm pose

(a) (b)

FIGURE 3.5. (a) Both grasp poses are successful, but a configuration with a smaller opening between fingers would provide a better grasp. (b) The case where occlusion is too large leading to failure due to collisions. It demonstrates a limitation of our algorithm in dealing with significant occlusions.

	SplitPSO (Kiatos et al., 2021)	Ours (a)	Ours (b)
Cat	86%	100%	100%
Bottle	95%	100%	90%
Can	97%	100%	100%
Detergent	92%	100%	100%
Dwarf	93%	100%	79%
LetterP	59%	100%	100%
Milk	91%	100%	100%
Shampoo	92%	100%	87%
Spray Flask	83%	100%	77%
Wine Glass	75%	100%	88%
Average Success Rate	86.3%	100%	92.1%

TABLE 3.2. Average success rate for grasping 10 objects. SplitPSO: success rate reported by Kiatos et al. (2021). Ours (a): success rate for best grasp poses of 20 trials, best grasp pose is the one with the least matching error from each trial's converged grasp poses. Ours (b): success rate of all converged grasp poses for a single trial.

	SplitPSO (Kiatos et al., 2021)	PPO-JPO Fan et al. (2019)	MDISF (Fan and Tomizuka, 2019)	Ours
Time (s)	4.25	3.263	4.47	0.71
	ISF (Fan et al., 2018a)		Ours	
Time (s)	2.33		1.56	

TABLE 3.3. Using Barrett hand, SplitPSO achieves 4.25 seconds on 10 objects; the iterative PPO-JPO method requires 3.263 seconds on 12 objects; the combination of MDISF and GTO averages 4.47 seconds for 10 objects, and ours achieves 0.71 seconds on 10 objects. Using a Parallel gripper, the ISF method accomplishes 2.33 seconds on 9 objects and our method achieves 1.56 seconds on 10 objects.

and joint position separately via particle swarm optimization with parallelization. It achieved an average success rate of 86.3% across 10 objects, as outlined in Table 3.2 (SplitPSO). In our study, employing a similar methodology to the KG3 gripper, we successfully grasped all objects, as detailed in Table 3.2 (Ours (a)). The generated grasp poses for the Barrett Hand closely resembled those of the KG3 gripper. Subsequently, Algorithm 3.1 was executed once, and all converged grasp poses were simulated. However, only 92.1% of the grasp pose candidates resulted in a successful grasp, as indicated in Table 3.2 (Ours (b)), highlights the importance of having multiple initializations. Our algorithm also significantly reduces the computation time due to parallelization and less parameters for optimization, as shown in Table 3.3.

3.4.2 Real Experiment

To validate the practical efficacy of our approach, preliminary real-world experiments were conducted using a Jaco Gen 2 Arm equipped with a KG-3 gripper. Object point clouds were captured with a RealSense D405 camera, and the data—cropped via a bounding box—was input directly into our algorithm without additional post-processing. As depicted in Figure 3.6, we tested the method on ten objects, executing 10 grasps per object with parameters consistent with our simulation study. The results showed a 100% success rate on this test set, with an average pose computation time of 0.925 seconds, as summarized in Table 3.4.

This initial validation has important scope limitations. The test set did not include more adversarial objects that are critically flat (Figure 3.5(a)) or involve large occlusions (Figure 3.5(b))—scenarios known to challenge ICP-based correspondence matching. Consequently, the high success rate reflects performance under favorable conditions rather than the method's absolute limit. A more nuanced performance characteristic is explored in Section 5.4.2, where we show that for the same input, the success rate of our generative method tends to be bimodal: it is either very high when the observable geometry is well-suited, or it fails completely when key geometric features are missing or ambiguous, with little middle ground.

3.5 Summary and Discussion

In this chapter, we have introduced a parallel SGD-ICP-based grasp optimization method that capitalizes on point cloud data from both the gripper and the object. Demonstrating its versatility across various gripper types, the approach exhibits remarkable efficiency, achieving an average computation time of

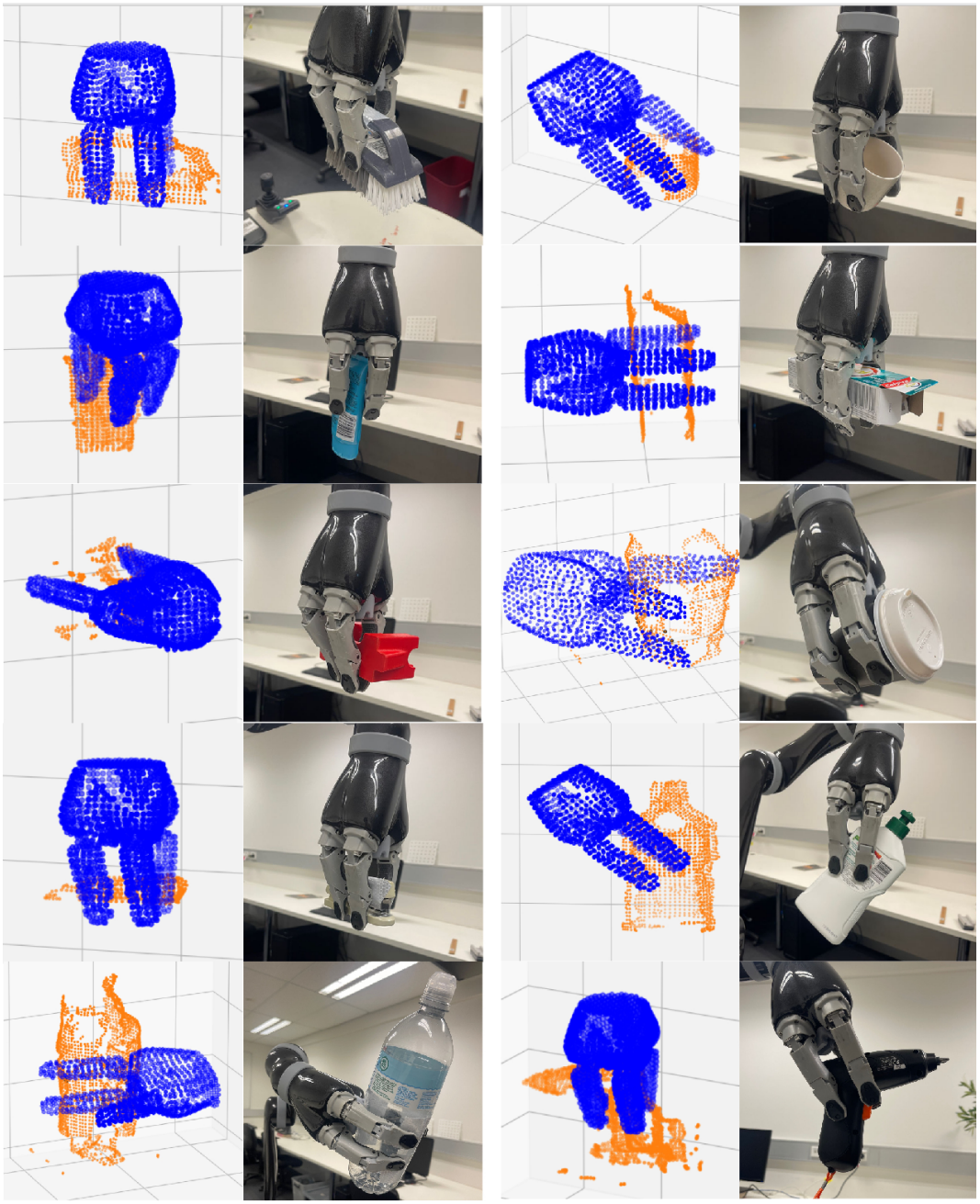

FIGURE 3.6. An illustration of real grasp with KG3 gripper. Object on the bottom right shows our algorithm having trouble with flat object with large occlusion.

0.925 seconds for the KG3 gripper in real experiments. This performance surpasses other point cloud-based analytic approaches. In a real experiment involving 10 distinct objects, the method successfully executes 10 consecutive grasps for each object. Our results underscore the potential of employing parallel optimization methods to enhance grasp quality and reduce computation time in robotic manipulation tasks, eliminating the need for intricate gripper joint angle optimizations.

	success rate	time (s)
Brush	10/10	0.932
Tea Cup	10/10	0.923
Sunscreen	10/10	0.926
Toothpaste box	10/10	0.897
Red Holder	10/10	0.892
Coffee Cup	10/10	1.05
Stapler	10/10	0.748
Detergent	10/10	0.927
Bottle	10/10	1.02
drill	10/10	0.935
Average	100%	0.925

TABLE 3.4. Computation time and success rate for KG3 real experiment.

Despite promising results in both simulation and real-world experiments, our current algorithm still faces challenges common to methods based on ICP and sampling-based data-driven approaches. In particular, the initialization step plays a crucial role in determining the final grasp outcome. Our approach requires sampling initial parameters around the target area, and further incorporates additional initializations based on the preshapes of the gripper. However, experiments reveal that with a fixed initialization, the algorithm consistently generates nearly identical grasps across multiple runs—resulting in extreme outcomes, where success rates are either 100% or 0%. Although generating different sets of initializations is possible, there is no systematic method guaranteeing an optimal or even successful distribution of grasp configurations.

To address this limitation, our next chapter will explore the integration of SVGD. The goal is to achieve a more diversified and robust distribution of initial parameters, mitigating the sensitivity to the initial guess and enhancing the overall robustness of the grasp synthesis process.

CHAPTER 4

Grasping with Annealed Stein ICP

4.1 Introduction

In this chapter, we address the grasping problem by formulating it as an optimization task where the final grasp pose is achieved by matching point clouds of the gripper and the object. Our focus is on a power grasp, where the grasping pose involves the whole palm of the gripper, and not a precision grasp, where only the fingers are involved. The optimization steps are visualized in blue in Fig 4.1. We explore the integration of Stein Variational Gradient Descent (SVGD). The goal is to achieve a more diversified and robust distribution of initial parameters and final grasp poses, mitigating the sensitivity to the initial guess and enhancing the overall robustness of the grasp synthesis process.

The main contribution of this chapter is a framework for generating efficient and stable grasp poses using GPU-based parallel Stein ICP between the gripper and object's point clouds. The algorithm is independent of the gripper type and does not rely on costly finger joint angle optimization typical of analytic approaches. Our approach generates robust optimal grasp poses with uniformly distributed initializations, demonstrating robustness in both simulations and complex real-world experiments with various objects and noisy, partial point clouds. For 11 different objects using the Kinova KG3 gripper, we achieve an average success rate of 87.3% and a computation time of 0.926s in real experiments.

4.2 Related Work

Data-driven approaches rely on large labeled datasets and requires significant training time. However, it achieves higher success rates, particularly with objects similar to those used during the training. Fang et al. (2023) proposes AnyGrasp, which can generate grasp poses dynamically. Xu et al. (2021) proposes AdaGrasp which learns a policy that can be used for different grippers. Most of these existing methods face challenges in generalizing to objects that are not present in the training datasets. In contrast, our

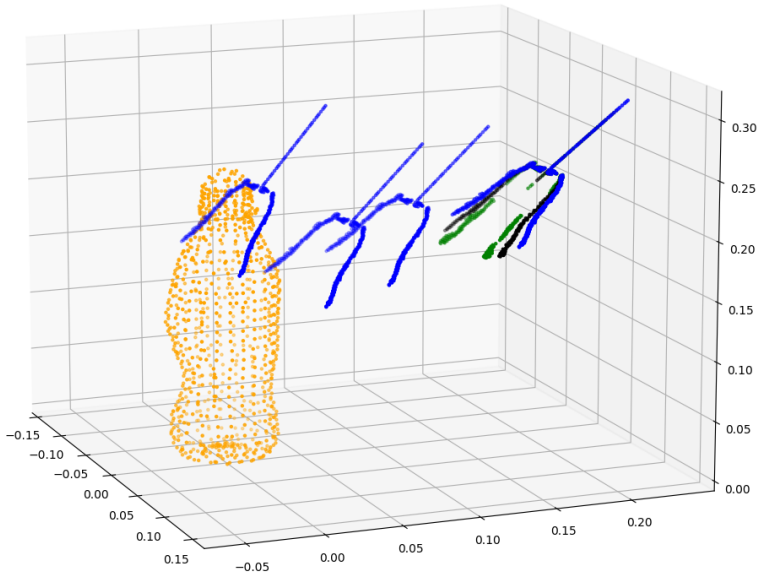

FIGURE 4.1. An illustration of the optimization process. Green, blue and black point clouds are the initial poses of three preshapes of the KG3 gripper. Blue plots show the optimization process to find the grasp pose of a singe preshape.

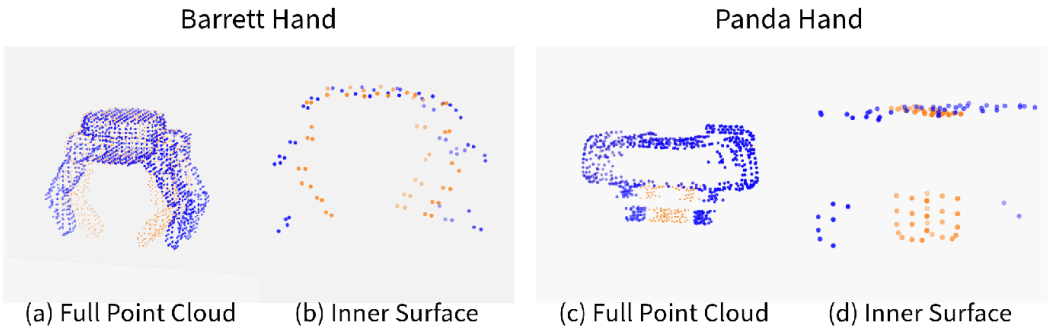

FIGURE 4.2. Preshapes used for simulation in this chapter. On the left, we have two preshapes of Barrett Hand. On the right, we selected two preshapes from the ten used for Franka Hand. The full point cloud generates SDF, and the partial point cloud is used to optimize.

approach eliminates the need for training and demonstrates an enhanced ability to generalize to unfamiliar objects. Moreover, our technique can be applied alongside these methods to enhance grasp quality. The analytic approach and evaluation metrics have been introduced in Section 3.2.

4.3 Methodology

We investigate grasp synthesis for unknown objects while avoiding the expensive optimization of finger joints. Our algorithm takes as input a gripper's inner surface point cloud $\mathcal{S}$ (Figure 4.2 (b, d)) and the object's 3-D point cloud $\mathcal{R}$ for matching, a complete gripper's point cloud $\mathcal{G}$ (Figure 4.2 (a, c)) for generating SDF and object's point cloud with the table's point cloud $\mathcal{C}$ for collision checking. Specifically, we utilize point clouds of different finger configurations as preshapes, which remain unchanged during optimization. The algorithm aims to determine an optimal grasp pose parameterized by θ. All transformations are with respect to the object's coordinate frame. The grasp pose is optimized using parallel AS-ICP with the following constrained loss:

$$\begin{aligned} \min_{\theta_i} \quad & \mathcal{L}(\mathcal{R}, T_\theta(\mathcal{S})) \\ s.t. \quad & \text{dist}(\mathcal{C}, \text{SDF}(T_\theta(\mathcal{G}))) > 0, \end{aligned} \tag{4.1}$$

where $T_\theta(*)$ is the transformed point cloud w.r.t. θ, and the loss function

$$\mathcal{L} = \mathcal{L}_{com} + \mathcal{L}_{ct} \tag{4.2}$$

is a combination of two grasp quality metrics, matching error ($\mathcal{L}_{ct}$) and distance between tool centre point and centre of mass ($\mathcal{L}_{com}$), which we will explain in the next Section 4.3.2.

The loss function is constrained by the distance between the object's point cloud and the gripper's SDF to ensure a physically plausible grasp pose. A collision occurs when the gripper's point cloud is inside the physically infeasible space of the object's point cloud, corresponding to a negative distance value from the SDF.

4.3.1 Stein ICP

In Stein ICP (Section 2.5), the optimal update direction for particles is:

$$\begin{aligned} \hat{\phi}^*(\theta) = \sum_{j=1}^{K} [& -\gamma(t)(N\bar{g}(\theta_j, \mathcal{M}) + \nabla_\theta \log p(\theta_j))k(\theta_j, \theta) \\ & + \nabla_\theta k(\theta_j, \theta)], \end{aligned} \tag{4.3}$$

where $\gamma(t)$ is the annealing schedule defined in Equation (4.6). $\nabla \log p(\theta)$ is represented by gradients from Gaussian priors for translations and von Mises priors for rotations.

The first gradient term in Equation (4.3), weighted by a kernel function, determines the steepest direction for the log probability. Conversely, the second term represents the gradient of the kernel function, acting as a repulsive force that promotes dispersion among particles and prevents them from converging to local modes of the log probability.

We use the RBF kernel for translation:

$$k(\theta'_{1:3}, \theta_{1:3}) = \exp\left(-\frac{1}{h}\|\theta_{1:3} - \theta'_{1:3}\|_2^2\right), \tag{4.4}$$

and the dot product as a kernel for rotation:

$$k(\theta'_{4:7}, \theta_{4:7}) = \text{abs}(\theta_{4:7} \cdot \theta'_{4:7}). \tag{4.5}$$

To address the mode collapse problem inherent in SVGD (Zhang et al., 2020), we adopt annealed SVGD (D'Angelo and Fortuin, 2020) with SGD ICP. The annealing schedule in Equation (4.6) encourages more exploration, resulting in a method known as Annealed Stein ICP, which improves the distribution of poses crucial for our grasping tasks.

$$\gamma(t) = \left(\frac{\text{mod}\ (t, T/C)}{T/C}\right)^p \tag{4.6}$$

where t is the current iteration, T is the maximum iteration, C is the number of cycles, and p is an exponent determining the transition speed between the two phases.

4.3.2 Cost Functions

Unlike traditional ICP problems where source and reference point clouds belong to the same entity with variations possibly arising from partial observations, point clouds of the gripper and the object are likely to be quite distinct. The proposed algorithm approximates shape matching by maximizing the selected grasp quality metrics rather than achieving a perfect match.

These grasp quality metrics are selected to achieve two main objectives: minimize the matching error to estimate the grasp pose ($\mathcal{L}_{ct}$) and minimize the moment at contact points to ensure a stable grasp ($\mathcal{L}_{com}$) as described in Section 3.3.2:

$$\mathcal{L}_{ct} = \frac{1}{m} \sum_{s'_i, \hat{r}_i \in \text{Pairs}} (\|s'_i - \hat{r}_i\|^2). \tag{4.7}$$

$$\mathcal{L}_{com} = \min_{\theta} \|(R \cdot TCP + t) - CoM\|^2. \tag{4.8}$$

Unlike in the previous chapter, we found that under the SVGD framework, an arbitrary weight for each cost is not necessary. However, an observation-based weight assignment could benefit the overall performance.

4.3.3 Collision Checking

The ICP algorithm does not account for collision checking during the optimization process, which is crucial in grasping tasks. Simple ICP matching may cause the gripper to penetrate the object, resulting in physically infeasible grasp poses. We keep the same collision avoidance strategy as in Section 3.3.4:

$$\begin{aligned} \mathcal{L}_{col} &= \frac{1}{N_{col}} \sum_{r_{col} \in \mathcal{C}} \min_{s' \in \mathcal{S}} \|r_{col} - s'\|^2, \\ s.t. \quad & \text{dist}(\mathcal{C}, \text{SDF}(T(\mathcal{G}_i))) < 0, \end{aligned} \tag{4.9}$$

where N_{col} is the number of colliding points in the object point cloud. In Equation (4.9), we do not account for $\mathcal{L}_{com}$ as the gripper needs to move away from the object's center of mass (CoM).

4.3.4 Algorithm

We perform two optimization processes: AS-ICP and SGD-ICP. AS-ICP ensures that the poses are well-distributed around the object. After generating a distribution of potential grasping poses, we refine each Stein particle by updating it with SGD-ICP to determine the optimal grasp pose.

A summary of the Parallel Shape matching-based grasp method is provided in Algorithm 4.1. Given the object's point cloud $\mathcal{R}$ as a reference and the complete gripper's point cloud $\mathcal{G}$, we first compute the SDF of each preshape of the gripper and the target object's centre of mass. We collect all preshapes' SDF as $\mathcal{SDF} = \{SDF_i\}_{i=1}^{N}$ by adding a constant offset ϵ between them, which can be considered as spreading them in space. All operations are performed in parallel for all initializations by combining

Algorithm 4.1: Parallel Shape Matching with AS-ICP

Input: Point cloud of: Gripper $\mathcal{S} = \{s_i\}_{i=1}^{N}$, Target Object $\mathcal{R} = \{r_i\}_{i=1}^{M}$, and Table $\mathcal{C}$;
complete gripper $\mathcal{G}$, initial parameters $\Theta_0 = \{\theta_0^j\}_{j=1}^{J}$,
number of iterations for AS-ICP k_{stein}, and total iteration k_{max}
Output: θ that minimizes the loss function

$CoM \leftarrow$ center of mass of target object
$SDF \leftarrow$ SDF of the gripper
while $k \leq k_{max}$ **do**
  for *each* $\theta_k^j \in \Theta_k$ *in parallel* **do**
    $\mathcal{S}_k^j \leftarrow$ Transform the source cloud with θ_k^j
    $\mathcal{M}_k \leftarrow$ Select a mini-batch from $\mathcal{R}$
    $\bar{g}_k^j \leftarrow$ Compute gradients (3.5) and (3.6)
    if $\mathrm{dist}(\mathcal{C}, \mathrm{SDF}(T_{\theta_k^j}(\mathcal{G}))) < 0$ **then**
      $\bar{g}_{col}^j \leftarrow$ Compute gradients using (2.27) and (2.28)
      $\bar{g}_k^j = \bar{g}_{col}^j \leftarrow$ Replace $\bar{g}_k^j$ with $\bar{g}_{col}^j$
    end
    if $k < k_{stein}$ **then**
      $\hat{\phi}^*(\theta) \leftarrow$ Compute using (4.3)
      $\theta_{k+1}^j \leftarrow$ Update θ_k^j with (2.17)
    else
      $\theta_{k+1}^j \leftarrow$ Update θ_k^j with (2.26)
    end
  end
  k = k +1
end
return $\theta = \mathrm{argmin}_{\theta^j} \mathcal{L}$

the point clouds and parameters into multidimensional tensors $\mathcal{S}$, $\mathcal{M}$, $\mathcal{C}$ and θ. The source point cloud is transformed with the J transformation parameters. Then, a mini-batch is sampled from the reference cloud. Next, corresponding closest points from the mini-batch are sought and stored in pairs for all points within each transformed source cloud. This is followed by computing loss, gradients, and relative error difference. Instead of transforming the SDFs of the gripper, we perform an inverse transformation on the object point cloud for each initialization, denoted as $\mathcal{R}' = \{r_i'\}$. These point clouds are then divided into different groups according to the SDF_i they are matching with. Then the same ϵ of the corresponding SDF_i is added to r'. In case of collision, $\bar{g}_{col}$ is computed, replacing the current gradients. If the relative error difference indicates convergence at this stage, it is set to a non-converged state to move the gripper out of the object. Initially, AS-ICP is executed for k_{stein} iterations. The resultant updated parameters θ are then used as the starting point for SGD-ICP. The parameter set θ undergoes further updates if it fails to converge.

(a) Gaussian mixture (b) Fibonacci sequence

FIGURE 4.3. (a) Initializations sampled from a mixture of Gaussian to provide some prior knowledge of the object. (b) Initializations sampled from the Fibonacci sequence and projected onto a quarter of the sphere, facing the direction of the robot arm.

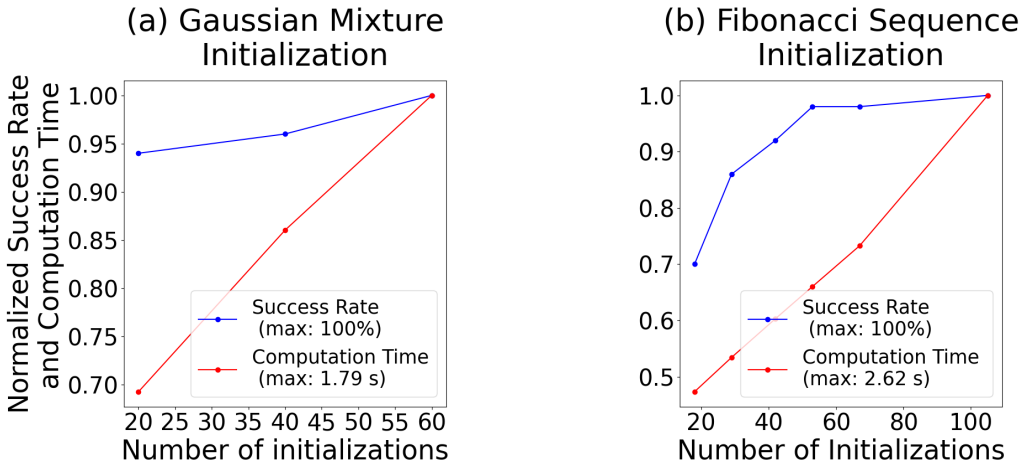

FIGURE 4.4. Plots of success rate and computation time with an increasing number of initializations for different sampling methods: (a) Gaussian and (b) Fibonacci. Sampling from a mixture of Gaussians leads to a higher success rate and faster computation time with fewer initializations as it provides some prior knowledge of objects.

4.4 Experiments

4.4.1 Ablation Study

Initialization plays a pivotal role in ICP-based algorithms. Given its non-convex nature, ICP is susceptible to converge at local minima (Biggie et al., 2023). This aspect is particularly critical in our problem setup,

where preshapes of the gripper vary significantly from the object geometry. Thus, precise initialization strategies are essential for achieving effective gripper-object matching.

AS-ICP effectively distributes initial poses around the object, increasing the likelihood of a successful grasp. Since we do not provide any prior knowledge of the objects, initializations are sampled from the Fibonacci sequence and projected onto a quarter of a sphere as shown in Figure 4.3 (b). However, this introduces many unnecessary initializations and leads to longer computation time, as discussed in the previous Section. A comparison between sampling from a mixture of Gaussians and Fibonacci sequences is provided in Figure 4.4. It is worth noting that as the number of initializations increases, the success rate approaches 100% at the cost of more computation time. By heuristically choosing four Gaussian means, our algorithm can achieve a higher success rate with fewer initializations, showing the possibility of learning some prior knowledge of the objects to reduce the dependency on initializations.

In our algorithm, we match the gripper's shape to the object's and do not optimize finger joints during the optimization process. Thus, choosing the right gripper preshapes becomes crucial. This chapter uses two preshapes for the Barrett Hand and ten for the Franka Hand, each corresponding to a set of fixed finger configurations, as shown in Figure 4.2. Additional configurations are used for the Franka hand to ensure the stability of the generated grasps.

4.4.2 Simulation Result

Our study begins with a comprehensive examination of our proposed method's success rate and efficiency, employing two distinct grippers—-the Franka Hand and the Barrett Hand within the Isaac Gym simulator (Makoviychuk et al., 2021). A list of parameters is provided in Table. 4.1. The optimization is performed on a laptop with an RTX 2070 GPU. Initializations are sampled from a quarter of a sphere using the Fibonacci sequence. Six additional initializations directly above the object are manually added, as the sampled initializations alone do not provide sufficient coverage. The parameter values in this work were chosen heuristically to lie within a reasonable operational range, rather than being optimized for any specific metric. Consequently, they represent a general compromise and do not necessarily yield the highest success rate or lowest computation time for a given object. A more sophisticated approach for future work would be to use a model, trained on object features, to predict and select an optimal parameter set per instance. We use Open3D (Zhou et al., 2018) for both SDF generation and collision checking. After reaching the maximum number of iterations, the converged θ with the minimum cost is selected as the final grasp pose.

Parameter	Franka Hand	Barrett Hand
Number of initializations per preshape	$\approx \frac{100-6}{4} + 6$	
Number of preshapes	**10**	**2**
Voxel size (Gripper's point cloud $\mathcal{S}$)	**0.005**	**0.025**
Number of points (Object with table $\mathcal{C}$)	1100 to 1900 points	
Voxel size (Object's point cloud $\mathcal{M}$)	0.005	
Mini-batch size	$N \times \frac{\min(k, 2k_{\max}/3)}{2k_{\max}/3}$	
Learning rate and cost weights for SGD	1	
Annealing factor	$\left(\frac{\text{mod }(k, k_{\max}/5)}{k_{\max}/5}\right)^2$	
Convergence criterion	0.02%	
Number of iterations for SVGD	15	
Number of iterations for SGD	25	

TABLE 4.1. Parameters used for Alg. 4.1. The initialization count (6) sets the number of top-down starting poses.

	AnyGrasp		Ours		AnyGrasp+Ours	
Object	**Rate (%)**	**Time (s)**	**Rate (%)**	**Time (s)**	**Rate (%)**	**Time (s)**
Cat	46	**0.046**	**98**	2.42	20	1.13
Detergent	78	**0.047**	**96**	2.57	88	1.08
Bottle	82	**0.044**	**96**	1.92	94	1.04
LetterP	84	**0.052**	**96**	2.63	86	0.93
Shampoo	28	**0.052**	**94**	2.54	38	1.11
Spray	68	**0.050**	**92**	2.45	80	1.02
Glass	68	**0.049**	**96**	2.70	74	1.15
Mug	14	**0.051**	**96**	2.32	78	1.14
Shark	**20**	**0.043**	2	2.17	28	1.21
Shoe	6	**0.060**	**54**	2.51	6	1.17
Average	49.5	**0.049**	**82.0**	2.42	59.2	1.10

TABLE 4.2. Simulation results comparing AnyGrasp and our method using the Franka Hand in the Isaac Gym simulator. Success rates (rate) and computation times (seconds) are shown.

A successful grasp is defined as when the gripper can lift and hold the object for 5 seconds. Computation time excludes the time required to generate the SDF of the gripper. The gripper's point cloud is sampled from the mesh for the Franka Hand. The gripper's point cloud is acquired through simulation with a depth camera for the Barrett Hand. Objects are selected from the KIT object models database (Kasper

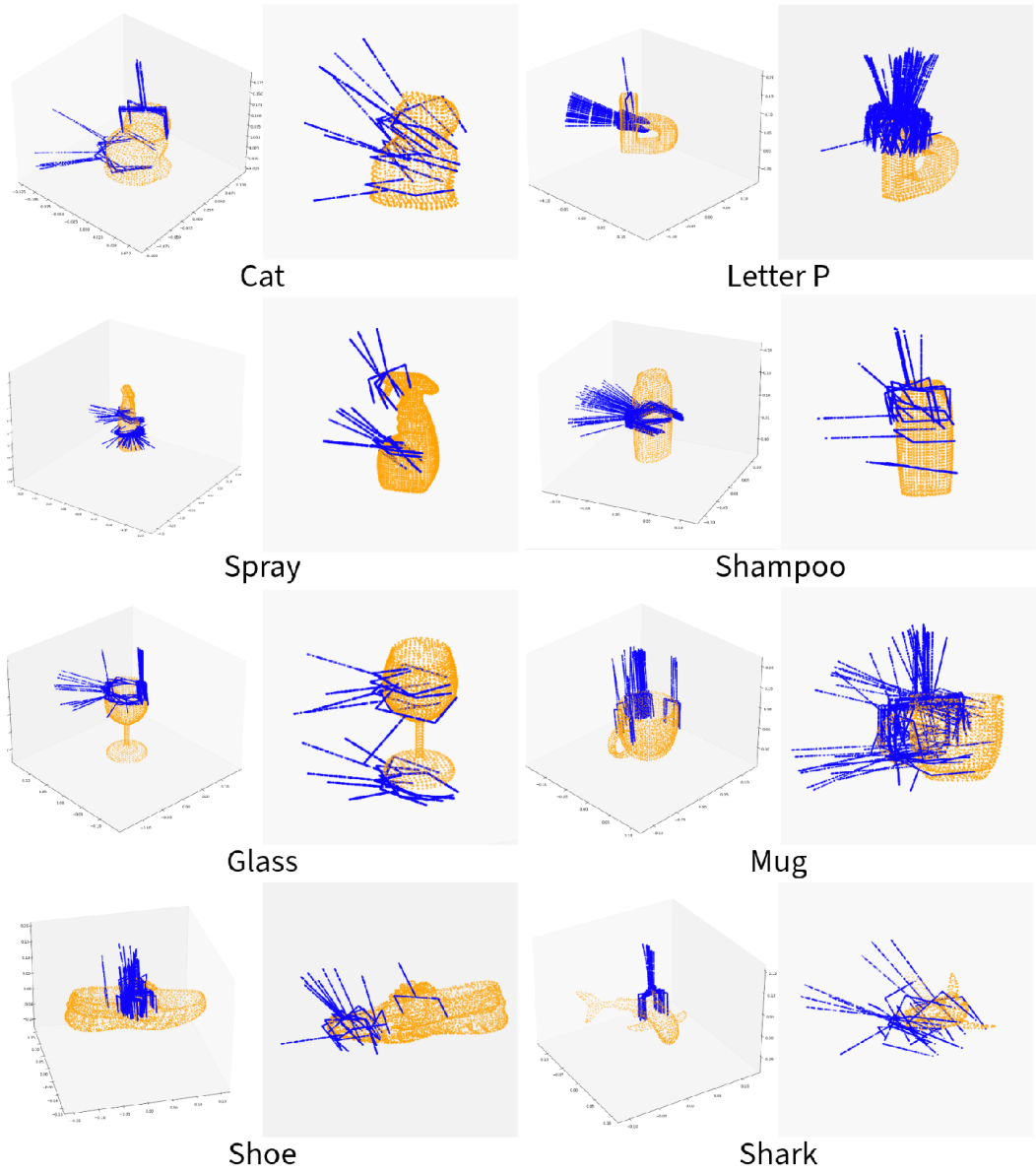

FIGURE 4.5. Best grasp poses with Franka Hand for 50 trials. On the left is the pose generated by our algorithm, and on the right is the pose generated by AnyGrasp. This visualization highlights a key advantage of our gradient-based optimization: it inherently optimizes for suitable gripper preshapes that align the fingers with the object's surface geometry to maximize contact area. In contrast, AnyGrasp appears to generate poses better suited for a fully opened, neutral gripper state.

et al., 2012) and Google Scanned Objects (Downs et al., 2022). The grasp poses generated are illustrated in Figure 4.5 for Franka Hand and Figure 4.6 for Barrett Hand. A summary of the success rate and optimization time is provided in Table. 4.2 and Table. 4.3. The success rate and optimization time are the averages of 50 simulations for each object.

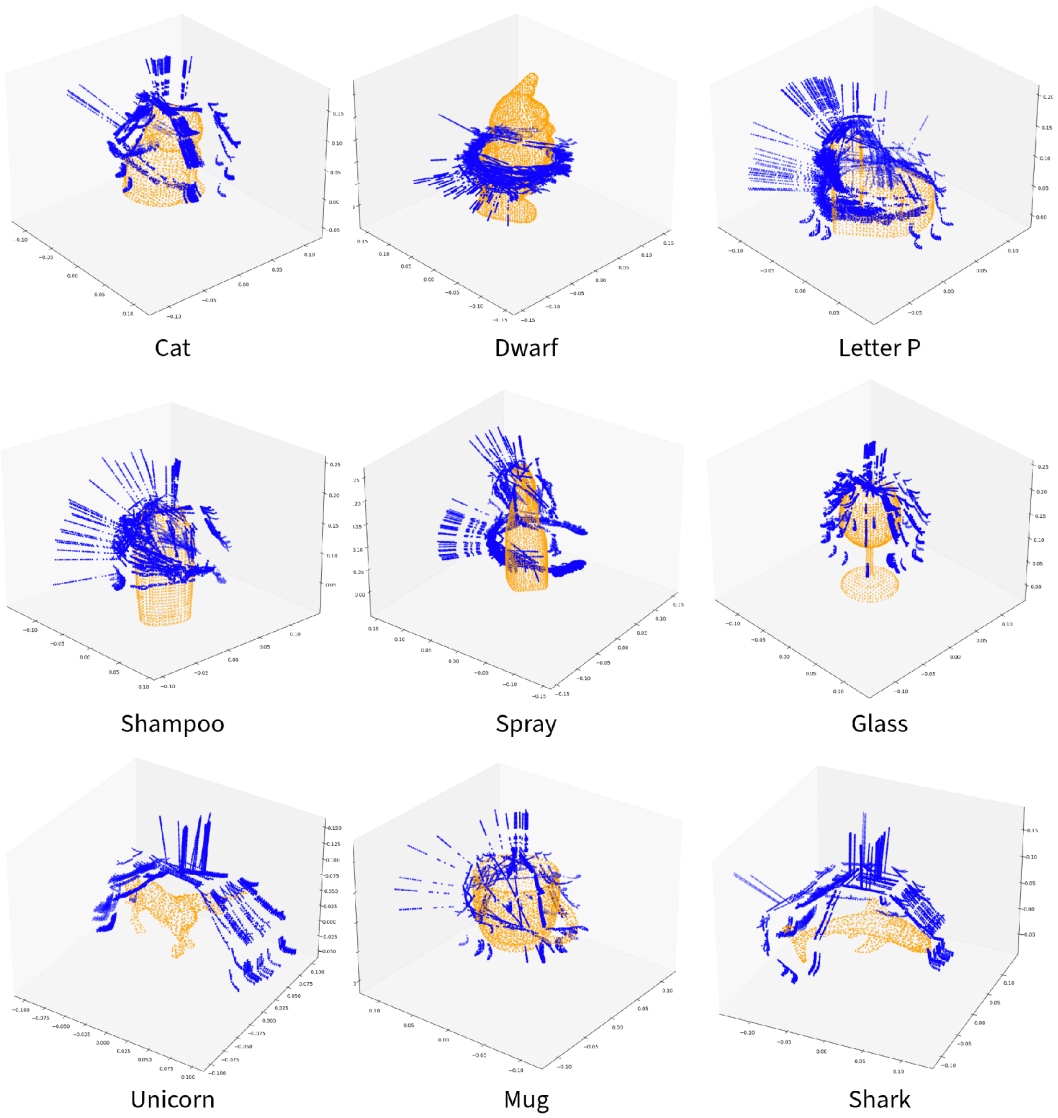

FIGURE 4.6. Best grasp poses with Barrett Hand for 50 trials.

Table. 4.2 provides a comparison with AnyGrasp (Fang et al., 2023). The grasp poses for AnyGrasp were generated using the code from GraspNet (2024) with pre-trained weights. Instead of using a point cloud captured from a camera, the full point cloud of the object was utilized. Fifty grasps were generated with random object positions to compute the metrics. As a data-driven method, AnyGrasp has a shorter computation time than ours, whereas our method achieves an overall higher success rate.

Figure 4.5 provides a qualitative comparison of grasp poses generated by our method and the AnyGrasp baseline. This visualization highlights a key advantage of our gradient-based optimization: it inherently

	SplitPSO		Ours	
Object	**Rate (%)**	**Time (s)**	**Rate (%)**	**Time (s)**
Cat	90	8.10	**100**	**0.996**
Detergent	90	10.4	**100**	**1.28**
Dwarf	50	9.55	**100**	**1.56**
LetterP	80	12.7	**100**	**1.55**
Milk	**100**	11.1	**100**	**1.43**
Shampoo	85	10.6	**92**	**1.20**
Spray	85	9.45	**86**	**1.40**
Glass	50	8.05	**96**	**1.24**
Unicorn	30	8.30	**98**	**1.43**
Mug	95	9.00	**96**	**0.972**
Shark	0	25.4	**14**	**1.12**
Average	68.6	11.2	**89.3**	**1.29**

TABLE 4.3. Simulation result comparison between SplitPSO and our method. The simulation uses Barrett gripper and is carried out in Isaac Gym simulator.

optimizes for suitable gripper preshapes that align the fingers with the object's surface geometry to maximize contact area. In contrast, AnyGrasp appears to generate poses better suited for a fully opened, neutral gripper state, which can result in suboptimal finger alignment upon closure. The improved contact from our poses reduces slipping (e.g., Cat, Shampoo, Glass, and Mug) and unstable grasps (e.g., Spray, Shampoo, and Shoe), whereas AnyGrasp leads to more unstable grasps and a higher likelihood of failure, especially for novel objects in diverse scenarios.

To evaluate the effectiveness of our algorithm as a post-processing module, we used 10 grasps generated by AnyGrasp as initializations for our method (Table 4.2). Except for the Cat, the combined method achieved a higher success rate than AnyGrasp alone and had a lower computation time compared to our method alone. The failure with the Cat was due to slip, highlighting our method's dependence on good initializations, similar to other ICP-based methods. However, the improvement in success rate with shorter computation time is promising, suggesting that better initialization parameters could further enhance performance, as discussed in Section 4.4.1.

Table. 4.3 provides a comparison for SplitPSO (Kiatos et al., 2021), with grasp poses being generated from Kiato (2024). Instead of using the proposed point cloud completion module, the object's full point cloud is used. Twenty grasps for each object are simulated. Our method outperforms SplitPSO in both success rate and computation time.

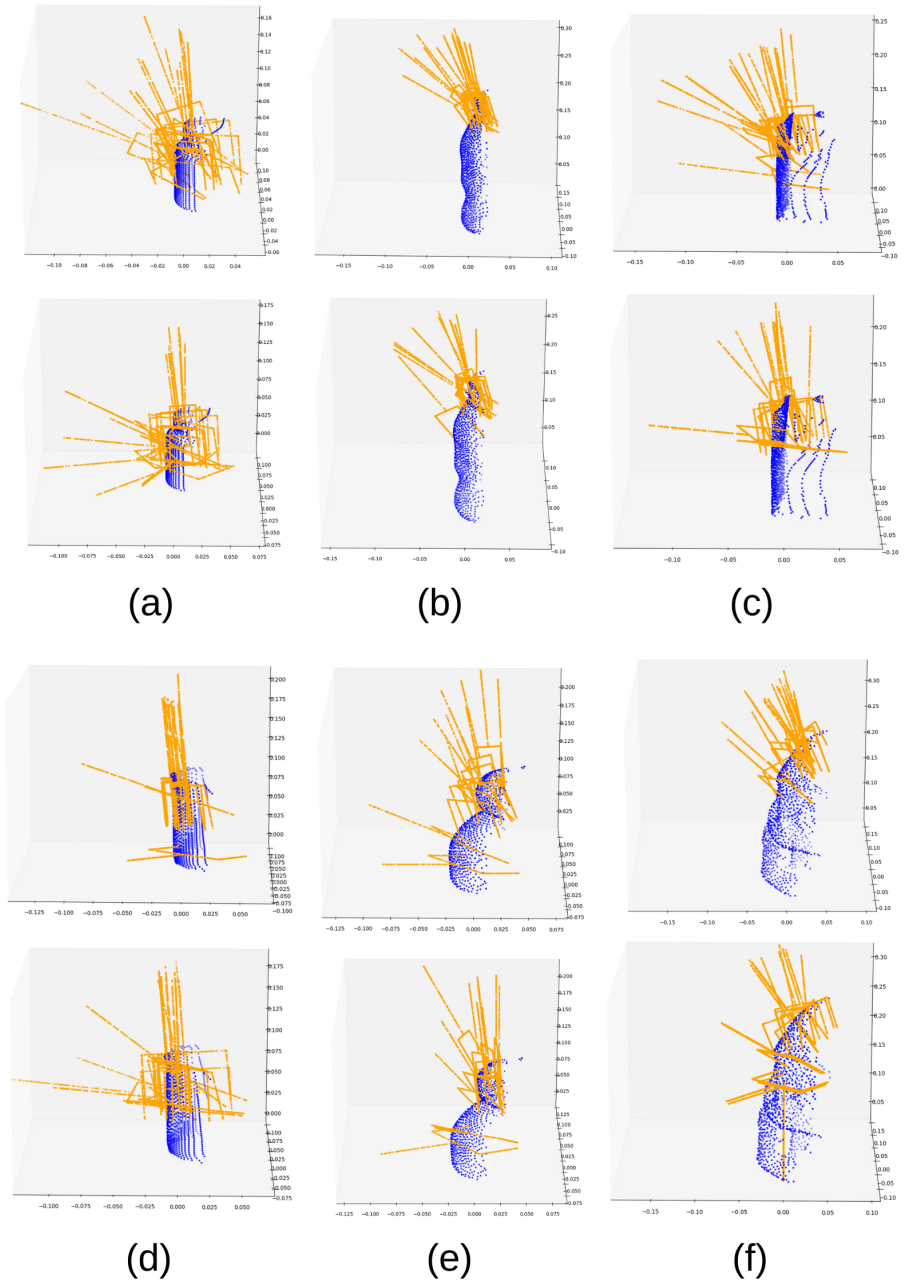

FIGURE 4.7. The best grasp poses identified across 30 trials for SGD-ICP (top) and AS-ICP (bottom). The results show that AS-ICP, which incorporates SVGD, produces a more diversified set of poses, especially in rotational space. This aligns with the core objective of using SVGD to enhance exploration.

With limited configurations and a full gripper's inner-surface point cloud, our algorithm focuses on a power grasp rather than a precision grasp. SplitPSO also faces this problem, as it uses the full point cloud of the gripper's inner surface. Thus, both methods fail to grasp the Shark, where a precision grasp with fingertips is more suitable. In contrast, AnyGrasp has a higher success rate, indicating a good precision

Object	Success Rate	Time (s)
Tea Cup	100%	1.01
Coffee Cup	100%	0.95
Detergent	80%	0.62
Brush	100%	0.99
Sunscreen	100%	0.84
Holder	60%	1.06
Case	80%	1.06
Washing Liquid	100%	0.81
Toy	80%	1.11
Hand	80%	0.79
Helping Hand Tool	80%	0.95
Average	87.3%	0.93

TABLE 4.4. Number of successes for five grasps, where the input for each grasp corresponds to a different orientation of the object.

grasp. Although the influence of a parallel gripper should be less significant, using the palm point cloud still affects our success rate with the Franka Hand. This can be seen from the grasp poses generated for the Shark, where our method fails due to contact with the Shark's fin. A potential solution is to sample point clouds from the finger surface with some distribution, as proposed by Fan and Tomizuka (2019). However, this object-dependent parameter requires data-driven methods to acquire the best preshapes for different objects.

Figure 4.7 shows the best grasp poses from 30 trials for SGD-ICP (top) and AS-ICP (bottom) across six objects. While AS-ICP (which incorporates SVGD) yields greater rotational diversity—consistent with SVGD's exploratory goal—its simple uniform prior fails to guide optimization effectively. This is evident in poses that collide with unseen object parts, and in Figure 4.7(f), where a high-scoring pose lies inside the object. These failures motivate the learned prior we introduce via an EBM in the next chapter.

4.4.3 Real Experiment

We conducted real experiments using the Jaco 2 Arm equipped with the KG3 gripper to test our approach with noise and partial point clouds. We captured objects' point clouds using a wrist-mounted RealSense D405 Camera. Unlike in simulation, where the algorithm has access to the entire point cloud, we used a single viewpoint partial point cloud for the real experiment.

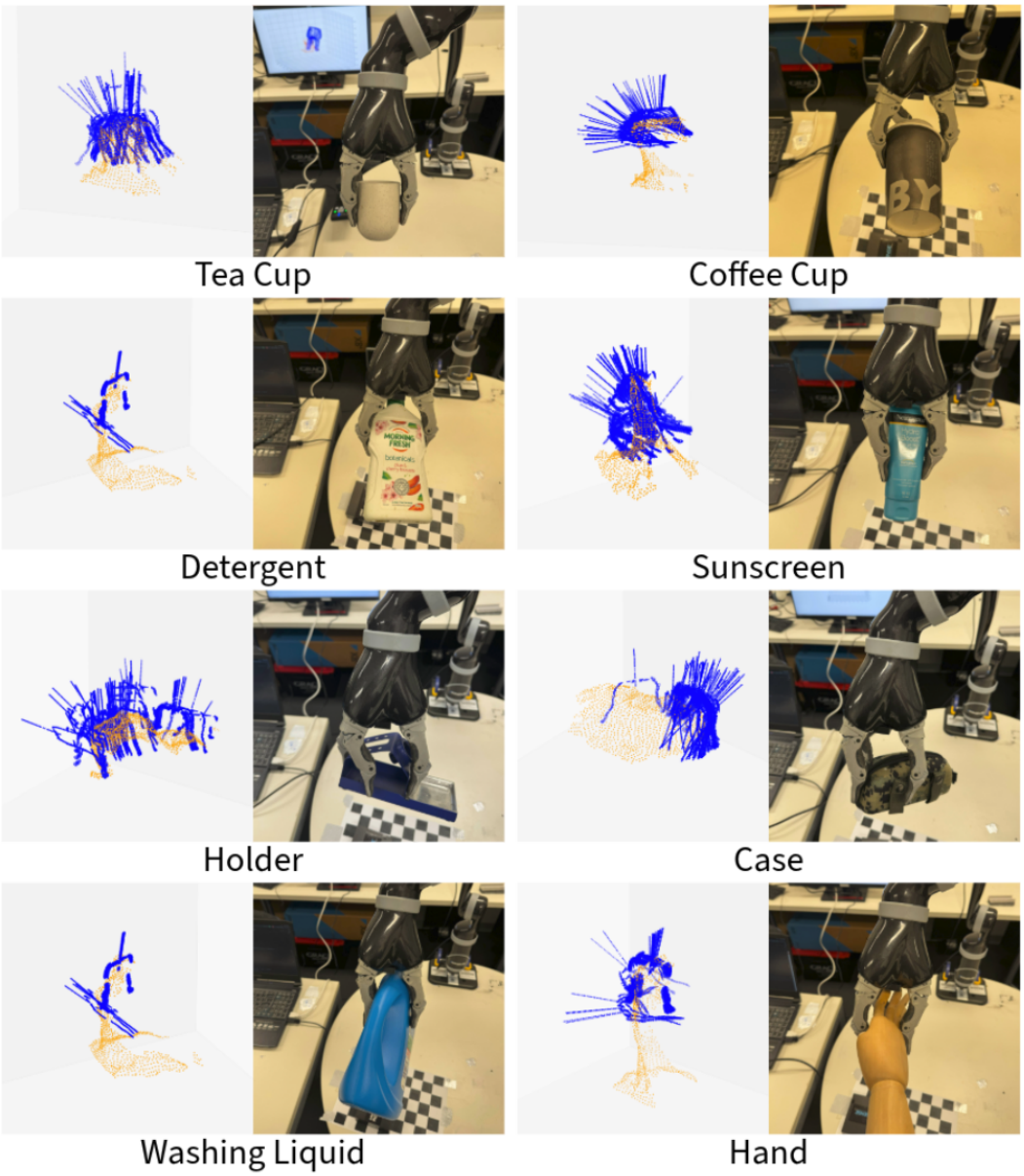

FIGURE 4.8. The picture on the left illustrates fifty grasp poses generated with the same point cloud. The photo on the right is the actual grasp for one of the five grasps in Table. 4.4. Our algorithm can grasp objects with large occlusions and noisy point clouds.

The experimental results, shown in Figure 4.8, demonstrate the performance of our approach on eight objects. Five grasps were executed for each object using the same parameters and initializations as in the simulation. Each trial involved placing the object in a different orientation, resulting in varied occlusions. A waypoint was set 15 cm away from the grasp pose to prevent collisions during motion. After grasping,

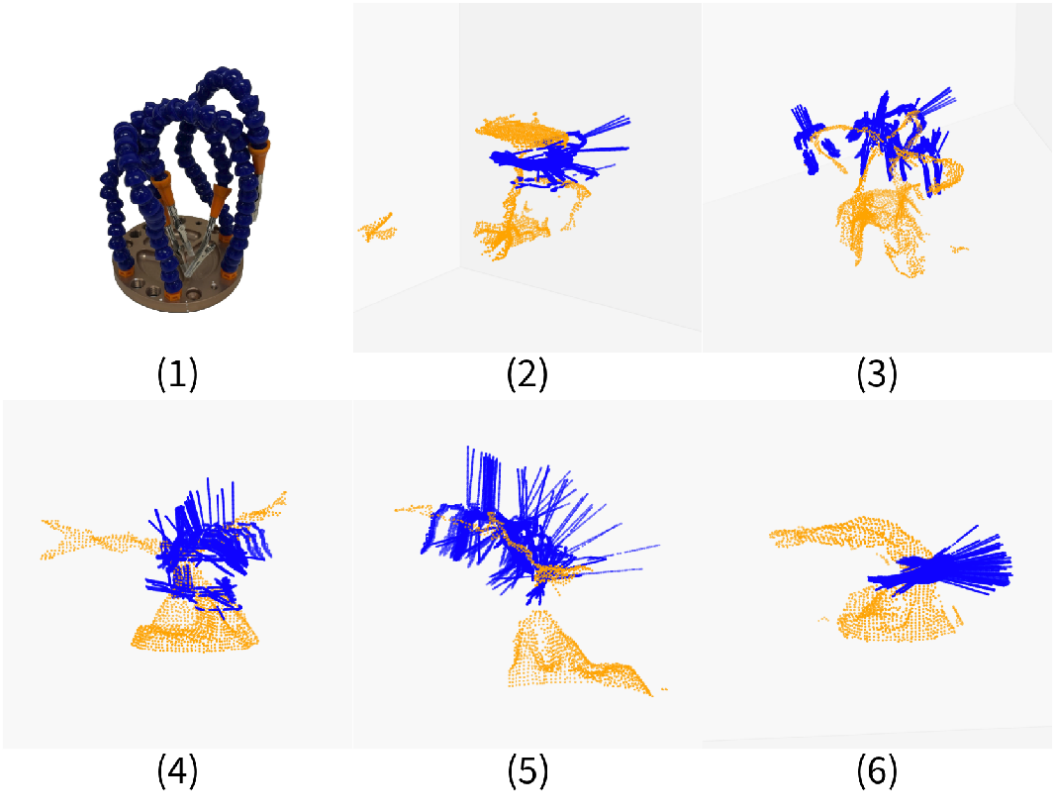

FIGURE 4.9. (1) Hand Helping Tool (2)-(6) Grasp poses for Hand Helping Tool with five different shapes.

the gripper returned to the location indicated in Figure 4.8. On average, we could grasp 4.37 out of the 5 grasps, with an average computation time of 0.926 seconds, as summarized in Table 4.4.

Notably, challenges arise in scenarios with large occlusions in the object point cloud, contributing to most fail cases with objects such as Detergent and Washing Liquid. To further challenge our algorithm, we tested it with multiple shapes of the Hand Helping Tool, as shown in Figure 4.9. As the point cloud does not provide physical properties of the object, some grasp poses will fail on the flexible regions of the object, as indicated by the failure of the Hand and Hand Helping Tool, where the former failed due to flexible fingers and the latter failed due to a false joint, as shown in Figure 4.9 (3). In addition, objects requiring a precision grasp also pose challenges in real experiments. For example, our algorithm can grasp the Toy while standing but fails when lying on the table.

4.5 Summary and Discussion

We have introduced a rigid shape-matching grasp optimization method that utilizes the parallel AS-ICP algorithm. Demonstrating its versatility across various gripper types, the approach exhibits remarkable robustness against noisy and partial point clouds, achieving an average computation time of 0.926 seconds and an average success rate of 87.3% for the KG3 gripper in real experiments surpassing other point cloud-based analytic approaches. Our results underscore the potential of parallel optimization methods to enhance grasp quality and reduce computation time in robotic manipulation tasks. Furthermore, we also observed that optimizing gripper joint angles is not compulsory to ensure a successful grasp.

In this chapter, we propose leveraging SVGD to establish a robust distribution prior for our SGD-ICP optimization process. This hybrid approach not only reduces the reliance on highly precise initializations around the object but also enables the incorporation of prior knowledge and task-related constraints to generate more effective initial conditions. Moreover, it facilitates the generation of a more diverse set of grasps compared to the previous chapter, where the resulting distribution was heavily concentrated around similar poses.

However, relying solely on SVGD for the entire optimization process is not feasible. A major challenge lies in the definition and utilization of gripper preshapes, which were introduced to circumvent the expensive optimization of finger joint configurations. Each preshape corresponds to a distinct set of parameter initializations; when applying parallel SVGD, these preshape parameters can become disrupted, resulting in inappropriate attractive and repulsive forces that degrade the update quality. Furthermore, while the use of a Gaussian distribution for translation combined with a von Mises distribution for rotation helps to ensure a broad spread of particles, these assumptions do not necessarily provide meaningful directional guidance for the updates.

These limitations motivate the work presented in our next chapter, where we collect data from the current algorithm and train an EBM. The EBM is intended to eliminate the reliance on preshape initializations and supply more meaningful update directions than those derived from the previously assumed distributions.

CHAPTER 5

Stein Energy-Based Grasp Synthesis

5.1 Introduction

Grasp synthesis remains a challenging problem in robotic manipulation, particularly when dealing with previously unseen objects and partial observations. Traditional approaches relying on complete object models often struggle to generalize to the wide variety of everyday objects (Tang et al., 2025). To address this challenge, we propose a novel grasp synthesis method that integrates a data-driven Energy-Based Model (EBM) (Dawid and LeCun, 2023) with Iterative Closest Point (ICP) (Besl and McKay, 1992) within the Stein Variational Gradient Descent (SVGD) framework (Liu and Wang, 2016).

Point cloud completion is typically used for partially observed objects, but generating precise shapes remains challenging due to the vast variety of daily-life objects (Kiatos et al., 2021). Some data-driven methods can generate grasps using partial point clouds (Fang et al., 2023); however, these approaches require full object models to generate the necessary training data (ten Pas et al., 2017). In Chapter 4, we proposed an optimization-based method Annealed Stein ICP (AS-ICP) (Zhang et al., 2024) that can generate grasps from partial point clouds. When the gripper point cloud is captured in a fully opened configuration, the resulting optimized grasp poses can become misaligned due to the large aperture. Conversely, using a partially opened gripper point cloud may cause some feasible grasp poses to be missed. To balance these issues, multiple predefined gripper shapes are incorporated to ensure more robust and comprehensive grasp coverage.

To enhance generalization, provide a meaningful prior for SVGD, and reduce the reliance on accurate object models or predefined gripper preshapes, we propose a hybrid framework that integrates the complementary strengths of learning-based and optimization-based approaches. Learning-based methods excel at capturing complex patterns from sensory data and generalizing across diverse objects, while

optimization-based methods leverage analytical formulations to guarantee physically consistent and geometrically precise solutions. First, the EBM is trained using data generated from single-view point clouds through the AS-ICP algorithm. The EBM assigns low energy to successful grasps and high energy to unsuccessful ones, effectively capturing global grasp-quality information learned from data. Then, by integrating the EBM gradient into the SVGD optimization, we iteratively refine grasp poses to minimize the energy. This integration not only leverages global cues from the learned EBM but also exploits local geometric properties, enhancing the overall robustness of the grasp synthesis.

We conducted extensive experiments on a dataset of 57 objects from the Google Scanned Objects and 10 objects from the KIT dataset, totaling 5360 grasp attempts. Our method achieves an average success rate of 60.9%, outperforming baselines such as AnyGrasp (31.1%), Grasp Pose Detection (GPD, 48.4%) and AS-ICP (56.6%).

The main contributions of this chapter are:

- **Hybrid optimization framework:** We propose a novel hybrid approach that integrates a learning-based method (EBM) with an analytical method (ICP) within a SVGD optimization process, enabling robust grasp synthesis from partial point clouds.
- **Comprehensive ablation study:** We systematically investigate the impact of model architecture, loss functions, and dataset composition on grasping performance, providing insights into the contribution of each component within the hybrid framework.
- **Extensive experimental validation:** Through large-scale simulations, we demonstrate that coupling an optimization-based method with a data-driven approach significantly improves robustness, repeatability, and generalization compared to either method alone.

5.2 Related Work

Grasp synthesis approaches can be broadly categorized into two major groups (Kleeberger et al., 2020): optimization-based methods, which rely on analytic formulations of contact and object geometry, and data-driven methods, which leverage learning from large-scale datasets. A detailed related work of these approaches is provided in Section 3.2 and Section 4.2. In this chapter, we adopt two representative learning-based baselines for comparison: AnyGrasp (Fang et al., 2023), a generative method that produces grasps by directly modeling the grasp distribution, and GPD (ten Pas et al., 2017), a sampling-based method that evaluates large sets of candidate grasps to identify feasible solutions.

EBMs (Dawid and LeCun, 2023) have attracted significant attention for their flexibility in modeling complex probability distributions. However, training EBMs in high-dimensional spaces remains a challenge. The intractability of the partition function becomes increasingly problematic as the dimensionality grows (Du and Mordatch, 2019). Moreover, high-dimensional energy landscapes are highly non-convex, containing many local minima. This can trap optimization in a single region, leading to mode collapse, in which solutions concentrate on a single mode and ignore others. Local minima restrict exploration, while mode collapse reflects the failure to capture the full multimodal structure of the landscape. Grathwohl et al. (2020) demonstrates how SVGD (Liu and Wang, 2016) can be used to efficiently update particles within the EBM framework, mitigating some of these challenges. SVGD has proven its effectiveness in robotic applications, such as trajectory planning (Yin et al., 2024), point cloud registration (Maken et al., 2022a), and localization (Maken et al., 2022c).

In this chapter, we integrate an EBM with an ICP algorithm, using SVGD to combine global grasp quality cues (learned by the EBM) with local geometric alignment information (provided by ICP). This hybrid approach is designed to address the limitations of both optimization-based and data-driven methods, particularly in scenarios where only a partial point cloud is available.

5.3 Methodology

In this chapter, we address the problem of grasp synthesis for unknown objects using a hybrid optimization framework that integrates data-driven and analytical approaches. Given the gripper's inner surface point cloud, $\mathcal{S}$, and the object's point cloud captured from a single viewpoint, $\mathcal{R}$, our goal is to determine an optimal grasp pose, parameterized by $\theta = (t, q)$, where $t = x, y, z$ represents the 3D translation and $q = q_w, q_x, q_y, q_z$ represents the unit quaternion (with q_w being the scalar and q_x, q_y, q_z the vector part). The initial pose is refined using SVGD to minimize a hybrid cost function, combining geometric alignment and learned grasp-quality assessments.

Formally, we define our grasp optimization as follows:

$$\min_{\theta} \quad \mathcal{L}(y, x) + E(y, x) \quad s.t. \quad \text{dist}(y, \text{SDF}(x)) > 0, \tag{5.1}$$

where $x = T_\theta(\mathcal{S})$ denotes the transformed gripper surface point cloud under pose θ, and $y = \mathcal{R}$ denotes the object's observed point cloud. Here, $\mathcal{L}(y, x)$ represents the geometric loss computed via ICP matching (Maken et al., 2019). $E(y, x)$ is the learned energy term, computed from an EBM trained on grasp outcomes (success/failure) evaluated in simulation, reflecting learned probabilistic knowledge of grasp quality. The constraint ensures that the resulting grasp pose is physically plausible and collision-free, enforced by evaluating the distance from the object points to the gripper's SDF.

This formulation explicitly combines the analytical prior (ICP geometric alignment) with a data-driven posterior (EBM energy), thereby harnessing the complementary strengths of both analytic geometric reasoning and learned grasp outcomes. A summary of our method is provided in Figure 5.1.

5.3.1 Data Generation

To build our dataset, we use the AS-ICP algorithm (Zhang et al., 2024) on single-view point clouds of objects. Specifically, we select 57 distinct objects from the Google Scanned Objects dataset (Downs et al., 2022) and 10 distinct objects from the KIT dataset (Kasper et al., 2012), capture eight different point clouds per object with four different orientations (0, 90, 180 and 270 degrees), each with two camera elevations (0.1 and 07 m). For each point cloud, AS-ICP is applied to optimize collision-free grasp poses. These optimized poses are subsequently validated in the Isaac Gym simulator (Makoviychuk et al., 2021). A grasp is considered successful if the gripper maintains a firm hold on the object even after it is subjected to shaking. This process yields a labeled dataset of grasp poses.

5.3.2 Network Architecture

Our EBM is designed to capture the interaction between the object and the gripper by processing their point clouds and the associated grasp pose information. The model consists of two primary modules:

Point Cloud Encoder

We adopt a PointNet-based encoder (Qi et al., 2017), as point clouds are unordered sets of 3D points without a regular grid structure. PointNet handles this by using permutation-invariant operations, making it well-suited for extracting robust geometric features for grasping. The Point Cloud Encoder module applies five one-dimensional convolutional layers (with kernel size 1), each followed by batch normalization and Rectified Linear Unit (ReLU) activations. After these convolutions, a global max pooling

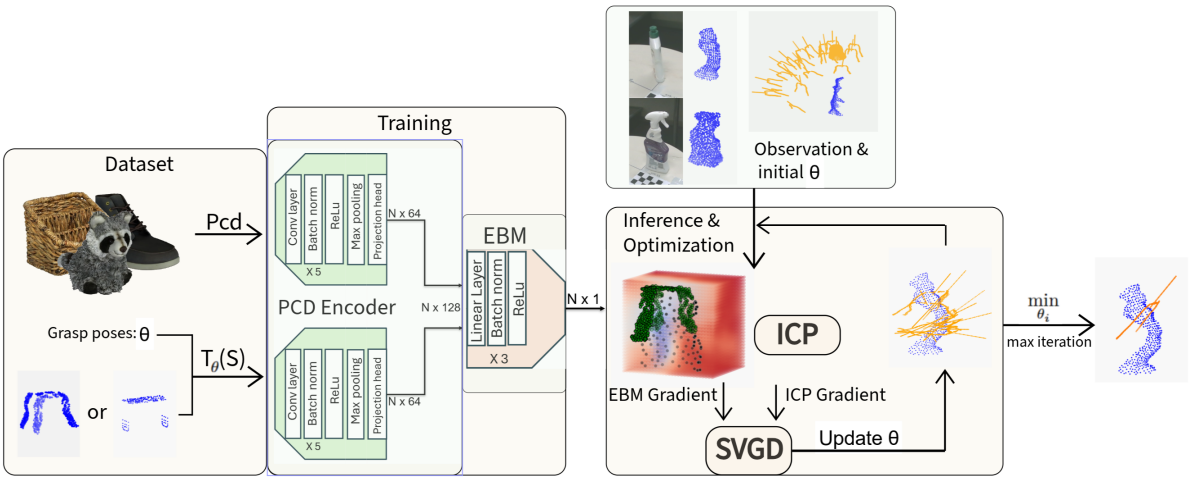

FIGURE 5.1. A brief summary of our algorithm. Both the object and gripper point clouds are fed into separate PCD Encoders to produce 64-dimensional features. These features are then concatenated and scored by the EBM to output a single energy value. The system uses SVGD to iteratively update the transformation parameters θ by leveraging gradients from the EBM and the cost function of ICP. The best pose is selected with minimum energy and matching error.

operation aggregates features across all points to produce a fixed-dimensional embedding. This embedding is further projected into a lower-dimensional feature space via a fully connected layer, layer normalization, and a ReLU activation.

Energy Function Network

The features are then fed into the Energy Function Network (Section 2.6.3), which is implemented as a multi-layer perceptron. This network comprises three fully-connected layers (interleaved with batch normalization and ReLU activations) and culminates in a single linear output that represents the energy. The energy score is not a binary decision variable but a continuous measure learned from data. Lower energy values indicate relatively more plausible grasp poses compared to higher ones, but there is no fixed threshold that separates "good" from "bad" grasps. Instead, EBMs are trained with contrastive learning: successful grasps are assigned lower energy, while failures are pushed toward higher energy. As a result, inference relies on ranking or optimization—selecting poses with minimal energy—rather than thresholding as in classifiers.

5.3.3 Network Training

Loss Function

We employ a contrastive loss function to train our EBM for grasp synthesis. The objective is to assign low energy values to positive grasp samples and high energy values to generated negative samples. The pairwise hinge loss is defined as follows (Dawid and LeCun, 2023):

$$L_{\text{hinge}} = E(y, x) - E(y, \hat{x}) + m, \tag{5.2}$$

where $E(y, x)$ is the energy for positive samples, $E(y, \hat{x})$ is the energy for negative samples, and m is a margin to enforce a clear separation between positive and negative energies.

We use a smooth, differentiable variant of the hinge loss as follows (Dawid and LeCun, 2023):

$$L = \log\left(1 + \exp\left(E^{+} - E^{-}\right)\right). \tag{5.3}$$

Here, E^{+} represents the energy of a positive sample, while E^{-} denotes the energy of a corresponding negative sample. In the context of grasping, even a small deviation in the positive grasp pose may lead to failure, resulting in many complex negative samples. Thus, preserving this specific pairing is crucial for effective model performance.

Data

The dataset generated in Section 5.3.1 is highly imbalanced, containing significantly more failure cases than successful grasp poses. This imbalance can negatively impact the performance of our EBM training. In Section 5.4, we discuss various data-processing strategies to mitigate such imbalances.

For training, we selected 50 objects. Given that most objects are non-symmetrical, each point cloud captured under different orientations and elevations is treated as a distinct group. From each group, we randomly select 80 successful grasp poses. In cases where a group contains fewer than 80 successful examples, we duplicate the available examples until the total reaches 80. Thus, the total number of positive data used for training is 32000.

Training

For training our EBM, we used a batch size $\mathcal{N}$ = 64 with the ADAM optimizer, setting the learning rate to 1×10^{-3} and applying a weight decay of 1×10^{-5}. The model was trained for 25 epochs, and the resulting model was then used to compute the gradient for SVGD. Our primary goal is not to achieve the best performance from the EBM but rather to demonstrate how the integration of a data-driven approach with optimization can effectively overcome the limitations inherent to each method.

5.3.4 SVGD Optimization

SVGD (Liu and Wang, 2016) is a particle-based variational inference method that approximates a complex target distribution using a set of interacting particles. Unlike traditional variational inference methods that assume a fixed parametric form for the approximate posterior, SVGD represents the target distribution non-parametrically as an empirical distribution over particles.

In the SVGD framework, the jth particle, θ_j, is updated as follows (Section 2.4):

$$\theta_j \leftarrow \theta_j + \eta\, \hat{\phi}^*(\theta_j), \tag{5.4}$$

where η is the step size and the optimal update direction $\hat{\phi}^*(\theta)$ is given by:

$$\hat{\phi}^*(\theta) = \frac{1}{K} \sum_{j=1}^{K} \left[k(\theta_j, \theta)\, \nabla \log p(\theta_j) + \nabla_{\theta_j} k(\theta_j, \theta) \right]. \tag{5.5}$$

Here, $k(\cdot, \cdot)$ is a positive definite kernel that couples the particles. The first term can be seen as an attractive force that pulls the particles toward regions of high probability density, and the second term acts as a repulsive force that prevents the particles from collapsing to a single mode, thereby encouraging diversity in the particle set.

In our work, we adapt SVGD within the AS-ICP framework (Zhang et al., 2024) for robust grasp synthesis. In Stein ICP (Maken et al., 2022a; Zhang et al., 2024), the gradient $\nabla \log p(\theta)$ in Equation (5.5) is replaced by the gradient of the SGD-ICP cost function with respect to the transformation parameters (Maken et al., 2019, 2022b):

$$\nabla \log p(\theta) = -\gamma(t) \left(N\, \bar{g}(\theta, \mathcal{M}) + \nabla_\theta \log p(\theta) \right), \tag{5.6}$$

where $\bar{g}(\theta, \mathcal{M})$ denotes the averaged gradient of the ICP cost function over a mini-batch $\mathcal{M}$, and N is a normalization factor equal to the number of particles. The prior gradient term, $\nabla_\theta \log p(\theta)$, is computed using Gaussian priors for translation and von Mises priors for rotation.

Additionally, the annealing schedule $\gamma(t)$ in AS-ICP periodically reduces the attractive force during optimization, enhancing particle diversity and accelerating convergence. However, AS-ICP relies on fixed analytic prior assumptions, which may not provide sufficient guidance for robust grasp optimization.

To address this limitation, we propose integrating a learned data-driven model within the Stein ICP framework. Specifically, we replace the prior gradient $\nabla_\theta \log p(\theta)$ in Equation (5.6) with the gradient derived from an EBM, denoted by $\nabla_\theta E(y, x)$:

$$\nabla \log p(\theta) = -\gamma(t) \left(N \, \bar{g}(\theta, \mathcal{M}) + \nabla_\theta E(y, x) \right), \tag{5.7}$$

We used the RBF kernel for translations and the dot product kernel for rotations. For the dot product kernel, the bandwidth parameter was set using the median heuristic:

$$\sigma = \sqrt{\frac{\text{median}(*)}{2 \log(n+1)}} \tag{5.8}$$

As shown in Figure 5.2, the optimization outcome is sensitive to the kernel bandwidth, particularly for the translation component. We empirically found the theoretically-motivated median heuristic to be unsuitable for translation in our domain, likely due to the high-dimensional, non-standard structure of the grasp pose space where pairwise distances between particles are not a reliable scale indicator. Within a manually tested range, performance variation was not substantial. Therefore, we selected a fixed bandwidth of $\sigma = 3$ for translation as it provided stable, empirically sound results across our test objects. For rotation, where results were less sensitive, we retained the median heuristic. We acknowledge that the optimal σ may be object- or scenario-dependent; a learning-based adaptive kernel is a promising direction but lies outside the scope of this thesis, which focuses on the integration framework itself.

We observed that the magnitudes of the learned EBM gradient $\nabla_\theta E(y, x)$ and the matching gradient $\bar{g}(\theta, \mathcal{M})$ differ significantly. To prevent one term from dominating the SVGD update and to ensure stable convergence, we introduced a dynamic scaling factor w:

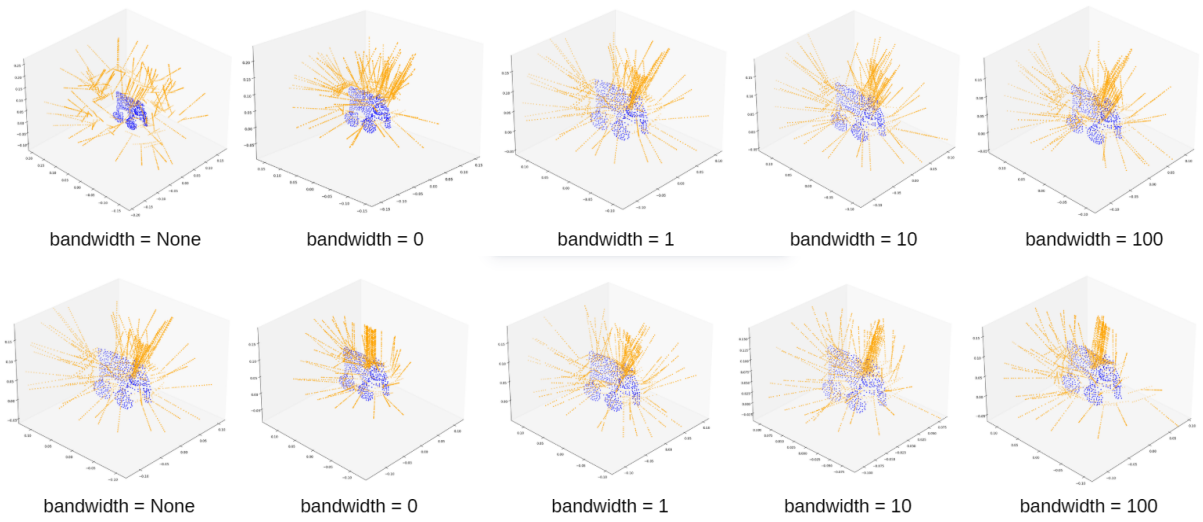

FIGURE 5.2. Influence of kernel bandwidth on SVGD optimization performance for translation (top) and rotation (bottom). The median heuristic (labeled "None") was found to be inadequate for the translation component. The performance within a fixed bandwidth range was relatively stable. We selected $\sigma = 3$ for translation as it performed reliably across our experimental objects, while retaining the median heuristic for rotation to minimize manual hyperparameter.

$$w = 100 \times \tanh\left(\frac{\|\nabla_\theta E(y,x)\|}{\|\bar{g}(\theta,\mathcal{M})\|}\right), \tag{5.9}$$

The $\tanh$ function provides a smooth, bounded scaling. The constant factor of 100 was empirically determined to approximately equalize the effective step sizes contributed by both gradient terms in our experimental setup, creating a balanced hybrid update. This factor is inherently tied to the scale of our specific EBM's outputs and geometric cost function. While network normalization or a more principled adaptive scheme could resolve this in future work, the current formulation provides a stable and effective solution for our proof-of-concept integration, demonstrating the viability of the hybrid approach.

The final combined gradient for parameter updates is then defined as:

$$\nabla \log p(\theta) = -\gamma(t)\left(w \times N\bar{g}(\theta,\mathcal{M}) + \nabla_\theta E(y,x)\right). \tag{5.10}$$

We utilize the ADAM optimizer with a learning rate of 1×10^{-2} for parameter updates. Although the optimization requires a large number of iterations to converge, further increases in the learning rate result in parameter divergence and numerical instability. Additionally, we note that the ICP matching gradient

provides strong guidance for translation parameters when the gripper is distant from the object, but its magnitude diminishes significantly as the gripper approaches the object's surface.

To take advantage of this behavior, we explicitly incorporate the matching gradient as a regularization term to accelerate early-stage translation optimization, gradually reducing its influence in later iterations. The final adaptive update rule for the transformation parameters is as follows:

$$\theta_{t+1} = \theta_t + \hat{\phi}_t^*(\theta_k) + \left(1 - \frac{k}{K}\right) \bar{g}(\theta_k, \mathcal{M}), \tag{5.11}$$

where $\hat{\phi}_t^*(\theta_k)$ denotes the SVGD update, and the adaptive term $\left(1 - \frac{k}{K}\right)$ progressively decreases the regularization effect of the matching gradient as the iteration index k progresses toward the max iteration K.

At the end of the optimization process, we evaluate the matching error for each particle by selecting the minimum non-collision error across a predefined set of gripper preshapes, following the procedure described in Zhang et al. (2024). The complete optimization procedure is summarized in Algorithm 5.1.

Algorithm 5.1: Stein Energy-Based Grasp

Input: Point cloud of: Gripper $\mathcal{S} = \{s_i\}_{i=1}^N$, Target Object $\mathcal{R} = \{r_i\}_{i=1}^M$, initial parameters $\Theta_0 = \{\theta_0^j\}_{j=1}^J$, number of iterations K

Output: θ that minimizes the sum of energy and matching error

while $k \leq K$ **do**
  for *each* $\theta_k^j \in \Theta_k$ *in parallel* **do**
    $\mathcal{S}_k^j \leftarrow$ Transform the source cloud with θ_k^j
    $\nabla_\theta E(y, x)^j$ and $\bar{g}(\theta_k, M)^j \leftarrow$ Compute gradients (5.10) and (5.5)
    Collision avoidance using SDF
    Update θ (5.11)
  end
  k = k + 1
end
for *each* $\theta^j \in \Theta_K$ **do**
  matching error$^j \leftarrow \min(\text{matching error}_{preshapes}^j)$
end
return $\theta = \text{argmin}_{\theta^j}\ (\text{norm}(\text{energy}) + \text{norm}(\text{matching error}))$

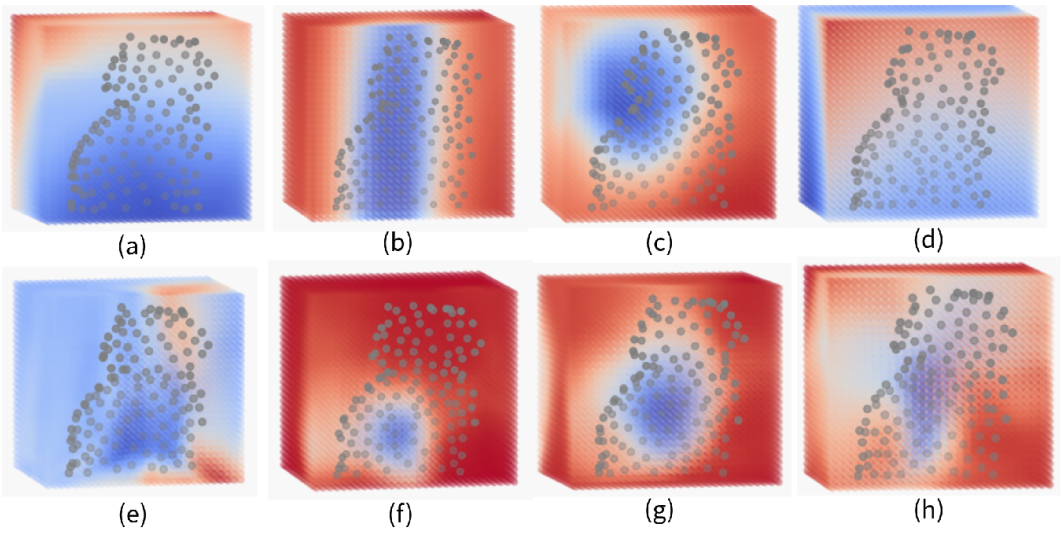

FIGURE 5.3. Energy plots for the EBM trained under different data processing schemes for a top-down grasp. In these plots, blue indicates low energy (favorable grasps), and red indicates high energy (unfavorable grasps). Panels (a) and (e) show results using the entire dataset. Panels (b) and (f) display results when training with only positive examples, paired with an equal number of negative examples sampled from the dataset. Panels (c) and (g) illustrate the outcome when negative examples are generated by uniform sampling around each positive instance. Panels (d) and (h) demonstrate the best performance, achieved by balancing the number of positive examples within each group (defined by object orientation and elevation). The top row corresponds to training without the gripper point cloud, whereas the bottom row includes the gripper point cloud, resulting in significantly improved performance.

5.4 Ablation

5.4.1 Data Processing

Our experiments reveal that both the collection and utilization of the training dataset have a significant impact on the performance of our EBM. After generating data using AS-ICP, we evaluated several data processing strategies:

Raw Data

Initially, we trained the EBM using the raw dataset. However, the dataset was imbalanced—containing 28555 positive data in a total of 72461 data—which hindered the contrastive loss from effectively distinguishing between positive and negative samples (see Figure 5.3(a) and (e)).

To mitigate the initial class imbalance, we retained all positive examples and randomly sampled an equal number of negative examples from the pool of unsuccessful optimization outcomes. However, this straightforward balancing strategy led to a collapsed, object-invariant model (see Figure 5.3(b) and (f)). We hypothesize this occurs due to a randomly paired negative and positive may originate from a completely different region of the grasp space (e.g., a left-sided grasp paired with a right-sided failure). This excessive, unstructured pose difference makes it difficult for the model to learn the subtle geometric distinctions that separate stable from unstable grasps, resulting in a flattened energy field that fails to generalize across objects.

Localized Negative Sampling

To overcome the above issue, we generated negative examples by uniformly sampling around each positive example. This method provided more localized negative samples, resulting in improved performance. It also generates samples with collisions, which the original dataset does not contain. However, the model consistently failed to capture the correct vertical positioning of the grasp due to dataset bias, where specific grasp configurations were overrepresented during training, as illustrated in (Figure 5.3(c) and (g)).

Group-based Sampling

Finally, we partitioned the dataset into groups according to the different orientations and elevations defined in Section 5.3.3. For each group, we randomly selected an equal number of positive examples—duplicating them if necessary to meet the required count—and then generated negative examples around each positive instance. This strategy focuses more on objects that are more difficult to grasp and improves overall energy distribution. (Figure 5.3(d) and (h)).

Additionally, we evaluated the effect of including the gripper's point cloud as input. Models trained with the gripper point cloud (bottom row of Figure 5.3) outperformed those trained without it (top row of Figure 5.3). Based on these results, we use the configuration in Figure 5.3(h) for training our final model.

It is important to note that even in the configuration shown in Figure 5.3(h), the regions of lowest energy are still shifted downward, resulting in a vertical bias that is lower than expected. This observation indicates that a top-down grasp selected solely based on the lowest energy criterion could result in the gripper intruding into the object. We also notice that the energy closer to the bottom of the objects tends

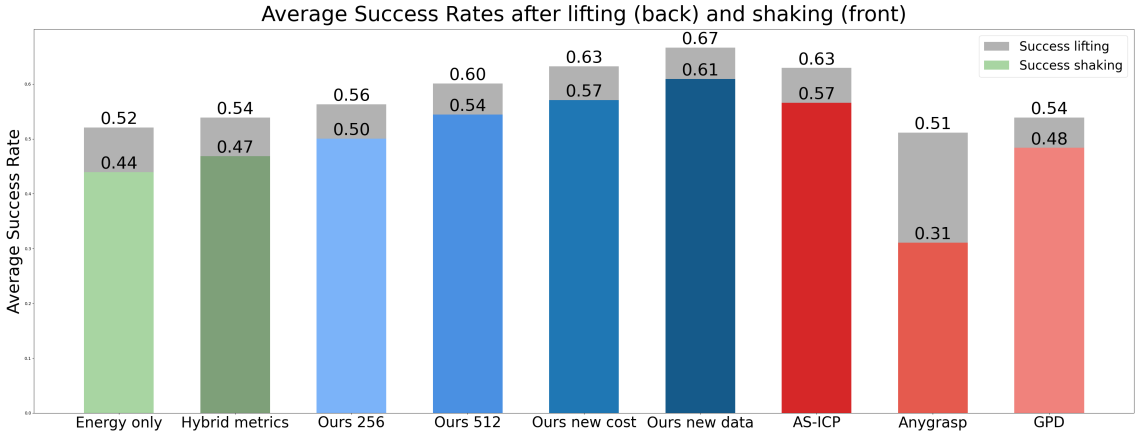

FIGURE 5.4. Plots of average success rate after lifting and shaking. The results demonstrate a clear trend of increasing performance as the model improves. Our method outperforms individual analytical and data-driven methods, other hybrid approaches, and baseline methods.

to be less accurate. We hypothesize that this discrepancy is due to a lack of sufficient positive examples for both top-down grasps and grasps close to the table surface in the dataset generated by the AS-ICP algorithm.

This analysis suggests that the quality of the data plays a more critical role than its sheer size when training an EBM. In other words, a well distributed dataset that better captures the full variability and nuance of successful grasp configurations will lead to a more accurate and reliable energy landscape, even if the overall number of training examples is lower.

5.4.2 Hybrid Model

The main focus of this chapter is to introduce and evaluate a hybrid optimization framework for grasp synthesis. While ICP and EBM are employed here as representative analytical and data-driven components, the framework is designed to be modular and easily extended to incorporate other methods. Our goal is not to train the single best possible EBM, but to demonstrate that as the learned model improves, overall grasping performance correspondingly increases.

To assess the framework, we conduct an extensive ablation study and compare it systematically against individual analytical (ICP) and learning-based (EBM, GPD, AnyGrasp) methods, as well as an established hybrid method (Hybrid metrics). A common hybrid baseline is to use ICP as a post-processing step for

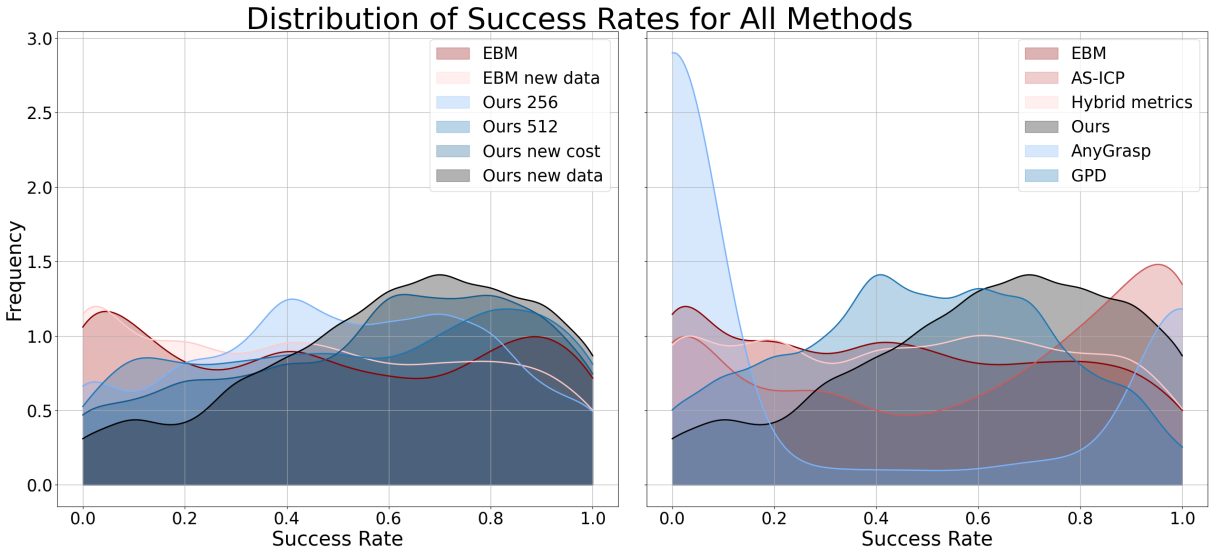

FIGURE 5.5. Kernel density estimates (KDEs) of success rates for all evaluated methods. The KDEs transform discrete success rate measurements into smooth, continuous probability distributions. The vertical axis shows the estimated probability density, where a higher density at a given success rate indicates that more experimental trials resulted in that performance level. **Left:** Set 1 compares various hybrid model variants, including ablations of different architectures and training data. **Right:** Set 2 compares key baselines and representative methods, including analytical, learning-based, and hybrid approaches. Our hybrid model's performance improves as the model becomes better, achieving higher and sharper peaks near a high success rate, indicating improved reliability and overall performance compared to baseline methods in Set 2.

Method	Mean (↑)	Std (↓)	≥ 0.1 (↑)	≥ 0.5 (↑)	≥ 0.9 (↑)
Energy only	0.439	0.318	0.756	0.381	0.056
Hybrid metrics	0.468	0.312	0.795	0.431	0.054
Ours 256	0.500	0.285	0.864	0.451	0.057
Ours 512	0.544	0.310	0.851	0.513	0.093
Ours new cost	0.571	0.291	0.890	0.586	0.082
Ours new data	**0.609**	0.270	**0.920**	**0.623**	0.095
AS-ICP	0.566	0.367	0.784	0.567	0.192
AnyGrasp	0.311	0.440	0.347	0.306	**0.254**
GPD	0.484	**0.259**	0.873	0.424	0.022

TABLE 5.1. Comparison of methods by group-level mean, standard deviation, and fraction of groups exceeding selected success thresholds. The arrows indicate whether higher (↑) or lower (↓) values are preferred. Thresholds (0.1, 0.5, 0.9) denote the fraction of groups whose success rate is above 10%, 50%, and 90%, respectively, providing a sense of performance across low, moderate, and high success regimes.

learning-based grasps. However, as shown in our earlier work Zhang et al. (2024), if the initial poses from the learned model are poor, simple ICP post-processing does not substantially improve performance.

Our ablation study examines the influence of both model architecture and dataset composition. We train several variants of the hybrid method, spanning two datasets:

- **Dataset 1:** A set of 32,000 positive examples generated by AS-ICP—effectively corresponding to a uniform prior with ICP providing the posterior. Models trained on this dataset include:
  - **EBM**: Baseline with a 256-dimensional gripper encoder, evaluated using energy only.
  - **Ours 256**: Same model, evaluated using both energy and matching error.
  - **Ours new cost**: Same as above, but with two additional loss terms during evaluation.
  - **Ours 512**: Higher-capacity variant with a 512-dimensional encoder and an additional combined encoder for point cloud features, evaluated using energy and matching error.
- **Dataset 2:** A refined set of 3,280 positive examples generated by grasps from the improved "Ours new cost" model:
  - **EBM new data**: 256-dimensional encoder, evaluated on energy only.
  - **Ours new data**: Same model, evaluated on energy, matching error, and the additional loss terms.

The additional loss terms used are defined as follows:

$$\text{PartialError}(R_i) = \text{sigmoid}\left(10\left(\left|(R_i\mathbf{e}_z)_y\right| - 0.5\right)\right), \quad i = 1, \ldots, N \tag{5.12}$$

where $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$, R_i is the i-th rotation matrix, and $\mathbf{e}_z = [0, 0, 1]^T$. This loss serves as a proxy for grasp confidence under partial observation: it penalizes horizontal grasp poses, which are more likely to attempt a grasp from an unobserved region.

$$\text{PointInGrasp} = \exp\left(-\frac{1}{10}\sum_{i=1}^{N}\left(p_i \in \mathcal{B}\right)\right) \tag{5.13}$$

where $\mathcal{B}$ is the volume within the grasp (i.e., the region enclosed by the gripper). This loss encourages at least part of the target object to be within the gripper configuration.

Figure 5.4 and the left panel of Figure 5.5 show a clear trajectory of improvement. The initial EBM trained on AS-ICP data exhibits a broad spread of success rates, reflecting limited robustness and generalization. Adding matching error and additional loss terms progressively shifts the distribution toward higher, more peaked success rates—indicating greater consistency. Increasing model capacity and refining the dataset

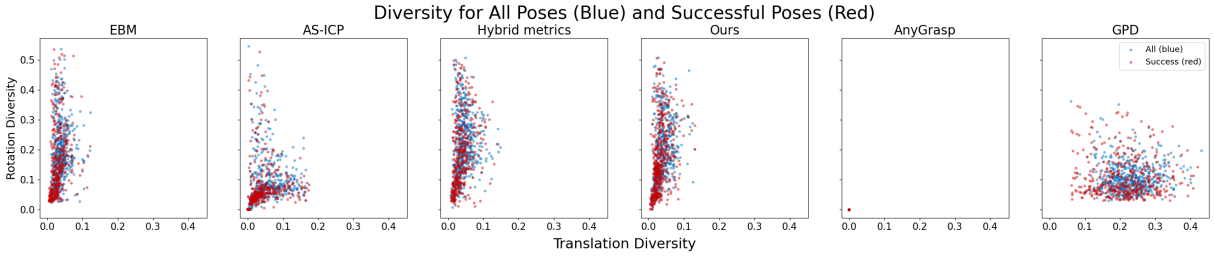

FIGURE 5.6. Comparison of grasp diversity across different methods. Translation diversity (x-axis, up to 98th percentile) and rotation diversity (y-axis) are shown for all attempted (blue) and successful (red) grasps, each are computed separately. AnyGrasp produces identical poses, while GPD yields high translation but low rotation diversity. AS-ICP results are concentrated at low diversity, and EBM at high rotation diversity.

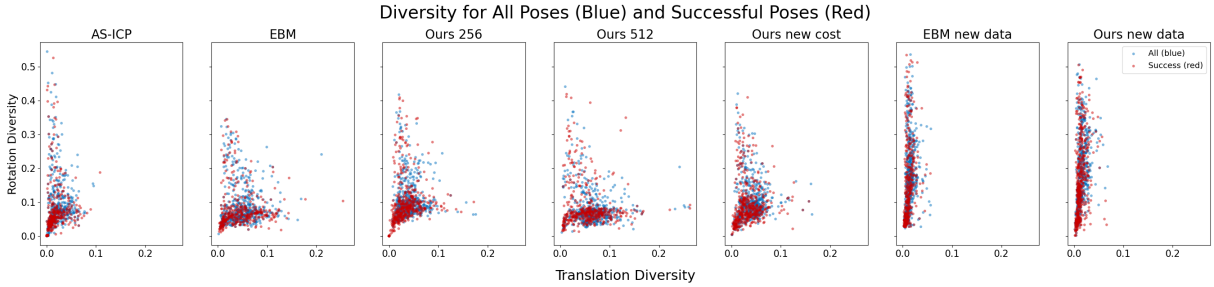

FIGURE 5.7. Comparison of grasp diversity for various hybrid models. The EBM trained on the initial dataset closely follows the distribution of ICP and exhibits high translation diversity. In contrast, the model trained on the second dataset shows lower translation diversity but substantially higher rotational diversity. Hybrid models overall tend to mirror the distribution of the learned EBM.

further improves performance, with “Ours new data” achieving the highest frequency near high success rates.

In contrast, the right panel of Figure 5.5 compares our hybrid method against AS-ICP, the Hybrid metrics baseline, and learning-based methods (energy only, GPD, AnyGrasp). Analytical methods such as AS-ICP show broad distributions and many low-success cases. AnyGrasp exhibits extreme cases similar to AS-ICP, while GPD produces a distribution more similar to our best hybrid models.

Both figures demonstrate that our hybrid framework achieves higher mean success rates (Figure 5.4) and tighter high-performance distributions, indicating greater reliability. However, Table 5.1 shows that for the very highest success rate thresholds, our method yields fewer near perfect (90%) cases, suggesting that the inherent stochasticity of the learned component can limit absolute performance in rare cases.

In Figure 5.6 and Figure 5.7, we present the distribution of samples in terms of rotation and translation diversity. Diversity is computed as the mean pairwise distance between each successful grasp pose, using standard deviation for translation and dot product for rotation. The rotation term is weighted by translation distance, i.e., rotation differences are emphasized more for nearby poses. Because this is computed from the best grasps of ten trials, it reflects the repeatability of each method.

Figure 5.7 shows the diversity distributions for different hybrid models. Their repeatability patterns largely mirror those of the underlying learning-based component. In Figure 5.6, the sampling-based method (GPD) exhibits high diversity, particularly in translation, reflecting its exploratory sampling strategy. In contrast, the generative approach (AnyGrasp) demonstrates extremely high repeatability, often producing nearly identical grasp poses. Our best-performing model (Ours new data) strikes a balance between stability and variability: it consistently selects similar grasp locations on the object while maintaining substantial variation in approach angles, resulting in higher rotation diversity.

The ablation study confirms that each architectural and data refinement step yields tangible performance gains. The proposed hybrid optimization framework consistently outperforms its individual components as well as the established baselines (GPD and AnyGrasp) under the tested conditions of grasping isolated objects from partial point clouds. The diversity analysis demonstrates that the hybrid method effectively balances repeatability with pose diversity. Overall, these results validate the proposed deeply integrated hybrid approach as a robust and effective strategy for grasp synthesis in the target domain.

5.5 Experiments

5.5.1 Simulation

We selected 67 objects from the Google Scanned Objects dataset and captured their point clouds in the Isaac Gym simulator (Makoviychuk et al., 2021). Some objects were rescaled to ensure that the Franka robotic arm could grasp them securely—large enough for a stable hold, yet not so small as to fit entirely within the gripper. All training and optimization experiments were conducted on a laptop equipped with an RTX 2070 GPU. The grasp pose's origin is aligned with the known object origin. In real experiments, the origin is approximated by the centroid of the segmented object point cloud, computed from a stable reference view to avoid inconsistencies caused by occlusions and changing viewpoints.

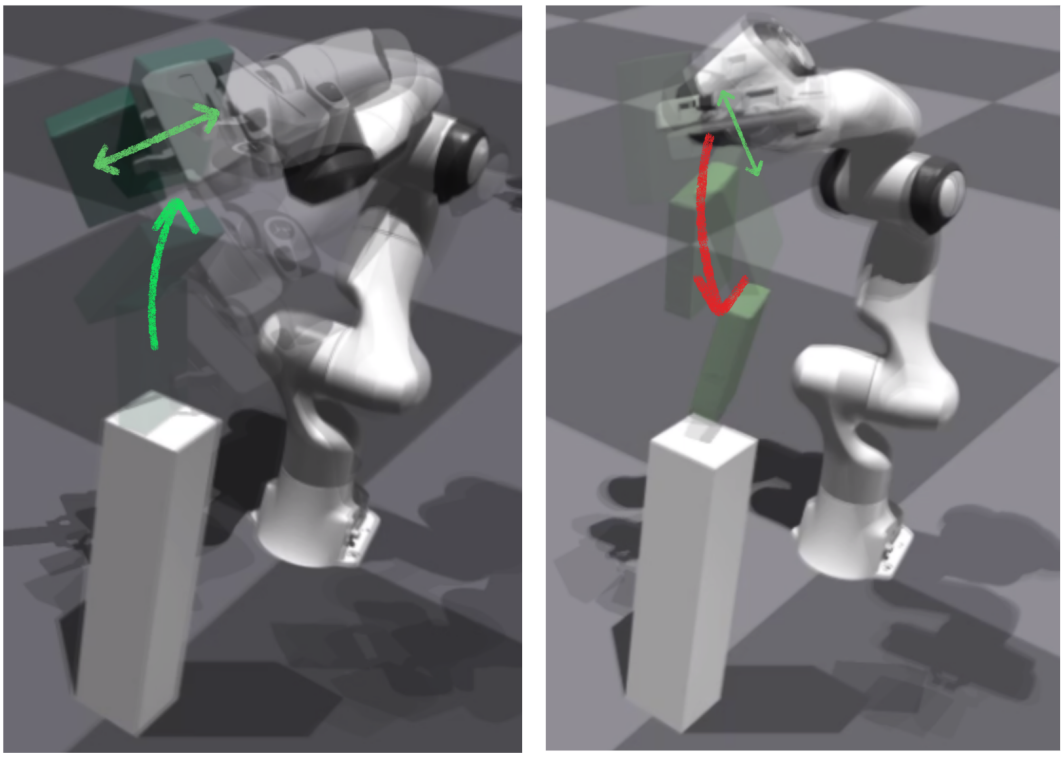

FIGURE 5.8. Examples of grasp robustness under dynamic testing. Left: A successful grasp that withstands vigorous shaking. Right: A grasp that succeeds in lifting but fails when shaken, with the object falling from the gripper (trajectory shown in red). The green arrows indicate the shaking trajectory applied in both tests.

For evaluation, we compared our approach against three baseline methods—AnyGrasp (Fang et al., 2023), Grasp Pose Detection (GPD) (ten Pas et al., 2017), and AS-ICP (Zhang et al., 2024)—all of which generate grasp poses from partial point clouds. While Average Precision (AP) (Fang et al., 2020) is a standard metric for assessing grasp quality, real-world grasping typically allows only a single attempt. Therefore, for each method, we generated a set of candidate grasps, selected the top-scoring pose, and executed it. This process was repeated ten times to compute the average success rate.

It is worth noting that the maximum achievable success rate in our setup is inherently below 100%, due to both object size constraints and the fixed approach angle for partial observations. As illustrated in Figure 5.8, even two nearly identical grasp poses on a large box can yield different outcomes—failure after shaking due to weight distribution.

Simulation results for the 67 objects are shown in Figure 5.9. Our method achieved an average success rate of 60.9% over 5,360 grasps, outperforming AnyGrasp (31.1%), GPD (48.4%), and AS-ICP (56.6%). We also achieved higher object-based minimum and maximum success rates, as indicated by the error

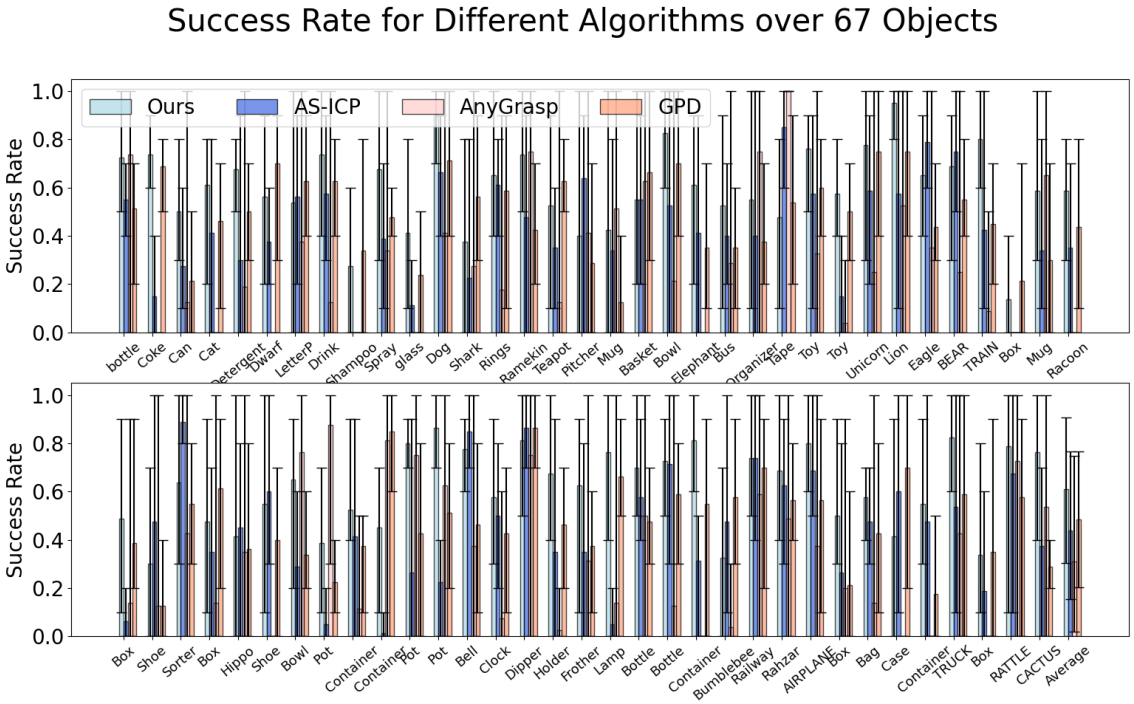

FIGURE 5.9. Simulation results comparing our approach with baseline methods. Our method achieved an average success rate of 60.9%, outperforming AnyGrasp (31.1%), GPD (48.4%), and AS-ICP (56.6%). We also achieved higher object-based minimum and maximum success rates, as indicated by the error bars.

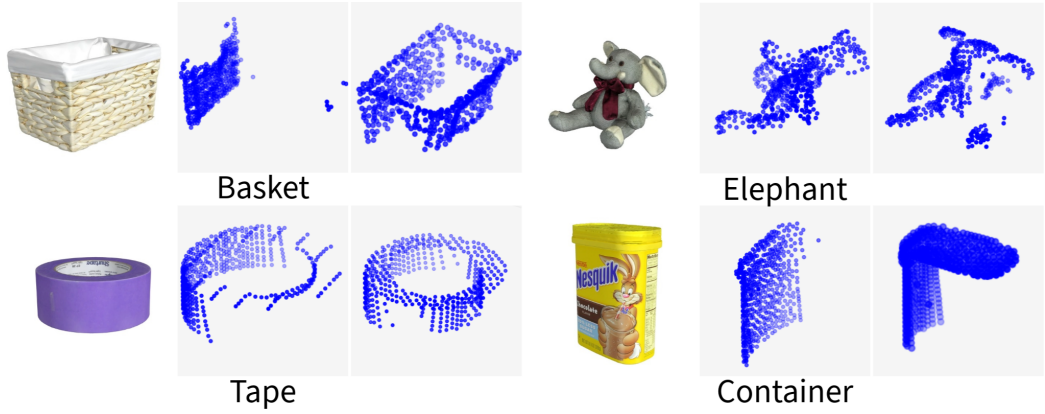

FIGURE 5.10. Examples of objects and their corresponding point clouds from different viewpoints, which can differ significantly.

bars. However, conventional object-level analysis is less meaningful in our setting, since partial point clouds from different viewpoints can differ significantly (Figure 5.10). For example, Figure 5.11 shows that there is no obvious visual distinction between object categories with less than 50% success rate and those above 70%, suggesting that occlusion and viewpoint-specific visibility are key factors.

FIGURE 5.11. Examples of objects used in simulation that our methods achieves less then 50% success rate (left) and more than 70% (right). There is no obvious visual distinction between object categories with less than 50% success rate and those above 70%, suggesting that occlusion and viewpoint-specific visibility are key factors.

Our approach also demonstrates strong generalization ability: although the model was trained on only the first 41 objects, it achieved a higher success rate on unseen objects (64.6%) than on the training set (58.6%).

Computation Step	**Time (s)**
Overall computation time (hybrid)	8.77
Model loading time	1.17
Energy gradient computation time	0.458
Overall computation time (original ICP)	2.42

TABLE 5.2. Computation time breakdown for the hybrid method compared to the original AS-ICP.

As summarized in Table 5.2, the current computation time (8.77 seconds) is considerably higher than that of the original AS-ICP (2.42 seconds), despite the simplifications of using a single preshape and fewer iterations. Likely dominant factors include:

- **Iterative Data Transfer:** The sequential querying of the EBM and gradient computation within the SVGD loop may involve repeated and suboptimal data transfers between the CPU and GPU.
- **Python Overhead:** The research prototype is implemented in Python, which can introduce significant interpretation and memory management overhead for tight loops compared to optimized C++ code.
- **Auxiliary Operations:** Costs from tasks such as point cloud preprocessing, coordinate transformations, and repeated memory allocations that are not captured in the high-level timing.

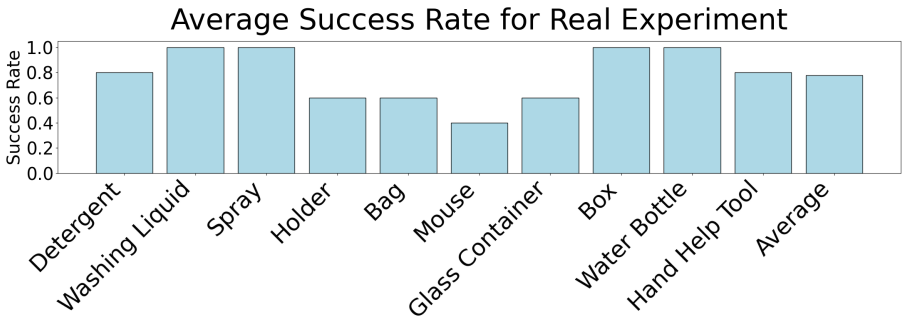

FIGURE 5.12. Average success rate of five grasps for ten objects, where our method achieves an average success rate of 72 percent.

Thus, the current timing reflects a functional research prototype rather than an optimized deployment runtime. The performance gap highlights a clear and actionable path for future engineering work, where profiling, code optimization, and potential CUDA kernel fusion could dramatically reduce latency, bringing the hybrid method's efficiency in line with its superior grasp quality.

5.5.2 Real Experiment

To validate our algorithm, we conducted experiments using a KG3 gripper mounted on a Kinova arm. Ten common household objects were selected for evaluation. To ensure perceptual variety, each object was placed in five distinct orientations. A wrist-mounted Intel RealSense D405 camera captured the point cloud for each scenario. The raw data was then pre-processed through a standard pipeline involving bounding-box cropping and voxel-grid down-sampling. Overall, we achieved a 72% success rate across fifty grasps, as summarized in Figure 5.12. Figure 5.13 displays a single view of each object along with the corresponding point clouds and generated grasps.

Our algorithm demonstrates robust performance with noisy and partial point clouds and adapts well to different grippers. However, its performance is bounded by two distinct classes of limitations: fundamental sensing constraints and inherent algorithmic constraints. First, the method's effectiveness is fundamentally tied to input data fidelity—a constraint shared by all geometry-driven grasp synthesis. When the perceived geometry is critically corrupted, optimization operates on misleading information. This explains failures like the Glass Container (g), where transparency causes severe, inherent sensor noise. Second, beyond these sensing limits, our method faces specific algorithmic challenges. The lower success rates for the Holder (d) and Mouse (f) highlight two issues:

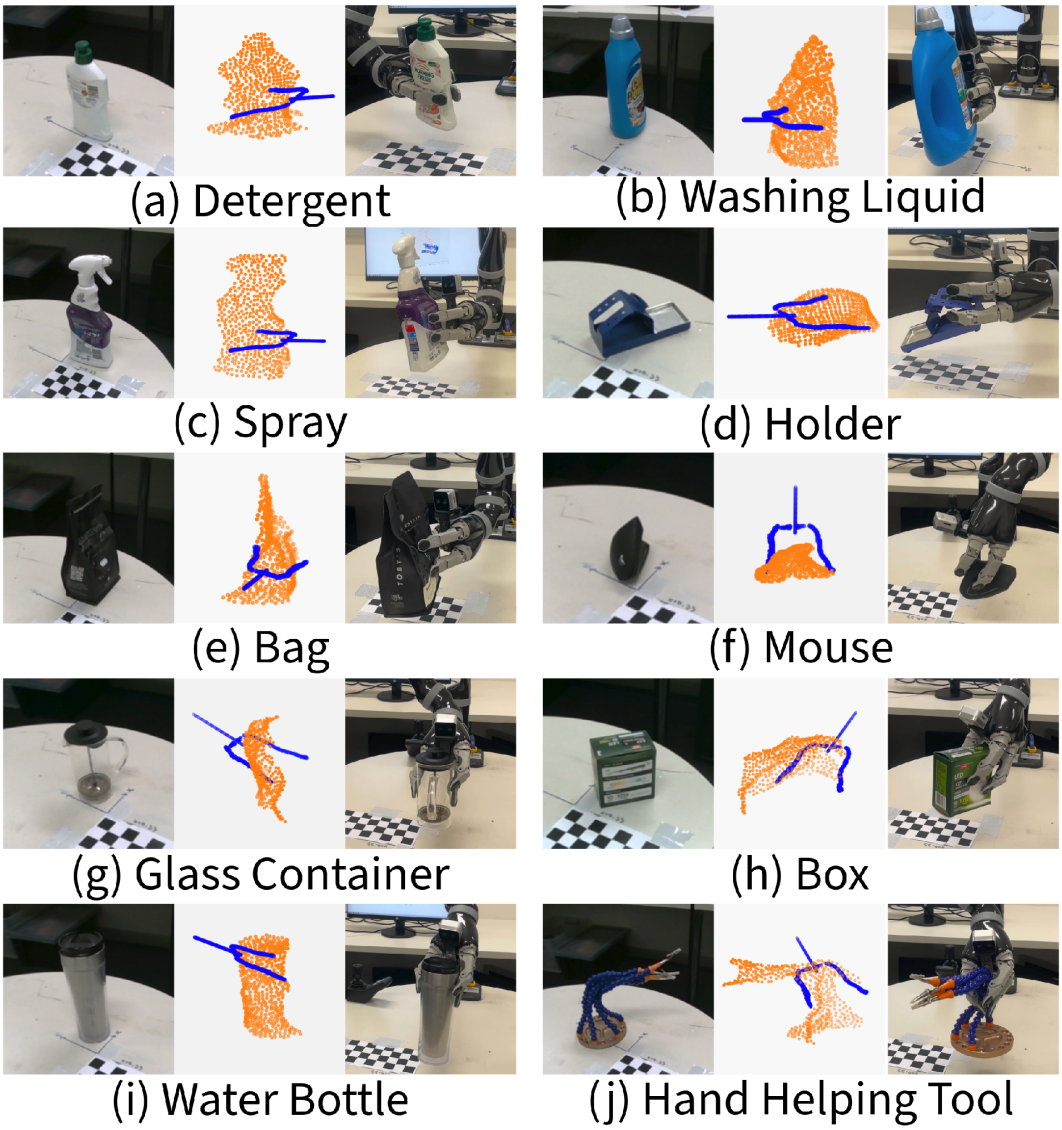

FIGURE 5.13. Illustration of the real experiment. For each object, the left panel shows the camera's view, the middle panel presents the scanned point cloud with the grasp pose predicted by our method, and the right panel depicts the KG3 gripper executing the actual grasp.

- **Contextual Noise:** Performance degrades with "severe noise near the table," where the algorithm must distinguish object geometry from environmental artifacts.
- **Inherited Model Bias:** The EBM is trained on data generated by AS-ICP. It therefore inherits and can amplify its predecessor's biases, such as a preference for power grasps and reduced efficacy on flat or very small objects.

Thus, while the framework is robust to structured incompleteness (partial views), it remains susceptible to unstructured corruption (severe noise) and the specific geometric biases embedded in its learned components.

In addition, we performed 20 grasps using AS-ICP on the detergent, washing liquid, holder, and hand-helping tool, and found that the average success rate for each object was the same as that achieved by our method.

5.6 Summary and Discussion

In this chapter, we introduced a hybrid grasp synthesis framework that integrates an EBM with ICP to generate reliable grasp poses from partial point cloud data. The EBM is trained on data produced by the AS-ICP algorithm and its gradients are incorporated within a SVGD optimization scheme. This design enables our method to capture both global grasp quality and fine-grained local geometric details.

Through ablation studies, we demonstrated the critical role of structured datasets in training EBMs and showed that the hybrid formulation consistently outperforms each component when used in isolation. Experimental results highlight that our method achieves high success rates on both seen and unseen objects, surpassing several state-of-the-art baselines. Overall, the proposed framework provides a robust and generalizable approach to grasp synthesis, combining the complementary strengths of analytic optimization and data-driven modeling.

CHAPTER 6

Conclusions and Future Work

This thesis addressed the challenge of robust grasp synthesis from partial observations by formulating it as a rigid shape-matching task between gripper and object point clouds. We developed a hybrid optimization framework that integrates analytical ICP with a learning-based EBM within a SVGD pipeline. This design leverages fine-grained local geometry for precise contact alignment while using the learned prior to infer global grasp quality from partial data. The framework demonstrated strong performance in single-view scenarios, successfully generalizing to object shapes do not present in the EBM's training set and surpassing two established baselines (GPD and AnyGrasp) under the tested conditions. Furthermore, parallel computation was incorporated, improving the efficiency of the optimization. The work establishes a foundation for deeply integrated hybrid grasping methods, with the core limitations identified—sensitivity to severe perceptual corruption and inherited data biases—providing clear directions for future work.

This concluding chapter summarizes the key contributions of the thesis and outlines promising avenues for future research.

6.1 Summary of Contributions and Related Chapters

Grasping as Rigid Shape Matching We formulated grasp synthesis as rigid shape matching using point clouds of both the object and gripper. This allowed us to utilize the gripper's SDF for efficient collision avoidance, reducing dependence on precise object models. Additionally, we introduced a parallelized SGD-ICP algorithm, significantly accelerating optimization compared to existing approaches.

Grasping with Annealed Stein ICP We proposed integrating SVGD with our SGD-ICP optimization to provide a robust prior distribution. This hybrid approach reduced sensitivity to initializations around the object and enabled incorporating prior knowledge and task-specific constraints. Crucially, it generated

a diverse distribution of grasp poses rather than repetitive outcomes typically observed in conventional optimization.

Stein Energy-Based Grasp Synthesis We developed a novel hybrid framework combining learning-based (EBMs) and analytical (ICP) approaches within the SVGD optimization pipeline, specifically tailored for robust grasp synthesis from partial point clouds. We provided a detailed analysis of how dataset quality and selection impact the learned energy function, highlighting that curated datasets often yield superior performance compared to larger but less focused datasets. Extensive simulation results and ablation studies demonstrated that blending optimization methods with data-driven models significantly enhances generalization and overall grasp quality.

6.2 Future Work

6.2.1 Task-Oriented Grasp Initialization and Evaluation Metrics

Current grasp synthesis methods often rely on general metrics such as force closure, friction, or learned scoring functions, with initial grasp sampling typically centered around objects. However, this approach may yield grasp poses unsuitable for specific manipulation tasks. For example, placing a bottle vertically or horizontally on a shelf requires distinctly different grasps. Although Chapter 4 introduced potential ways to identify better initial sampling regions, a clear integration with task-specific trajectory planning remains an open challenge. A promising future direction is combining grasp synthesis explicitly with task-driven trajectory planning, where trajectory planners provide feasible approach directions used as initialization regions. Developing evaluation metrics that jointly consider grasp quality and task compatibility is essential for more effective robotic manipulation.

6.2.2 Data Collection and Processing for Learning

Chapter 5 highlighted the significance of data selection, preprocessing and data bootstrapping, illustrating that increased dataset size alone does not guarantee improved model performance. This observation raises critical questions: What constitutes an optimal dataset? Is it possible to achieve high performance using fewer but carefully selected examples? Furthermore, can we develop a model capable of actively selecting and weighting data during training to optimize learning efficiency? Future work should explore adaptive data evaluation strategies, particularly in the context of EBMs, which prioritize meaningful geometric

relationships over random sampling. Developing algorithms that dynamically assess data relevance and pair positive and negative samples based on geometric properties could significantly enhance model learning.

6.2.3 Sensor Feedback Control for Grasping

The core contribution of this thesis is an offline optimization process that computes a high-quality initial grasp pose from a static observation. Executing this full optimization during the dynamic grasping motion is computationally prohibitive. However, this does not preclude real-time adaptation. The framework can transition from a convergence-seeking process to a correction-seeking one. Specifically, the SVGD optimizer could be applied for a fixed, small number of steps using the latest sensor data. This would not recalculate a global optimum but would provide efficient, gradient-based corrections to refine the execution online. These targeted corrections could provide meaningful guidance for training or augmenting Reinforcement Learning (RL) policies, offering a physically informed bias that could significantly improve sample efficiency and policy robustness in learning-based control.

Bibliography

2013. Grasping in robotics. [Online; accessed 2026-01-11].

K. S. Arun, T. S. Huang, and S. D. Blostein. 1987. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700. [Online; accessed 2026-01-11].

Barrett Technology. 1990–. Barrett hand: Advanced robotic gripper. `https://www.barrett.com/robotic-hand/`. [Online; accessed 2026-01-11].

Janusz Będkowski and Andrzej Masłowski. 2012. GPGPU computation in mobile robot applications. *International Journal on Electrical Engineering and Informatics*, 4(1):15–26. [Online; accessed 2026-01-11].

José M. Bernardo and Adrian F. M. Smith. 1994. *Bayesian Theory*, volume 586 of *Wiley Series in Probability and Statistics*. John Wiley & Sons, Chichester, UK. [Online; accessed 2026-01-11].

Paul J. Besl and Neil D. McKay. 1992. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–606. SPIE, Boston, MA, USA. [Online; accessed 2026-01-11].

Antonio Bicchi. 2000. Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on Robotics and Automation*, 16(6):652–662. [Online; accessed 2026-01-11].

H. Biggie, A. Beathard, and C. Heckman. 2023. BO-ICP: Initialization of iterative closest point based on bayesian optimization. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14774–14780. IEEE, London, UK. [Online; accessed 2026-01-11].

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, USA. [Online; accessed 2026-01-11].

Jan Blumenkamp, Steven Morad, Jennifer Gielis, and Amanda Prorok. 2023. CoViS-Net: A cooperative visual spatial foundation model for multi-robot applications. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, pages 1233–1245. [Online; accessed 2026-01-11].

Boston Dynamics. 2024. An electric new era for Atlas. https://bostondynamics.com/blog/electric-new-era-for-atlas/. [Online; accessed 2026-01-11].

Randy C. Brost. 1991. *Analysis and Planning of Planar Manipulation Tasks*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA. [Online; accessed 2026-01-11].

Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, and Wolfram Burgard. 2005. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Intelligent Robotics and Autonomous Agents Series. MIT Press, Cambridge, MA. [Online; accessed 2026-01-11].

Vassilios N. Christopoulos. 2010. *Characteristic Information Required for Human Motor Control: Computational Aspects and Neural Mechanisms*. Ph.D. thesis, University of Minnesota, Minneapolis, MN. [Online; accessed 2026-01-11].

Matei Ciocarlie, Corey Goldfeder, and Peter K. Allen. 2007. Dimensionality reduction for hand-independent dexterous robotic grasping. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3270–3275. IEEE, San Diego, CA, USA. [Online; accessed 2026-01-11].

F. D'Angelo and V. Fortuin. 2020. Annealed stein variational gradient descent. In *Proceedings of the 3rd Symposium on Advances in Approximate Bayesian Inference (AABI 2020)*, pages 1–13. PMLR, Online. [Online; accessed 2026-01-11].

Naël Daoud, Jean-Pierre Gazeau, Saïd Zeghloul, and Marc Arsicault. 2011. A fast grasp synthesis method for online manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 421–427. IEEE, San Francisco, CA, USA. [Online; accessed 2026-01-11].

A. Dawid and Y. LeCun. 2023. Introduction to latent variable energy-based models: A path towards autonomous machine intelligence. arXiv preprint arXiv:2306.02572. Les Houches Summer School Lecture Notes 2022 Preprint. [Online; accessed 2026-01-11].

Dan Ding, Yun-Hui Lee, and Shuguo Wang. 2001. Computation of 3-d form-closure grasps. *IEEE Transactions on Robotics and Automation*, 17(4):515–522. [Online; accessed 2026-01-11].

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. 2022. Google scanned objects: A high-quality dataset of 3D scanned household items. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, Philadelphia, PA, USA. [Online; accessed 2026-01-11].

Yilun Du and Igor Mordatch. 2019. Implicit generation and modeling with energy-based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3603–3613. NeurIPS Foundation, Vancouver, Canada. [Online; accessed 2026-01-11].

Özge Ekrem and Bekir Aksoy. 2023. Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm. *Engineering Applications of Artificial Intelligence*, 122:106099. [Online; accessed 2026-01-11].

Y. Fan, H.-C. Lin, T. Tang, and M. Tomizuka. 2018a. Grasp planning for customized grippers by iterative surface fitting. In *2018 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1396–1401. IEEE, Munich, Germany. [Online; accessed 2026-01-11].

Y. Fan, T. Tang, H.-C. Lin, and M. Tomizuka. 2018b. Real-time grasp planning for multi-fingered hands by finger splitting. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5340–5346. IEEE, Madrid, Spain. [Online; accessed 2026-01-11].

Y. Fan and M. Tomizuka. 2019. Efficient grasp planning and execution with multifingered hands by surface fitting. *IEEE Robotics and Automation Letters*, 4(2):720–727. [Online; accessed 2026-01-11].

Y. Fan, X. Zhu, and M. Tomizuka. 2019. Optimization model for planning precision grasps with multi-fingered hands. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1088–1094. IEEE, Macau, China. [Online; accessed 2026-01-11].

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. 2020. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11441–11450. IEEE, Seattle, WA, USA (Virtual). [Online; accessed 2026-01-11].

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. 2023. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(4):3029–3046. [Online; accessed 2026-01-11].

Carlo Ferrari and John Canny. 1992. Planning optimal grasps. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2290–2295. IEEE, Nice, France. [Online; accessed 2026-01-11].

Franka Emika GmbH. 2016–. Franka emika: The next generation of collaborative robots. `https://www.franka.de`. [Online; accessed 2026-01-11].

X. Gao, W. Tang, P. Xie, and G. Wang. 2022. Novel representation of robotic grasp detection based on residual hourglass architecture. In *2022 International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 49–54. IEEE, Dalian, China. [Online; accessed 2026-01-11].

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA. [Online; accessed 2026-01-11].

Jackson Gorham and Lester Mackey. 2015. Measuring sample quality with stein's method. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 226–234. NeurIPS Foundation, Montreal, Canada. [Online; accessed 2026-01-11].

Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. 2021. RGB matters: Learning 7-DoF grasp poses on monocular RGBD images. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13459–13466. IEEE, Xi'an, China. [Online; accessed 2026-01-11].

GraspNet. 2024. Anygrasp sdk. `https://github.com/graspnet/anygrasp_sdk`. [Online; accessed 2026-01-11].

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. 2020. Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 3732–3743. PMLR, Online (originally Vienna, Austria). [Online; accessed 2026-01-11].

Li Han, Jeffrey C. Trinkle, and Zexiang Li. 2000. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*, 16(6):663–674. [Online; accessed 2026-01-11].

Honda Motor Co., Ltd. 2001. Honda introduces ASIMO humanoid robot for rental business. `https://global.honda/en/newsroom/news/2001/c011112-eng.html`. [Online; accessed 2026-01-11].

Radu Horaud and Fadi Dornaika. 1995. Hand-eye calibration. *The International Journal of Robotics Research*, 14(3):195–210. [Online; accessed 2026-01-11].

Berthold K. P. Horn. 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642. [Online; accessed 2026-01-11].

Itseez. 2015. Open source computer vision library. `https://github.com/itseez/opencv`. [Online; accessed 2026-01-11].

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233. [Online; accessed 2026-01-11].

Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. 2012. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934. [Online; accessed 2026-01-11].

E. S. Kennedy and W. R. Stanton, editors. 1924. *The treatise of al-Jazari on automata: leaves from a manuscript of the Kitāb fī marifat al-ḥiyal al-handasiyya in the Museum of Fine Arts, Boston*. Museum

of Fine Arts, Boston, Massachusetts, USA. Facsimile with typescript translation.

Michael Kiato. 2024. Geometric object grasper. `https://github.com/mkiatos/geometric-object-grasper`. [Online; accessed 2026-01-11].

M. Kiatos, S. Malassiotis, and I. Sarantopoulos. 2021. A geometric approach for grasping unknown objects with multifingered hands. *IEEE Transactions on Robotics*, 37(5):1752–1763. [Online; accessed 2026-01-11].

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. [Online; accessed 2026-01-11].

Kinova Robotics Inc. 2006–. Kinova: Robotic manipulation solutions. `https://www.kinovarobotics.com/`. [Online; accessed 2026-01-11].

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. [Online; accessed 2026-01-11].

Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F. Huber. 2020. A survey on learning-based robotic grasping. *Current Robotics Reports*, 1(4):239–249. [Online; accessed 2026-01-11].

K. Komoda, P. Jiang, H. Han, J. Ooga, H. Eto, S. Tokura, H. Chatani, K. Sawa, Y. Oka, and K. Konda. 2024. Hybrid-AI grasp planning system that integrates rule-based and DNN-based methods for throughput improvement of picking robots. In *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 691–696. IEEE, Boston, MA, USA. [Online; accessed 2026-01-11].

Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86. [Online; accessed 2026-01-11].

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444. [Online; accessed 2026-01-11].

Yann LeCun and Fu Jie Huang. 2005. Loss functions for discriminative training of energy-based models. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, pages 206–213. PMLR, Barbados. [Online; accessed 2026-01-11].

Ian Lenz, Honglak Lee, and Ashutosh Saxena. 2015. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724. [Online; accessed 2026-01-11].

Susanna Leveroni and Kenneth Salisbury. 1996. Reorienting objects with a robot hand using grasp gaits. In Georges Giralt and Gerhard Hirzinger, editors, *Robotics Research: The Seventh International Symposium*, International Symposium on Robotics Research, pages 39–51. Springer-Verlag, London. [Online; accessed 2026-01-11].

Juncheng Li and David J. Cappelleri. 2024. Sim-grasp: Learning 6-DOF grasp policies for cluttered environments using a synthetic benchmark. *IEEE Robotics and Automation Letters*, 9(9):7645–7652. [Online; accessed 2026-01-11].

Zexiang Li and S. Shankar Sastry. 2002. Task-oriented optimal grasping by multifingered robot hands. *IEEE Transactions on Robotics*, 18(4):32–44. [Online; accessed 2026-01-11].

Wenwei Lin, Peidong Liang, Guantai Luo, Ziyang Zhao, and Chentao Zhang. 2022. Research of online hand–eye calibration method based on charuco board. *Sensors*, 22(10):3805. [Online; accessed 2026-01-11].

Qiang Liu and Dilin Wang. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2378–2386. NeurIPS Foundation, Barcelona, Spain. [Online; accessed 2026-01-11].

Zygmunt Łuniewicz and Marzanna Jagiełło. 2024. The silesian impact of Hero's treatise. Salomon de Caus and the Wrocław garden of Laurentius Scholz. *Quarterly Journal of the History of Science and Technology*, 2024(3):41–67.

Raymond R. Ma and Aaron M. Dollar. 2011. On dexterity and dexterous manipulation. In *Proceedings of the 15th International Conference on Advanced Robotics (ICAR)*, pages 1–7. IEEE, Tallinn, Estonia. [Online; accessed 2026-01-11].

Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. 2017. Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *Proceedings of Robotics: Science and Systems (RSS)*. [Online; accessed 2026-01-11].

Jeffrey Mahler, Florian T. Pokorny, Brian Hou, Melrose Roderick, Michael Laskey, Mathieu Aubry, Kai Kohlhoff, Torsten Kröger, James Kuffner, and Ken Goldberg. 2016. Dex-Net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1957–1964. IEEE, Stockholm, Sweden. [Online; accessed 2026-01-11].

F. A. Maken, F. Ramos, and L. Ott. 2019. Speeding up iterative closest point using stochastic gradient descent. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1276–1282. IEEE, Montreal, QC, Canada. [Online; accessed 2026-01-11].

F. A. Maken, F. Ramos, and L. Ott. 2022a. Stein ICP for uncertainty estimation in point cloud matching. *IEEE Robotics and Automation Letters*, 7(4):10285–10292. [Online; accessed 2026-01-11].

Fahira Afzal Maken, Fabio Ramos, and Lionel Ott. 2022b. Bayesian iterative closest point for mobile robot localization. *The International Journal of Robotics Research*, 41(9-10):851–874. [Online;

accessed 2026-01-11].

Fahira Afzal Maken, Fabio Ramos, and Lionel Ott. 2022c. Stein particle filter for nonlinear, non-gaussian state estimation. *IEEE Robotics and Automation Letters*, 7(2):5421–5428. [Online; accessed 2026-01-11].

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. 2021. Isaac gym: High performance GPU-based physics simulation for robot learning. arXiv preprint arXiv:2108.10470. [Online; accessed 2026-01-11].

Ravi Malladi, James A. Sethian, and Baba C. Vemuri. 1995. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175. [Online; accessed 2026-01-11].

Allison Marsh. 2022. In 1961, the first robot arm punched in. *IEEE Spectrum*. [Online; accessed 2026-01-11].

Matthew T. Mason. 2001. *Mechanics of Robotic Manipulation*. Intelligent Robotics and Autonomous Agents Series. MIT Press, Cambridge, MA. [Online; accessed 2026-01-11].

Matthew T. Mason. 2018. Toward robotic manipulation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:1–28.

Andrew T. Miller and Peter K. Allen. 1999. Examples of 3D grasp quality computations. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1240–1246. IEEE, Detroit, MI, USA. [Online; accessed 2026-01-11].

Andrew T. Miller and Peter K. Allen. 2004. GraspIt!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122. [Online; accessed 2026-01-11].

Brian Mirtich and John Canny. 1994. Easily computable optimum grasps in 2D and 3D. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation (ICRA)*, pages 739–747. IEEE, San Diego, CA, USA. [Online; accessed 2026-01-11].

H. Mnyussiwalla, Pascal Seguin, Philippe Vulliez, and Jean-Pierre Gazeau. 2022. Evaluation and selection of grasp quality criteria for dexterous manipulation. *Journal of Intelligent & Robotic Systems*, 104(2):20. [Online; accessed 2026-01-11].

Seyed S. Mohammadi, Nuno F. Duarte, Dimitrios Dimou, Yiming Wang, Matteo Taiana, Pietro Morerio, Atabak Dehban, Plinio Moreno, Alexandre Bernardino, and Alessio Del Bue. 2023. 3DSGrasp: 3D shape-completion for robotic grasp. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3815–3822. IEEE, London, UK. [Online; accessed 2026-01-11].

Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 2019. 6-DoF graspnet: Variational grasp generation for object manipulation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2901–2910. IEEE, Seoul, South Korea. [Online; accessed 2026-01-11].

Richard M. Murray, Zexiang Li, and S. Shankar Sastry. 1994. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL. [Online; accessed 2026-01-11].

Ken Nakahara and Roberto Calandra. 2025. Learning gentle grasping using vision, sound, and touch. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3367–3374. [Online; accessed 2026-01-11].

Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. 2023. Deep learning approaches to grasp synthesis: A review. *IEEE Transactions on Robotics*, 39(5):3994–4015. [Online; accessed 2026-01-11].

V. D. Nguyen. 1987. Constructing force-closure grasps in 3D. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 240–245. IEEE, Raleigh, NC, USA. [Online; accessed 2026-01-11].

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. arXiv preprint arXiv:2303.08774. [Online; accessed 2026-01-11].

Frank C. Park and Bryan J. Martin. 1994. Robot sensor calibration: Solving ax=xb on the euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5):717–721. [Online; accessed 2026-01-11].

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174. IEEE, Long Beach, CA, USA. [Online; accessed 2026-01-11].

Carlo Pedretti and Mark E. Rosheim, editors. 2006. *Leonardo da Vinci's Robots*. Springer, Berlin, Heidelberg. Reinterprets scattered manuscript fragments to reconstruct designs for functioning automata, including the mechanical knight.

Martin Pfanne. 2022. *In-Hand Object Localization and Control: Enabling Dexterous Manipulation with Robotic Hands*, volume 149 of *Springer Tracts in Advanced Robotics*. Springer International Publishing, Cham, Switzerland. [Online; accessed 2026-01-11].

Domenico Prattichizzo and Jeff C. Trinkle. 2008. *Grasping*, chapter 29, pages 671–700. Springer-Verlag, Berlin, Heidelberg. [Online; accessed 2026-01-11].

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85. IEEE, Honolulu, HI, USA. [Online; accessed 2026-01-11].

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2019. On the convergence of adam and beyond. arXiv preprint arXiv:1904.09237. [Online; accessed 2026-01-11].

Máximo A. Roa and Raúl Suárez. 2009. Computation of independent contact regions for grasping 3D objects. *IEEE Transactions on Robotics*, 25(4):839–850. [Online; accessed 2026-01-11].

Máximo A. Roa and Raúl Suárez. 2015. Grasp quality measures: Review and performance. *Autonomous Robots*, 38(1):65–88. [Online; accessed 2026-01-11].

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407. [Online; accessed 2026-01-11].

Dan Roth. 2016. Neural networks 1. `https://www.cis.upenn.edu/~danroth/Teaching/CS446-17/LectureNotesNew/neuralnet1/main.pdf`. CS446 Lecture Notes. [Online; accessed 2026-01-11].

Carlos Rubert, Beatriz León, Antonio Morales, and Joaquín Sancho-Bru. 2018. Characterisation of grasp quality metrics. *Journal of Intelligent & Robotic Systems*, 89(2):319–342. [Online; accessed 2026-01-11].

A. Sahbani, S. El-Khoury, and P. Bidaud. 2012. An overview of 3D object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3):326–344. [Online; accessed 2026-01-11].

Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. 2008. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173. [Online; accessed 2026-01-11].

Karun B. Shimoga. 1996. Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, 15(3):230–266. [Online; accessed 2026-01-11].

Charles Stein. 1972. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, volume 6, pages 583–603. University of California Press, Berkeley, CA, USA. [Online; accessed 2026-01-11].

Vignesh Subramaniam, Snehal Jain, Jai Agarwal, and Pablo Valdivia y Alvarado. 2020. Design and characterization of a hybrid soft gripper with active palm pose control. *The International Journal of Robotics Research*, 39(14):1668–1685. [Online; accessed 2026-01-11].

Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. 2021. Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, Xi'an, China. [Online; accessed 2026-01-11].

Chao Tang, Dehao Huang, Wenlong Dong, Ruinian Xu, and Hong Zhang. 2025. Foundationgrasp: Generalizable task-oriented grasping with foundation models. *IEEE Transactions on Automation Science and Engineering*, 22(1):xxx–xxx. [Online; accessed 2026-01-11].

Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. 2017. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473. [Online; accessed 2026-01-11].

The MathWorks Inc. 2023. What is inverse kinematics? `https://www.mathworks.com/discovery/inverse-kinematics.html`. [Online; accessed 2026-01-11].

Dongying Tian, Xiangbo Lin, and Yi Sun. 2024. Adaptive motion planning for multi-fingered functional grasp via force feedback. In *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, pages 835–842. IEEE, Nancy, France. [Online; accessed 2026-01-11].

Roger Y. Tsai and Reimar K. Lenz. 1989. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358. [Online; accessed 2026-01-11].

Teng Wan, Shaoyi Du, Yiting Xu, Guanglin Xu, Zuoyong Li, Badong Chen, and Yue Gao. 2019. RGB-D point cloud registration via infrared and color camera. *Multimedia Tools and Applications*, 78(24):33223–33246. [Online; accessed 2026-01-11].

Kaimeng Wang, Yongxiang Fan, and Ichiro Sakuma. 2024. Robot grasp planning: A learning from demonstration-based approach. *Sensors*, 24(2):436. [Online; accessed 2026-01-11].

Shaochen Wang, Zhangli Zhou, and Zhen Kan. 2022. When transformer meets robotic grasping: Exploits context for efficient grasp detection. *IEEE Robotics and Automation Letters*, 7(3):8170–8177. [Online; accessed 2026-01-11].

Albert Wu, Michelle Guo, and C. Karen Liu. 2022. Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, volume 205, pages 1744–1754. PMLR. [Online; accessed 2026-01-11].

Zhenjia Xu, Beichun Qi, Shubham Agrawal, and Shuran Song. 2021. Adagrasp: Learning an adaptive gripper-aware grasping policy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4626. IEEE, Xi'an, China. [Online; accessed 2026-01-11].

Zeya Yin, Tin Lai, Subhan Khan, Jayadeep Jacob, Yonghui Li, and Fabio Ramos. 2024. Stein movement primitives for adaptive multi-modal trajectory generation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11901–11908. IEEE, Abu Dhabi, UAE. [Online; accessed 2026-01-11].

Qunchao Yu, Weiwei Shang, Zengzhi Zhao, Shuang Cong, and Zhijun Li. 2020. Robotic grasping of unknown objects using novel multilevel convolutional neural networks: From parallel gripper to dexterous hand. *IEEE Transactions on Automation Science and Engineering*, 18(4):1730–1741. [Online; accessed 2026-01-11].

Lazher Zaidi, Juan Antonio Corrales, Belhassen Chedli Bouzgarrou, Youcef Mezouar, and Laurent Sabourin. 2017. Model-based strategy for grasping 3D deformable objects using a multi-fingered robotic hand. *Robotics and Autonomous Systems*, 95:196–206. [Online; accessed 2026-01-11].

Hanbo Zhang, Jian Tang, Shiguang Sun, and Xuguang Lan. 2022. Robotic grasping from classical to modern: A survey. arXiv preprint arXiv:2202.03631. [Online; accessed 2026-01-11].

J. Zhang, R. Zhang, L. Carin, and C. Chen. 2020. Stochastic particle-optimization sampling and the non-asymptotic convergence theory. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1466–1476. PMLR, Online (originally Palermo, Italy). [Online; accessed 2026-01-11].

W. Zhang, F. A. Maken, T. Lai, and F. Ramos. 2024. Grasping by parallel shape matching. In *2024 Australasian Conference on Robotics and Automation (ACRA)*, pages xxx–xxx. Australasian Conference on Robotics and Automation, Brisbane, Australia. [Online; accessed 2026-01-11].

Xin Zhang and Andrew Curtis. 2020. Seismic tomography using variational inference methods. *Journal of Geophysical Research: Solid Earth*, 125(4). [Online; accessed 2026-01-11].

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. 2018. Open3d: A modern library for 3D data processing. arXiv preprint arXiv:1801.09847. [Online; accessed 2026-01-11].

Xiaojin Jerry Zhu. 2005. Semi-supervised learning literature survey. `http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf`. Technical Report 1530, Department of Computer Sciences, University of Wisconsin-Madison. [Online; accessed 2026-01-11].