AnyDexGrasp: Learning General Dexterous Grasping for Any Hands with Human-level Learning Efficiency

Hao-Shu Fang[1], **Hengxu Yan**[1], **Zhenyu Tang**[1], **Hongjie Fang**[1], **Chenxi Wang**[1] **and Cewu Lu**[2]
[1]Department of Computer Science, Shanghai Jiao Tong University, [2]School of Artificial Intelligence, Shanghai Jiao Tong University

We introduce an efficient approach for learning dexterous grasping with minimal data, advancing robotic manipulation capabilities across different robotic hands. Unlike traditional methods that require millions of grasp labels for each robotic hand, our method achieves high performance with human-level learning efficiency: only hundreds of grasp attempts on 40 training objects. The approach separates the grasping process into two stages: first, a universal model maps scene geometry to intermediate contact-centric grasp representations, independent of specific robotic hands. Next, a unique grasp decision model is trained for each robotic hand through real-world trial and error, translating these representations into final grasp poses. Our results show a grasp success rate of 75-95% across three different robotic hands in real-world cluttered environments with over 150 novel objects, improving to 80-98% with increased training objects. This adaptable method demonstrates promising applications for humanoid robots, prosthetics, and other domains requiring robust, versatile robotic manipulation. Project website: https://graspnet.net/anydexgrasp/.

Summary

A general framework for learning visually guided dexterous grasping across various robotic hands, achieving human-level learning efficiency and accurate performance in cluttered environments.

1. Introduction

Grasping, as a fundamental problem of prehensile manipulation, holds significant importance in robotics. Over the past decades, diverse mechanical structures for robotic hands have been developed. Visually guided dexterous grasping is in high demand to enable robots to interact effectively with their environments. This ability also plays a crucial role in the context of intelligence. Throughout human evolution, early humans developed the capability for precise grip (Skinner et al., 2015), which enabled tool use and is believed to have facilitated the evolution of the human species (Almecija et al., 2010; Kivell, 2015). From the perspectives of both advancing robotics and promoting embodied intelligence, it is essential to design a learning framework that efficiently equips different robotic hands with visually guided dexterous grasping capabilities.

To make such a grasping system practically useful, it should use a single commodity camera, observe environments with cluttered objects, handle perception noise and generate a set of dexterous grasp poses that can be selected by subsequent tasks. Due to the challenges of the problem, early research focused on generating dexterous grasp poses given a single, complete object mesh, utilizing either analytical (Miller and Allen, 2004; Rosales et al., 2011; Liu et al., 2021; 2020) or learning-based approaches (Li et al., 2023). The idea is to decouple the grasping system into 6D pose estimation and grasp poses generation based on the object CAD model. However, the requirement for the object mesh limits its ability to handle new object shapes.

It is challenging to detect grasp poses for unseen objects based on partial-view perception. Some recent methodologies pursue mesh completion using partial point clouds (Lundell et al., 2021; Wei et al., 2022; 2024), followed by grasps generation on the complete mesh. However, the error introduced by perception noise and mesh completion often results in inaccurate grasp analysis. An increasing amount of research has attempted to learn the mapping from raw partial observation to grasp poses within a single network. Due to the highly nonlinear property of this mapping, extensive training data is required. Two data sources are commonly

Correspondence: Hao-Shu Fang (fhaoshu@gmail.com), Cewu Lu (lucewu@sjtu.edu.cn).

adopted: human grasping demonstrations (Gupta et al., 2016; Christen et al., 2019; Qin et al., 2022; Mandikal and Grauman, 2022; Wei et al., 2024; Shaw et al., 2024) or data from simulated environments (Brahmbhatt et al., 2019; Corona et al., 2020; Grady et al., 2021; Li et al., 2023; Wang et al., 2023). However, both methods have their limitations. The former approach struggles to accurately capture hand gestures and is confined to robotic hands resembling human anatomy. The latter requires substantial effort to build the simulation environment, annotate grasp poses, and transfer algorithms from simulation to the real world. These challenges limit current grasping systems to simple scenarios, typically involving a single object at a time from a limited set. No prior work demonstrates robust grasping in cluttered environments from partial-view perception in the real world.

Most critically, even if these challenges are overcome, the policy obtained with substantial efforts is only suitable for a specific robotic hand each time. The end-to-end learning paradigm implicitly encodes the information about hand kinematic structure, relevant state information and grasp quality in the weights, making it difficult for models to share computation between different hands. Consequently, we need to repeat the tedious data generation and policy training pipeline for each hand.

We identified two main bottlenecks for efficiently learning visually guided dexterous grasping for different robotic hands: the requirements for extensive training data for each hand, and the inability to share computation across different hands. These bottlenecks arise from attempting to learn the mapping from raw observation to grasp poses with an end-to-end network. In this paper, we revisit this paradigm. We hypothesize that if there exists a low-dimensional intermediate state space that encapsulates grasp information, then the mapping from this state space to grasp poses can be learned more efficiently than the original mapping, requiring less training data. Moreover, if such a state space is transferable across different robotic hands, it could be shared without the need to retrain a state estimator each time. Note that such a state space should not require object knowledge during inference, in order to generalize to unseen objects.

Recognizing this potential, we aimed to identify such a state space. For the grasping problem, the robot needs to decide its grasp forces on each finger, based on the grasp matrix, surface normals of contact points and friction coefficient (Dai et al., 2018). From visual perception, the information we can extract is the positions and normals of potential contact points, where positions are linked to the grasp matrix and normals determine the orientations of friction cones. Based on this observation, we introduce a novel intermediate representation for multi-finger grasping, referred to as the Contact-centric Grasp Representation (CGR), which encapsulates contact information on the object's surface and possesses SE(3)-equivalent property.

Based on this representation, we present AnyDexGrasp, a novel methodology that can effectively learn dexterous grasping for different hands on a modest set of training objects. In this method, the multi-finger grasp detection problem that maps raw perception to grasp poses is divided into two steps. In the first step, we train a general representation model that maps single-view partial observations to contact-centric grasp representations. A large-scale dataset is annotated to train this model. After training, it can be applied to different hands without fine-tuning. In the second step, we map the contact-centric grasp representations to a set of grasp proposals through a hand-specific mapping, and then learn a hand-specific classifier to evaluate each grasp proposal. This classifier takes a contact-centric grasp representation and a grasp proposal as input and maps them to the probability of grasp success. The training data is collected by real-world trial and error. We empirically observed that this mapping is significantly easier to learn, requiring merely hundreds of trial-and-error attempts. It dramatically reduces the cost of real-world learning and allows our approach to work for different types of robotic hands efficiently.

We evaluate the effectiveness of our method using three different robotic hands, each featuring three to five fingers. Our system is first trained on 144 objects, with approximately 2,000 to 8,000 grasp attempts, depending on the robotic hand. On a diverse set of 150 previously unseen objects, including deformable and adversarial items, our approach achieves an average grasp success rate ranging from 80% to 98% across different hands. Notably, this performance is achieved in cluttered scenarios, demonstrating the effectiveness of our approach.

In addition we explore further reductions in training samples required for our grasp learning paradigm. We limit the training objects to 40 and reduce the grasp attempts to approximately 400 to 1,000 depending

on the robotic hand. Even with this limited amount of training data, our system consistently achieves grasp success rates ranging from 75% to 95% during real-world testing. Notably, our experiments also highlight the potential for further reductions of training samples, with the ability to decrease the training object number to 30 and the total grasp attempts to 200 for a three-finger hand, without decreasing the grasp performance by a large margin. Such learning efficiency allows robots to master visually guided grasping in a matter of hours in the real world, surpassing the learning efficiency of human infants.

We conduct a series of analyses to clarify why our two step learning method is so efficient. In the first step, we perform a geometry coverage analysis, showcasing that by scaling up data in the correct dimension, the local geometries on just 40 objects can effectively cover a wide range of unseen objects. This explains the generalization capabilities with a small number of training objects. In the second step, we provide various perspectives illustrating how our proposed contact-centric grasp representation serves as a robust state space for grasp decision, which allows the model to learn from just hundreds of real-world trial-and-error attempts.

This paper represents a significant step toward the efficient realization of dexterous robotic grasping, with the potential to revolutionize various applications, from advanced humanoid robots to prosthetic hands.

2. Results

2.1. Formulation

A multi-finger grasp pose g is formally defined as:

$$g = [\mathbf{R}\ \mathbf{t}\ \mathbf{q}], \tag{1}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ represents the robotic hand's rotation, $\mathbf{t} \in \mathbb{R}^{3\times1}$ denotes the hand's translation, and $\mathbf{q} \in \mathbb{R}^{n\times1}$ characterizes the joint configuration of a n-DoF multi-finger hand. The goal of the grasp pose detection problem is to predict a set of grasp poses from a scene perception. Conventionally, data-driven methods have employed a single network $f(\cdot)$ to map the partial point cloud of the scene $\mathcal{P} \in \mathbb{R}^{k\times3}$ to a set of candidate poses, $\mathbf{G} = \{g_i\}_{i=1}^{|\mathbf{G}|}$. In contrast, our approach decouples the mapping into two distinct steps: a state embedding step and a grasp decision step.

In the state embedding step, we extract a collection of contact-centric grasp representations from the partial point cloud $\mathcal{P}$. This is achieved by using a hand-agnostic representation model $\Phi(\cdot)$, which generates the scene representation $\mathcal{R}$:

$$\mathcal{R} = \Phi(\mathcal{P}), \tag{2}$$

where $\mathcal{R} = \left\{r_j\right\}_{j=1}^{|\mathcal{R}|}$ is a set of contact-centric grasp representations.

The grasp decision step consists of two distinct procedures: a mapping process that converts contact-centric grasp representations into a set of candidate grasp poses (referred to as grasp candidates) and a quality estimation process for each candidate. For each grasp representation r_j, we generate a set of grasp candidates based on the specific robotic hand. A hand-dependent mapping function $\mathcal{K}(\cdot)$ takes a grasp representation r_j and a hand specification h as input, and output $\mathbf{G}_j$:

$$\mathbf{G}_j = \mathcal{K}(r_j, h), \tag{3}$$

where $\mathbf{G}_j = \left\{g_j^{(i)}\right\}_{i=1}^{|\mathbf{G}_j|}$ denotes the set of grasp candidates for each r_j.

To estimate the quality of a grasp, we use a hand-dependent grasp decision model $\Psi(\cdot)$, which predicts the probability of success β given a grasp representation r_j, a grasp pose $g_j^{(i)}$, and a hand specification h:

$$\beta = \Psi(r_j, g_j^{(i)}, h). \tag{4}$$

Objective: Our goal is to find a set of grasp poses $\mathbf{G}^*$ that maximizes the grasp success rate given a desired number of grasp poses K:

$$\mathbf{G}^* = \underset{\mathbf{G}\subset\bigcup_j \mathbf{G}_j,\ |\mathbf{G}|=K}{\arg\max}\ \mathbb{E}_{r_j\in\mathcal{R},\ g_j^{(i)}\in\mathcal{K}(r_j,h)\cap\mathbf{G}}[\Psi(r_j, g_j^{(i)}, h)]. \tag{5}$$

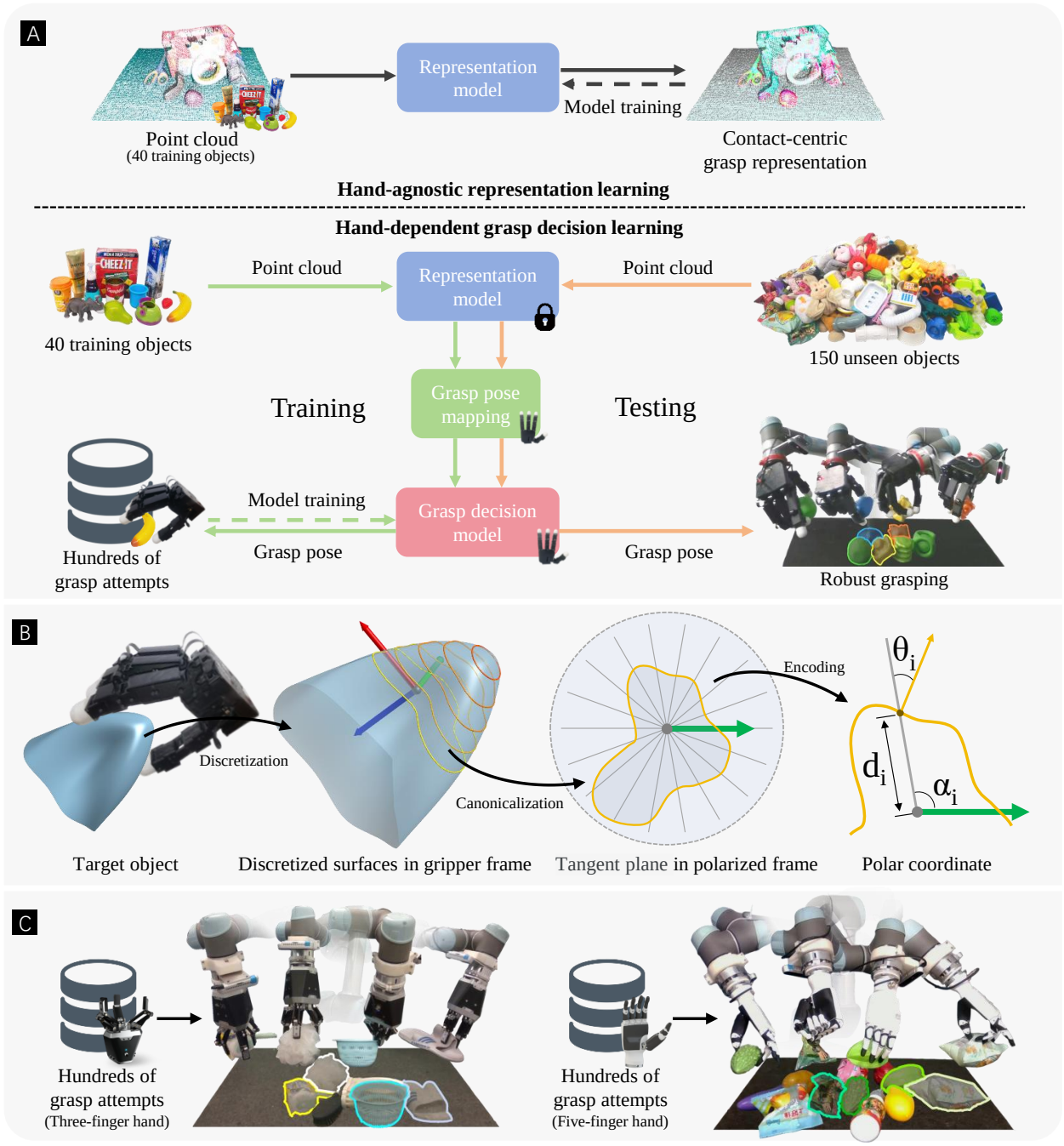

Figure 1: The overview of our method. **(A)**: Our method consists of two steps. The first step is to train a representation model on partial-view point cloud. The training set only consists of 40 objects. The second step would fix the representation model, and train a grasp decision model that takes the grasp-centric contact representation as input and outputs the grasp success score, based on hundreds of real-world trial-and-error attempts. The grasp algorithm is tested thoroughly on hundreds of unseen objects. **(B)**: Illustration of contact-centric grasp representation. A local geometry is discrete into several tangent planes along the approach direction of a robotic hand. Each tangent surface is transformed into the polarized coordinate frame of the robotic hand. The shape of the surface is encoded into discretized points and normal representation in the polar coordinate. **(C)**: Our experiments are also carried out on a three-finger hand and a five-finger hand and demonstrate excellent performance.

Figure 1A shows the pipeline of our methodology.

2.2. Contact-centric Grasp Representation

We initiate our approach with the development of a contact-centric grasp representation. Initially, consider a 2D object, we can represent it as a set comprising surface points and their corresponding normals:

$$r_{2d} = \{(p_i, n_i) \mid i = 1, 2, \cdots, N\}. \tag{6}$$

In this representation, p_i denotes the position of a surface point, and n_i represents the normal vector associated with that surface point. For clarity, the object's surface is discretized into N bins.

For the task of grasp pose detection, it is common to represent the object shape in a local coordinate frame (ten Pas et al., 2017; Mousavian et al., 2019), as the classification of grasp quality depends primarily on the geometry within a localized area. This step, referred to as canonicalization, equips the representation with SE(3)-equivalent property and makes subsequent learning easier. For the 2D example, when we employ a polar coordinate system and sample the pole coordinate $\mathbf{t}_{2d}$ and the polar axis $\mathbf{R}_{2d}$, the discrete object shape representation is refactored accordingly. In this system, a surface point p_i is represented by an angle α_i from the polar axis and a distance d_i from the pole. Additionally, the surface normal is encoded as the angle between the normal n_i and α_i:

$$\begin{aligned} r_{2d} = \Bigg\{ & (\alpha_i, d_i, \theta_i) \mid i = 1, 2, \dots, N,\ p_i' = \mathbf{R}_{2d}(p_i - \mathbf{t}_{2d}),\ n_i' = \mathbf{R}_{2d} n_i, \\ & \alpha_i = \frac{p_i'}{\|p_i'\|}, d_i = \|p_i'\|, \\ & \theta_i = \arccos\left(\frac{\alpha_i \cdot n_i'}{\|\alpha_i\|\|n_i'\|}\right); \mathbf{R}_{2d}, \mathbf{t}_{2d} \Bigg\}. \end{aligned} \tag{7}$$

A benefit of adopting a polar coordinate system is that the in-plane rotation angles $\{\alpha_i\}$ can be uniformly sampled across the polar angle range, resulting in constant values for $\{\alpha_i\}$ across different representations. Therefore, we move $\{\alpha_i\}$ to the right side of the set notation to make the representation more compact. Since the values of d_i and θ_i depend on α_i, we rewrite them as d_{α_i} and θ_{α_i}:

$$r_{2d} = \left\{ (d_{\alpha_i}, \theta_{\alpha_i}) \mid \alpha_i = 0, \frac{2\pi}{N}, \dots, \frac{2\pi(N-1)}{N}; \mathbf{R}_{2d}, \mathbf{t}_{2d} \right\}. \tag{8}$$

Extending this representation to a real-world 3D object and a 3D coordinate system with rotation $\mathbf{R}_{3d}$ and translation $\mathbf{t}_{3d}$ involves decoupling the object's geometry along a chosen axis and composing multiple 2D representations. By selecting a specific axis in the 3D coordinate system (*e.g.*, the z-axis), we discretize the object along this axis into M sections. Each section corresponds to a cross-sectional slice of the object at a particular coordinate along the axis.

Within each cross-sectional slice, the same polar coordinate system is employed as in the 2D case. We apply the same angular sampling and the local geometry is represented in terms of distance and normal angle at each sampled angle α_i. The 3D representation is then formulated as:

$$r_{3d} = \left\{ (d_{\alpha_i}, \theta_{\alpha_i})_j \,\middle|\, \alpha_i = 0, \frac{2\pi}{N}, \dots, \frac{2\pi(N-1)}{N},\ j = 1, 2, \dots, M; \mathbf{R}_{3d}, \mathbf{t}_{3d} \right\} \tag{9}$$

In the following sections, we use r as a shorthand for r_{3d}. In Figure 1B, we illustrate the process of representing a 3D geometry in the contact-centric representation format within a robotic hand's local coordinate frame.

2.3. Robotic Hands and Grasp Types

In our experiments, we utilize three distinct robotic hands:

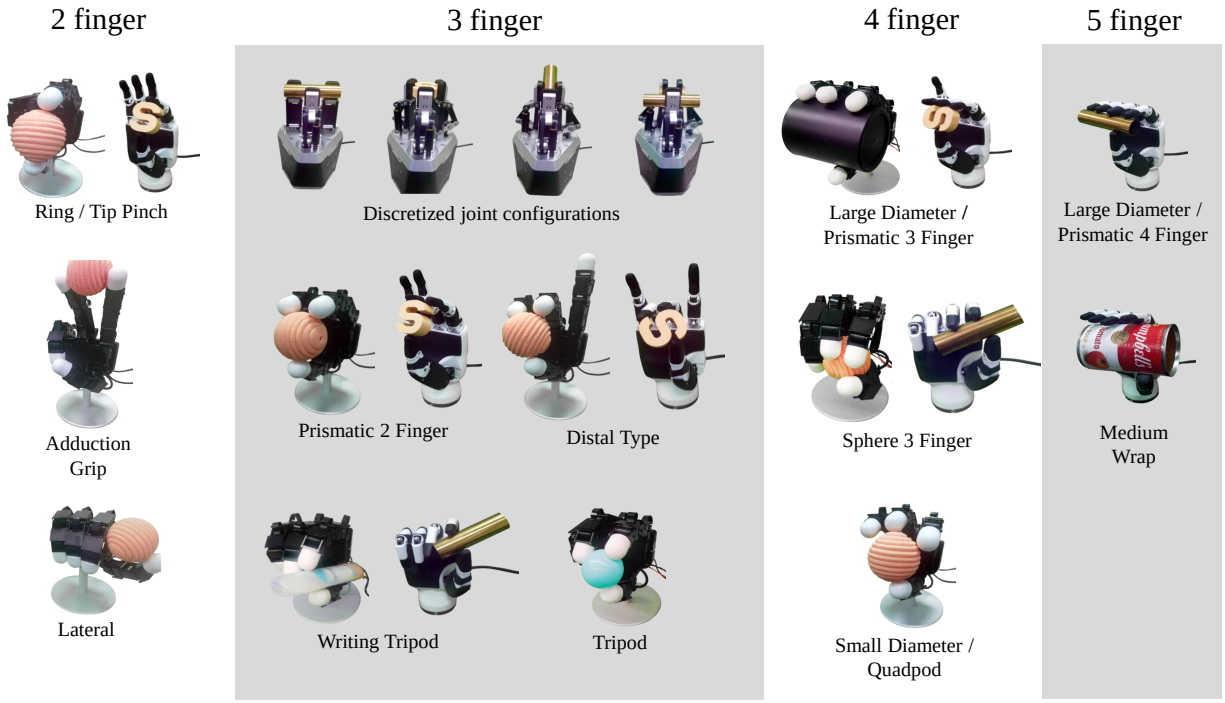

Figure 2: Illustration of the predefined grasp types for three robotic hands. The types are categorized by the number of fingers involved in the grasping procedure. Some types can be categorized into multiple taxonomies defined in previous literature (Feix et al., 2015) when the grasping depths differ.

- DH-3: A three-finger robotic hand comprises 4 degrees of freedom and 2 motors, operating in an underactuated manner.
- Allegro: A four-finger robotic hand comprises 16 degrees of freedom and 16 motors, designed for full actuation.
- Inspire: A five-finger robotic hand equipped with 12 degrees of freedom and 6 motors, operating in an underactuated manner.

These robotic hands represent a variety of applications, including industrial tasks, dexterous manipulation, and underactuated prosthetic hand functionalities.

One challenge with dexterous grasping is the complexity introduced by the high degrees of freedom in these robotic hands, which creates a vast joint configuration space. However, when humans grasp objects, we typically rely on only a small subset of these configurations, which can be categorized into specific taxonomies (Cutkosky et al., 1989; Feix et al., 2015). To address this complexity and make grasp pose detection more manageable, we discretize the continuous joint configurations of the multi-fingered hands into a finite set of predefined grasp types. This is represented as $\mathbf{q} \in \{\mathbf{q}_1, ..., \mathbf{q}_c\}$, where c denotes the total number of grasp types specific to each hand.

For the three-finger hand, we discretize the entire joint space into several bins, while for the four- and five-finger hands, we select grasp types from the human grasp taxonomy that can be executed by these dexterous robotic hands. The predefined grasp types are illustrated in Figure 2. While this approach simplifies the grasp pose detection process, it still provides sufficient flexibility for subsequent manipulation tasks.

It is important to note that these grasp types serve as anchor poses prior to contact. Once the hand reaches its target position, it undergoes a closure process, where the fingers progressively move toward each other until the forces exerted on the finger joints reach predefined limits.

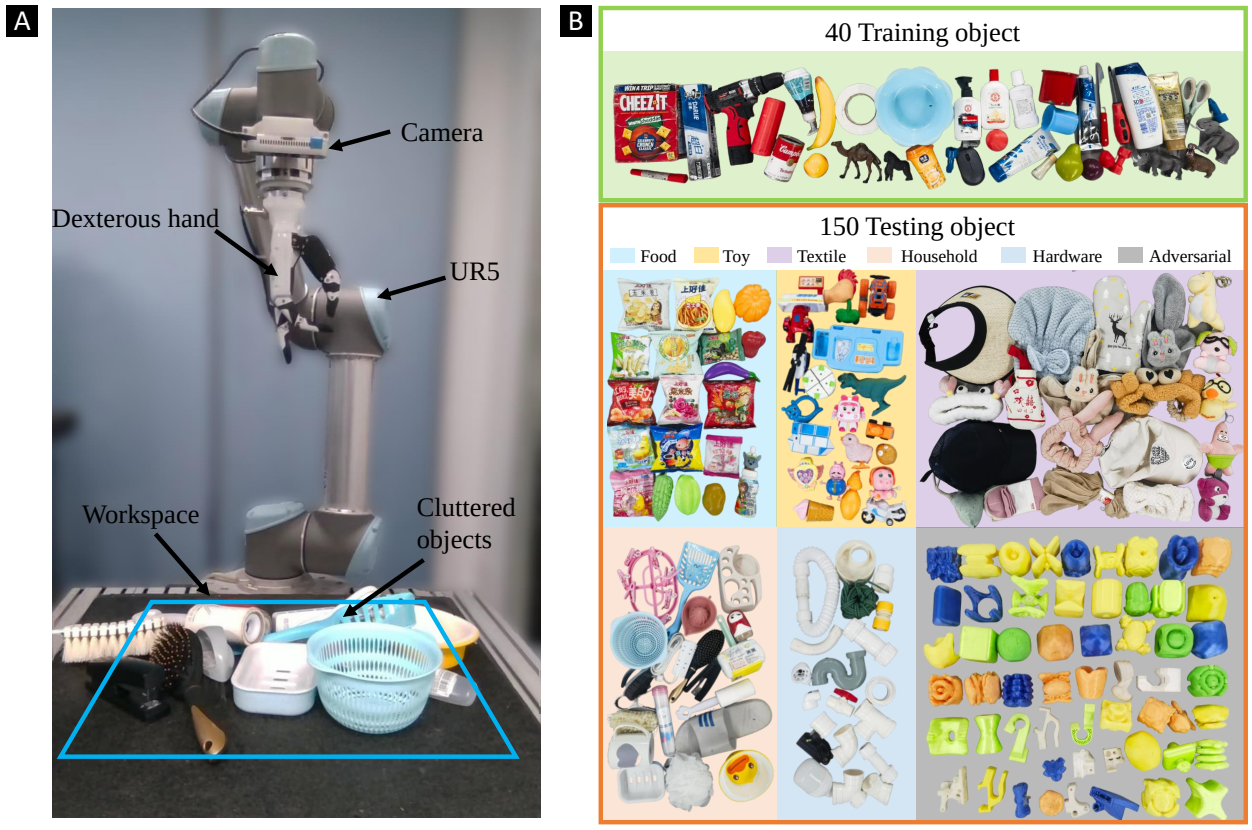

Figure 3: Experimental setup. (A): Platform setting of our dexterous grasping experiments. **(B)**: Illustration of our 40 training objects and 150 testing objects. The testing objects are much more diverse than the training objects, including deformable and adversarial objects not presented in the training set.

2.4. Overview of Experiments

To assess the performance of our multi-finger grasping model, we established a real-world experimental platform. Our hardware setup includes a UR5 robotic arm and an Intel RealSense D415 camera, positioned at the robot's end-effector. The initial camera pose is vertical to the table and is approximately 60 cm above it. Figure 3A illustrates the setup of our robotic platform.

We first learn a general hand-agnostic representation model based on an offline annotated, large-scale dataset. Once the representation model is learned, we use the predicted contact-centric grasp representation as a new state space for the problem of multi-finger grasping. For each robotic hand, we can learn grasping in the real world directly through trial and error. We start with thousands of trial-and-error grasp attempts, and gradually reduce the number to hundreds of attempts in later experiments to demonstrate the efficiency of our learning paradigm. We also vary the number of training objects from 144 to 40 and even 30, to verify the generalization ability of our grasp system.

To assess the multi-finger grasp performance thoroughly, we construct a comprehensive real-world test set, featuring objects commonly encountered in everyday life. These objects encompass diverse shapes, materials, and textures and are categorized into hardware, food, textile, household, toy, and adversarial items. The test set comprises nearly 150 objects ranging in size from $2.5 \times 2.5 \times 2.5$ cm^3 to $8 \times 8 \times 5$ cm^3.

During real-world testing, objects from each category are randomly placed on a table in a cluttered way, and the robots attempt to grasp all the objects and clear the table. This process is repeated twice for accuracy. We also establish a baseline that aligns the principal closing axis of the grasp types with antipodal grasp poses, followed by collision detection, to compare with our proposed method. The success rate is determined by dividing the number of successful grasp attempts by the total number of grasp attempts.

Ultimately, our grasp system is successfully evaluated on three different robotic hands, where the

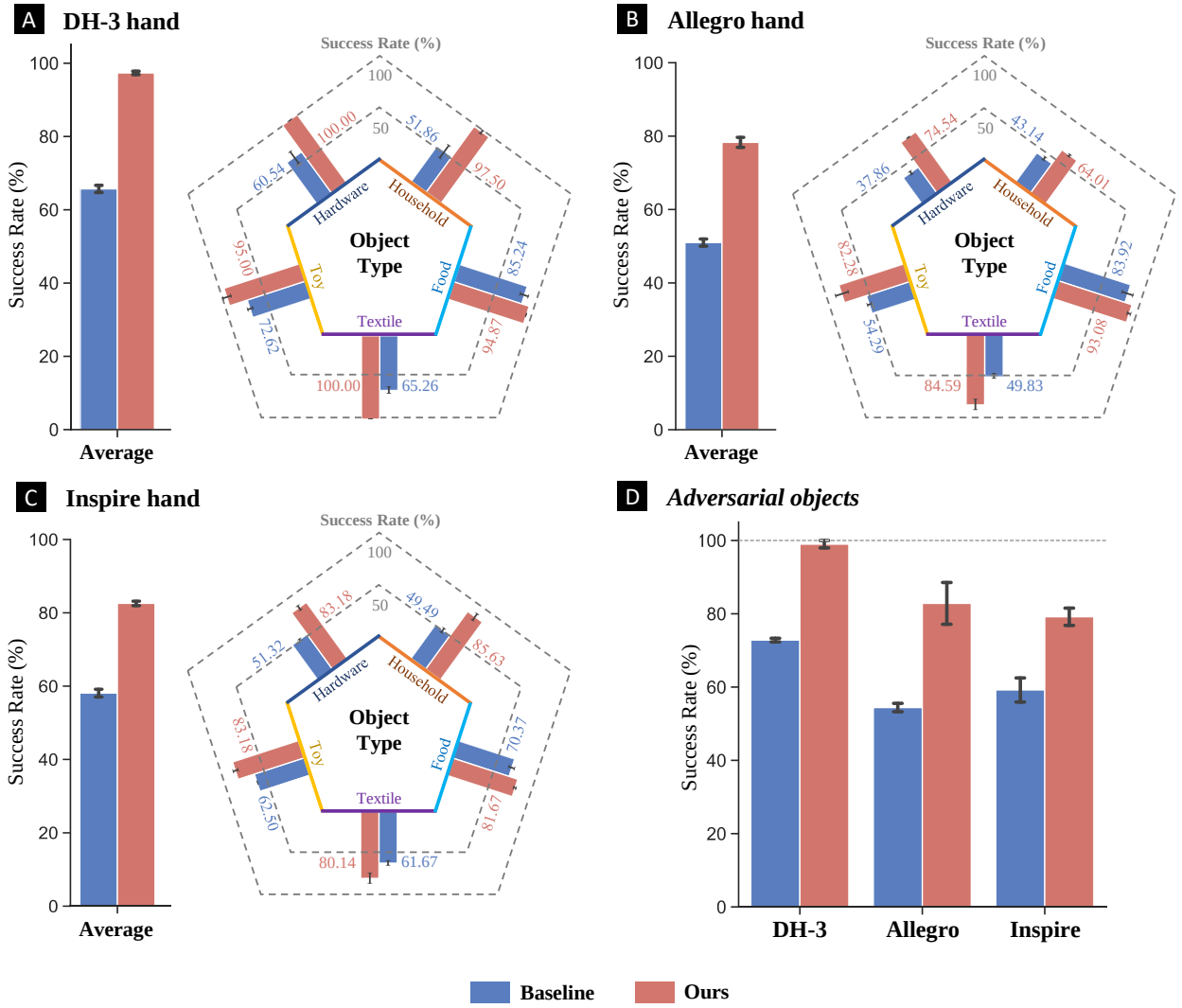

Figure 4: Success rates on the testing set after training on abundant real-world data. (A): The averaged and detailed success rates of the DH-3 hand on five object categories commonly encountered in our daily activities. **(B)**: The averaged and detailed success rates of the Allegro hand. **(C)**: The averaged and detailed success rates of the Inspire hand. **(D)**: The success rates on the adversarial objects of three robotic hands.

whole system is trained on a limited dataset comprising merely 40 objects and hundreds of grasp attempts, and tested on a broader spectrum of 150 previously unseen objects. Notably, it represents a pioneering achievement in the literature where a grasping algorithm is evaluated on a significantly larger set of objects than those included in its training dataset. The final training and testing objects are illustrated in Figure 3B for reference.

2.5. Learning Dexterous Grasping in the Real World

Based on the contact-centric grasp representations output by a trained representation model, we first employ a training set of 144 objects to train the grasp decision model. Approximately 1,000 grasp samples are collected for each grasp type, forming the basis for our learning process. The amount of training objects and grasp samples would be gradually reduced in later sections to verify the effectiveness of our method.

Dexterous Grasping on Daily Objects

We systematically evaluate the success rates of our approach on testing objects from the first five categories commonly encountered in our daily activities. The average success rates achieved by the three distinct

robotic hands are 97%, 78%, and 83%, respectively. Movies S1, S2 and S3 record the grasping process. In contrast, the success rates of the baseline method using heuristic sampling and collision detection reach only 66%, 51%, and 58%. A detailed breakdown of success rates for each object category is presented in Figure 4A. Compared to the baseline method, the substantial improvements across this extensive test set demonstrate the effectiveness of our proposed representation and approach.

Several noteworthy points are hereby highlighted. Firstly, the 3-finger gripper attains an average success rate of 97% across over 100 real-world objects, surpassing even the performance of previous state-of-the-art parallel-gripper algorithm (Fang et al., 2023b). Secondly, for deformable objects within the textile and food categories, the grasp success rates across different grippers show no significant degradation. In some cases, they even slightly outperform the overall success rate, despite the absence of explicit training on deformable objects. This observation emphasizes the remarkable generalization capacity of data-driven methods. We observe that deformable objects tend to comply with the gripper during the grasping process, making them easier to be successfully grasped.

Regarding grasping speed, our system takes an average of 0.5 second to generate 200 grasp poses in a cluttered scene. Additionally, an extra collision detection step utilizing scene partial point cloud and hand mesh is performed. It takes 20 seconds on our CPU using Open3D library (Zhou et al., 2018). Although this step could be accelerated through advanced collision detection technology or hardware acceleration, this aspect falls outside the scope of this paper.

Dexterous Grasping on Adversarial Objects

In addition to daily objects, we extend our method's evaluation to more challenging adversarial objects. These objects encompass 13 human-selected items from DexNet (Mahler et al., 2017) and 49 program-generated objects from EGAD! evaluation set (Morrison et al., 2020), characterized by distinct shapes and varying grasp difficulties. Prior literature shows a performance degradation of parallel grasping on adversarial versus daily objects (Fang et al., 2023b). To the best of our knowledge, this is the first comprehensive evaluation of a multi-finger grasping algorithm on adversarial objects in real-world scenarios.

Success rates for the three distinct robotic hands are reported in Figure 4B, where our system achieves 99%, 82%, and 79% success rates, respectively. Movies S4, S5 and S6 record the grasping process. In contrast, the baseline method achieves success rates of 72%, 54%, and 59%. Remarkably, the performance on adversarial objects is on par with daily objects for all the robotic hands, highlighting the promising generalization ability of our dexterous grasping system. It surprises us since previous results from parallel grippers show a dramatic performance degradation. We presume that the additional fingers can improve the grasping ability, and the adversarial objects designed for parallel grippers do not pose significant challenges in multi-finger cases.

In the subsequent sections, unless otherwise stated, we proceed to conduct experiments on all 150 objects including daily ones and adversarial ones.

2.6. Reducing Real-World Training Burden

In previous experiments, we collected a considerable volume of real-world training data, approximately 1,000 trials per grasp type, on the full set of 144 training objects. In this section, we aim to alleviate the demands of real-world training by assessing the model's performance under reduced object and trial conditions. For the sake of simplicity, our evaluation concentrates on the 3-finger hand in this section.

We initiate this exploration by reducing the number of objects utilized in real-world trial-and-error attempts. Two smaller sets consisting of 40 and 30 objects were adopted for our experiments. The details of the training object sets are given in Materials and Method. To ensure a fair comparison with the original object set, we collect an equivalent number of around 1,000 grasp samples for each grasp type, ensuring the convergence of the grasp decision model.

Figure 5A presents the experimental results. As the training object count reduces from 144 to 40, the grasp success rate achieves 96.7% on all testing objects, representing only a marginal decrease of 1.1%. Such subtle performance degradation, given a nearly 3/4 reduction in training objects, showcases the robustness of our approach. A further reduction to 30 training objects results in an overall grasp success of 95.1%. This

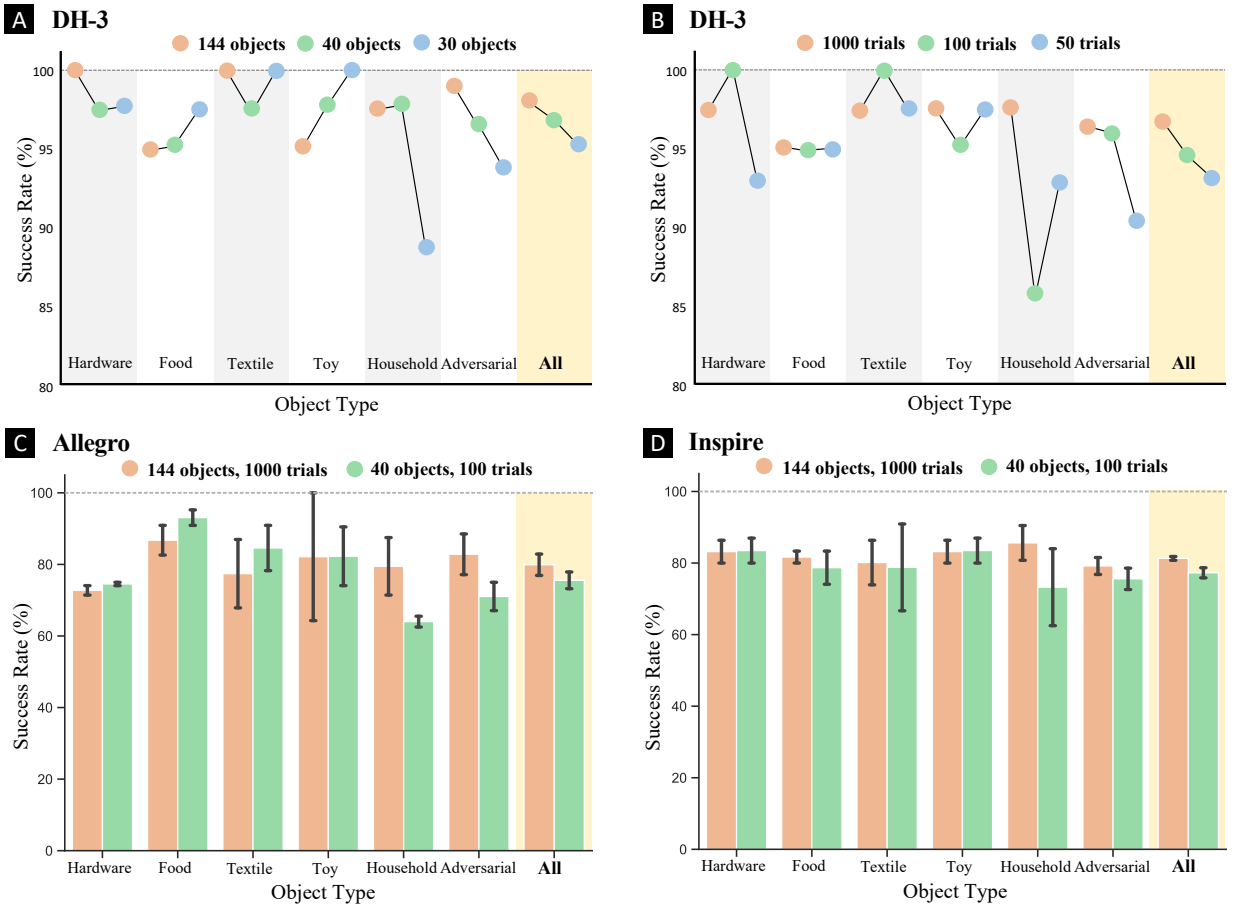

Figure 5: Success rates on the testing set after training on reduced real-world data. (A): We reduce the training object number from 144 to 40 and 30 respectively and test the success rates on different categories of objects. **(B)**: With 40 training objects, we reduce the data from around 1000 trials per grasp type to 100 trials and 50 trials respectively. **(C) and (D)**: When reducing the training data on fewer training objects and fewer grasp attempts, success rates on both Allegro hand and Inspire hand only decrease slightly, showing good generalization ability and high learning efficiency of our method.

translates to a further decrease of 1.6%, yet the performance remains notably promising. Our experiments demonstrate that, with proper learning methods, the thousands of training objects adopted in previous systems(Mahler et al., 2019; Wang et al., 2023) are not necessary.

Since the performance degradation when reducing the object set from 144 to 40 and from 40 to 30 is comparable, while reducing the training object set from 40 to 30 does not significantly lower the training burden, we opt to proceed with the 40-object training set for subsequent experiments.

We then explore the impact of reducing the number of trials and errors for each grasp type. On the 40 training objects, we reduce trials and errors from approximately 1,000 attempts per grasp type to 100 attempts and 50 attempts, respectively. Figure 5B presents the real robot testing results. When training with 100 trials per grasp type, the success rate reaches 94.5% on average on all objects. We show the whole grasping process in Movie S7. This success rate is strikingly high given the limited number of real-world training samples. Previous literature (Xu et al., 2023; Liu et al., 2023) often required millions of grasp attempts in simulation to achieve grasping proficiency. Further reducing the trials to 50 attempts per grasp type yields a success rate of 93.1% on all objects. These results demonstrate the high learning efficiency of our method, which requires only a small number of grasp attempts for convergence. In our following experiments, given the already high efficiency of 100 trials per grasp type, we adopt this setting for learning.

2.7. Dexterous Grasp Learning with 40 Objects and 100 Attempts

In the previous section, we demonstrated the robust grasping policy acquired by the 3-finger gripper through a significantly limited amount of training data and real-world attempts. In this section, we extend the validation of such a learning paradigm to the other two robotic hands utilized in this study.

We directly assess the performance of training using 40 objects with 100 trials for each grasp type on the four-finger and five-finger hands. Depending on the number of grasp types, the total real-world training samples amount to 1,000 and 800 for these two robotic hands, respectively. This significantly reduced volume of real-world training samples, nearly 1/10 of the original experiments, presents a territory in grasp learning that is unexplored by previous work.

Figure 5C and Figure 5D display the detailed success rates of real-world experiments. The average success rates stand at 75%, and 77% for all objects. The grasping process is recorded in Movies S8, S9, S10 and S11. It's striking that the success rates show minimal decreases compared to the original performance. This observation demonstrates the substantial learning efficiency enabled by our methodology. Such proficiency allows diverse robotic hands to acquire dexterous grasping ability in real-world settings.

Notably, this efficiency surpasses that observed in human infants, who typically require months of practice to develop visually guided grasping skills. The grasp success rates for human infants reach 61.9% at 8 months old (Domellöf et al., 2015), which involves thousands of practice attempts starting at 4 months old (Newell et al., 1989). It is noteworthy that our grasping results are achieved based solely on visual perception, with no tactile feedback.

2.8. Influence of Grasp Types

Accuracy of Different Grasp Poses

In the above experiments, we have shown that our method can enable efficient grasp learning with high success rates. Here we further analyze the success rates of each robotic hand with a detailed breakdown according to their respective grasp types. For clarity purposes, we number each grasp type, as illustrated in Figure 9. The results trained on 40 objects and 100 grasp attempts per grasp type are adopted for analysis, as depicted in Figure 6A. For each hand, we can see that different grasp types have different difficulties in dealing with grasping. Usually, the success rates after learning are dramatically higher than the baseline method. However, there also exist some exceptions. For example, the grasp type 1 of the five-finger Inspire hand after training yields a close success rate to the baseline. After inspection, we found that this grasp type was selected fewer times after the training. We anticipated that other grasp types might be more confident to grasp if the objects can be grasped by multiple types, which leaves some hard cases for this grasp type.

Distribution of Grasp Types

A natural question that arises is whether the system learned by our method can demonstrate a variety of grasp types. From the example above, it is possible that the system may achieve high success rates by favoring one or two grasp types while ignoring diversity. To address this, we analyze how frequently each grasp type is selected during testing to verify whether our system indeed learns diverse grasp poses. To quantify the frequency of each grasp type, we normalize by dividing the number of grasp attempts for each type by the total number of grasp attempts across all types.

To establish a baseline, we examine the frequency of grasp types obtained by the baseline method, reflecting the inherent frequency determined solely by collision detection. Grasp types prone to collision with the scene naturally constitute a smaller fraction among all types. This baseline grasp type frequency serves as a reference for natural distribution. The top row of Figure 6B illustrates the grasp type frequency for different robotic hands. Notably, the three-finger hand exhibits a balanced distribution, whereas the four- and five-finger hands display more unbalanced distributions. This discrepancy arises from the fact that the fingertips of the three-finger hand consistently point in the same direction along the approach vector, resulting in similar collision situations across different types. Conversely, the four- and five-finger hands exhibit types with greater variance, including some that are prone to collision with the scene.

Then, we present the frequency of grasp pose after employing our learned system. The statistics are

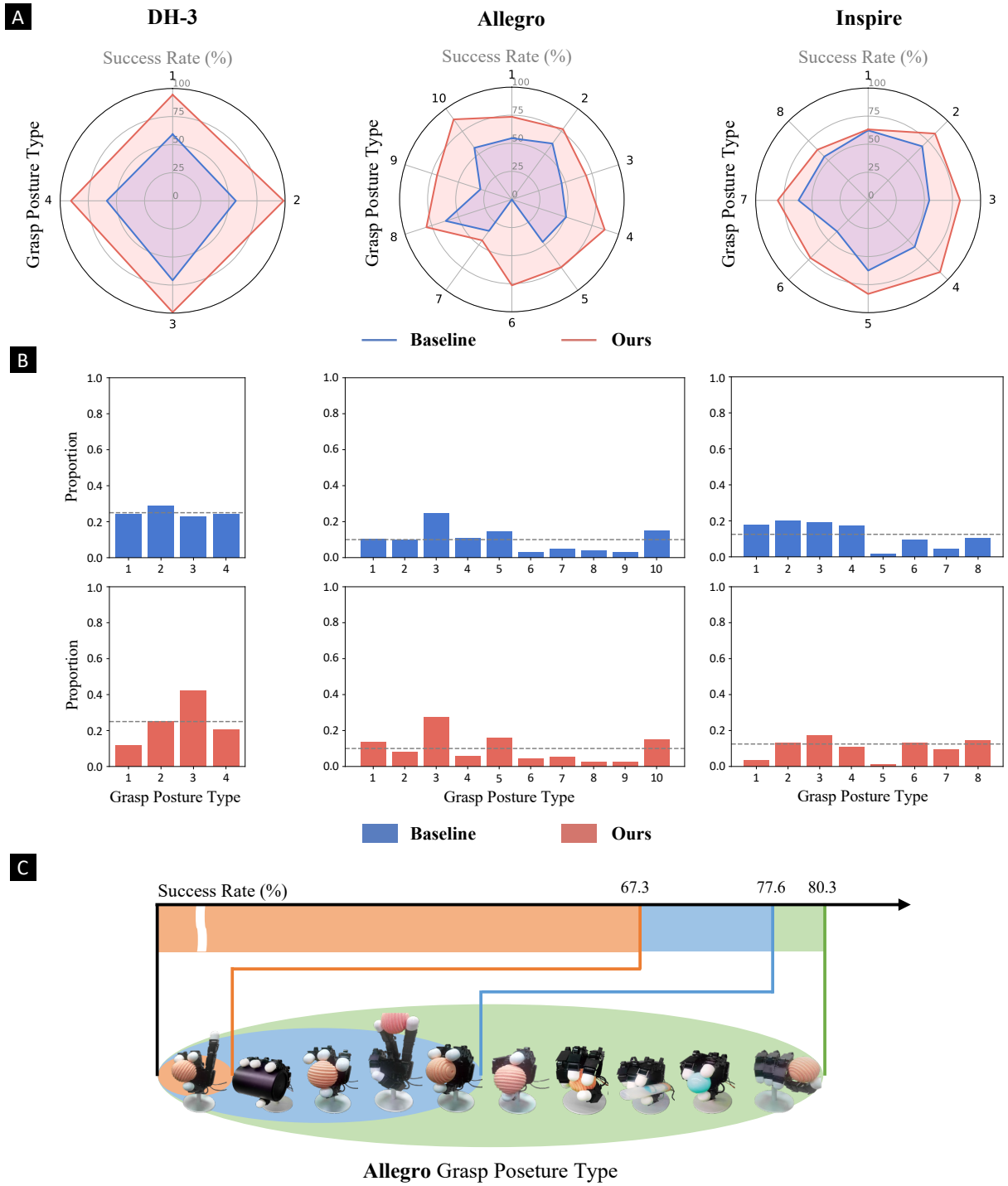

Figure 6: Analysis of the influence by grasp type. (A) A breakdown analysis of grasp success rates on different grasp types for each robotic hand. **(B)** The selected frequency of different grasp types for each robotic hand during testing. **(C)** Grasping success rates when using different portions of grasp types for the allegro hand.

given in the second row of Figure 6B. For the three-finger hand, type 3 presents an increasing ratio among all grasp types. The reason is that this grasp type presents a higher success rate, and usually has a higher grasp quality score than other grasp types. However, the other three grasp types are also frequently selected. For the four- and five-finger hand, the grasp frequency is similar to the baseline method. These results affirm that our learned system adeptly captures diverse grasp poses, achieving high success rates without compromising grasp diversity.

Reducing Grasp Types

Another question for multi-finger grasping is whether employing multiple grasp types is necessary, given the argument that a single power grasp might be sufficient for good results. However, we argue that incorporating multiple types enhances flexibility, particularly when faced with cluttered scenarios. To prove that, we conducted a targeted experiment to compare grasping outcomes with varying numbers of grasp types. Specifically, we employed the best grasp model trained with 144 objects for the Allegro hand, initially defined with 10 grasp types. In our experiment, we compared the original model with two modified versions that use fewer grasp types. The first version was limited to a single grasp type, specifically the one that achieved the highest overall success rate across all types. The second version used a subset of the five most effective grasp types, chosen based on their individual success rates. For simplicity, the evaluation focused exclusively on adversarial objects due to the performance similarity with that on the entire object set. The resulting success rates are detailed in Figure 6C.

The original method, employing 10 grasp types, achieved an 80.3% success rate on the test set. In contrast, utilizing only a single grasp type led to a reduction in the grasp success rate to 67.3%. Employing five grasp types performed better, resulting in a success rate of 77.6%, but is still inferior to the original method. Our experimental results show that increasing the number of grasp types can improve overall grasp success rates. One reason for this improvement is that a greater variety of grasp types provides more flexibility, enabling the hand to better adapt to different object shapes, sizes, and orientations. Additionally, using multiple grasp types can increase tolerance for collisions, allowing the hand to adjust its grasping strategy based on spatial constraints, particularly in cluttered environments. It is noteworthy that, on the other hand, our results also reveal that the benefits derived from further adding grasp types would eventually saturate. Thus, it is reasonable to adopt diverse yet limited grasp types, which optimizes both grasp success rates and learning efficiency.

3. Discussion

3.1. Efficiency Analysis

It is surprising to see that our grasp system can be learned for different hands so efficiently. Previous work for multi-finger grasping usually require thousands of objects and millions of grasp samples (Mahler et al., 2019; Eppner et al., 2021). And in the deep learning era, it seems to be an underlying rule that we need to train a robot system on as many objects as possible to have good generalization ability. However, the satisfactory performance of our system breaks this intuition. What are the key factors for our method to learn such efficiently and generalize so well? There are two aspects of learning efficiency in our system, the first is that we only need 40 objects and the second is that we only need hundreds of trials for each hand. Here we discuss how our method achieves efficiency in these two aspects.

Representation Model

Representation plays a crucial role in our system for real-world learning, as it must map different geometries into a contact-centric grasp representation. How can the system, trained on only 40 objects, generalize to hundreds of unseen objects? We address this by conducting a geometry coverage analysis, revealing that **scaling up data along the right dimension** is key to improving the model's generalization ability.

Our representation model takes a scene point cloud as input and outputs contact-centric grasp representations (CGRs). To train this model, we need a dataset containing scene point clouds with annotated CGRs across various geometries. Since CGRs depend on local geometry, a representative dataset must include

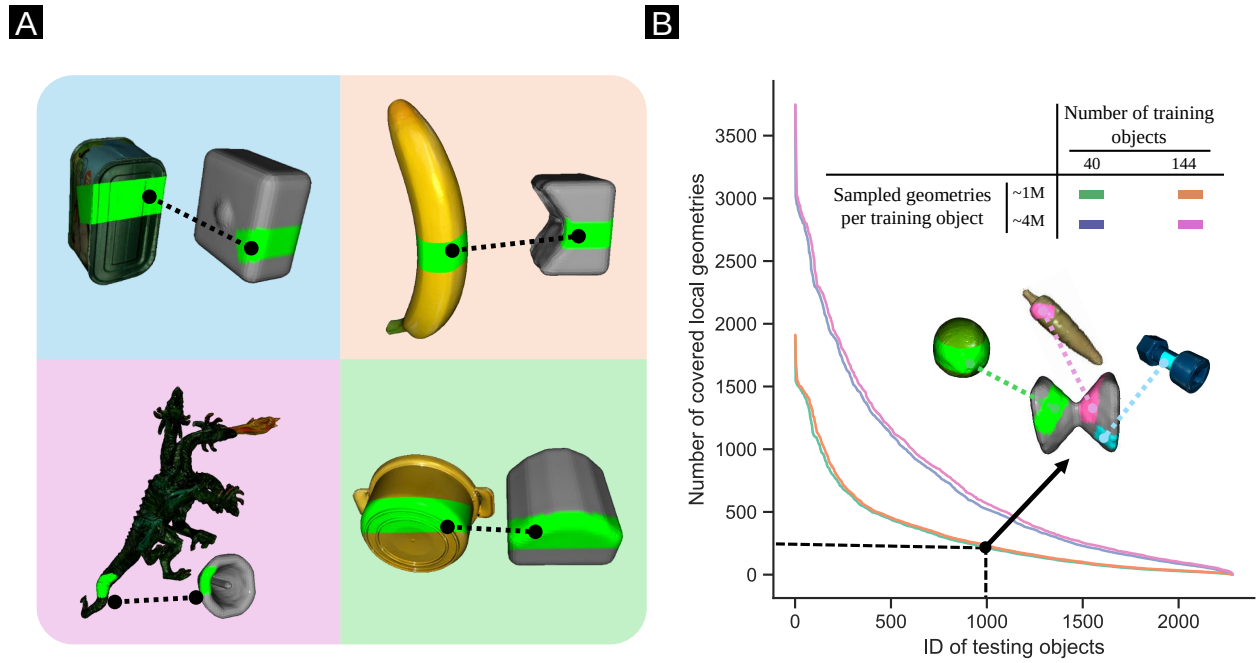

Figure 7: Geometry coverage analysis. (A): The colored objects are our training objects and the gray objects are in the EGAD! object set. The surfaces highlighted in green and connected by a dotted line have similar local geometries. We see that although the training object and testing object have very different overall shape, we can find local geometries on them that are pretty similar. **(B)**: The local geometry coverage curves on the testing set given different choices of scaling up the training set. The x-axis denotes the ID of each testing object, and the y-axis denotes the number of covered local geometries on each testing object. An example is given where the 1000-th testing object has around 250 covered local geometries. We only draw 3 of them for illustration.

diverse local geometries to effectively train deep networks. While many researchers intuitively attempt to collect more training objects to achieve this, our geometry coverage analysis demonstrates that more objects do not necessarily lead to richer local geometries.

We begin by defining the local geometry used in our analysis. Specifically, the representation network operates in a partial observation scenario, where it infers contact positions and normals on unobserved surfaces based on the observed geometry. Although each normal and contact point is predicted independently, we consider the minimal continuous components of local patches, enclosed by the simplest form of grasping—an antipodal grasp—as a foundational element in our analysis for consistency.

Next, we define geometry coverage in the analysis. Given a training and testing object set, a local geometry on a test object is considered "covered" if it closely resembles a local geometry from the training object set. We define similarity by a chamfer distance smaller than 1mm, with examples illustrated in Figure 7A. For a training dataset, we can assess the diversity of local geometries by counting the number of covered geometries on test objects.

In practice, when constructing a training dataset, we need to generate labels for a fixed number of local geometries selected from the training objects, constrained by computational resources. There are two possible dimensions along which to collect more local geometries: increase the number of training objects or increase the sample density on each training object. To assess which dimension is more effective for increasing local geometry diversity, our analysis is conducted as follows. We collect two training object sets: $\mathbb{S}$, with 40 objects, and $\mathbb{L}$, with 144 objects. For each object set, we sample 1 million and 4 million local geometries from each object on average, respectively (sampling details given in supplementary material). This combination results in four different training datasets. The testing object set for the coverage analysis is the EGAD! test set, which contains over 2000 complex, program-generated objects. We sample around

400 local geometries on each test object and evaluate if they are covered. Figure 7B shows the number of covered geometries on each test object, considering different training sets and sampling densities.

Surprisingly, we found that increasing the number of training objects does not significantly increase the coverage rate on the test set. However, increasing the sampling density of local geometries per object leads to a dramatic increase in coverage—*even when the total number of sampled geometries is similar to increasing the object count.* This result demonstrates that increasing sample density for each training object is far more impactful than increasing the number of objects.

Based on this analysis, we prioritize scaling up the label density of CGRs on each training object, rather than increasing the number of training objects, when constructing our dataset. By training on over a billion CGRs, our model has learned to map local geometries to grasp representations effectively, thereby enhancing its ability to generalize to novel objects.

Grasp Decision Model

In the previous section, we discussed how our representation model can generalize well to novel scenes despite being trained on only 40 objects. Now, we turn to the grasp decision model and discuss why it can learn grasp success from just hundreds of trial-and-error attempts. Here, we highlight a few possible reasons.

First, the representation captures all the relevant information about force closure that can be extracted from vision. For a point-to-plane contact problem, the force-closure condition must satisfy the following criteria (Dai et al., 2018):

$$\begin{aligned} &Gf = 0, \\ &GG^\top > \epsilon I_{6\times 6}, \\ &f_i^\top n_i > \frac{1}{\sqrt{\mu^2+1}}|f_i|, \end{aligned} \tag{10}$$

where f is the vector of contact forces acting at each contact point, G is the grasp matrix determined by the positions of the contact points, and n_i represents the surface normal at the i-th contact point. The latter two parameters are the only aspects that a vision model can estimate, and are generated by our representation model. The grasp decision model needs only to learn whether the forces f exerted by the gripper for different grasp types can satisfy Equation (10), given a friction coefficient μ. Although the friction coefficient is unknown, the model tends to learn an average behavior from the training set.

Second, the representation is compact. Instead of dealing with high-dimensional data like images or point clouds, we reduce the input to a 1D vector that represents the shape. This compactness simplifies the mapping from input to grasp quality, making it easier for the grasp decision model to learn.

3.2. Method Positioning

The evolution of visually guided dexterous grasping methodologies within robotics has developed two prominent paradigms: the 6D pose estimation paradigm and the end-to-end grasp learning paradigm. The former relies on the precise estimation of an object's 6D pose and then calculates the hand pose accordingly. It can transfer across different robotic hands easily, but requires prior knowledge of the object's model. On the other hand, the end-to-end grasp learning models do not require explicit object knowledge, yet the trained models lack transferability across different robotic hands.

Our proposed approach explores a middle ground between these two paradigms, which ombining the advantages of both. By developing a contact-centric grasp representation that encapsulates the scene's contact information, we eliminate the need for an object's model beforehand. The CGR preserves critical information pertinent to grasp quality, endowing our system with adaptability and applicability across different morphologies of robotic hands. Moreover, by eliminating the need for an accurate kinematic model, which was frequently used in previous work learned in simulation (Xu et al., 2023; Wan et al., 2023), our method is suitable for soft hand grasp learning.

3.3. Integration with Tactile Sensors

Future directions for research entail expanding the scope of the contact-centric grasp representation model to include a wider array of tactile and sensory information, enabling a more comprehensive understanding of object manipulation. Tactile sensors encapsulate rich information concerning contact positions and contact point normals, mirroring the fundamental attributes of our representation model. This alignment highlights the potential for our method to work well with tactile sensing technology. By leveraging this alignment, the incorporation of tactile sensors can augment the current representation, further refining the contact-centric information available for decision-making in our learning paradigm.

4. Materials and Method

4.1. Baseline Method

Here we introduce our baseline method of multi-finger grasping. Currently, our community can achieve human-level robotic grasping with a parallel-jaw gripper (Fang et al., 2023b). An intuitive approach for multi-finger grasping is to mimic the behavior of parallel grasping. Thus, we propose a baseline method that discovers the principal closing axis of a robotic hand and aligns it with a parallel grasp pose. First, for each grasp type of a robotic hand, we manually designate its principal closing axis, which is the primary direction along which the fingers converge when the hand closes to grasp an object. Then, given a parallel grasp pose and a grasp type of a hand, we can align the multi-finger hand's principal closing axis to the parallel grasp pose. Previous literature (Fan et al., 2019; 2018) also explored similar ways to initialize a multi-finger grasp. In Figure 10 we illustrate the alignment example.

When grasping with a selected robotic hand, we first generate multiple high-score antipodal grasp poses for a single-view point cloud using the AnyGrasp library (Fang et al., 2023b). Then, for each antipodal grasp pose, we align the robotic hand configured in all grasp types with the antipodal pose. It means that for each antipodal grasp pose, we would have multiple multi-finger grasp candidates with different types. For all of the multi-finger grasp candidates across the scene, we run a collision detection based on the partial-view point cloud and robotic hand model. We select the grasp type assigned with the highest antipodal grasp score for the remaining grasp candidates without collision. If multiple grasp candidates have the same grasp score, we randomly select one as the final grasp pose.

4.2. Algorithm Details

Next, we introduce the details of our algorithm, which consists of three steps: learning the representation model, mapping from representation to grasp pose, and learning the grasp decision model.

Representation Model

Our representation model takes a partial-view point cloud as input and generates the contact-centric grasp representation r for different rotation $\mathbf{R}_{3d}$ and translation $\mathbf{t}_{3d}$ across the scene. It may seem initially challenging to establish the representation model, given that this representation demands full surface information, and the r needs to be predicted for SE(3) space across the scene. However, recent advancements in grasp pose detection algorithms have successfully learned the mapping from partial-view point clouds to antipodal grasp poses across the scene, unveiling the feasibility of learning the mapping from partial-view point clouds to the proposed intermediate representation. Specifically, prior works, such as graspnet-baseline (Fang et al., 2020b) and GSNet (Wang et al., 2021), have predicted the gripper opening widths and antipodal scores for discretized rotation $\mathbf{R}_{3d}$ and translation $\mathbf{t}_{3d}$ across the scene:

$$s = \left\{ (w_{\alpha_i}, \mu_{\alpha_i})_j \middle| \alpha_i = 0, \frac{2\pi}{N}, \ldots, \pi - \frac{2\pi}{N},\ j = 1, 2, \ldots, M; \mathbf{R_{3d}}, \mathbf{t_{3d}} \right\}, \tag{11}$$

where $w_{\alpha_i} = 2 \times \max(d_{\alpha_i}, d_{\alpha_i+\pi})$ and $\mu_{\alpha_i} = \max(\tan(\theta_{\alpha_i}), \tan(\theta_{\alpha_i+\pi}))$ are the gripper opening width and antipodal grasp quality metric defined in (Fang et al., 2020b). This representation shares a structural resemblance with our contact-centric grasp representation in Equation (9). Thus, we opt to build our representation model upon the GSNet (Wang et al., 2021) architecture.

When predicting the representation, it is intractable to account for every possible $\mathbf{R}_{3d}$ and $\mathbf{t}_{3d}$ in continuous space. Previous work (Fang et al., 2020b; Wang et al., 2021) addressed this by selecting orientation $\mathbf{R}_{3d}$ from 300 discretized directions, voxelizing the scene, and selecting only $\mathbf{t}_{3d}$ that lies on object surfaces. However, the total number of resulting combinations still remains quite large. In (Wang et al., 2021), a metric called "graspness" is proposed as a heuristic to bias sampling towards $\mathbf{t}_{3d}$ and $\mathbf{R}_{3d}$ values that have a higher probability of generating successful grasp poses. This metric includes two components: "point-wise graspness" and "view-wise graspness." Point-wise graspness is calculated by counting the ratio of high-score antipodal grasp poses among all poses at a given $\mathbf{t}_{3d}$, while view-wise graspness counts this ratio among all grasp poses at a given $\mathbf{R}_{3d}$ for a sampled $\mathbf{t}_{3d}$. These two scores are learned jointly within the grasp pose detection network and guide sampling during inference.

In this work, since we aim to train a hand-agnostic representation model, we define a new "graspness" score for each r to indicate its suitability for subsequent grasping across different robotic hands. Intuitively, for a point $(\alpha_i, d_i, \theta_i)$, a robotic hand achieves better contact when θ_i is small, meaning the surface normal n_i is opposite to the contact direction (assuming the robotic finger approaches towards the polar pole of the local coordinate frame). Additionally, geometries with many high-score antipodal grasp poses tend to be easier for dexterous hands to grasp. Thus, we define "graspness" in this paper as the sum of θ_i values below a threshold and the number of antipodal grasp poses in r. This definition helps the model reduce candidates for representation prediction within a scene without significantly affecting accuracy.

Similar to GSNet, our representation model consists of three cascaded modules. Firstly, a Minkowski Engine (Choy et al., 2019) backbone takes the single-view point cloud as input, encodes their geometric features, and outputs a computed feature vector for each input point. Then a multi-layer-perception (MLP) takes the features of each point and generates a point-wise graspness heatmap. We sample 1024 seed points with high graspness, and forward these points to another MLP block. It outputs the view-wise graspness scores for 300 approach directions towards each seed point respectively. We then select the direction with the highest graspness score for each point, group the features with cylinder grouping (Fang et al., 2020a) along that direction and forward the grouped features for each point through a final MLP block. This final layer outputs r for $N = 48$ in-plane rotations and $M = 5$ grasp depths, which are 0.005m, 0.01m, 0.02m, 0.03m and 0.04m respectively.

Mapping from Representation to Multi-Finger Grasp Candidates

After we obtain the representation r at different positions across the scene, we link them with different grasp types of a robotic hand to generate multi-finger grasp candidates. In theory, since we have predicted the contact information, we can already generate suitable multi-finger grasp candidates through optimization (Miller and Allen, 2004; Liu et al., 2021). However, for simplicity, we follow the same technique adopted in the baseline method to generate multi-finger grasp candidates. Such a design also facilitates fair comparison with the baseline method and shows how our grasp decision network improves the grasping ability.

For each predicted CGR with the form of Equation (9), we calculate the corresponding antipodal grasp representation defined in Equation (11). Then the CGRs with top-500 antipodal grasp scores are selected. These representations are associated with different multi-finger grasp candidates following the same procedure of the baseline method. After this process, we query the orientation $\mathbf{R}_g$ and translation $\mathbf{t}_g$ of the multi-finger grasp candidates associated with the CGRs (more details in supplementary material). Together with the associated grasp types, we map the CGRs to multi-finger grasp candidates.

Learning Multi-Finger Grasping

For each sampled grasp candidate g_i, we learn a mapping from its corresponding CGR r_i to grasp success probability. This mapping is approximated through the grasp decision model, using training data collected via trial and error:

$$\alpha = \Psi(r_i, g_i, h).$$

In practice, we train different decision models for different robotic hands, denoted as $\Psi_h(r_i, g_i)$. Since the grasp types of each hand are discretized, we further decompose the classification of different grasp types

Algorithm 1 Multi-finger Grasping Data Collection for Robotic Hand h

Input: the expected size K of the grasp dataset.
Output: the collected grasp dataset $\mathbf{G}$.

$\mathbf{G} \leftarrow \emptyset$
while $|\mathbf{G}| < K$ **do**
  The ***robot*** moves to the ready pose
  $\mathcal{P} \leftarrow$ ***camera***.perception ▷ capture RGBD images and transform into point cloud
  $\mathcal{R} \leftarrow \Phi(\mathcal{P})$ ▷ generate scene representation from the point cloud
  Sample a CGR $r \in \mathcal{R}$ in the scene
  Sample a grasp type $\mathbf{q} \in \{\mathbf{q}_1, ..., \mathbf{q}_c\}$
  $[\mathbf{R}_g\ \mathbf{t}_g] \leftarrow$ compute_grasp_pose(r) ▷ Map CGR to the grasp pose
  if collision_detection($[\mathbf{R}_g\ \mathbf{t}_g\ \mathbf{q}], \mathcal{P}; h$) **then**
    ▷ Check if the multi-finger grasp pose will collide with the scene point cloud
    continue
  end if
  The ***robotic hand*** executes the multi-finger grasp pose $[\mathbf{R}\ \mathbf{t}\ \mathbf{q}]$
  Record the grasp result S ▷ Collect trial-and-error results
  $\mathbf{G} \leftarrow \mathbf{G} \cup \{\langle r, \mathbf{R}_g, \mathbf{t}_g, \mathbf{q}, S \rangle\}$
end while
return the collected grasp dataset $\mathbf{G}$

into different sub-models:

$$\Psi_h(r_i, g_i) = \sum_{\mathbf{q}} \mathbb{I}(g_i, \mathbf{q})\, \Psi_{h,\mathbf{q}}(r_i, \mathbf{R}_g, \mathbf{t}_g),$$

where $\mathbb{I}(g_i, \mathbf{q})$ is an indicator function that is 1 when g_i matches the grasp type $\mathbf{q}$ and 0 otherwise. Since $\mathbf{R}_g$ and $\mathbf{t}_g$ are functions of r_i, we can simplify the input to the sub-models by removing $\mathbf{R}_g$ and $\mathbf{t}_g$. Thus, $\Psi_{h,\mathbf{q}}(r_i, \mathbf{R}_g, \mathbf{t}_g)$ can be reformulated as $\Psi_{h,\mathbf{q}}(r_i)$, where the computation of $\mathbf{R}_g$ and $\mathbf{t}_g$ is implicit in the model. We empirically found that using different sub-models for different grasp types gives better performance. The input to the model consists of the CGR of a selected grasp candidate. The model's output is a score of whether the selected grasp would be successful. Details of the model is given in supplementary material. For simplicity, we regard the combination of all sub-models for each robotic hand as a single model and still refers to it as the grasp decision model.

Detection Post-Processing

Following the grasp decision model's output, we select grasp poses with high-quality scores, typically exceeding 0.9. Collision detection is then performed by voxelizing the pre-shaped multi-finger hand and examining intersections between the hand voxels and the scene point cloud using the Open3D library. The final grasp pose is chosen from those grasp poses without collision with the scene, with the highest grasp quality score.

4.3. Training Environment

Training Object Set

Our experiments involve three different training object sets, the larger dataset $\mathbb{L}$ contains 144 objects, the smaller one $\mathbb{S}$ contains 40 objects, and the tiniest one $\mathbb{T}$ contains 30 objects. $\mathbb{L}$ is the training set collected in AnyGrasp (Fang et al., 2023b). $\mathbb{S}$ encompasses the 40 training objects featured in the original GraspNet-1Billion dataset, and $\mathbb{T}$ includes 30 randomly selected objects from $\mathbb{L}$. In Figure 8 we detail the three training object sets.

Data Annotation and Collection

To facilitate the training of our representation model, we re-annotate the GraspNet-1Billion dataset. The training set consists of 100 scenes made up of 40 objects. Each scene includes 256 RGBD images, each of which can be transformed into a single-view point cloud. Instead of the original antipodal grasp representation (illustrated in Equation (11)), we annotate the contact-centric grasp representation as per Equation (9) for the 100 training scenes.

Our process begins by voxelizing the 3D mesh of each training object with a resolution of 0.005 m. We collect all points on the voxelized object surface, denoted as $\{\mathbf{t}_{3d}^{(i)}\}$, where i indexes each individual surface point. For each surface point $\mathbf{t}_{3d}^{(i)}$, we sample 300 approach directions $\{\mathbf{R}_{3d}^{(j)}\}$, where j indexes the sampled directions. We then compute the CGR r for each combination of $\mathbf{t}_{3d}^{(i)}$ and $\mathbf{R}_{3d}^{(j)}$. This computation relies on the complete mesh of the object. The computed CGRs are then projected from each training object to the training scenes based on the object's 6D pose provided in the original dataset.

After generating the CGRs for each scene, we apply a simple post-processing step to verify grasp feasibility. For each CGR, we check whether a cylindrical region extending backward along the approach direction collides with the tabletop or other objects in the scene. If a collision is detected, we set the CGR to a zero vector, indicating it is not a viable grasp candidate. This post-processing step helps reduce the likelihood of robotic hand collisions within the scene.

To train the grasp decision model, we collect grasping data by trial and error. Previously, most of the grasp attempts related to multi-finger grasping were collected within a simulation environment. Nevertheless, significant gaps may arise due to the inherent differences between the simulation and real environments. Thus, in this paper, we directly collect grasping data in a real-world environment.

We provide an overview of the complete data collection pipeline, summarized in Algorithm 1. Initially, we randomly place objects on the table. We then run the representation model to generate dense contact-centric grasp representations for the scene. We sample a CGR and a grasp type of the robotic hand, and map the CGR to a multi-finger grasp pose. Collision detection is performed to ensure that the grasp pose does not collide with the scene. If no collision happens, we execute the grasp process. During this process, we record whether the grasp is successful and store the necessary information in the dataset.

Training Details

For the representation model, the input point clouds are down-sampled with a voxel size of 0.005m. In practice, we set the parameters of the 3D representation N and M in Equation (9) to 48 and 5 respectively. The model is trained on the re-annotated GraspNet-1Billion dataset using one Nvidia A100 GPU with Adam optimizer (Kingma and Ba, 2014) and an initial learning rate of 0.001. The learning rate follows a descent strategy and we adopt "poly" policy with $power = 0.9$ for learning rate decay. The model is trained from scratch with a batch size of 4. For data augmentation, we randomly flip the scene horizontally and randomly rotate the points by Uniform$[-30^\circ, 30^\circ]$ around the z-axis (in the camera coordinate frame). We also randomly translate the points by Uniform$[-0.2\text{m}, 0.2\text{m}]$ in the x- or y-axis and Uniform$[-0.1\text{m}, 0.2\text{m}]$ in the z-axis.

For the grasp decision model, since we have a relatively limited amount of collected data, our model is trained for only 20 epochs to avoid overfitting. We leverage the Adam optimizer (Kingma and Ba, 2014). The learning rate follows a segmented descent strategy starting from 0.0001, and the batch size $\mathbf{Z}$ is set to 128 to optimize training efficiency. Since the network is quite small, we train the model on a laptop with NVIDIA 1650 GPU.

4.4. Experimental Procedure

In each experiment, we randomly distribute objects from different categories in the robot workspace. During the grasping process, the partial-view point cloud captured by the camera is fed into our representation model. When collecting training data, we follow the procedure in Algorithm 1. During testing, we first choose 100 CGRs from the outcome of the representation model, which has the top-100 antipodal grasp scores. These CGRs are mapped to multi-finger grasp candidates, and given the number of predefined types

for each robotic hand, the total number of multi-finger grasp candidates varies (*e.g.*, we define 4 grasp types for the three-finger hand, thus it has 400 grasp candidates). These grasp candidates are fed into our grasp decision model. The grasp candidates with the top-200 grasp quality scores then undergo collision detection post-processing. The grasp pose that passes collision detection and has the highest grasp score is selected as the final multi-finger grasp pose in the camera's coordinate system. It is subsequently converted into the world coordinate system and sent to the UR5 robot through socket communication. The UR5's embedded motion planner navigates it to the grasp pose, where we set a waypoint 10 cm backward from the final grasp along the approach direction to avoid collision during movement. Simultaneously, the robotic hand is configured to the selected grasp type. After the robot arm reaches the target pose, the robotic hand closes the fingers until the grasping force reaches a predefined limit. The robot then lifts the object and moves it to the top of the bin and drops the object. The experiment concludes with manually recording whether the robotic hand successfully move the object to target position.

Acknowledgments

The authors would like to thank Xiaolin Fang for the helpful revision and Antonia Bronars and Jiang Zou for the helpful discussion.

References

Sergio Almecija, Salvador Moya-Sola, and David M Alba. Early origin for human-like precision grasping: a comparative study of pollical distal phalanges in fossil hominins. *PLoS One*, 5(7):e11727, 2010.

Samarth Brahmbhatt, Ankur Handa, James Hays, and Dieter Fox. Contactgrasp: Functional multi-finger grasp synthesis from contact. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2386–2393. IEEE, 2019.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019.

Sammy Christen, Stefan Stevšić, and Otmar Hilliges. Guided deep reinforcement learning of control policies for dexterous human-robot interaction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2161–2167. IEEE, 2019.

Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Grégory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5031–5041, 2020.

Mark R Cutkosky et al. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on robotics and automation*, 5(3):269–279, 1989.

Hongkai Dai, Anirudha Majumdar, and Russ Tedrake. Synthesis and optimization of force closure grasps via sequential semidefinite programming. *Robotics Research: Volume 1*, pages 285–305, 2018.

Erik Domellöf, Marianne Barbu-Roth, Louise Rönnqvist, Anne-Yvonne Jacquet, and Jacqueline Fagard. Infant manual performance during reaching and grasping for objects moving in depth. *Frontiers in Psychology*, 6: 1142, 2015.

Clemens Eppner, Arsalan Mousavian, and Dieter Fox. Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227. IEEE, 2021.

Yongxiang Fan, Te Tang, Hsien-Chung Lin, and Masayoshi Tomizuka. Real-time grasp planning for multi-fingered hands by finger splitting. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4045–4052. IEEE, 2018.

Yongxiang Fan, Xinghao Zhu, and Masayoshi Tomizuka. Optimization model for planning precision grasps with multi-fingered hands. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1548–1554. IEEE, 2019.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11444–11453, 2020a.

Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020b.

Hao-Shu Fang, Minghao Gou, Chenxi Wang, and Cewu Lu. Robust grasping across diverse sensor qualities: The graspnet-1billion dataset. *The International Journal of Robotics Research*, 42(12):1094–1103, 2023a.

Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023b.

Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on human-machine systems*, 46(1):66–77, 2015.

Patrick Grady, Chengcheng Tang, Christopher D Twigg, Minh Vo, Samarth Brahmbhatt, and Charles C Kemp. Contactopt: Optimizing contact to improve grasps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1471–1481, 2021.

Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3786–3793. IEEE, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tracy L Kivell. Evidence in hand: recent discoveries and the early evolution of human manual manipulation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1682):20150105, 2015.

Puhao Li, Tengyu Liu, Yuyang Li, Yiran Geng, Yixin Zhu, Yaodong Yang, and Siyuan Huang. Gendexgrasp: Generalizable dexterous grasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8068–8074. IEEE, 2023.

Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep differentiable grasp planner for high-dof grippers. In *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, Massachusetts, 2020.

Qingtao Liu, Yu Cui, Qi Ye, Zhengnan Sun, Haoming Li, Gaofeng Li, Lin Shao, and Jiming Chen. Dexrepnet: Learning dexterous robotic grasping network with geometric and spatial hand-object representations. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3153–3160. IEEE, 2023.

Tengyu Liu, Zeyu Liu, Ziyuan Jiao, Yixin Zhu, and Song-Chun Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *IEEE Robotics and Automation Letters*, 7(1):470–477, 2021.

Jens Lundell, Francesco Verdoja, and Ville Kyrki. Ddgc: Generative deep dexterous grasping in clutter. *IEEE Robotics and Automation Letters*, 6(4):6899–6906, 2021.

Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio, and Ken Goldberg. 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)*, Cambridge, Massachusetts, July 2017.

Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26), 2019.

Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In *Conference on Robot Learning*, pages 651–661. PMLR, 2022.

Andrew T Miller and Peter K Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, 2004.

Douglas Morrison, Peter Corke, and Jürgen Leitner. Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 5(3):4368–4375, 2020.

Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *International Conference on Computer Vision (ICCV)*, 2019.

Karl M Newell, Deirdre M Scully, PV McDonald, and Renée Baillargeon. Task constraints and infant grip configurations. *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology*, 22(8):817–831, 1989.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022.

Carlos Rosales, Lluís Ros, Josep M Porta, and Raúl Suárez. Synthesizing grasp configurations with specified contact regions. *The International Journal of Robotics Research*, 30(4):431–443, 2011.

Kenneth Shaw, Shikhar Bahl, Aravind Sivakumar, Aditya Kannan, and Deepak Pathak. Learning dexterity from human hand motion in internet videos. *The International Journal of Robotics Research*, 43(4):513–532, 2024.

Matthew M. Skinner, Nicholas B. Stephens, Zewdi J. Tsegai, Alexandra C. Foote, N. Huynh Nguyen, Thomas Gross, Dieter H. Pahr, Jean-Jacques Hublin, and Tracy L. Kivell. Human-like hand use in australopithecus africanus. *Science*, 347(6220):395–399, 2015. doi: 10.1126/science.1261735.

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

Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. *arXiv preprint arXiv:2304.00464*, 2023.

Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15964–15973, 2021.

Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023.

Wei Wei, Daheng Li, Peng Wang, Yiming Li, Wanyi Li, Yongkang Luo, and Jun Zhong. Dvgg: Deep variational grasp generation for dextrous manipulation. *IEEE Robotics and Automation Letters*, 7(2):1659–1666, 2022.

Wei Wei, Peng Wang, Sizhe Wang, Yongkang Luo, Wanyi Li, Daheng Li, Yayu Huang, and Haonan Duan. Learning human-like functional grasping for multi-finger hands from few demonstrations. *IEEE Transactions on Robotics*, 2024.

Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

Supplementary Methods

Query 6D Grasp Pose for CGR

When we map a CGR to grasp pose, we first calculate the antipodal grasp representation of the CGR. Given a CGR

$$r = \left\{ (d_{\alpha_i}, \theta_{\alpha_i})_j \,\middle|\, \alpha_i = 0, \frac{2\pi}{N}, \ldots, 2\pi - \frac{2\pi}{N},\ j = 1, 2, \ldots, M; \mathbf{R}_{3d}, \mathbf{t}_{3d} \right\}, \tag{12}$$

the antipodal grasp representation is calculated by

$$s = \left\{ (w_{\alpha_i}, \mu_{\alpha_i})_j \middle| \alpha_i = 0, \frac{2\pi}{N}, \ldots, \pi - \frac{2\pi}{N},\ j = 1, 2, \ldots, M; \mathbf{R_{3d}}, \mathbf{t_{3d}} \right\},$$

where $w_{\alpha_i} = 2 \times \max(d_{\alpha_i}, d_{\alpha_i+\pi})$ and $\mu_{\alpha_i} = \max(\tan(\theta_{\alpha_i}), \tan(\theta_{\alpha_i+\pi}))$. After we obtained s, we choose the α_i and j that has the maximum antipodal grasp score:

$$(\alpha_i^*, j^*) = \arg\max_{\alpha_i, j} (\mu_{\alpha_i})_j.$$

We add the rotation α_i^* and translation corresponds to the j^*-th section along approach direction to $\mathbf{R_{3d}}$ and $\mathbf{t_{3d}}$:

$$\begin{aligned} \mathbf{R_g} &= \mathbf{R_{3d}} \cdot \mathbf{R}_z(\alpha_i^*), \\ \mathbf{t_g} &= \mathbf{t_{3d}} + \mathbf{d}(j^*) \cdot \mathbf{R_{3d}} \cdot \mathbf{R}_z(\alpha_i^*) \cdot \mathbf{z}, \end{aligned}$$

where:

$$\mathbf{R}_z(\alpha_i^*) = \begin{bmatrix} \cos(\alpha_i^*) & -\sin(\alpha_i^*) & 0 \\ \sin(\alpha_i^*) & \cos(\alpha_i^*) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$\mathbf{d}(\cdot)$ is a function that maps the index j of the section to its actual depth along the approach direction (maps $\{1, 2, 3, 4, 5\}$ to $\{0.005\text{m}, 0.01\text{m}, 0.02\text{m}, 0.03\text{m}, 0.04\text{m}\}$), and:

$$\mathbf{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The updated rotation $\mathbf{R_g}$ and translation $\mathbf{t_g}$ are the 6D grasp pose that corresponds to this CGR.

Details of Grasp Decision Model

Each grasp decision sub-model is learned by a neural network. It takes a contact-centric grasp representation as input and outputs a score ranging from 0 to 1 to indicate whether the corresponding grasp candidate would be successful. The input size is $2 \times 5 \times 48 = 480$, which is composed of distances and normal angles on 5 sections along 48 in-plane rotations.

The network comprises seven fully connected layers with a skip connection for improving robustness. Each intermediate layer consists of a fully connected layer with 1024 neurons, a batch normalization layer, and a ReLU activation function. The output of the second intermediate layer is also forwarded to the fifth intermediate layer with a skip connection. Networks for different grasp types are trained separately. We employ a loss function defined as:

$$L = -\frac{1}{\mathbf{Z}} \sum_{z=1}^{\mathbf{Z}} y_z \log(p_z). \tag{13}$$

In this equation, L is the loss, y denotes the binary label of whether the real robot trial and error succeeded or not, and p represents the predicted grasp success probability by the network. $\mathbf{Z}$ denotes the batch size.

Local Geometry Sampling for Grasp Coverage Analysis

The local geometries are cropped using 3D boxes defined by valid antipodal grasp poses. To obtain the grasp candidates, each object is voxel-downsampled to get grasp points in uniform distributions. V approach directions are sampled on the grasp point. A inplane rotation angles are sampled uniformly for each direction. On the training objects, we set V=100 and A=12 for the dense set, and V=50 and A=6 for the sparse set, respectively. In these two cases, the average numbers of local geometries for each training object are around 1M and 4M. For testing objects in the EGAD dataset, we set V=100 and A=12.

Supplementary Text

Training Object Collection

The 144 training objects are collected from supermarkets and grocery stores, which is extended from the 40 training objects collected in GraspNet-1Billion (Fang et al., 2023a). The principle of choosing objects is that they have a roughly different shape or some local geometries from other objects, and they are chosen by authors heuristically. We provide the 3D scanned models of the objects to support reproducible research. Figure 8 shows an overview of the training object.

Grasp Types for Different Hands

The index number for each grasp type of different robotic hands is given in Figure 9.

Principal Closing Axis for Different Grasp Types

We illustrate the principal closing axis for different grasp types in Figure 10. The x-axis (in red) in the local coordinate frame is the approach direction and the y-axis (in green) is the principal closing axis of the hand. The two-finger gripper (in blue) is the corresponding antipodal grasp pose for each grasp type.

Supplementary Movies

We believe that presenting the complete process of our robotic grasping experiments can provide valuable insights into potential improvements for the grasping system. Additionally, it is essential to demonstrate the system's robustness, which requires running it for an extended period. Therefore, we recorded the entire grasping process, retaining all original content without cuts, but with speed adjustments to keep the video at a reasonable length. The grasping process for each robotic hand lasts over 3 hours, with the total time across all three hands exceeding 15 hours. We applied a 20x speed-up for the collision detection phase and a 2x speed-up for the grasp execution phase. Even after these adjustments, the resulting videos still exceed 6 hours in length. Consequently, we have hosted the videos on YouTube, with the links provided below:

- Movie S1 - Grasping with 3-finger DH-3 hand on daily objects, after training on 144 objects: https://youtu.be/GGBesshyfxk
- Movie S2 - Grasping with 4-finger Allegro hand on daily objects, after training on 144 objects: https://youtu.be/HkrvWm_TTGo
- Movie S3 - Grasping with 5-finger Inspire hand on daily objects, after training on 144 objects: https://youtu.be/3Om7G8nMJPg
- Movie S4 - Grasping with 3-finger DH-3 hand on adversarial objects, after training on 144 objects: https://youtu.be/GGBesshyfxk?t=1837
- Movie S5 - Grasping with 4-finger Allegro hand on adversarial objects, after training on 144 objects: https://youtu.be/E7i3pqxA4RM
- Movie S6 - Grasping with 5-finger Inspire hand on adversarial objects, after training on 144 objects: https://youtu.be/o6LQwRgu82s
- Movie S7 - Grasping with 3-finger DH-3 hand on daily and adversarial objects, after training on 40 objects: https://youtu.be/--5wIHfPoZs
- Movie S8 - Grasping with 4-finger Allegro hand on daily objects, after training on 40 objects: https://youtu.be/uhaC8NORqm4

- Movie S9 - Grasping with 4-finger Allegro hand on adversarial objects, after training on 40 objects: https://youtu.be/5pN6BYOH4xw
- Movie S10 - Grasping with 5-finger Inspire hand on daily objects, after training on 40 objects: https://youtu.be/GQDLTVjXPQk
- Movie S11 - Grasping with 5-finger Inspire hand on adversarial objects, after training on 40 objects: https://youtu.be/B7qc7qRw4ss

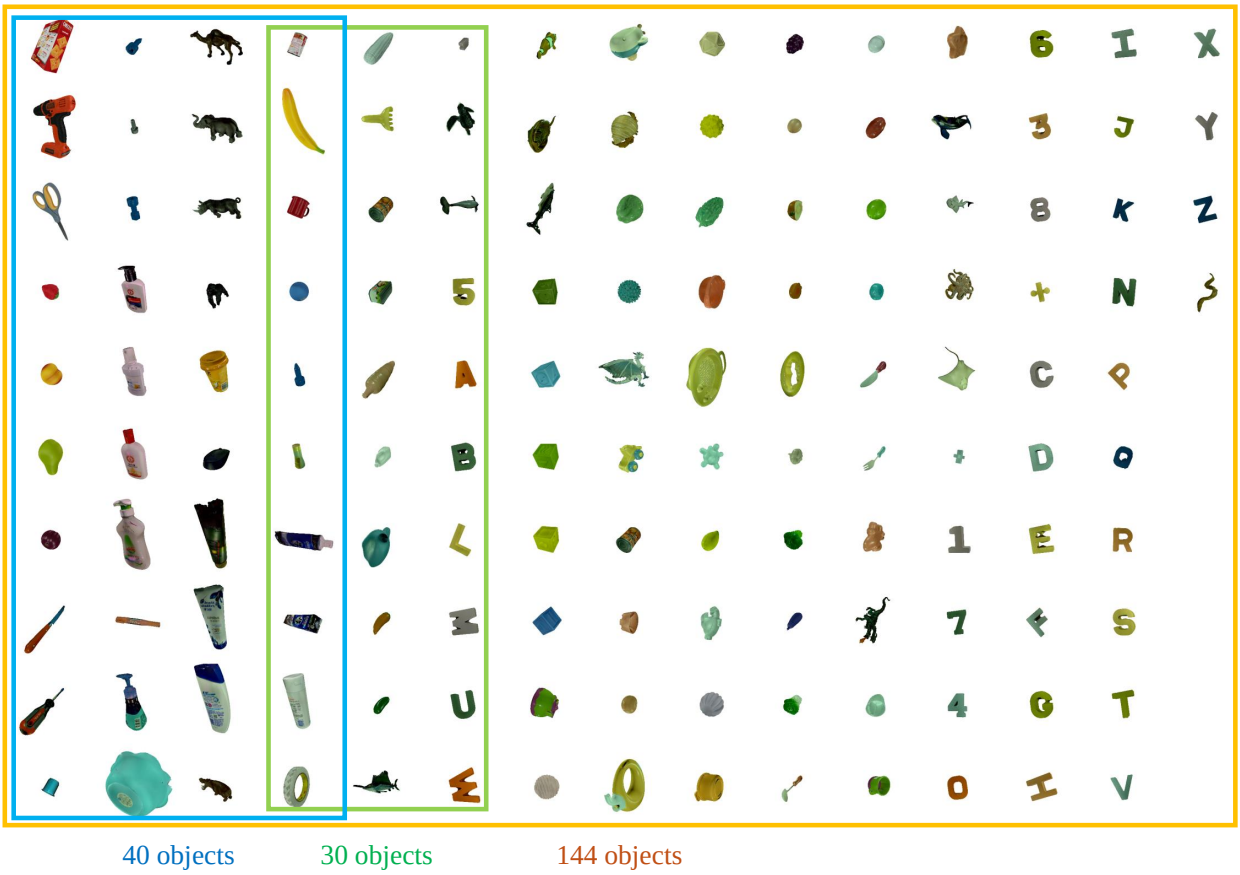

Figure 8: Training object set. The set $\mathbb{L}$ with 144 training objects are enclosed by the orange rectangle, the set $\mathbb{S}$ with 40 training objects are enclosed by the blue rectangle, and the set $\mathbb{T}$ with 30 training objects are enclosed by the green rectangle. Their CAD models are available upon request.

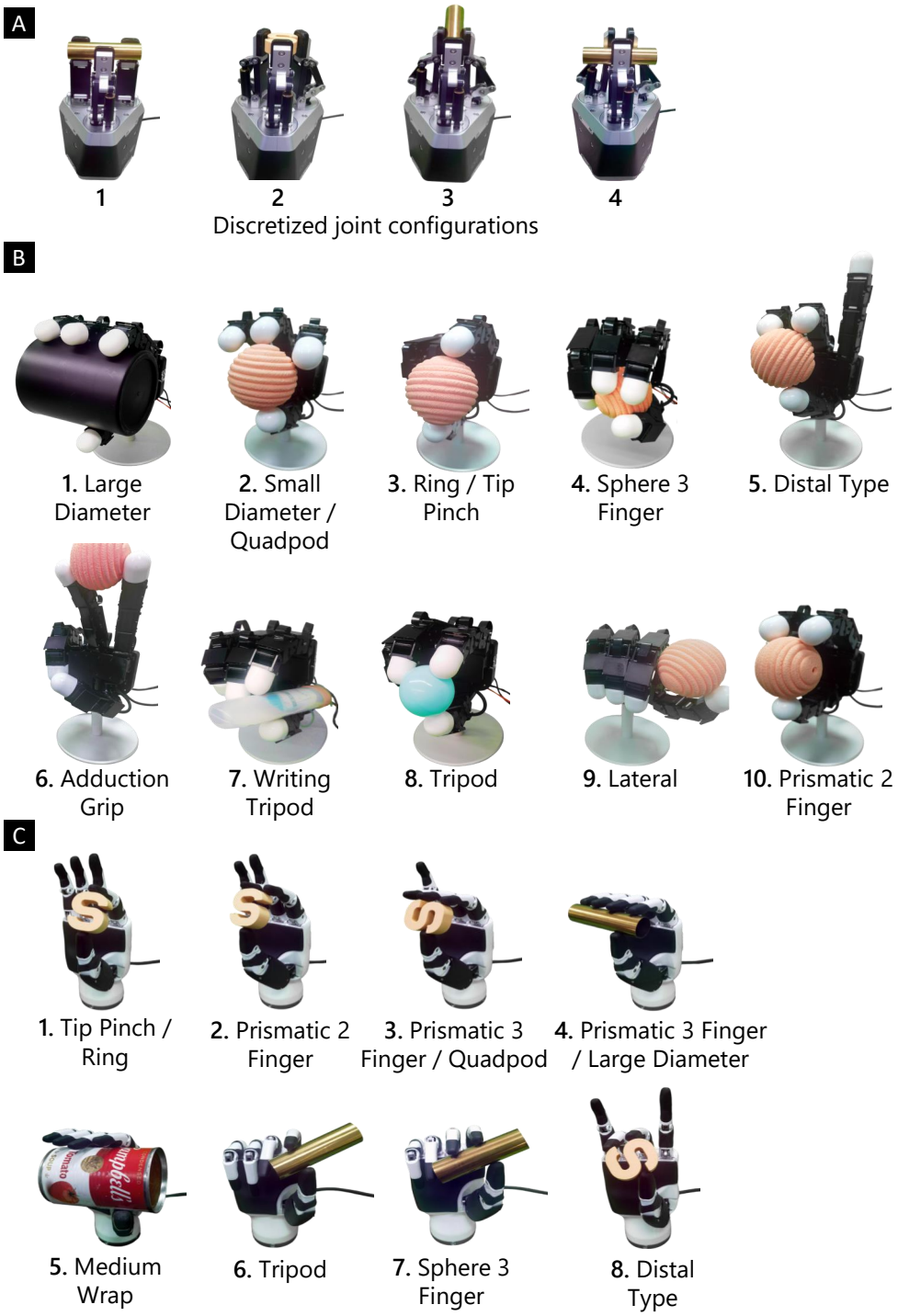

Figure 9: Grasp type numbering. **(A)**, **(B)** and **(C)** give the index numbers of different grasp types for the three-finger, four-finger, and five-finger hands, respectively.

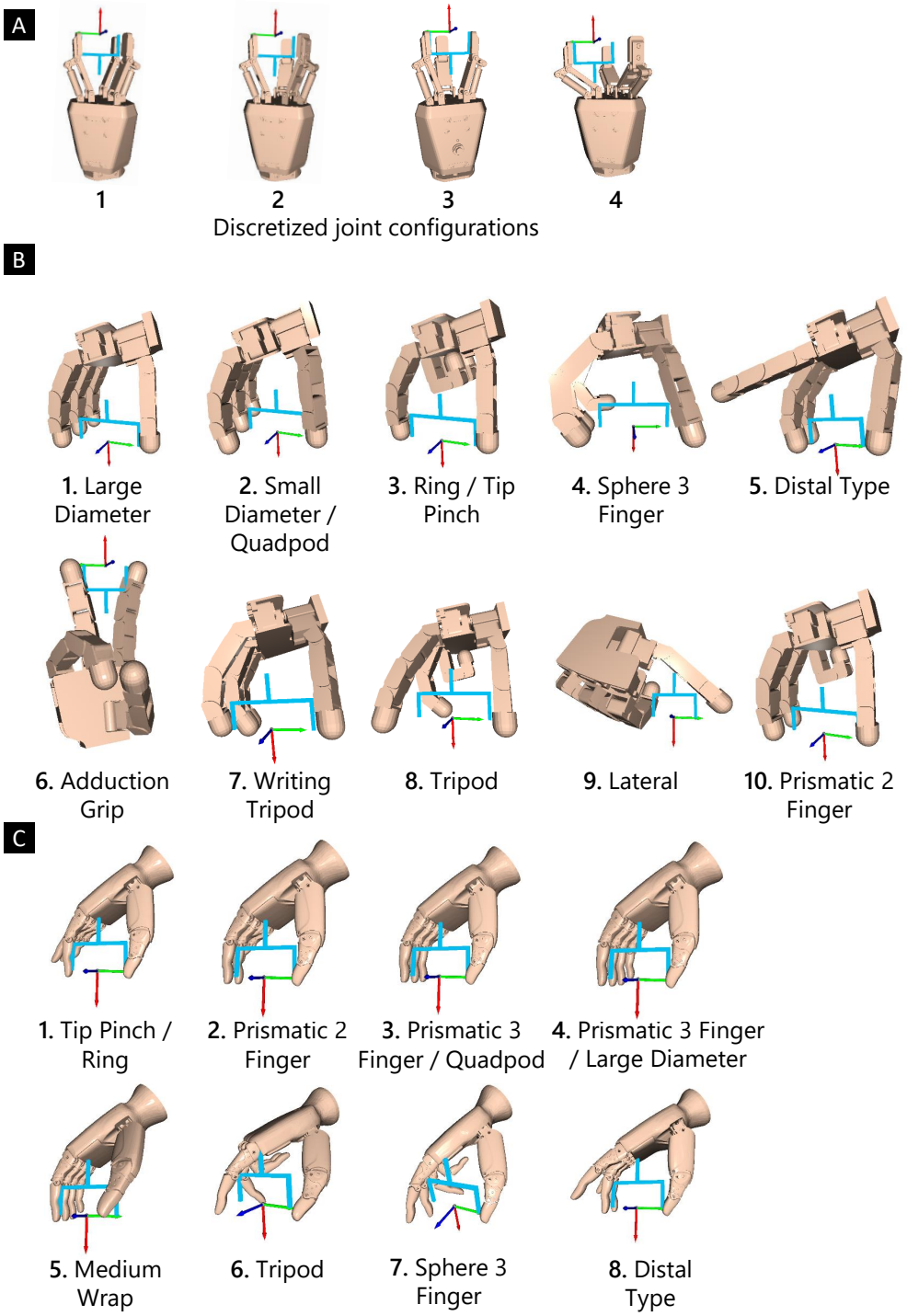

Figure 10: Identifying principal closing axis. We show the designated principal closing axis and corresponding antipodal grasp pose for each predefined grasp type. **(A)**, **(B)** and **(C)** shows the results for the three hands we used.