Grasp it Like a Pro 2.0: A Data-Driven Approach Exploiting Basic Shapes Decomposition and Human Data for Grasping Unknown Objects

Alessandro Palleschi[§,†], *Student Member, IEEE*, Franco Angelini[§,†], *Member, IEEE,* Chiara Gabellieri*, *Member, IEEE,* Do Won Park[§], Lucia Pallottino[§,†], *Senior Member, IEEE,* Antonio Bicchi[§,†,‡], *Fellow, IEEE,* Manolo Garabini[§,†], *Member, IEEE*

***Abstract*—With the improvements in their computational and physical intelligence, robots are now capable of operating in real-world environments. However, manipulation and grasping capabilities are still areas that require significant improvements. To address this, we introduce a new data-driven grasp planning algorithm called Grasp it Like a Pro 2.0. This algorithm utilizes a small number of human demonstrations to teach a robot how to grasp arbitrary objects. By decomposing objects into basic shapes, our algorithm generates candidate grasps that can generalize to different object's geometry. The algorithm selects the grasp to execute based on a selection policy that maximizes a novel grasp quality metric introduced in this work. This metric considers the complex interdependencies between the predicted grasp, the local approximation produced by the basic shape decomposition, and the gripper used. We evaluate our approach against multiple baselines using different grippers and objects. The results demonstrate the effectiveness of our method in generating and selecting high-quality and reliable grasps. With a soft underactuated robotic hand, our algorithm achieves a 94.0% success rate in 150 grasps across 30 different objects. Similarly, with a rigid gripper, it achieves an 85.0% success rate in 80 grasps across 16 different objects.**

***Index Terms*—Grasping, Multifingered hands, Perception for Grasping and Manipulation, Human-driven Grasping**

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreements No. 871237 (Sophia), No. 101017274 (DARKO), and No. 101016970 (NI), in part by the Ministry of University and Research (MUR) as a part of the PON 2014-2021 "Research and Innovation" resources—Green/Innovation Action—DM MUR 1062/2021, and by the Italian Ministry of Education and Research in the framework of the CrossLab and FoReLab projects (Departments of Excellence).

Corresponding Author: Alessandro Palleschi `alessandro.palleschi@phd.unipi.it`

[§]Centro di Ricerca "Enrico Piaggio", Università di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy

[†]Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy

*Robotics and Mechatronics Lab, EEMCS Faculty, University of Twente, 7500 AE Enschede, The Netherlands

[‡]Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, via Morego, 30, 16163 Genova, Italy

Fig. 1. The ability to grasp previously unseen objects with different grippers, adapting to imperfectly known, highly dynamic, and unstructured situations is crucial to enable general-purpose robots to be effective in a large field of use cases.

I. Introduction

Grasping objects is a fundamental skill that humans acquire easily, allowing them to manipulate a wide range of objects effortlessly. However, current robotic manipulation and grasping capabilities still lag behind human abilities [1], hindering the development of general-purpose robots capable of operating in unstructured and dynamic environments [2], [3]. Because of this, much effort has been, and is currently being, devoted by the robotics community in studying the theory of grasping with a focus on grasp planning for unknown objects [4].

Grasp planning methods are classically divided into two categories [5]: analytical and data-driven approaches. Analytical methods rely on well-established force and form closure theory to plan contact points for a stable grasp assuming complete knowledge of the object's geometry and physics [6]–[9] and thus lack in flexibility and robustness when the robot does not have access to a model of the object to grasp. For this reason, data-driven approaches have become more and more popular within the last 20 years [10]–[21], as they show greater flexibility and performance in uncertain settings.

These methods rely on the generation of a set of candidate grasps using, e.g., heuristics or learning from

data, and on the consequent ranking of the grasps within this set [22]. Typically based on deep learning approaches, such methods exploit large datasets of objects and labeled grasps (generally designed for parallel rigid grippers) to train neural networks for grasp-detection [23] or grasp evaluation [15]. Other approaches perform the training using synthetic datasets [17], [24] relying on a model of the gripper used; or exploit human-grasp demonstrations [18], [19] to generate human-like grasps. While deep learning-based approaches are promising tools for grasp planning, they might require a large amount of training data and time [25], [26] to be robust and adaptable to novel objects and to more complex grippers with many degrees of freedom or provided with compliant elements. Indeed, for these cases, datasets of labeled grasps might not be available or reliable models for use in simulators could be difficult to obtain. The development of lightweight and data-efficient algorithms that can be adapted to different grippers and that can synthesize valid grasps for a wide variety of objects is still an open problem.

In this paper, leveraging on the framework proposed by the authors in [19], we present *Grasp it Like a Pro 2.0* (GLP 2.0), a data-driven grasp planning algorithm that is able to generate grasps for unknown objects with different grippers (see Fig.1). The method only requires human demonstrations of grasps, composed of 6-DoF hand poses and interaction forces, of basic shapes. The collected demonstrations are used to learn a model used for grasp synthesis.

Given the point cloud of an unknown object as input, GLP 2.0 decomposes it into the same basic shapes used for training. It then uses the learned model to generate candidate grasps for these basic shapes. A global analytical grasp quality score is introduced to evaluate and select the grasps. The score takes into consideration the characteristics of the gripper used by the robot, the acquired point cloud, possible collisions with the environment, and an estimate of the grasp interaction forces obtained through the learned model.

The main contributions of the paper are:

1) the design and implementation of the data-driven method, GLP 2.0, to generate 6 DoF grasp poses for unknown objects;
2) the design of a novel grasp selection policy based on an analytical grasp quality score to rank and select the generated grasps;
3) the extensive experimental validation of GLP 2.0 on a compliant underactuated robotic hand, the Pisa/IIT SoftHand [27], with a direct comparison with our previous approach [19]. The method achieves relevant performance in terms of grasping success rate on a total of 30 objects and 150 grasps (5 per object), showing a 25% improvement;
4) an experimental comparison, using the Pisa/IIT SoftHand, of GLP 2.0 with two state-of-the-art algorithms [15], [28] on 16 objects and 80 grasps. Results show that our approach outperforms the two baselines in terms of grasping success rate;
5) an implementation of GLP 2.0 with a more standard, rigid, gripper, showing that the framework can be transferred and applied to different robotic hands;
6) an experimental comparison of GLP 2.0 applied to a two-finger rigid gripper, the Franka Emika Hand [29], on 16 objects and 80 grasps. The results show that GLP 2.0 achieves good performance, comparable to the ones obtained with the compliant hand for the same set of objects, and it outperforms the two baselines [15], [24] in terms of grasping success rate;
7) a detailed and critical discussion of the limitations of GLP 2.0.

The structure of the paper is the following. In Section II we review the relevant literature, Section III describes the main component of the algorithm, while in Sections IV and V we describe the experimental setup and protocol used to validate the method with both compliant and rigid grippers, discussing the results obtained. Eventually, we draw our conclusions and discuss directions for future research in Sections VI and VII.

II. Related Works

Grasping is one of the most popular research topics in the robotics community, and over the years many approaches and solutions have been proposed for grasp synthesis. Beside the distinction between analytical and data-driven methods, they are classified in [12] according to: the information they assume to have about the target object (known, familiar, unknown), the features used for the synthesis (2D, 3D, or multi-modal), the object-grasp representation (local or global object attributes), and the specific hand used (standard grippers, multi-fingered hands, or underactuated/soft end-effectors).

A. Known Objects

Grasp synthesis for known objects relies on a complete knowledge of the target object. This knowledge is used to generate offline a set of grasps from which select a feasible candidate. Then, once an object belonging to the database is encountered, the problem is to select a feasible grasp given the environmental conditions [10], [30]–[32].

However, human environments are characterized by a large variety of objects, with different shapes, sizes, and materials. This high variability makes it problematic to use techniques that require a complete knowledge of the object. Indeed, this would lead to the long-lasting and time-consuming process of providing the robot with a model for each possible object it might encounter.

B. Familiar Objects

The limitations of the grasping methods based on the full knowledge of the object can be overcome by approaches that exploit the fact that many every-day objects share similar/familiar and common characteristics [18], [33]–[38]. By exploiting this familiarity, it is possible to train on a set of objects and generalize to novel objects

that fall within one of the categories in the training set. This relaxes the necessity of having an exact model for every object to be grasped.

Grasp synthesis for familiar objects can help increasing the generality of the grasp creation process. Nonetheless, their performance depends on the quality and variety of the data used for training. An erroneous categorization of a novel object could produce unreliable grasps [4], but acquisitions of large datasets is a time-consuming and non-trivial operation [26]. The use of synthetic datasets [24] generated through simulations with reliable simulators like Graspit! [39] could ease the data generation phase, but relying on a model of the hand used for grasping their applicability to more complex hands other than rigid ones is still an open problem [40], [41].

Therefore, a great effort has been put by the scientific community in developing grasping algorithms for unknown objects, i.e., not relying on any prior information on the object, but only data acquired from perception. The method we propose falls into this category.

C. Unknown Objects

Given an unknown object, grasp candidates can be generated from acquired partial and not-complete point-clouds. The approach proposed in [4] uses an approximation of the gripper shape (using a two-layer C-shape cylinder) and then searching for this shape on the partial point-cloud. This is conceptually similar to the method proposed in [42], which however did not exploit depth measures. Other approaches exploit the inherent symmetry of many commonly used objects to generate a full model from a partial point-cloud using geometric considerations [43], [44] or deep-learning [45]. The shape is then used to generate a set of grasp candidates. In [46], starting from a noisy point-cloud, grasps for a multi-fingered hand are generated based on a shape complementarity metric between the cloud of the object and the shape of the hand, whose kinematic model is assumed to be known.

A different approach that relies only on 2D images is presented in [47]. They use curvature information obtained from the silhouette, combined with a visual-servoing control to maximize the curvature at the grasping points, to achieve a correct grasping pose. The work proposed in [15] generates grasp hypotheses for a 2-fingered gripper on any visible surface of the input point cloud. It also proposes a new grasp descriptor takes into account local surface normals and different viewpoints. In [48], the authors propose a new grasp planning algorithm that takes into account both object geometry and gripper characteristics as inputs. A deep neural network is used to predict a set of contact points from the point cloud of the target object that are in force closure and reachable by the hand. The use contact points as output allows to transfer between different multi-fingered hands, assuming that a kinematic model is available.

Besides, there exists a class of methods that attempt to resolve the problem of grasping unknown and potentially irregularly shaped objects using soft and compliant grippers [49], [50]. The planning and control of the grasp is simplified, using, e.g., simple top-down grasps [50], and the embodied intelligence and adaptability of the soft gripper is exploited to increase the robustness.

The approaches presented so far use either global information about the shape or low-level local features to generate the grasp hypothesis. A different solution, like the one proposed in this work, is instead to use approximations of the object using basic primitive shapes. Approaches of this type mainly differ based on the type and number of primitive shapes employed.

A single quadric, estimated from multiple views, is used in [51] to approximate the object shape and plan grasp poses for a multi-fingered hand. The approach presented in [52] uses a single superquadric model to approximate both the shape of the unknown object and the volume graspable by an anthropomorphic hand. The grasping pose is then obtained as the solution of an optimization problem. In [53] a partial view of the object is used to generate a superquadric model. They assume symmetry to complete the object model to fit for the superquadrics representation. The grasp is then designed to maximize the stability and force balance, using the fitted parameters to determine the best contact points for a two-fingered gripper.

However, decomposition via a single primitive shape often fails to provide an accurate approximation of the object [30], [54]. In [54], the authors decompose the object into a multilevel tree of superquadrics used to select subspaces likely to contain good grasps. Sampling of these subspaces and evaluation using Graspit! are then used to find stable grasps. Multiple superquadrics are also used in [55] to estimate the surface of an object from 2.5D data.

The approach we propose does not use superquadrics to approximate the object shape, but, as well as other methods [19], [56], [57], uses the minimum volume bounding box (MVBB) decomposition. This choice represents an effective trade-off between computational effort and quality of the approximation [30] and have been widely applied as supportive method for grasp synthesis. The MVBB decomposition has been used in [56] to generate a grasping pose for a given box based on a user-defined geometric heuristic. Random local variations of the selected pose are then tested in simulation on a set of synthetic point clouds of unknown objects.

Based on this work, in [19] we proposed a data-driven method for grasping unknown objects. In [19] the grasping pose of the robotic hand associated to a box was no longer based on a geometric heuristic, but used a small set of demonstrations provided by a human operator operating the same hand to generate human-like poses. The quality of a grasp was then evaluated, considering the relative box-hand alignment and the possible collisions with the environment. The algorithm showed good performance, being able to generate valid grasps for an underactuated compliant robotic hand with 19 degrees of freedom without needing any information about the object other than

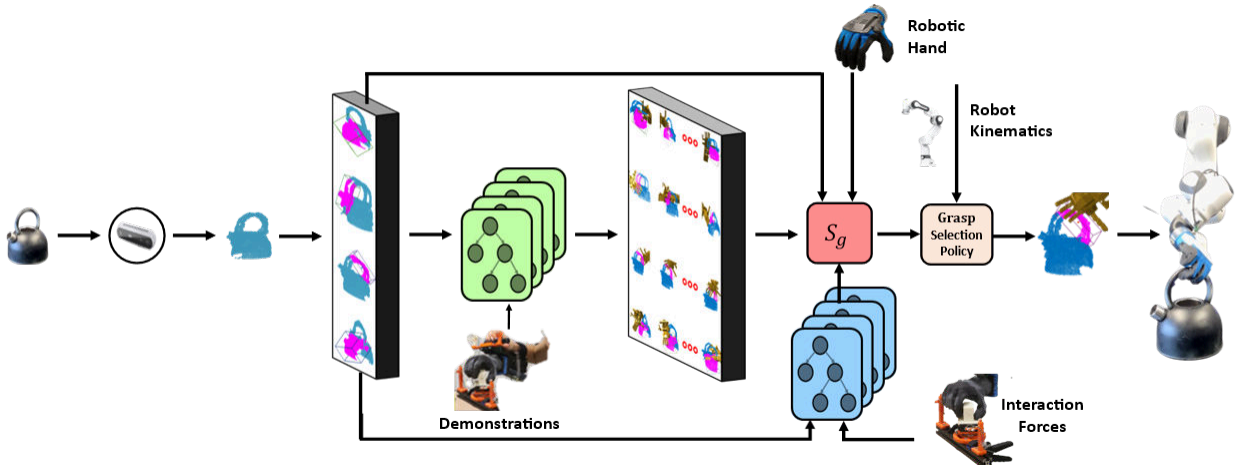


Fig. 2. Pipeline of the proposed method Grasp it Like a Pro 2.0. GLP 2.0 starts with the acquisition of a point cloud of the target object. The cloud is decomposed into a generic number N of minimum volume bounding boxes. A model learned from human demonstrations of grasps for exemplary boxes is used to generate a set of 6-DoF grasp poses for the obtained decomposition. A novel grasp quality score, $\mathbb{S}_g$, is introduced based on information about the robotic gripper, the point cloud of the object and the environment, and of the grasp interaction forces estimated using a learned model to rank and select the best grasp among the candidate set.

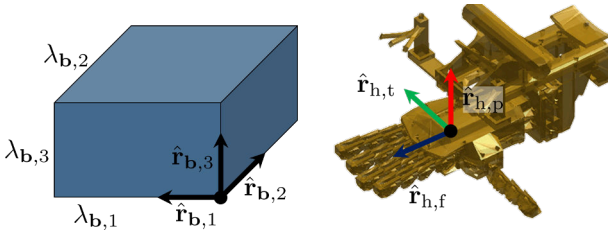


Fig. 3. (Left) Box-related reference frame. (Right) Gripper-related reference frame.

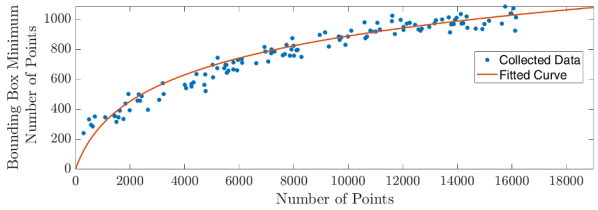


Fig. 4. Collected data and fitted curve used to select μ.

those acquired with its perception, or to model the robotic hand, but using only a limited set (648) of demonstrations. Nonetheless, it was validated only for the Pisa/IIT SoftHand.

In this paper, building upon this framework, we propose a data-driven grasp planning algorithm for grasping unknown objects that can be adapted to different grippers. In the following section, a detailed description of the proposed method is reported.

III. The Grasp Planning Algorithm

In this section, we describe the main components of the proposed grasp planning algorithm, GLP 2.0. Figure 2 reports a schematic visualization of the proposed pipeline.

The approach starts with the acquisition of a point cloud of the target object. The cloud is decomposed into a generic number N of minimum volume bounding boxes. Secondly, exploiting a Decision Regressor Tree (DTR) trained on recorded data of a skilled human grasping sample boxes with the same gripper, a set of candidate grasps is generated from the obtained box decomposition. These poses are then ranked according to a specific metric, that takes into consideration the geometry and properties of the robotic gripper and an estimate of the interaction forces, and the best grasp is selected for execution.

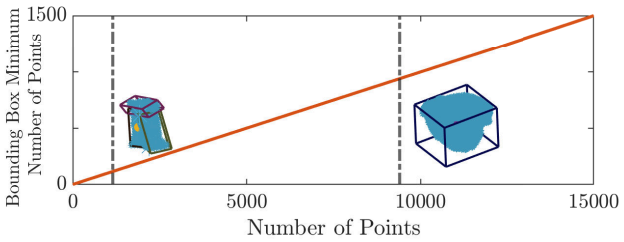
The approach proposed in this paper is inspired by the method we presented in [19], but differs from it in three key aspects: the approximation of the object shape with the **box-decomposition** algorithm, the **grasp-generation** policy, i.e., the definition of the set of candidate grasps and finally the **grasp-selection** policy, i.e., the metric used to select the best grasp.

In the following, we present a detailed description of all the components of the proposed algorithm.

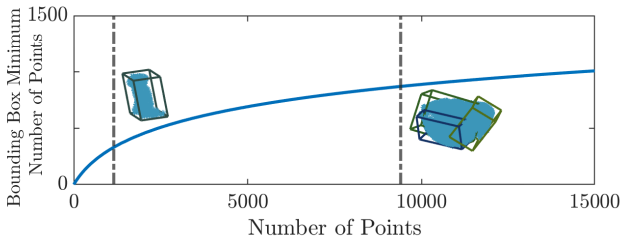
A. Object Acquisition and Shape Approximation

The first step of the planning algorithm is the acquisition of the object point cloud through the use of RGB-D sensors. Following [19], the point cloud is processed in order to decompose the object into a number N of bounding boxes. We decided to use a box-decomposition method for approximating the shape of the acquired point cloud because we found out cuboids represent a good tradeoff between computational effort for obtaining a decomposition and capability of approximating the object shape for grasp planning purposes. Indeed, decomposition based on minimum-volume bounding boxes have been already successfully used in the literature [19], [30], [56] for this kind of problem. In addition, as we will highlight in the next section, this solution allows us to increase the data efficiency of the proposed approach that can use a small set of human demonstrations. The decomposition is performed using a modified C++ implementation[1] of

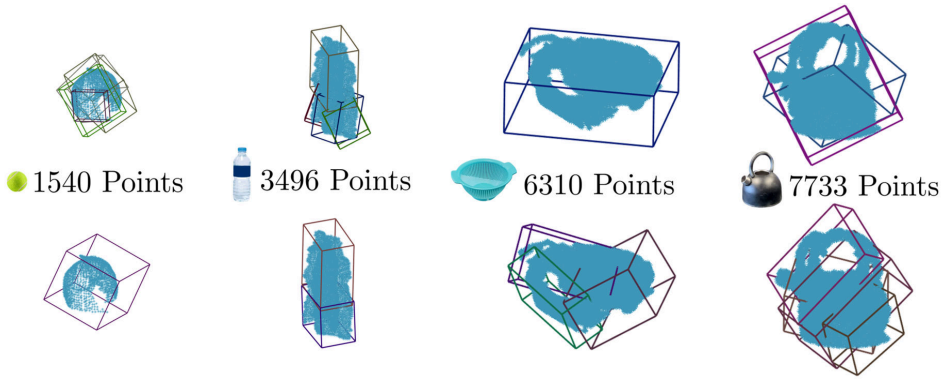
[1]https://github.com/manuelbonilla/pacman_bbox



(a) $\mu_{\mathbf{b}}(X) = X/10$: Using a linear function, small objects are decomposed into a larger number of boxes, whereas large objects might be approximated with a single box.



(b) $\mu_{\mathbf{b}}(X) = a_1 \log(a_2 X + 1)$: Using a log function, small objects are decomposed into a single box, whereas large objects might be approximated by a larger number of boxes.



(c) Examples of box decomposition for different objects. Top [19], bottom our approach.

Fig. 5. Effects of the two different methods to define the minimum number of points of the point cloud contained within a single bounding box.

the algorithm proposed in [30] that was already used in our previous work [19]. A description of the decomposition procedure is reported in the Appendix. The final output of the algorithm is a set $\mathscr{B}$ of bounding boxes. Each element $\mathbf{b} \in \mathscr{B}$ is a pair formally defined as

$$\mathbf{b} \triangleq \langle \mathrm{T}_{\mathbf{b}}, \boldsymbol{\lambda}_{\mathbf{b}} \rangle, \tag{1}$$

where $\boldsymbol{\lambda}_{\mathbf{b}} \triangleq [\lambda_{\mathbf{b},1}, \lambda_{\mathbf{b},2}, \lambda_{\mathbf{b},3}]^\top \in \mathbb{R}^3$ represents the vector of the box dimensions, while

$$\mathrm{T}_{\mathbf{b}} \triangleq \left[\begin{array}{ccc|c} \hat{\mathbf{r}}_{\mathbf{b},1} & \hat{\mathbf{r}}_{\mathbf{b},2} & \hat{\mathbf{r}}_{\mathbf{b},3} & \mathbf{y}_{\mathbf{b}} \\ 0 & 0 & 0 & 1 \end{array}\right], \tag{2}$$

is the transformation matrix expressing the pose of the box in the world frame. Specifically, $\mathbf{y}_{\mathbf{b}}$ is the position in world frame of a reference frame like the one depicted in Fig. 3, while $\hat{\mathbf{r}}_{\mathbf{b},j}, j \in \{1,2,3\}$, are the versors aligned with the box sides.

The box decomposition is a crucial step in the entire process of the algorithm. A poor decomposition, which does not adequately approximate the object, can potentially lead to the generation and then selection of inefficient and non-robust grasps.

The decomposition algorithm has three user-selectable parameters, the minimum volume admissible for a box, the gain threshold used to evaluate if a split has to be enforced, and the minimum number of points $\mu_{\mathbf{b}}$ of the point cloud each box should contain. In [19] the minimum volume was set to a fixed low value, while the minimum number of points of the point cloud was set proportional to the total number of points (constant of proportionality set to 0.1). This latter choice carries with it some problems and limitations when decomposing both small objects (associated to point clouds composed of few points) and large objects (with a higher number of points). Small objects are approximated with numerous boxes of small size. Larger objects are instead approximated by a single bounding box, with dimensions that can be potentially out of the graspable range for the chosen gripper.

To overcome these issues and achieve an approximation of the shape of the object more suited for generating high-quality grasps, we propose in this work a different law for selecting the minimum number of points. Instead of a linear trend, we opted for a logarithmic curve described by the following equation

$$\mu_{\mathbf{b}} \triangleq a_1 \log(a_2 X + 1), \tag{3}$$

where $\mu_{\mathbf{b}}$ is the minimum number of points contained by a bounding box $\mathbf{b}$, X is the total number of points, while a_1 and a_2 are design parameters, regressed from experimental data. Indeed, we collected a set of 120 point clouds, and for each cloud we saved the pair $\langle |\text{cloud}|, \mu_{\mathbf{b}} \rangle$ containing the number of points of the cloud, $|\text{cloud}|$, and the value of $\mu_{\mathbf{b}}$ leading to a good box decomposition. We used these pairs to fit the logarithmic curve (3) and compute the two parameters. The collected data and the fitted curve are shown in Fig. 4.

Using this function, small objects are approximated with a reduced number of boxes (or even a single one) while larger objects are decomposed in such a way that better approximation of their shape is obtained, as shown in Fig.5.

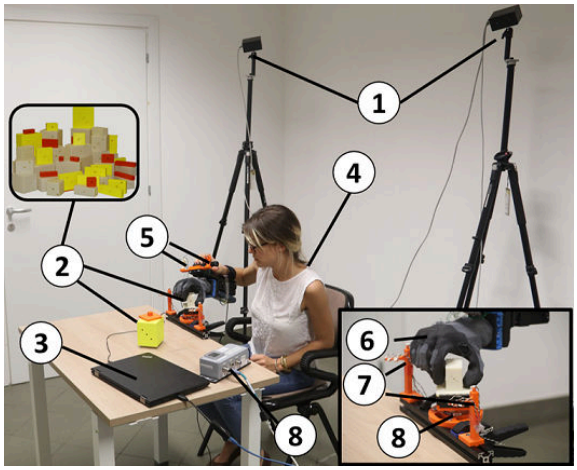


Fig. 6. Experimental setup used to record the set of human demonstrations. The PhaseSpace cameras (1) are used to track the pose of the hand using eight hand-fixed markers (5) and eight box-fixed markers (7). A human operator (4) grasps a set of boxes (2) with the Pisa/IIT SoftHand (6). A force torque-sensor (8) is used to record the interaction wrenches. The recorded data are saved on a PC (3), and later used to train a DTR model. Training on a small set of recorded demonstrations of a skilled operator grasping cuboids, the robot is able to generate a grasp pose for a generic object.

B. Grasp Prediction from Human Data

One of the main feature of the approach is the exploitation of human grasping skills to let the robot learn human-like poses for the specific gripper used. Here, we used the Pisa/IIT SoftHand [27] as gripper, but the considerations reported in the following can be adapted to different gripper choices.

1) Robotic Gripper: First we present the information about the robotic gripper needed by the proposed method. Indeed, we only need a few information of the specific gripper. In particular, we define the object hand as

$$\mathtt{h} \triangleq \langle O_{\mathrm{h}}, \mathscr{O}, \mathscr{C}, \boldsymbol{\delta} \rangle \ . \tag{4}$$

The first element of $\mathtt{h}$ is the reference frame $O_{\mathrm{h}} = (\hat{\mathbf{r}}_{\mathrm{h,p}}, \hat{\mathbf{r}}_{\mathrm{h,t}}, \hat{\mathbf{r}}_{\mathrm{h,f}})$ attached to the hand, which is the one depicted on the right in Fig. 3. Then, we have a representation of both the gripper shape, $\mathscr{O} \subset \mathbb{R}^3$, and a model of the closing region of the gripper, $\mathscr{C} \subset \mathbb{R}^3$. The definition of $\mathscr{C}$ is inspired by [58], where they define the closing region of a gripper as *"the volumetric region swept out by the fingers when they close"*. Finally, the vector $\boldsymbol{\delta} \in \mathbb{R}^5$ is used to encompass a series of gripper-related thresholds on the maximum graspable dimensions, the relative gripper-box alignment, and the collisions thresholds.

These elements will be described and used in Sec.III-C to define the grasp-quality score and for the selection of the best grasp.

2) Learning Human Grasping Skills: First, we collected a small set of grasp demonstrations from a human operator. With the setup depicted in Fig.6, a commercial Phase Space Motion Capture[2] system is used to track the position of 8 markers placed on the robotic gripper (a manually operated Pisa/IIT SoftHand in our case) w.r.t. other 8 world-fixed markers. This system allows for registration of the correct hand pose when grasping a set of 56 cuboid sample boxes, whose dimensions have been chosen to cover the feasible grasp range of the gripper[3].

We also recorded the interaction wrenches exerted during the grasp. Indeed, during the demonstrations, the sample boxes were rigidly fixed and aligned to a sensorized platform. The platform was equipped with a force-torque sensor ATI mini45. The sensor, zeroed and calibrated properly before each trial, is used to record the interaction wrenches (forces and torques) during the grasping phase.

For each attempt, the operator was asked to exert for 3 seconds a force along the three spatial directions and a torque along all the axes of the box. The force/torque measurements for the specific axis on which the operator was asked to act were extracted from the recorded data and then used to define a metric for the total interaction wrench of the grasp as $\mathbf{w} = [f, \tau]^\top \in \mathbb{R}^2$. Specifically, the recorded data have been processed extracting the maximum value for each component of both the force and the torque, and the elements f and τ have been computed as the norm of the vectors composed of these maximum values for the force and torque, respectively.

Following [19], the recorded set of 648 grasping attempts is used to let the robot learn a model $\mathbf{p} = \psi(\boldsymbol{\lambda}_\mathbf{b})$ to predict a human-like pose $\mathbf{p} = [p_1, \ldots, p_6]^\top \in \mathbb{R}^6$ (relative to the box), given the box dimensions $\boldsymbol{\lambda}_\mathbf{b}$. In [19] this was made possible through the use of a Decision Tree Regressor (DTR).

Applying this learned model $\psi(\boldsymbol{\lambda}_\mathbf{b})$ to the boxes generated by the minimum volume bounding box decomposition algorithm, we are able to generate a set of human-like candidate grasp poses for the gripper. Indeed, using a small set of human demonstrations, we are potentially able to grasp any object, regardless of shape or size, without any prior information or knowledge about it apart from an acquired point cloud.

a) Inclusion of interaction wrenches: In [19], we only use the pose of the hand and box dimensions from the demonstrated data to train the DTR and infer a grasp pose given a vector of box dimensions. However, as pointed out, these were not the only data recorded during the demonstrations that could be exploited for planning purposes. The recorded demonstrations also provide information on the interaction forces generated during the grasp, that we decided to actively include into the planning pipeline. The choice to include theses interaction forces in the grasp planning algorithm is dictated by the considerations made in [19], where it is pointed out as the grasping success rate was quite low for heavy objects. Our insight is that the inclusion of interaction forces can help to select more robust grasps.

In order to include this information into the grasp planning algorithm, we propose to let the robot learn an

[2]http://phasespace.com/

[3]https://qbrobotics.com/wp-content/uploads/2021/07/qb-SoftHand-Research-datasheet-r200.pdf

augmented model $\tilde{\psi}(\boldsymbol{\lambda}_\mathbf{b})$ that is able to infer from the vector of box dimensions $\boldsymbol{\lambda}_\mathbf{b}$ not only the hand pose $\mathbf{p}$, but also a model $\nu(\boldsymbol{\lambda}_\mathbf{b})$ for the interaction wrenches metric $\mathbf{w}$. Therefore, the complete model learned by the robot is

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{w} \end{bmatrix} = \tilde{\psi}(\boldsymbol{\lambda}_\mathbf{b}) = \begin{bmatrix} \psi(\boldsymbol{\lambda}_\mathbf{b}) \\ \nu(\boldsymbol{\lambda}_\mathbf{b}) \end{bmatrix}. \tag{5}$$

b) Global Grasp Generation Policy: In [19], a set of 48 candidates poses is generated for a specific box, selected according to a hand-designed heuristic. The best feasible grasp in this set is then selected and executed, without considering the grasps generated from other boxes in $\mathscr{B}$. The original method is inherently locally optimal because of the greediness introduced by evaluating grasps for only one box, and it could result in suboptimal choices.

We now take a step toward building a global method by removing the box selection step. The prediction of the grasp poses is thus made over all the boxes generated by the decomposition algorithm. Hence, we increase the set of candidate grasps from which to select the best one (see Fig.2). We can also guarantee to select the optimal grasp, according to the specific metric and the feasibility constraints, among all the possible grasps originated from the given box decomposition.

3) Generation of the candidate set: We present now how the set of candidate grasps $\mathscr{G}$ is generated. As explained in the previous section, given a decomposition of a point cloud into a set of cuboid bounding boxes $\mathscr{B}$, we are able to predict a set of grasp poses $\mathbf{p}$ and interaction wrenches $\mathbf{w}$. Each of these pairs $\langle \mathbf{p}, \mathbf{w} \rangle$ is associated with the specific box $\mathbf{b} \in \mathscr{B}$ from which it was generated. We therefore define a *grasp* $\mathtt{g}$ as a tuple composed by the specific gripper $\mathtt{h}$ used for the demonstrations, a predicted grasp pose $\mathbf{p}$, the box $\mathbf{b}$ used for prediction, and eventually the predicted interaction wrench $\mathbf{w}$, i.e., as

$$\mathtt{g} \triangleq \langle \mathtt{h},\, \mathbf{p},\, \mathbf{b}, \mathbf{w} \rangle. \tag{6}$$

All the grasps predicted for a given box $\mathbf{b} \in \mathscr{B}$ are collected into a set $\mathscr{G}_\mathbf{b}$. Finally, the complete set of candidate grasps $\mathscr{G}$ is simply built as the union of the grasps set for each box in $\mathscr{B}$

$$\mathscr{G} = \bigcup_{\mathbf{b} \in \mathscr{B}} \mathscr{G}_\mathbf{b}. \tag{7}$$

C. Grasp Evaluation

As typical for data-driven approaches, the grasp to execute among the candidate set is chosen so to maximize a properly defined metric [22], [59]. In this work, we propose a metric embedded into a global *quality score* ($\mathbb{S}_\mathtt{g}$). This score maps each grasp to a real number between zero and one, i.e., $\mathbb{S}_\mathtt{g} : \mathscr{G} \to [0, 1]$, and has been designed so to take into account:

i) the box from which the grasp has been generated;
ii) the predicted interaction wrench;
iii) the relative alignment between the box and the gripper;
iv) possible collisions with the environment.

In the following, we describe in detail the heuristics used to embed the effects from (i) to (iv) into proper score functions, used to construct the global score $\mathbb{S}_\mathtt{g}$.

1) Box score: The first index is related to the specific box $\mathbf{b}$ used for predicting the grasp, and in particular to its density of points belonging to the point cloud, $\rho_\mathbf{b}$, and its distance from the point cloud centroid, $d_\mathbf{b}$.

This score is used to favor grasps originated from outer boxes, that can approximate handle-like parts and are associated with a higher number of collision-free grasps [19], but with a high density of points, i.e., that provide a good local approximation of the object shape. The box score can then be computed as

$$\mathbb{J}_\mathbf{b} \triangleq \frac{1}{2}\left(\frac{\rho_\mathbf{b}}{\max\limits_{\mathbf{b}_k \in \mathscr{B}} \rho_{\mathbf{b}_k}}\right)^2 + \frac{1}{2}\left(\frac{d_\mathbf{b}}{\max\limits_{\mathbf{b}_k \in \mathscr{B}} d_{\mathbf{b}_k}}\right)^2, \tag{8}$$

where the density and centroid distance have been normalized w.r.t. the maximum density and maximum distance of all the boxes in $\mathscr{B}$.

Note that the score $\mathbb{J}_\mathbf{b}$ is equal for all the grasps $\mathtt{g} \in \mathscr{G}_\mathbf{b}$.

2) Wrench score: The second score, $\mathbb{J}_\mathbf{w}$, is used to embed the learned wrench into the grasp-selection procedure. It is used to favor grasps associated with high interaction wrenches $\mathbf{w}$. Indeed, our hindsight is that $\mathbf{w}$ can be used as an indirect measure of the robustness and stability of the grasp, i.e., grasps associated with high interaction wrenches $\mathbf{w}$ have a higher probability of being more robust.

The score has been designed as

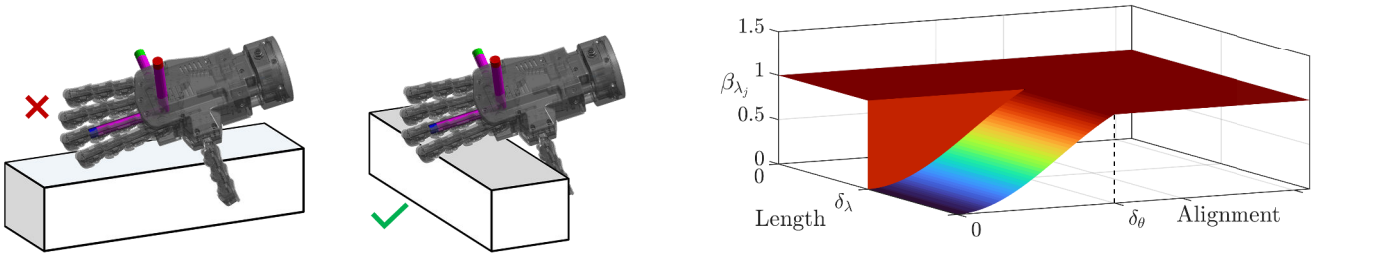
$$\mathbb{J}_\mathbf{w} \triangleq \frac{1}{2}\left(\frac{f_\mathtt{g}}{\max\limits_{\mathtt{g}_k \in \mathscr{G}} f_{\mathtt{g}_k}}\right)^2 + \frac{1}{2}\left(\frac{\tau_\mathtt{g}}{\max\limits_{\mathtt{g}_k \in \mathscr{G}} \tau_{\mathtt{g}_k}}\right)^2, \tag{9}$$

where the force and torque of the selected grasp have been normalized w.r.t. the maximum force and maximum torque considering all the grasps in $\mathscr{G}$.

3) Alignment score: This index considers the relative alignment between the gripper and the object to grasp, and depends on both the specific gripper and the object (or its local approximation used to sample the grasp). This score is used to penalize grasps that are highly aligned with sides that exceed the physical limits of the robotic hand. Grasps in which the fingers are highly aligned with long sides (see Fig.7(a)) are indeed discarded in favor of grasps less aligned (see Fig.7(b)).

In our previous work, we used the metric proposed in [56] to select the to-be-executed grasp, searching the pose with the thumb of the SoftHand more aligned with the longest side of a candidate box. This actually leads to a greedy iterative procedure that starting from the longest box side searches for the grasp with the thumb most aligned without considering the alignment with the other sides.

In this work, we exploit similar considerations for the definition of the specific score $\mathbb{J}_\mathrm{a}$. First, we remove the iterative procedure to select the grasp with the thumb more aligned with the longest side, introducing a term $\alpha \in$



(a) Fingers highly aligned to the longest side of the box. (b) Fingers loosely aligned to the longest side of the box. (c) Tridimensional representation of β_{λ_j} as function of the alignment and of the length.

Fig. 7. Alignment of the fingers w.r.t. the box dimensions.

$[0, 1]$ that combines both information on the dimensions of the box and the alignment of the thumb w.r.t. all the dimensions. More formally, we define α as

$$\alpha \triangleq \frac{1}{\bar{\lambda}} \max_{j \in \{1,2,3\}} \lambda_{\mathbf{b},j} |\hat{\mathbf{r}}_{\mathbf{b},j} \cdot \hat{\mathbf{r}}_{\mathrm{h,t}}| \,, \tag{10}$$

where $\bar{\lambda} = \max_{\mathbf{b} \in \mathscr{B}} \max_{j \in \{1,2,3\}} \lambda_{\mathbf{b},j}$ is the longest dimension among all the boxes in $\mathscr{B}$, $\hat{\mathbf{r}}_{\mathrm{h,t}}$ is the versor of the hand frame O_{h} aligned with the thumb (see Fig.3), and $\cdot$ is the scalar product operator.

Second, we include into $\mathbb{J}_{\mathrm{a}}$ information on the constraints provided by the maximum opening of the hand. Indeed, the dimensions of the hand provide a natural limit on the maximum dimensions graspable.

Being $\hat{\mathbf{r}}_{\mathrm{h,f}}$ the unitary versor of the hand frame directed along the fingers (as in Fig.3), the relative angle between the gripper fingers and a box side versor $\hat{\mathbf{r}}_{\mathbf{b},j}$ can be simply computed as $\theta_j \triangleq \mathrm{acos}(|\hat{\mathbf{r}}_{\mathbf{b},j} \cdot \hat{\mathbf{r}}_{\mathrm{h,f}}|)$, being $j = 1 \dots 3$. Note that, given the definition of θ_j we have that $\theta_j \in [0, \pi/2]$. We introduce the following function to evaluate the alignment of the hand w.r.t. the sides of the box

$$\beta_{\lambda_j} \triangleq \begin{cases} 1 - \mathrm{sinc}\left(\frac{\theta_j}{\delta_\theta}\right), & (\theta_j \leq \delta_\theta) \wedge (\lambda_{\mathbf{b},j} > \delta_\lambda) \\ 1 & \text{otherwise,} \end{cases} \tag{11}$$

where $\delta_\theta \in \boldsymbol{\delta}(\mathrm{h})$ is a user-defined threshold on the maximum allowed relative alignment and $\delta_\lambda \in \boldsymbol{\delta}(\mathrm{h})$ is the maximum length graspable by the gripper (25 deg and 100mm, respectively, for the SoftHand), and sinc is the unnormalized sinc function defined as

$$\mathrm{sinc}(x) \triangleq \begin{cases} \dfrac{\sin(x)}{x}, & x \neq 0 \\ 1. & x = 0 \end{cases} \tag{12}$$

A tridimensional representation of β_{λ_j} as function of the relative alignment θ_j and of the length $\lambda_{\mathbf{b},j}$ is reported in Fig.7(c).

Combining (10) and (11), we eventually define $\mathbb{J}_{\mathrm{a}}$ as follows

$$\mathbb{J}_{\mathrm{a}} \triangleq \alpha \prod_{j=1}^{3} \beta_{\lambda_j} \,. \tag{13}$$

It can be noted that, if the alignment is above the desired threshold and/or the length is within the gripper physical limits, $\mathbb{J}_{\mathrm{a}}$ will be equal to α, i.e., ranking the grasps depending on the thumb alignment and the normalized length of the box side.

4) Collision score: An important aspect to consider when selecting a grasp is the potential collision of the gripper with the object and the environment (such as a table) in the selected pose. In addition, it is important to consider possible collisions that may happen during the closure of the gripper fingers. We propose to include into $\mathbb{S}_{\mathbf{g}}$ information on these collisions, introducing a collision score to penalize and discard grasps that are not collision-free.

The definition of this score exploits two auxiliary indexes designed to take into account the two class of collisions presented before, i.e., collisions of the whole gripper and collisions of the fingers during the closure.

To model the former, we introduce $\kappa_{\mathscr{O}}$, a function that exploits the prior knowledge on the geometric representation of the shape of the gripper h, i.e., $\mathscr{O}(\mathrm{h})$ to detect collisions of the gripper at the grasping pose.

Given $\mathscr{O}(\mathrm{h})$, it is possible to define the density $\rho_{\mathscr{O}}$ of the occupancy volume as the ratio between the number of points of the object (and the environment) inside the said volume and the total volume of $\mathscr{O}$. From this, $\kappa_{\mathscr{O}}$ is defined as

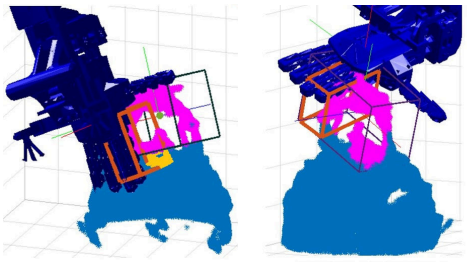
$$\kappa_{\mathscr{O}} \triangleq \begin{cases} 0, & \text{if } \rho_{\mathscr{O}} \geq \delta_{\mathscr{O}} \\ 1, & \text{otherwise,} \end{cases} \tag{14}$$

where $\delta_{\mathscr{O}} \in \boldsymbol{\delta}(\mathrm{h})$ is a user-defined threshold that determines when the gripper is considered to collide with the environment. This threshold is used to account for the resolution and presence of noise in the acquired point cloud. The effect of $\kappa_{\mathscr{O}}$ is, hence, to filter out the grasp poses for which the gripper would be in collision with the object or the table on which the object lies.

For the sake of simplicity, we assume that the occupancy volume $\mathscr{O}$ of the gripper can be approximated by a union of cuboids, to speed up the computation of $\rho_{\mathscr{O}}$. More complex and detailed choices can be made, e.g., employing ellipsoids or superquadrics [60], but this increases the time needed to check the collisions.

To include the fingers' range of movements and the possible collisions during closure, we introduce $\kappa_{\mathscr{C}}$, a function that exploits the model of the closing region of the gripper h, $\mathscr{C}(\mathrm{h})$.

Figure 8 shows examples of two different grasp poses generated for a boiler. The blue points are the point cloud acquired by the camera, the pink points are the points



(a) (b)

Fig. 8. Two examples of grasp poses generated for a specific box output of the box decomposition algorithm using a boiler point cloud as input. The orange box originated from the fingers of the hand is a simple model of the closing region $\mathscr{C}(\mathrm{h})$.

of the cloud that are inside the box used to generate the two grasps. The yellow points represent the points of the cloud that do not belong to the aforementioned box but are inside the closing region $\mathscr{C}(\mathrm{h})$ of the gripper.

In the example, the gripper used was the Pisa/IIT SoftHand. Due to its compliance and its synergy-based closure mechanism, we modelled the closing region as a single cuboid originated from the index and middle fingers. Different, and more fine, approximations can be made to describe $\mathscr{C}(\mathrm{h})$, depending on the specific gripper used [52], [58], [60].

It is possible to see how, for the grasp on the left, the closing region contains points of the object (yellow points) that do not belong to the box used to generate the grasp. Thus, while closing, the fingers would collide with the object. This could reduce the likelihood of a successful grasp, due to, e.g., a displacement of the impacted object or the impeded closure of the hand around the target box. For the grasp on the right, the closing region contains only points part of the target box, hence providing a collision-free closure.

We designed $\kappa_{\mathscr{C}}$ to include a heuristics into the grasp planning algorithm favoring grasps as in Fig.8(b) over grasps as in 8(a).

Given a grasp $\mathbf{g}$ for a specific box $\mathbf{b} \in \mathscr{B}$, we define the density $\rho_{\mathscr{C}}$ as the ratio between the number of points of the object that are not contained into the target box $\mathbf{b}$ but are inside the closing region $\mathscr{C}(\mathrm{h})$ and the total volume of the closing region $\mathscr{C}(\mathrm{h})$. From this, $\kappa_{\mathscr{C}}$ is defined as

$$\kappa_{\mathscr{C}} \triangleq \begin{cases} 0, & \text{if } \rho_{\mathscr{C}} > \bar{\delta}_{\mathscr{C}} \\ \zeta(\delta_{\mathscr{C}}), & \text{if } \underline{\delta}_{\mathscr{C}} \leq \rho_{\mathscr{C}} \leq \bar{\delta}_{\mathscr{C}} \\ 1, & \text{otherwise,} \end{cases} \tag{15}$$

where $\bar{\delta}_{\mathscr{C}} \in \boldsymbol{\delta}(\mathrm{h})$ and $\underline{\delta}_{\mathscr{C}} \in \boldsymbol{\delta}(\mathrm{h})$ are two user-defined thresholds, and where $\zeta(\delta_{\mathscr{C}})$ is a smooth and monotonic function such that $\zeta(\bar{\delta}_{\mathscr{C}}) = 0$ and $\zeta(\underline{\delta}_{\mathscr{C}}) = 1$, e.g., a cubic spline. The collision score $\mathbb{J}_{\mathrm{c}}$ is then modeled as the combination of the two contributions, $\kappa_{\mathscr{O}}$ and $\kappa_{\mathscr{C}}$, related to $\mathscr{O}$ and $\mathscr{C}$. More formally, $\mathbb{J}_{\mathrm{c}}$ is defined as

$$\mathbb{J}_{\mathrm{c}} \triangleq \kappa_{\mathscr{O}} \kappa_{\mathscr{C}}. \tag{16}$$

5) Global grasp quality score: Having defined four cost indexes to model the effects from i) to iv), we can eventually define the score $\mathbb{S}_{\mathrm{g}}$ as the product of these four elements

$$\mathbb{S}_{\mathrm{g}}(\mathrm{g}) \triangleq \mathbb{J}_{\mathbf{b}}\ \mathbb{J}_{\mathbf{w}}\ \mathbb{J}_{\mathrm{a}}\ \mathbb{J}_{\mathrm{c}}. \tag{17}$$

The designed metric is able to effectively describe the complex interdependencies between the predicted grasp, the box it is related to, and the gripper used for the grasp, by encompassing all these factors into a global score.

It has to be noted as, from the definition, $\mathbb{S}_{\mathrm{g}}(\mathrm{g}) = 0$ only in case of collisions, i.e., $\mathbb{J}_{\mathrm{c}} = 0$, and/or in case of grasps aligned with non-graspable sides, i.e., $\mathbb{J}_{\mathrm{a}} = 0$.

D. Grasp Selection Policy

Using (17), the grasp to be executed is selected as the solution of the constrained optimization problem

$$\begin{aligned} \mathrm{g}^* = & \arg\max_{\mathrm{g} \in \mathscr{G}} \mathbb{S}_{\mathrm{g}}(\mathrm{g}) \\ & \text{s.t.} \\ & \mathrm{g} \in \mathscr{F}_{\mathrm{r}} \\ & \mathbb{S}_{\mathrm{g}}(\mathrm{g}) > 0, \end{aligned} \tag{18}$$

where

$$\mathscr{F}_{\mathrm{r}} = \{\mathrm{g} \,|\, \mathbf{p}(\mathrm{g}) \in \mathscr{W}_{\mathrm{r}}\}$$

is used to denote the set of grasps g for which the grasp pose $\mathbf{p}(\mathrm{g})$ belongs to the reachable workspace $\mathscr{W}_{\mathrm{r}}$ of the specific robot r used to reach the pose. The problem is solved using a two-step approach. First, find the grasp in the set $\mathscr{G}$ that maximizes $\mathbb{S}_{\mathrm{g}}$. Then, the resulting pose is passed as input to a dedicated and robot-specific inverse kinematic block that will check for the feasibility. If the selected grasp is not feasible, it is removed from $\mathscr{G}$ and the algorithm select the next best grasp. It is worth noting that the last constraint, $\mathbb{S}_{\mathrm{g}}(\mathrm{g}) > 0$, implies that all grasps that are in collisions or are not feasible for the gripper due to the size of the box-side it is aligned with, are automatically discarded.

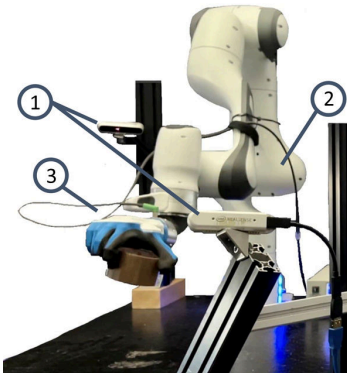
IV. Experimental Validation: Soft Hand

In this section, we describe the setup used for the experimental validation, and we present and comment the results of the experiments. First we report and discuss the performance of GLP 2.0 compared to our previous work [19]. Then, we compare the performance with two different state-of-the-art algorithms for planning grasps for soft hands.

A. Experimental Setup

The setup we used for the experiments is shown in Figure 9(a). It is composed by two RGB-D Intel® Realsense™D415 Camera (1), used to acquire the point cloud of the target object. The grasp pose is executed using a Franka EMIKA Panda (2) equipped with a Pisa/IIT SoftHand (3), the gripper used to collect the training data for the DTR (see Sec.III-B2 and [19]).

The manipulator and the end-effector are controlled using ROS. The point cloud acquisition is made using the



(a) Experimental Setup.

(b) Objects from [19].

(c) Additional objects tested.

Fig. 9. Experimental setup and complete set used for the experimental validation (a). 30 objects, of which 21 are the ones used in [19] (b) and 9 additional objects (c), from left to right: Big Colander, Small Colander, Controller, Brush, Pot, Boiler, Bottle, Foam Brick, Tennis Ball.

TABLE I
STATISTICS ON THE EXPERIMENTAL VALIDATION WITH THE PISA/IIT SOFTHAND.

	Rate [%]	**Time** [s]	**Boxes**	**Points**	$\bar{T}_{PC}$ [s]	$\bar{T}_{BB}$ [s]	$\bar{T}_{BS}$ [s]	$\bar{T}_{GP}$ [s]	$\bar{T}_{GS}$ [s]
GLP	75.3	3.15 ± 1.39	2 ± 1	4160 ± 2450	0.40 ± 0.23	1.29 ± 1.00	0.08 ± 0.02	1.14 ± 0.05	0.48 ± 0.24
		(1.66, 7.54)	(1, 6)	(651, 10690)	(0.08, 1.08)	(0.37, 4.61)	(0.04, 0.16)	(1.11, 1.36)	(0.02, 3.47)
GLP 2.0	**94.0**	5.42 ± 3.41	2 ± 1	4347 ± 2537	0.40 ± 0.23	1.39 ± 1.22	–	1.24 ± 0.12	2.71 ± 2.36
		(1.88, 15.17)	(1, 5)	(540, 10346)	(0.08, 1.09)	(0.38, 5.57)		(1.13, 2.01)	(0.22, 9.14)

ROS interface provided by Intel. The box decomposition is implemented in C++, while the grasp prediction using the DTR is implemented with a Python script. We used SciKit-Learn to train the DTR with a maximum tree depth of 8 (as in [19]). It has to be remarked that the DTR is trained only on the set of grasp demonstrations provided by a human operator for the set of cuboid boxes shown in Fig.6, and thus it does not require to be trained on real objects. All the other steps have been implemented in MATLAB, including the inverse kinematics step through the reverse priority inversion algorithm proposed in [61] and [62].

B. Object Dataset and Experimental Protocol

We evaluate the performance of the algorithm on a dataset composed of 30 objects, for which the robot does not have any model. We use the 21 objects used for the experimental evaluation in [19] (chosen as they have characteristics similar to the ones proposed in [63] as benchmark set), and we include 9 novel objects. The complete dataset is shown in Fig.9(b)-(c): on top the original set of testing objects, on the bottom the new objects. The new set presents three objects that are similar to the ones already used in [19] (the ball, the pot, and the foam) with slightly differences in terms of size and characteristics, but also includes a set of large and/or heavy objects such as the colander and the boiler which were not in [19].

In each test, the object is the only element in the scene, and it is randomly placed on a table in the reachable workspace of the robot. The robot always starts from the same position above the object. The approaching trajectory is composed by a 5-order polynomial for the translational part, while spherical linear interpolation (SLERP) is used to connect the initial and final orientation. For each object, we tested 5 grasps, for a total of 150 grasps. After the object is grasped, the robot lifts the object 150mm to evaluate the robustness of the closure. A grasp is considered successful if the robot is able to complete the task (grasping and lifting) without losing the object or without stopping. If the algorithm does not return any feasible grasp, the task is marked as a failure.

C. Results

In this section, we comment the results of the experimental validation. We first evaluate and compare the performance against the original method, which acts as baseline, in terms of overall grasping success rate and time needed to select a candidate grasp. We also compare the number of boxes obtained using the original and the modified box decomposition algorithm presented in Sec.III-A. A summary of these results is reported in Tab.I, our approach is denoted as GLP 2.0, where the overall success rate is reported, together with statistics (in the form mean±standard deviation (min, max)) on the execution time, the number of boxes, the number of points of the acquired point cloud, and the time for each step of the two algorithms. The values for the boxes have been rounded to the closest integer.

1) Grasping Performance: The first evaluation is about the grasping success rate of GLP 2.0 for the 30 objects. The method achieves an overall grasping success rate over the 150 grasps of 94.0% compared to the 75.3% obtained

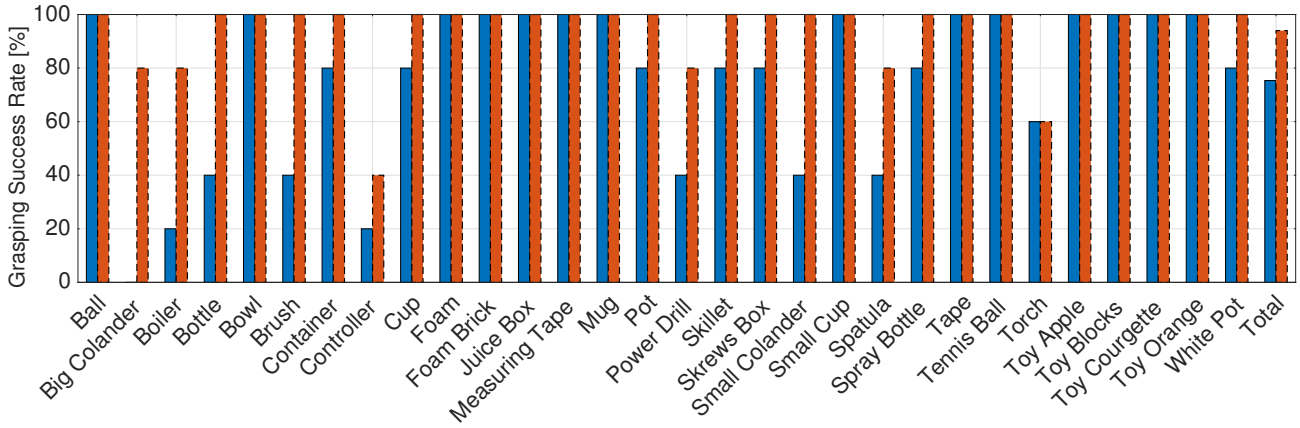


Fig. 10. Grasping Success Rate [%] with the Pisa/IIT SoftHand. Red dashed bars the results using GLP 2.0, the approach proposed in this paper, blue solid bars using [19]. GLP 2.0 shows a 25% improvement of the overall grasping success rate over the set of 30 objects (94% compared to 75.3%).

by the baseline algorithm. Figure 10 reports the success rate for each object in the dataset.

For every object our approach outperforms, or at least evens out, the baseline. Clear improvements can be seen when grasping large and/or heavy boxes, such as the Power Drill, the Big Colander, the Boiler, and the Brush. In particular, the baseline is not able to generate a successful grasp for the Big Colander, probably due to a poor box decomposition (large objects are often decomposed into a single box using [19]). Even if we remove this object from the statistics, the overall success rate using [19] is lower than the one we achieve (77.9% versus 94.5%).

In Fig.11 we show some frames of the robot while grasping eight objects from the complete dataset: the Big Colander, the Boiler, the Bottle, the Brush, the Controller, the Pot, the Small Colander, and the Spatula.

It can be noted that the robot tries to grasp handle-like parts for most of these objects, e.g., the Boiler, the Brush, and the Pot. In addition, lateral grasps are selected for slender objects such as the Bottle and the Brush, while top-down grasps seems more likely for objects with a different aspect ratio, i.e., larger than taller.

2) Box Decomposition: Following the considerations reported in Sec.III-A on the modified box decomposition algorithm, we compare the differences between GLP 2.0 and the work in [19] in terms of the number of boxes used to approximate the object. In Fig.12 we report the number of boxes N as a function of the number of points of the acquired point cloud. It is worth noting as the different method for determining the minimum number of points within each box leads to objects with few points generally approximated with one or two boxes at most, while the original approach can generate up to six boxes. Objects with many points are instead decomposed more finely.

3) Execution Time: We then compare the timing performance of the proposed approach and of the original algorithm. Both algorithms have been executed on a Laptop PC equipped with an Intel Core i7 Processor (6×2.20 GHz) and 16 GB DDR4 RAM.

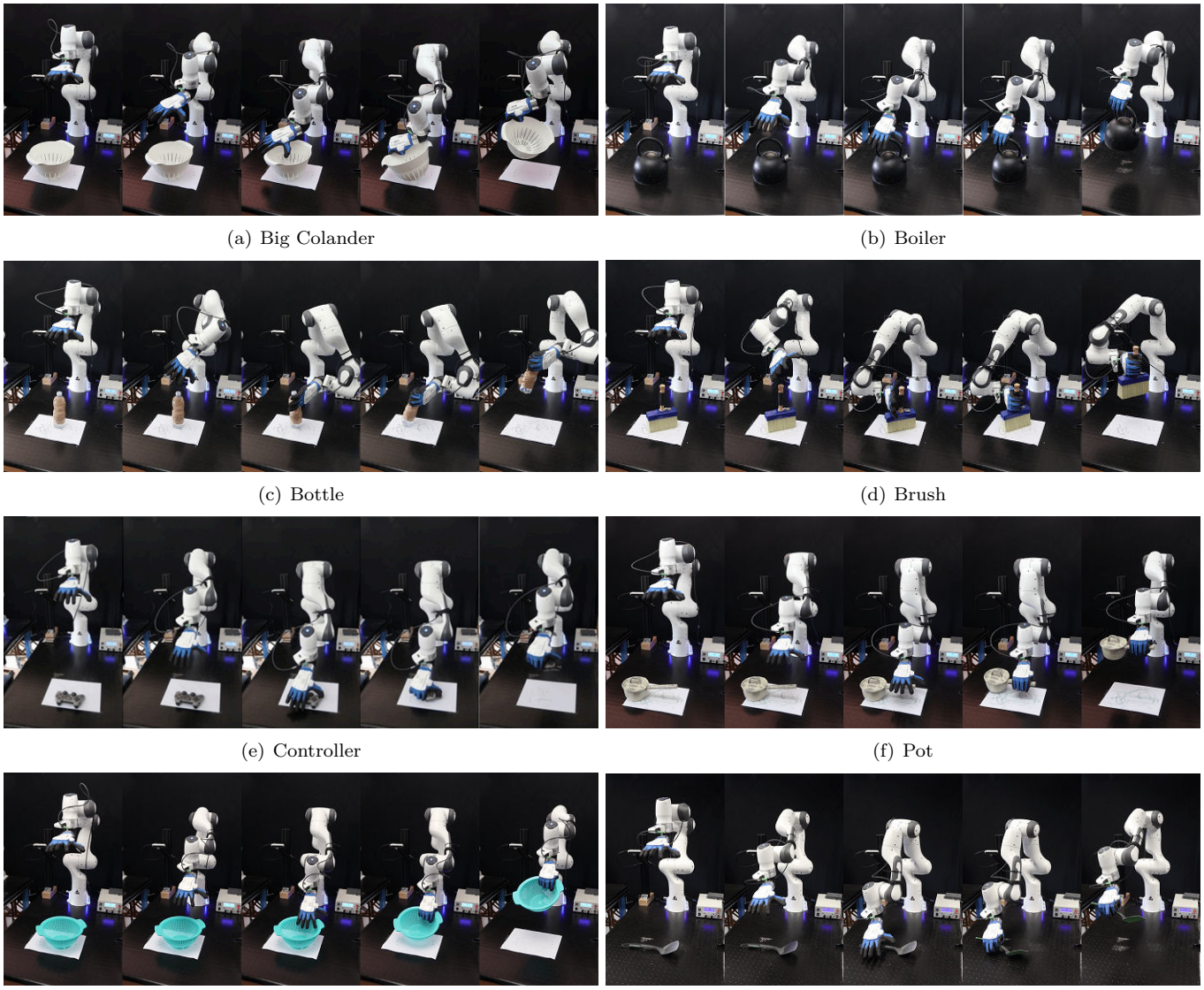
As shown in Fig.13(a), our approach proved to be about 1.7 times slower on average, with a worst-case execution time that is around twice the one of the baseline. In particular, it can be noted from Fig.13(c) that while the previous approach is weakly affected by the number of boxes (almost constant with a linear regression coefficient of -0.03), GLP 2.0 takes longer as the number of boxes increases (linear regression coefficient of 2.34). It is worth to recall that the cardinality of the candidate set $\mathscr{G}$ grows linearly with the number of boxes N ($\#\mathscr{G} = 48N$).

Analyzing the average contribution of each step of the two algorithms it can be noted from Fig.13(b) as the main difference between the two methods relies on the time spent during the analysis of the candidate grasps and the grasp selection. Indeed, the algorithm proposed in this paper evaluates a larger set of grasps with a more complex metric, so we can expect an increased computational effort. Indeed, if we normalize the time needed by the two algorithms for the cardinality of the respective grasp sets (accounting also for the cases in which [19] is required to select a different candidate box due to the lack of feasible grasps) we obtain that our approach takes on average 20ms per grasp compared to 4ms per grasp using [19]. Nonetheless, the current MATLAB implementation has not been optimized for performance, since optimizing the throughput of the entire system was out of the scope of this work. We are confident that an optimized implementation will be able to reduce the time needed to select the candidate grasp.

D. Comparison with other approaches

While the previous section presented the results of the proposed method compared to [19], this section presents the result of the comparison with other two grasp planning methods for soft hands.

First, we compare the performance with the method presented in [28], called **CS-GQ-CNN** in the following, that combines the grasp quality convolutional neural



(a) Big Colander (b) Boiler

(c) Bottle (d) Brush

(e) Controller (f) Pot

(g) Small Colander (h) Spatula

Fig. 11. Selections of frames of the robot grasping the Big Colander, the Bottle, the Brush, the Controller, the Pot, the Small Colander, and the Spatula.

network (GQ-CNN) module presented in [24] with the *closure signature* (CS) concept presented in [57], to plan grasps with the Pisa/IIT SoftHand. The GQ-CNN module estimates the optimal grasp (center and direction) to be performed with a parallel-jaw gripper given an observed depth image of the object to grasp. The CS module, instead, is used to plan the hand-object alignment given the estimated grasp center and direction. The CS provides a simplified way to plan grasps with soft hands, as it characterizes a specific closing motion that the hand can achieve through a direction of maximum closure applied at a certain point. As highlighted in [28], the center and direction defined by the CS can be compared to the grasping center and direction of a parallel-jaw gripper, thus it allows fast adaptation of models trained for parallel-jaw grippers to different hands.

Then, using a similar approach, we exploit the CS concept to adapt the **Grasp Pose Detection** (**GPD**) method presented in [15] to the Pisa/IIT SoftHand gripper. GPD takes a point cloud as input and produces 6D pose estimates of viable grasps as output. The GPD process involves two primary stages: generating a vast selection of potential grasps through sampling, and subsequently evaluating with a four-layer convolutional neural network (CNN) which of these candidates are good grasps.

Note that while GPD, similar to our methods, takes as input the object point cloud and outputs 6D grasp poses, the CS-GQ-CNN plans a grasp pose from a depth image obtained from an overhead camera and has been developed for top-down grasps. Since our setup uses two cameras, we decided to run the CS-GQ-CNN method on both depth maps, selecting the grasp with the highest Q-value among the two sets generated.

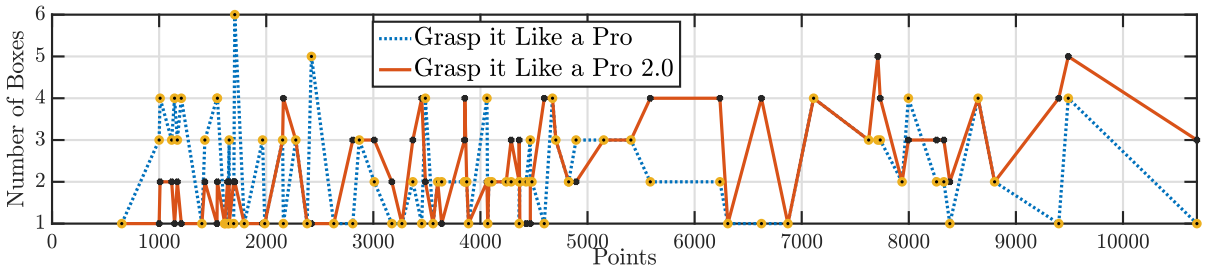
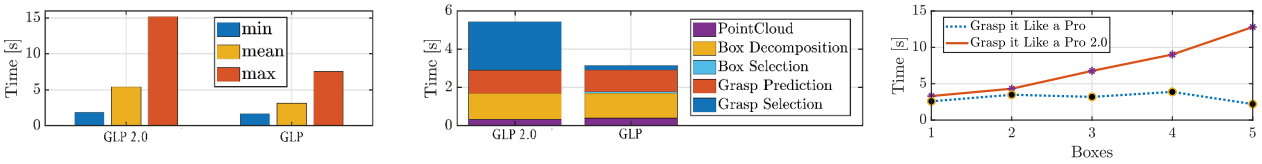


Fig. 12. Number of boxes as a function of the number of points of the point cloud. Solid our approach, dashed [19].



(a) Statistics on the Execution Time.

(b) Average contribution of each step of the algorithm.

(c) Mean execution time as a function of the number of boxes. Solid our approach, dashed [19].

Fig. 13. Execution Time Performance.

The comparison has been carried out on a subset of 16 objects taken from the ones listed in Tab. II. This subset contains objects covering various sizes, shapes, and weights. Indeed, it has large and heavy objects, the Boiler and the Power Drill, small objects, such as the Small Cup and the Toy Block, and even soft objects, like the Foam Brick. In addition, it also includes objects that are hard to grasp with our proposed method, e.g., the Controller. The experimental protocol is the same as the one described in the previous section, and for each object we perform 5 grasp attempts with the three methods.

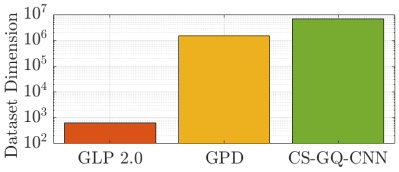
Figure 14 reports the results of the comparison. First, we report the dimensions of the dataset used to train the different machine learning models used by the three methods (Fig. 14(a)). It can be noted that GPD has been trained over 300,000 grasp poses randomly sampled over the created dataset of 1.5 million grasps (50,000 labeled grasps for each of the 55 objects of the BigBird dataset), while the GQ-CNN model presented in [24] has been trained over a dataset of 6.7 million synthetic point clouds, parallel-jaw grasps, and analytic grasp methods. In contrast, our method builds upon a small dataset of less than 1000 demonstrations of a human grasping sample cuboids. In fact, we use machine learning to generate the set of candidate grasp poses while the grasp evaluation is performed through a white-box method that exploits the grasp quality metric presented in Section III-C and does not require training a neural network.

Then, we show that the average time needed to obtain the grasp pose to be executed for each method (Fig. 14(b)) and the success rate for each object and the overall success rate over the testing set (Fig. 14(c)). The three algorithms have similar performance in terms of average time needed to generate and select the grasp to be executed. The results show that our method clearly outperforms both the baselines for every object in the testing set, realizing an overall 91.25% success rate while the two baselines remain around a 50% of successful grasps.

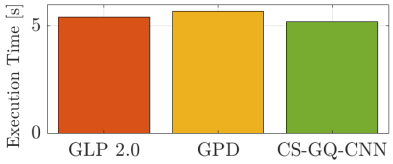
CS-GQ-CNN is the method achieving the worst performance. This result can be explained by the fact that it does not generate 6D grasps, but, exploiting the Dex-Net architecture, it predicts and evaluates grasps parameterized with the planar position, angle, and depth of a gripper relative to the depth image. In addition, the method has been designed for overhead cameras and top-grasps and struggles to adapt to more general settings and camera positions. On the other hand, GPD predicts 6D grasps and performs slightly better than the other baseline, but it has to be pointed out that the grasp quality network has been trained for rigid parallel grippers. Our approach, instead, is able to take into account the gripper structure and characteristics when generating and evaluating the grasps, as shown with the experimental validation on rigid gripper described in the following.

V. Experimental Validation: Rigid Gripper

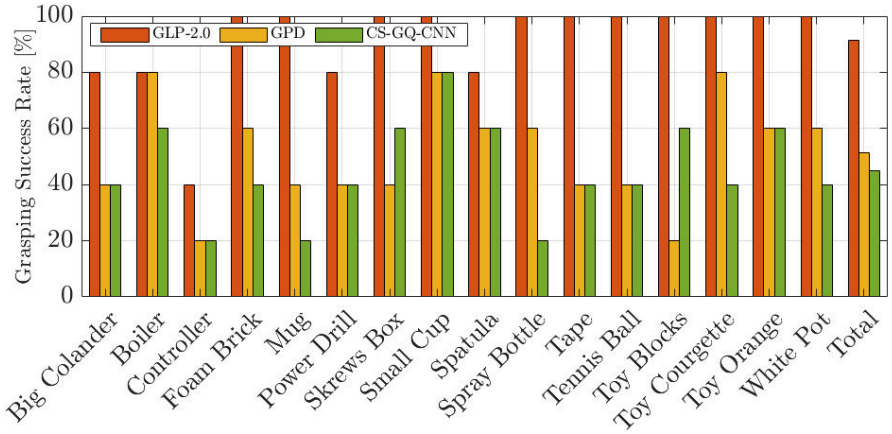
Some considerations about the general applicability of the method to different grippers are necessary. Indeed, the experimental validation has been carried out with the SoftHand, an underactuated and compliant gripper. The compliant nature of the SoftHand, and its capability to adapt during the closure, can affect, to some extent, the performance. Thus, an experimental validation using a rigid 2-Finger gripper (the Franka Hand) has been performed. This validation aims to assess if GLP 2.0 can be adapted to different grippers, and to evaluate the influence of compliance on the performance of the method. In the following, the procedure used to collect the data needed by the DTR for the grasp prediction and the considerations



(a) Dimensions of the collected training datasets. Data are displayed using a logarithmic scale.



(b) Execution times.



(c) Success Rate

Fig. 14. Results of the comparison with GPD and CS-GQ-CNN using the Pisa/IIT SoftHand. GLP 2.0 outperforms both baselines on the tested objects, relying on a smaller set of demonstrations.

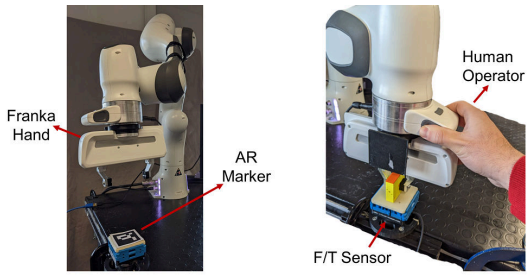


Fig. 15. Data acquisition with the Franka Hand

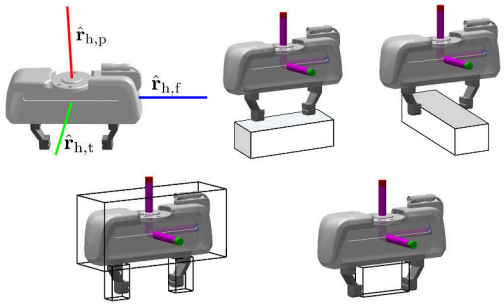


Fig. 16. Top: Definition of the local frame used to compute $\mathbb{J}_{\mathrm{a}}$ for the Panda Gripper and examples of alignment of the gripper with a box. Bottom left: approximation of the collision volume $\mathscr{O}$ as union of three cuboids. Bottom right: fingers' closing region $\mathscr{C}$.

used to provide a well-posed definition of the gripper to compute the grasp quality score are presented.

A. Acquisition of the human expertise

In order to apply the method to a different gripper, it is necessary to collect the data (grasping poses and wrenches) used to train the DTR. To this end, the setup depicted in Fig. 15 has been used to collect them. First, an ARuco marker is used to retrieve the pose of the box w.r.t. the robot frame. Then, a human operator uses the Franka Hand, through the hand-guiding interface of the Franka arm, to grasp the box attached to the F/T sensor and collect the data following the same protocol used for the Pisa/ITT SoftHand.

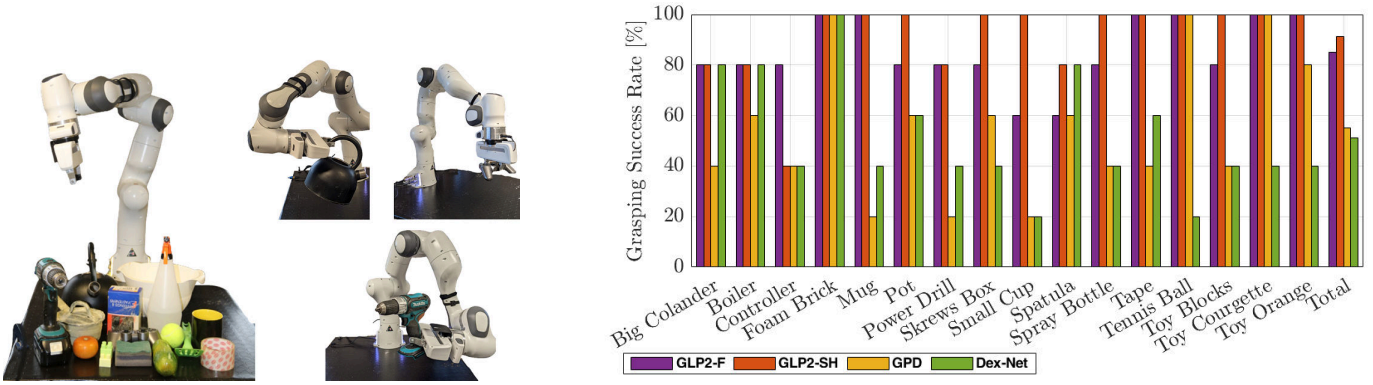
Considering the maximum opening of the gripper, it was not possible to demonstrate a grasp for each of the possible configurations of the 56 cuboids, so the dataset consists of 630 grasps, compared to the 648 grasps that were collected using the SoftHand.

1) Considerations on the grasp quality metric: Given the definition of each element of (17), while $\mathbb{J}_{\mathbf{b}}$ and $\mathbb{J}_{\mathbf{w}}$, as described in (8) and (9), are basically gripper-agnostic, $\mathbb{J}_{\mathrm{a}}$ and $\mathbb{J}_{c}$ are instead strongly gripper-dependent. Specifically, they depend on the definition of the quantities collected into the tuple $\mathtt{h}$ (see equation (4)).

For $\mathbb{J}_{c}$, being it related to the collisions of the gripper and its fingers during the grasping procedure, it should only be necessary to provide an approximation, even rough, of $\mathscr{O}$ and $\mathscr{C}$ to compute $\kappa_{\mathscr{O}}$ and $\kappa_{\mathscr{C}}$. More critical instead is the question of the box-gripper alignment. Indeed, the considerations provided in Sec.III-C3 are, at first glance arising from assumptions on the structure of the specific robotic hand used.

It is possible to use similar considerations for standard parallel grippers, as shown in Fig. 16 using the Franka Hand (a 2-finger parallel gripper). The top row shows how it is possible to define a local frame as the one defined for the SoftHand (see Fig. 3) to compute the gripper-box alignment score $\mathbb{J}_{\mathrm{a}}$. Following considerations analogous to the ones used for the SoftHand, grasps highly aligned with the longest side of a box might not be robust (or even unfeasible) given the limits on the gripper width (that for the Franka Hand is set to 0.08cm). The bottom row is instead a representation of the approximation using a combination of cuboids for $\mathscr{O}$ and the approximation of $\mathscr{C}$ through a single cuboid.

From these considerations, it is possible to provide a suitable definition of $\mathtt{h}$ even for the case of more standard grippers.



(a) Objects used for the validation and examples of generated grasps.

(b) Franka Hand: Grasping Success Rate (%).

Fig. 17. Experimental setup and grasping success rate using the Franka Hand. GLP 2.0 achieves an overall grasping success rate of 85.0%, comparable to the one obtained on the same objects with the Pisa/IIT SoftHand (91.25%). GLP 2.0 outperforms the other two baselines, which score a success rate lower than 60%.

B. Experimental setup and objects

The setup replicates the one used for the validation with the SoftHand: two Intel Realsense D415 cameras are used to collect the object point cloud. The manipulator and the gripper are controlled using ROS. The point cloud acquisition is made using the ROS interface provided by Intel. The box decomposition is implemented in C++, while the grasp prediction using the DTR is implemented with a Python script.

The performance of the algorithm is evaluated on a dataset composed of 16 objects, representing a subset of the complete dataset used for the validation with the SoftHand. A picture of the set of objects and a snapshot of executed grasps is shown in Fig. 17(a).

For each object, five grasps are attempted, with the target object placed randomly on the table at the beginning of each trial. The robot starts each trial from the same initial position. An RRT planner is used to plan a collision-free trajectory to reach the grasping pose. When the robot reaches the computed grasping pose, the gripper fingers are closed, and the manipulator tries to lift the object 150mm. A grasp is successful if the robot can keep the object grasped after the lifting phase. The trial is marked as a failure also in the case of the algorithm returning no feasible grasps.

C. Results

The results of the method applied to the Franka Hand (in the following denoted as GLP2-F) are compared to three different baselines: i) the proposed method using the Pisa/IIT SoftHand (GLP2-SH); ii) Dex-Net 2.0 [24]; iii) GPD [15].

The results of the experimental validation are reported in Fig. 17(b), where the success rate for each object is reported for the method and the three baselines. The method using the Franka Hand achieves an overall grasping success rate of 85.0%, compared to the 91.25% obtained with the same objects with the Pisa/IIT SoftHand. It is worth noting that it achieves satisfactory performance for almost every object in the dataset, except for the Small Cup and the Spatula, and clearly outperforms the other two baselines, GPD and Dex-Net, that score a success rate lower than 60%. As expected, Dex-Net is the method with the worst performance, as it has been developed assuming observations from an overhead camera. Compared to the results obtained with the SoftHand, it is possible to notice an improvement for the Controller. At the same time, there is a clear decrease for the Small Cup (60% against 100%).

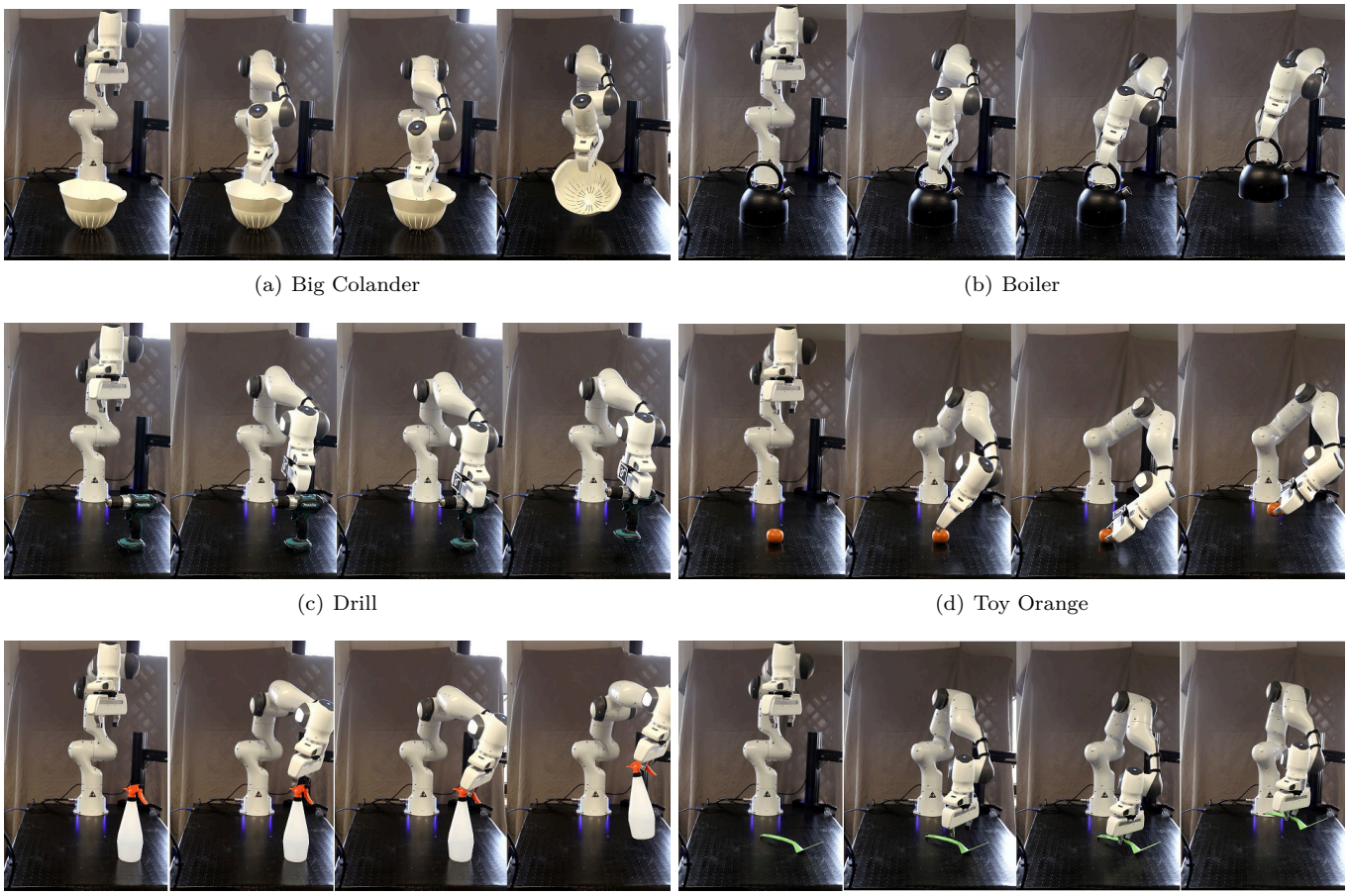
The reason for this reduction can be explained by the fact that such a small object is, in general, more affected by errors and artifacts in the acquired point cloud and depth maps. This type of artifact consequently influences the grasp pose generated and, for a rigid gripper such as the one used, can lead to failures and non-robust grasps. Indeed, it can be noted that also GPD and Dex-Net struggle with this particular object, since they are both able to grasp only once over the 5 trials. In these cases, the compliance of the gripper and its ability to adapt influence the performance positively.

Overall, these results seem to confirm that the proposed method is generally capable of generating and selecting good quality grasps even for non-compliant grippers, outperforming both GPD and Dex-Net. Furthermore, the results show that the performance are not strongly affected by the compliance of the gripper used initially. Nonetheless, using a compliant gripper can help increase robustness against uncertainties and errors related to poor perception.

VI. Limitations

This section will discuss the main limitations and shortcomings of the different parts of the proposed method.

a) Box decomposition: The main limitation is related to the robustness of the method to perception inaccuracies. Indeed, the performance of the box decomposition phase, and thus of the whole pipeline, strongly depends on the



(a) Big Colander (b) Boiler

(c) Drill (d) Toy Orange

(e) Spray (f) Spatula

Fig. 18. Selections of frames of the robot grasping the Big Colander, the Boiler, the Power Drill, the Toy Orange, the Spray, and the Spatula.

quality of the point cloud acquired by the sensors. Noisy or incomplete point clouds can lead to poor decomposition and therefore to reduced grasping performance. One possible solution to this shortcoming might be to use shape completion networks to reconstruct the acquired point cloud from partial views [64], [65]. This change would also work toward a more flexible framework because it could reduce the number of cameras needed to capture the point cloud.

b) Grasp generation: The method requires a dataset of grasp poses and interaction wrenches to train DTR. These data are gripper-dependent, therefore, to adapt the method to different grippers, it would be necessary to reacquire the data with the specific gripper. It has to be noted as this limitation is common to other data-driven algorithms that learn grasps for the specific gripper used for data collection. Further investigations are needed to increase the generality of the proposed approach, and to understand if the data collected with a specific gripper, e.g., a 2-finger parallel gripper, can be transferred to and used for other grippers that have similar properties, e.g., other 2-finger grippers.

c) Grasp selection: The grasp selection procedure depends on the evaluation of the quality score $\mathbb{S}_{\mathrm{g}}$. This score has been designed according to a set of heuristics that we believe are capable of evaluating the "quality" of a grasp. Nonetheless, some of these heuristics are based on the specific gripper definition $\mathtt{h}$ and have been described in Section III-C for the case of the SoftHand. We have later shown that it is possible to transfer the score and gripper definition to a completely different gripper, i.e., the Franka Hand. However, while some of the parameters like the ones describing the gripper's shape and closing regions, or the maximum graspable dimension, might be straightforward to compute and tune, the tuning of the other parameters, i.e., the collision's and alignment's thresholds, is less direct and it is impossible to provide a general automatic procedure.

In addition, in the current form the grasp selection policy is designed to select the best kinematically feasible grasps and does not take into consideration any other property related to the specific robot used to move the gripper, e.g., the manipulability of the grasp pose. In theory, it could be possible to modify the grasp selection policy presented in (18) by scaling the grasp quality metric by a measure of the manipulability of the specific grasp pose. This term could be computed using, e.g., the manipulability index introduced by Yoshikawa in 1985 [66]

$$\mu(\mathrm{g}) = \sqrt{\det\left(J(\mathbf{q}_{\mathrm{g}})J^T(\mathbf{q}_{\mathrm{g}})\right)},$$

where $\mathbf{q_g}$ is the joint configuration realizing the grasp pose $\mathbf{g}$, retrieved using a robot-specific inverse kinematics solver. This procedure would select a grasp considering both the quality of the grasp, measured by $\mathbb{S}_\mathbf{g}$, and the manipulability of this grasp given the specific manipulator.

d) Grasp execution: Eventually, the current framework only tackles the problem of generating a grasp pose for the gripper. It uses the interaction wrenches learned from the demonstrations only for planning purposes, i.e., to compute the associated score $\mathbb{J}_\mathbf{w}$, and does not use them during the execution of the grasp. The selected grasp is then executed in open-loop. However, especially in presence of uncertain and noisy perception, open-loop execution might cause failures because of early or inaccurate hand-object contacts [9]. Indeed, including closed-loop adaptive grasping strategies, capable of using tactile and contacts information [9], or the information of the learned interaction wrenches, into the proposed framework have the potential of improving the performance, especially when using rigid, non compliant, grippers.

VII. Conclusions

In this paper, we presented GLP 2.0, a data-driven grasp planning algorithm for grasping of unknown objects. The method leverages and improves a previous work from the same authors, which exploits an approximation of the target object into a generic number of basic shapes to generate a set of candidate grasps from demonstrations by a skilled human operator grasping the same shape. Based on the same philosophy of transferring the human grasping skills to the robot, we proposed an improved grasp-generation method and a novel grasp-selection policy. The experimental validation with a compliant underactuated hand shows that the method outperforms the original algorithm in terms of overall grasping success rate, at the expense of an increased execution time. We also showed that the method is transferable to more standard, rigid, grippers, providing an experimental validation of the method when applied to the Franka Hand, a two-finger gripper. Finally, we compared the method with several baselines, showing that GLP 2.0 achieves better performance in terms of grasping success rate. Future works will extend the method to multi-object cluttered scenarios and to task-oriented grasp planning.

Appendix

Object Dimensions

In Tab.II we reported the dimensions and the weights of the objects used for the experimental validation.

Box Decomposition Algorithm

Algorithm 1 describes the decomposition procedure. The algorithm uses a fit-and-split approach, that starts with fitting a 3D bounding box on the whole point cloud. This box is added to a list of candidate boxes, that are iteratively tested for potential splitting. If the volume of the candidate box or the number of points enclosed within it are below a user-specified threshold, the candidate box is added to the final set of boxes. Otherwise, the algorithm evaluates a best split of this box, by using 2D projections of the enclosed points to the box faces, according to the procedure reported in Algorithm 2. This split produces two boxes, box_1 and box_2, and if the reduction rate of the volume of the two new boxes compared with the original box is less then the user-specified gain, the two are added to the list of candidate boxes. Oterwhise, the split is not enforced and the original box is added to the final set of boxes.

Decision Tree Regressor

We use Decision Tree Regressors (DTRs) to learn the model $\tilde{\psi}(\boldsymbol{\lambda}_\mathbf{b})$ to predict the grasp pose and the associated interaction wrenches. In this work we trained two DTRs, one for predicting the grasp pose given a vector of box dimensions, and one for regressing the metric of the interaction wrenches described in Section III-B2. The Regression Tree model is represented as a binary tree, with each node representing a single input variable x_j and a split point on that variable. The tree's leaf nodes include an output variable $\mathbf{y}$, which is utilized to produce a prediction. Given a $(\mathbf{x}_i, \mathbf{y}_i)$ observation for $I = 1, 2 \ldots, n$, the regression tree construction is explained by the following stages: i) choose a splitting variable j and a splitting point s; ii) create two regions, R_1 and R_2: $R_1(j,s) = \{\mathbf{x}|x_j \leq s\}$, $R_2(j,s) = \{\mathbf{x}|x_j > s\}$; iii) for each $k \in \{1, \ldots, K\}$ find the splitting variable j and the split point s that solve the problem

$$\min_{j,s}[\min_{c_{1,k}} \sum_{\mathbf{x}_i \in R_1(j,s)} (y_{i,k} - c_{1,k})^2 + \min_{c_{2,k}} \sum_{\mathbf{x}_i \in R_2(j,s)} (y_{i,k} - c_{2,k})^2],$$

where $c_{1,k}, c_{2,k} \in \mathbb{R}$ are two constant decision variables that describe the model response. Given j and s, the solution of the minimization is

$$\hat{c}_{1,k} = \text{ave}(y_{i,k}|\mathbf{x}_i \in R_1(j,s)), \hat{c}_{2,k} = \text{ave}(y_{i,k}|\mathbf{x}_i \in R_2(j,s)),$$

where ave is the average of $y_{i,k}$ in region R_1 or R_2, and we define $\hat{c}_1 = [\hat{c}_{1,1}, \ldots, \hat{c}_{1,K}]^T$ and $\hat{c}_2 = [\hat{c}_{2,1}, \ldots, \hat{c}_{2,K}]^T$. iv) After determining the optimal split for each splitting variable, divide the data into two areas and perform the splitting procedure on each of the two regions recursively. The maximum tree depth is empirically chosen to 8 as a trade-off between model complexity and risk of overfitting. We used K-fold cross validation to verify the performance of the trained models, evaluating the MSE error between the models' predictions and the true labeled data in the validation set.

References

[1] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019.

[2] J. Mahler *et al.*, "Guest editorial open discussion of robot grasping benchmarks, protocols, and metrics," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1440–1442, 2018.

TABLE II
OBJECT DIMENSIONS

Bowl	Container	Cup	Foam Brick	Juice Box
				
∅140×65mm, 55g.	∅115×48mm, 35g.	∅74×84mm, 35g.	92×44×44mm, 6g.	60×38×80mm, 221g.
Measuring Tape	**Mug**	**Pot**	**Power Drill**	**Skillet**
				
75×35×65mm, 189g.	135×95×100mm, 345g.	280×133×74mm, 374g.	200×95×195mm, 1190g.	380×230×40mm, 316g.
Screws Box	**Small Cup**	**Spatula**	**Spray Bottle**	**Tape**
				
125×55×90mm, 444g.	80×50×60mm, 107g.	310×80×60mm, 45g.	120×95×290mm, 927g.	∅89×53mm, 62g.
Tennis Ball	**Torch**	**Toy Apple**	**Toy Blocks**	**Toy Courgette**
				
∅71mm, 46g.	185×∅78mm, 466g.	∅73×64mm, 140g.	87×87×47mm, 34g.	145×∅49mm, 15g.
Toy Orange	**Big Colander**	**Small Colander**	**Controller**	**Brush**
				
∅73mm, 18g.	∅270×160mm, 230g.	∅270×105mm, 87g.	157×95×55mm, 170g.	50×260×55mm,285g.
White Pot*	**Boiler**	**Bottle**	**Foam Brick***	**Ball***
				
290×135×130mm, 485g.	∅200×230mm, 690g.	∅67×199mm, 533g.	60×45×60mm, 5g.	∅65mm, 56g.

[3] M. Garabini *et al.*, "Wrapp-up: A dual-arm robot for intralogistics," *IEEE Robot. Autom. Mag.*, vol. 28, no. 3, pp. 50–66, 2020.

[4] Q. Lei *et al.*, "A survey of unknown object grasping and our fast grasping algorithm-c shape grasping," in *2017 3rd Int. Conf. Control Autom. Robot. (ICCAR)*. IEEE, 2017, pp. 150–157.

[5] A. Sahbani *et al.*, "An overview of 3d object grasp synthesis algorithms," *Rob. Auton. Syst.*, vol. 60, no. 3, pp. 326–336, 2012.

[6] A. Bicchi, "On the closure properties of robotic grasping," *Int. J. Robot. Res.*, vol. 14, no. 4, pp. 319–334, 1995.

[7] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings 2000 ICRA. Millennium Conference. IEEE Int. Conf. Robot. Autom.. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 1. IEEE, 2000, pp. 348–353.

[8] A. Rodriguez *et al.*, "From caging to grasping," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 886–900, 2012.

[9] G. J. Pollayil *et al.*, "Sequential contact-based adaptive grasping for robotic hands," *Int. J. Robot. Res.*, vol. 41, no. 5, pp. 543–570, 2022.

[10] A. T. Miller *et al.*, "Automatic grasp planning using shape primitives," in *2003 IEEE Int. Conf. Robot. Autom. (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1824–1829.

[11] C. Goldfeder *et al.*, "Grasp planning via decomposition trees," in *Proceedings 2007 IEEE Int. Conf. Robot. Autom.* IEEE, 2007, pp. 4679–4684.

[12] J. Bohg *et al.*, "Data-driven grasp synthesis—a survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, 2014.

[13] D. Kappler *et al.*, "Leveraging big data for grasp planning," in *2015 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2015, pp. 4304–4311.

[14] J. Mahler *et al.*, "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 *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2016, pp. 1957–1964.

[15] A. ten Pas *et al.*, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[16] J. Mahler *et al.*, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[17] A. Mousavian *et al.*, "6-dof graspnet: Variational grasp

Algorithm 1: MVBB Decomposition

```
Input: cloud, min_points, min_volume, gain
Output: Boxes
Boxes ← ∅ ;
points ← cloud.points ;
BoxTree ← FindBoundingBox(points) ;
while BoxTree ≠ ∅ do
    box ← BoxTree.pop ;
    if
     |box.points| ≤ min_points ∨ |box.volume| ≤ min_volume
     then
        Boxes.push(box);
    else
        faces ← nonOppositeFaces(box) ;
        (box1, box2) ←
          split(FindBestSplit(faces, box.points)) ;
        if gainVolume(box1, box2,box)<gain then
            BoxTree.push(box1);
            BoxTree.push(box2) ;
        else
            Boxes.push(box);
return Boxes ;
```

Algorithm 2: Find Best Split

```
Input: faces, points
Output: bestSplit
bestSplit ← ∅ ;
for f ∈ faces do
    p ← Project(points,f);
    for x ∈ p.x do
        (p1,p2) ← VerticalSplit(p,x) ;
        a1 ← Area(p1) ;
        a2 ← Area(p2) ;
        if a1+a2<minArea then
            minArea ← a1+a2 ;
            bestSplit←(i,x) ;
    for y ∈ p.y do
        (p1,p2) ← VerticalSplit(p,y) ;
        a1 ← Area(p1) ;
        a2 ← Area(p2) ;
        if a1+a2<minArea then
            minArea ← a1+a2 ;
            bestSplit←(i,y) ;
return bestSplit ;
```


generation for object manipulation," in *Proceedings of the IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2901–2910.

[18] C. Della Santina *et al.*, "Learning from humans how to grasp: a data-driven architecture for autonomous grasping with anthropomorphic soft hands," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1533–1540, 2019.

[19] C. Gabellieri *et al.*, "Grasp it like a pro: Grasp of unknown objects with robotic hands based on skilled human expertise," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2808–2815, 2020.

[20] H. Zhang *et al.*, "A real-time robotic grasping approach with oriented anchor box," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 5, pp. 3014–3025, 2021.

[21] R. Xu *et al.*, "Gknet: Grasp keypoint network for grasp candidates detection," *Int. J. Robot. Res.*, vol. 41, no. 4, pp. 361–389, 2022.

[22] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Auton. Robots*, vol. 38, no. 1, pp. 65–88, 2015.

[23] I. Lenz *et al.*, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4-5, pp. 705–724, 2015.

[24] J. Mahler *et al.*, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Robotics: Science and Systems (RSS)*, 2017.

[25] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016, pp. 3406–3413.

[26] S. Levine *et al.*, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 421–436, 2018.

[27] M. G. Catalano *et al.*, "Adaptive synergies for the design and control of the pisa/iit softhand," *Int. J. Robot. Res.*, vol. 33, no. 5, pp. 768–782, 2014.

[28] M. Pozzi *et al.*, "Hand closure model for planning top grasps with soft robotic hands," *Int. J. Robot. Res.*, vol. 39, no. 14, pp. 1706–1723, 2020.

[29] S. Haddadin *et al.*, "The franka emika robot: A reference platform for robotics research and education," *IEEE Robot. Autom. Mag.*, pp. 2–20, 2022.

[30] K. Huebner *et al.*, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *2008 IEEE Int. Conf. Robot. Autom.* IEEE, 2008, pp. 1628–1633.

[31] A. Herzog *et al.*, "Template-based learning of grasp selection," in *2012 IEEE Int. Conf. Robot. Autom.* IEEE, 2012, pp. 2379–2384.

[32] A. Gupta *et al.*, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *2016 IEEE/RSJ IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2016, pp. 3786–3793.

[33] A. Saxena *et al.*, "Robotic grasping of novel objects using vision," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 157–173, 2008.

[34] N. Vahrenkamp *et al.*, "Part-based grasp planning for familiar objects," in *2016 IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 919–925.

[35] Y. Xu *et al.*, "Graspcnn: Real-time grasp detection using a new oriented diameter circle representation," *IEEE Access*, vol. 7, pp. 159 322–159 331, 2019.

[36] M. Kokic *et al.*, "Learning task-oriented grasping from human activity datasets," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3352–3359, 2020.

[37] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb," in *2021 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2021, pp. 13 452–13 458.

[38] Y. Jiang *et al.*, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE Int. Conf. Robot. Autom.* IEEE, 2011, pp. 3304–3311.

[39] A. Miller and P. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[40] A. Rocchi and K. Hauser, "A generic simulator for underactuated compliant hands," in *2016 IEEE Int. Conf. Simul. Model. Program. Auton. Robots (SIMPAR)*. IEEE, 2016, pp. 37–42.

[41] R. Mengacci *et al.*, "An open-source ros-gazebo toolbox for simulating robots with compliant actuators," *Front. Robot. AI*, p. 246, 2021.

[42] E. Klingbeil *et al.*, "Grasping with application to an autonomous checkout robot," in *2011 IEEE Int. Conf. Robot. Autom.* IEEE, 2011, pp. 2837–2844.

[43] J. Bohg *et al.*, "Mind the gap-robotic grasping under incomplete observation," in *2011 IEEE Int. Conf. Robot. Autom.* IEEE, 2011, pp. 686–693.

[44] A. H. Quispe *et al.*, "Exploiting symmetries and extrusions for grasping household objects," in *2015 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2015, pp. 3702–3708.

[45] J. Lundell *et al.*, "Robust grasp planning over uncertain shape completions," in *2019 IEEE/RSJ IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2019, pp. 1526–1532.

[46] M. Kiatos *et al.*, "A geometric approach for grasping unknown objects with multifingered hands," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 735–746, 2020.

[47] B. Calli *et al.*, "Grasping of unknown objects via curvature maximization using active vision," in *2011 IEEE/RSJ IEEE Int. Conf. Intell. Robots Syst.*, 2011, pp. 995–1001.

[48] L. Shao *et al.*, "Unigrasp: Learning a unified model to grasp with multifingered robotic hands," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2286–2293, April 2020.

[49] C.-H. Liu *et al.*, "Optimal design of a soft robotic gripper for grasping unknown objects," *Soft Robot.*, vol. 5, no. 4, pp. 452–465, 2018.

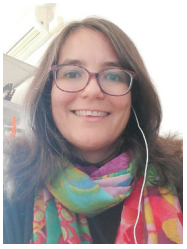
[50] F. Angelini *et al.*, "Softhandler: An integrated soft robotic system for handling heterogeneous objects," *IEEE Robot. Autom. Mag.*, vol. 27, no. 3, pp. 55–72, 2020.
[51] V. Lippiello *et al.*, "Visual grasp planning for unknown objects using a multifingered robotic hand," *IEEE/ASME Trans. Mechatron .*, vol. 18, no. 3, pp. 1050–1059, 2012.
[52] G. Vezzani *et al.*, "A grasping approach based on superquadric models," in *2017 IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2017, pp. 1579–1586.
[53] A. Makhal *et al.*, "Grasping unknown objects in clutter by superquadric representation," in *2018 Second IEEE Int. Conf. Robot. Comput. (IRC)*. IEEE, 2018, pp. 292–299.
[54] C. Goldfeder *et al.*, "Grasp planning via decomposition trees," in *Proceedings 2007 IEEE Int. Conf. Robot. Autom.*, 2007, pp. 4679–4684.
[55] T. T. Cocias, *et al.*, "Multiple-superquadrics based object surface estimation for grasping in service robotics," in *2012 13th Int. Conf. Optim. Electr. Electron. Equip. (OPTIM)*, 2012, pp. 1471–1477.
[56] M. Bonilla *et al.*, "Grasp planning with soft hands using bounding box object decomposition," in *2015 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2015, pp. 518–523.
[57] M. Pozzi *et al.*, "The closure signature: a functional approach to model underactuated compliant robotic hands," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2206–2213, 2018.
[58] A. ten Pas and R. Platt, *Using Geometry to Detect Grasp Poses in 3D Point Clouds*. Cham: Springer International Publishing, 2018, pp. 307–324.
[59] C. Rubert *et al.*, "On the relevance of grasp metrics for predicting grasp success," in *2017 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2017, pp. 265–272.
[60] M. Sorour *et al.*, "Grasping unknown objects based on gripper workspace spheres," in *2019 IEEE/RSJ IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2019, pp. 1541–1547.
[61] F. Flacco and A. De Luca, "A reverse priority approach to multi-task control of redundant robots," in *2014 IEEE/RSJ IEEE Int. Conf. Intell. Robots Syst.* IEEE, 2014, pp. 2421–2427.
[62] ——, "Unilateral constraints in the reverse priority redundancy resolution method," in *2015 IEEE/RSJ IEEE Int. Conf. Intell. Robots Syst. (IROS)*. IEEE, 2015, pp. 2564–2571.
[63] B. Calli *et al.*, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, 2015.
[64] X. Wang *et al.*, "Cascaded refinement network for point cloud completion," in *CVPR*, 2020, pp. 790–799.
[65] X. Yu *et al.*, "Pointr: Diverse point cloud completion with geometry-aware transformers," in *ICCV*, 2021.
[66] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.

Alessandro Palleschi is a Postdoctoral Researcher at the Research Center "E. Piaggio", University of Pisa. He received from the University of Pisa his BSc. in Electronics in 2015, his MSc. in Robotics in 2018, both cum laude, and his Ph.D. degree in Robotics in 2023, with honors. He has been a Visiting Researcher at the Interactive Perception and Robot Learning Lab, Stanford University, from April to September 2022. He is also co-founder of the company XStar Motion. His research interests include robotic manipulation and trajectory optimization.

Franco Angelini is an Assistant Professor at University of Pisa. He received the B.S. degree in computer engineering in 2013 and M.S. degree (cum laude) in automation and robotics engineering in 2016 from the University of Pisa, Pisa, Italy. University of Pisa granted him also a Ph.D. degree (cum laude) in robotics in 2020. His main research interests are control of soft robotic systems, impedance planning, grasping, and robotic environmental monitoring.

Chiara Gabellieri is a researcher in the RaM group at the University of Twente. She is a Marie Skłodowska-Curie postdoctoral fellow with the Flyflic project and work-package leader in the Horizon Europe coordination and support action AeroSTREAM. Her research interests include grasping, environmental robotics, and aerial manipulation. Chiara received her Ph.D. in Information Engineering in 2021 from the University of Pisa and, from the same institution, her MSc. in Robotics and Automation Engineering with honors in 2017 and her BSc. in Bioengineering in 2014. She was at LAAS-CNRS from November 2017 to May 2018 and a visiting Ph.D. student at the German Aerospace Center (DLR) from November 2019 to May 2020.

Do Won Park received his MSc. in Robotics and Automation Engineering from the University of Pisa in 2021, and his BSc. in Bioengineering in 2017, from the same institution. In his master thesis, he investigated grasp planning algorithms for robots equipped with anthropomorphic robotic hands.



Lucia Pallottino is Associate Professor at the Centro di Ricerca "E. Piaggio" and the Dipartimento di Ingegneria dell'Informazione at the University of Pisa. She received the "Laurea" degree in Mathematics (1998) and a Doctoral degree in Robotics and Industrial Automation (2002). She has been Visiting Scholar at M.I.T. (2000-2001) and Visiting Researcher at UCLA, (2004). She is Director of Centro di Ricerca "E. Piaggio" (since Jan. 2023), Co-founder of Proxima robotics srl and of XStar Motion, and Scientific collaborator in the DARKO European project. Her main research interests are motion planning and control, optimal control, coordination of multi–robot systems, distributed algorithms.

Antonio Bicchi is a scientist interested in robotics and intelligent machines. He holds a chair in Robotics at the University of Pisa, leads the Soft Robotics Laboratory at the Italian Institute of Technology in Genova, and is an Adjunct Professor at Arizona State University. His work has been recognized with many international awards and has earned him four prestigious grants from the European Research Council (ERC). He launched initiatives such as the WorldHaptics conference series, the IEEE Robotics and Automation Letters, and the Italian Institute of Robotics and Intelligent Machines.

Manolo Garabini graduated in Mechanical Engineering and received the Ph.D. in Robotics from the University of Pisa where he is currently employed as Associate Professor. His main research interests include the design, planning, and control of soft adaptive robots. Co-founder of qbrobotics and of XStar Motion, currently, he is the coordinator of the H2020 EU Research Project Natural Intelligence.