Tampere University

Kuisma Hannuksela

LEARNED MANIPULATION ON ROBOT ARMS WITH PARALLEL-JAW GRIPPERS

Bachelor's thesis
Faculty of Engineering and
Natural Sciences (ENS)
Examiner: Roel Pieters
May 2023

ABSTRACT

Kuisma Hannuksela: Learned manipulation on robot arms with parallel-jaw grippers
Bachelor’s thesis
Tampere University
Bachelor’s Degree Programme in Science and Engineering
May 2023

Utilizing machine learning is a promising prospect for various robotic manipulation applications and has been of growing interest in the research community recently. This thesis reviews research papers relevant to learned manipulation on robot arms with two-finger grippers, published since 2021. The focus was chosen to limit the mechanical complexity, and to find out how the field of research has developed since the latest literature reviews with a broad perspective on the topic. Pressing challenges listed in two earlier reviews were used as a basis for evaluating recent developments. The thesis is organized into chapters discussing subtasks common in robot manipulation, the usage of simulation for robot learning, and human-robot interaction and learning from demonstration.

Robots get information on their surroundings as prior information and observations. Visual and depth observations are commonly used, as well as the robot arm’s internal observations, such as joint torque and gripper finger tactile sensing. Various external signals can be used to convey human intent to the robot learning system. Notably, many of the approaches utilizing multiple sensory modalities are inspired by human perception.

Approaches to task planning beyond immediate action and over multiple subtasks vary extensively. The tasks are also different in each study, so directly comparing the performance of task planning methods was deemed unfeasible. Instead, the novelties and challenges of individual approaches are discussed in terms of performance as well as adaptability, explainability and generalizability. The reviewed planning methods demonstrate the benefits of structural and modular policies, as well as reinforcement learning, often with experience replay.

Grasp synthesis has been extensively studied. Grasping is typically based on observations with a camera mounted to either a fixed point or at the robot arms end effector. High grasping success rates have been achieved with both observation approaches when the task is to grasp a singular object. However, cluttered scenes pose a problem for grasping, and methods with push-grasp actions and scene rearrangement have been developed for the purpose.

Translation and rotation can be trivial subtasks, but to avoid collisions or for instance spilling coffee on a laptop more complicated path planning may be required. Various machine learning methods have proven successful in path planning and learning specific trajectory shapes for tasks like page-flipping. The challenges of moving while at contact, as well as the stability of learned robot arm movement have also been addressed in literature. Releasing grip is similarly trivial but preparing for the release by correctly orienting objects or ensuring successful assembly are challenges where machine learning has proven beneficial.

Simulation is utilized in most of the reviewed studies. The thesis discusses the benefits of simulation, simulation benchmarks, as well as the most encountered simulation environments and their development. The challenges of domain transfer from simulation to reality and the methods for reducing the reality gap are also addressed.

Humans can interact with robot arms physically, with teleoperation or by less direct means such as EEG-signal. Different forms of human demonstrations are also used in various ways for training robots.

The thesis concludes with a discussion on development regarding the list of challenges rephrased from prior reviews. The integration of learning approaches into more complete and generalizable systems is singled out as a key challenge for future research.

Keywords: machine learning, robotics, manipulation, robot arm, parallel-jaw gripper, two-finger gripper, literature review

TIIVISTELMÄ

Kuisma Hannuksela: Koneoppiminen kaksisormisen robottikäsivarren manipulointitehtävissä
Kandidaatintyö
Tampereen yliopisto
Teknisten tieteiden kandidaatin tutkinto-ohjelma
Toukokuu 2023

Koneoppimisen hyödyntäminen on lupaava mahdollisuus robotiikan alalla, ja kiinnostus aiheeseen on kasvanut tutkimusyhteisössä viime vuosina. Tässä kandidaatintyössä tarkastellaan koneoppimista kaksisormisen robottikäsivarren manipulaatiotehtävissä. Kirjallisuuskatsaus keskittyy vuodesta 2021 alkaen julkaistuihin tutkimuksiin, jotta voidaan selvittää kehitysaskelia viimeisimpien aihetta yleisesti käsittelevien kirjallisuuskatsausten jälkeen. Kahdessa aiemmassa katsauksessa listataan merkittäviä haasteita, joihin alan tutkimuksen tulisi vastata, ja niistä koostettua listaa käytetään tässä työssä viimeaikaisen kehityksen tarkastelun pohjana. Kandidaatintyö on jaettu lukuihin, joissa perehdytään robotiikan manipulaatiotehtävien osatehtäviin, simulaation hyödyntämiseen, ihmisen ja robotin vuorovaikutukseen sekä malliesimerkeistä oppimiseen.

Robotit saavat tietoa ympäristöstään ennakkotietoina ja havaintoina. Yleisimmin havainnointiin käytetään kameroita, syvyyskuvalla tai ilman, sekä robottikäsivarren sisäisiä mittauksia, kuten nivelten vääntömomentteja ja tarttujan kosketusanturimittauksia. Ihmisen aikomuksia voidaan viestiä robotille erilaisilla ulkoisilla signaaleilla. Huomionarvoista on, että monet useita erityyppisiä antureita hyödyntävät menetelmät ottavat mallia ihmisen havainnointitavoista.

Useita toimintavaiheita kattavien manipulointitehtävien suunnitteluun pyritään tutkimuksissa hyvin vaihtelevilla tavoilla. Myös tehtävätyypit vaihtelevat tutkimusten välillä, joten menetelmien suorituskykyä ei voitu suoraan vertailla. Eri tehtävänsuunnittelumenetelmiä tarkastellaankin suorituskyvyn lisäksi mukautuvuuden, selitettävyyden ja yleistettävyyden kannalta. Katsauksessa nousi esiin rakenteellisten ja modulaaristen menetelmien sekä vahvistusoppimisen hyötyjä tehtävänsuunnittelussa.

Kaksisormista tartuntaa on tutkittu kattavasti. Tartunta perustuu useimmiten kameralla saatuun havaintoon tartuttavasta kappaleesta. Kamera voi olla joko paikallaan, usein suoraan työtason yläpuolella, tai kiinnitettynä robottikäsivarteen lähelle tarttujaa, ja molemmilla tavoilla on saavutettu korkeita onnistumisasteita yksittäisen kappaleen tartunnassa. Kun työtasolla on lukuisia kappaleita epäjärjestyksessä, tehtävä on huomattavasti hankalampi. Tätä haastetta pyritään tutkimuksissa ratkomaan työntö- ja tartuntaliikkeen yhdistämisen sekä kappaleiden uudelleenjärjestelyn keinoin.

Siirto- ja kiertoliikkeen suunnitteluun voidaan tarvita koneoppimista, esimerkiksi jos on tarpeen välttää törmäyksiä tai kahvikupin siirtoa tietokoneen yli. Useat tarkastellut koneoppimismenetelmät toimivat hyvin reittisuunnittelussa, tai pystyvät oppimaan esimerkiksi sivunkääntöön vaadittuja liikeratoja. Kontaktin aikana liikkumista ja liikkeen stabiiliutta tarkastellaan myös kirjallisuudessa. Tartunnan irrotusta pitää joissain tapauksissa valmistella kääntämällä kappale haluttuun suuntaan tai varmistamalla kokoonpanon onnistuminen, ja myös näihin haasteisiin koneoppiminen on kyennyt vastaamaan.

Enemmistö viitatuista tutkimuksista hyödyntää simulaatiota. Työssä tarkastellaan simulaation hyötyjä, yleisimpiä simulaatioympäristöjä, sekä simulaation mahdollistamaa menetelmien suorituskykyvertailua. Simulaatiossa opitun siirto todelliseen maailmaan on haastavaa, ja työssä esitellään myös keinoja simulaation ja todellisuuden välisen kuilun pienentämiseen.

Ihminen voi vaikuttaa robottikäsivarren toimintaan fyysisesti, etäohjauksella, tai epäsuoremmilla keinoilla, kuten aivosähkökäyrän välityksellä. Erilaisia malliesimerkkejä voidaan käyttää robotin kouluttamiseen lukuisin eri tavoin.

Yhteenvetona palataan aiempien kirjallisuuskatsausten nostamiin haasteisiin. Koneoppimismenetelmien integraatio kokonaisvaltaisemmiksi ja paremmin eri manipulaatiotehtäviin yleistettäviksi järjestelmiksi on tärkeä yksittäinen kehityskohde alan tutkimukselle.

Avainsanat: koneoppiminen, robotiikka, manipulointi, robottikäsivarsi, kaksisorminen tarttuja, kirjallisuuskatsaus

PREFACE

I would like to thank associate professor Roel Pieters at Tampere University for supervising the thesis. Throughout the project, he provided valuable feedback and guidance on the thesis and insight on the topic of robot learning. My seminar opponent Aylin Pyykkö also contributed by giving feedback from a peer student's perspective.

Tampere, 5 May 2023

Kuisma Hannuksela

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

A-C	Actor-Critic
ANN, NN	(Artificial) Neural Network
CNN, ConvNet	Convolutional Neural Network
DNN	Deep Neural Network
EEG	Electroencephalography
ErrP	Error-related Potentials
GAN	Generative Adversarial Network
GTN	Geometric Task Network
HER	Hindsight Experience Replay
LfD	Learning from Demonstration
MDP	Markov Decision Process
RGB	Red-Green-Blue
RGB-D	Red-Green-Blue-Depth
RL	Reinforcement Learning
sim2real	Simulation to Reality transfer
Q	Action-value representation

.

1. INTRODUCTION

The utilization of machine learning has led to rapid development in the field of robotics in recent years. One area where learning has resulted in notable success is robotic arms performing manipulation tasks.

This thesis focuses on tabletop robot arms with simple two-finger grippers to avoid the mechanical complexity of multi-fingered and soft robot hands as well as the larger workspace and navigation challenges of mobile robots. However, many of the presented findings are at least partially applicable for other types of robots as well.

Relevant research was searched on the Andor search engine with the query “robot AND learn* AND manipulation AND arm NOT mobile NOT bimanual NOT dual-arm”, filtered by articles in peer-reviewed journals in 2021-2023. The search results were screened based on abstract, and further if unsure, resulting in a total of 49 research papers. Some of the referenced papers feature robots with for instance tools or three-fingered grippers instead of parallel-jaw grippers. None those articles include grasping tasks, and they were considered directly applicable and relevant to the review’s focus.

In addition, 6 earlier reviews on the topic, as well as some supporting articles, book chapters, and webpages mostly are referenced. While the references do not cover all relevant research in the area, they should be sufficient to form a general understanding of the latest developments in learned manipulation on robots with parallel-jaw grippers. The purpose of the thesis is to discuss and categorize these developments, comparing the principles behind them on a descriptive level.

The thesis begins with theoretical background, including earlier reviews on the topic and a brief description of common learning methodologies in robotic manipulation. The next chapters discuss recent developments in the subtasks involved with manipulation, the use of simulation, and human effect to robotic systems via interaction and demonstration. To conclude, the findings of previous chapters are summarized and challenges yet to be resolved are discussed.

2. THEORETICAL BACKGROUND

Learning for robotic manipulation has been subject to grown interest over recent years. Multiple review papers on the topic of robot manipulation have been published before. The following section outlines some of the most relevant reviews on the topic. The chapter concludes with a general description of some of the most popular methodologies utilized in learning robot manipulation.

2.1 Earlier reviews on learned robot manipulation

Several reviews on the topic of learned robot manipulation have been conducted previously. This area of research has seen rapid development over just a few years, which can be seen through the change of views of review papers on year-to-year basis.

Mason takes a broad view at manipulation in their review paper, published in 2018. The review starts with defining manipulation as “an agent’s control of its environment through selective contact”, which is adopted for the purposes of this thesis. He then describes human and animal manipulation before moving on to robotic manipulation and comparing the two. Interestingly, machine learning is named as a future direction of robotic manipulation showing that, despite some successful examples, learned robot manipulation was still in its infancy. [1]

In their 2019 review, Billard and Kragle note the progress in learning methods for manipulation tasks, particularly grasping. They discuss the problem of training data acquisition, pointing out image and video data available on the internet, training in simulation, and expert demonstration as possible avenues, arriving at the conclusion that there are problems in robotics that cannot be answered by learning. [2]

Kroemer et al. conduct a detailed and extensive review of robot learning for manipulation, published in 2020. They reference over 400 papers to discuss different aspects of learned robot manipulation, including perception, manipulation actions and their transition model as well as different model types and data collection strategies. They highlight the modular nature of manipulation problems, suggesting that hierarchical skill structures are useful for more complex problems and transferring learned skills to different tasks. For future research they point out that the versatility and robustness of manipulation skills should be improved, along with a list of other pressing challenges. [3]

Cui and Trinkle expand on the previous reviews by addressing the adaptability of learned robot manipulation. They discuss adapting to variations that may be internal or external to the robot, previously encountered or novel. Capturing generalized information, active learning and learning from demonstration are presented as potential solutions. Moreover, the review agrees with Kroemer et al. on the benefits of modularity. Similarly, Cui and Trinkle conclude their review with a list of open questions and potential avenues for future research. [4]

More recently, Mohammed et al. review deep reinforcement learning methods for manipulation of objects cluttered environments. The scope of their review is narrower than this thesis in both methodology and environment, but it provides valuable and detailed insight in that category of robot manipulation problems. The review covers articles between 2016 and 2022 stating that it has been the most productive period of research in the subject. [5]

Similarly, Elguea-Aguinaco et al. note a significant increase in the number of publications on reinforcement learning and robot manipulation over recent years. They focus their review on reinforcement learning in contact-rich manipulation, covering studies from 2017 to 2022. The review analyses the main trends and challenges of reinforcement learning in contact-rich tasks, and proposes a framework connecting the main concepts in the area. [6]

Earlier reviews have extensively mapped the development learned robot manipulation until 2021. Hence, this review discusses new research over 2021 and 2022, focusing on machine learning in manipulation tasks performed on robot arms with parallel-jaw grippers. The prior works that are closest to this domain, although broader in scope [3], [4], provide a combined list of challenges as follows:

- Multi-modal learning, exploiting various sensory modalities beyond vision [3], [4]
- Informed exploration strategies for active learning [3], [4]
- Continuing to improve on learned skills as they are being used [3], [4]
- Reducing sample complexity in policy learning, while not resorting to empirically tuned hyper-parameters [3]
- Better generalization and adaptation by enabling customization by modularity and domain adaptation, as well as improving transfer across robot embodiments [4] and different tasks [3]
- Computational improvements such as parallel active learning and real-time performance [4]

- Advanced simulators [4] and exploitation of physical knowledge [3]
- Safety guarantees for learning [3]
- Integration of learning into more complete systems [3]

The above list functions as valuable expert insight, as this thesis attempts to examine which of these challenges have been addressed, and to what extent, since their publishing.

2.2 Learning methodologies for manipulation

Technical details of robot manipulation learning methods are outside the scope of this thesis. However, it is necessary to establish a general understanding of the utilized methodologies to be able to compare the various proposed approaches. Hence, this section describes the types of learning methods most encountered in the following chapters.

Most studies reviewed in this thesis utilize (artificial) neural networks, ANN or NN for short, in their learning algorithms. A neural network comprises of connected layers of functions, inspired by neurons in the brain. The parameters of these functions are learned by gradually adjusting to desired output from the given input. Neural networks are called deep neural networks (DNN) if they have multiple hidden layers of neurons between the input and output layers. [7] Commonly used variations of neural networks in robot manipulation are convolutional neural networks (CNN, or ConvNet), with matrix-form input and cross-neuron convolutional kernels [8, p. 36], and recurrent neural networks (RNN), that handle timeseries data in a recurrent manner [9].

Reinforcement learning, RL for short, is a popular methodology in robotics applications. Technical details differ, but the basic idea is that an RL-policy learns a value function based on an external reward function and aims to maximize future rewards while executing repetitive trials. [10] While RL-methods are mostly built on neural networks, other policy search algorithms, such as the model-based relative entropy policy search (MORE), based on information theory, exist and have practical uses [11]. RL systems are generally modelled by a Markov Decision Process, MDP for short, or its variants. The MDP is a sequential representation of the agent’s states, actions, and goal. [6] **Figure 1** shows a flowchart of the reinforcement learning process.

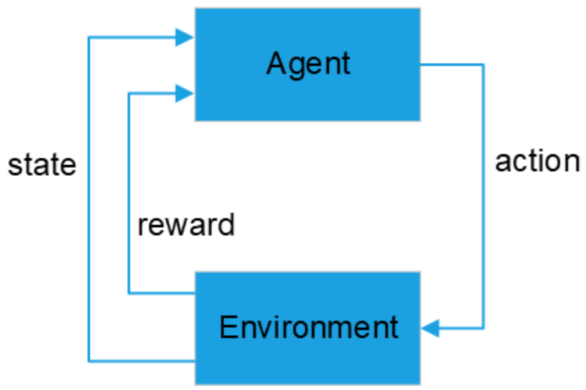

Figure 1. *Reinforcement learning process, adapted from [10]*

Q-learning is an RL methodology that learns an action-value representation, denoted by Q, instead of a value function. Consequently, learning a Q-function removes the need for action selection and learning models, making Q-learning a model-free approach. [10]

Another reinforcement learning type is actor-critic (A-C), where the policy network, actor, and value network, critic, are trained in parallel. The critic network learns to evaluate the current policy by estimating the value function, while the actor learns to improve the policy based on the critic's evaluation. [10]

Similarly, Generative Adversarial Networks (GAN) train a generative model and a discriminative model in parallel. The generative model attempts to generate data that fits the training data distribution, while the discriminative model estimates the generative model's performance. [12]

Some studies use the term meta-learning. Meta-learning refers to learning with the purpose of being able to quickly adapt to novelties either on- or off-policy, in essence, learning to learn [13, p. VI]. On a technical level, meta-learning approaches vary, but the referenced studies use common neural network components [14]–[17].

Besides various types of neural networks, a support vector machine (SVM) is used in one reviewed study, citing its ability to learn with less samples than neural networks [18]. Support vector machines are a family of machine learning classification and regression methods [19] based on fitting a hyperplane that maximizes margin to data classes [20].

3. SUBTASKS IN LEARNED MANIPULATION

The range of robot arm manipulation includes various tasks such as pushing objects off the table [21], needle threading [22] and garment unfolding [18]. A large portion of tasks in literature can be categorized as assembly [23]–[28] or pick-and-place [14], [15], [17], [29]–[38] problems. These types of tasks typically include grasping one or more objects on the working surface and moving them to a desired location, sometimes in a specific angle or from a specific direction. The main difference of these categories is the amount and quality of contact between objects. While pick-and-place task often aim to place the held object into a bin or on the working surface, assembly tasks such as peg-in-hole involve more contact between the held object and its target and millimetre-level accuracy [24]–[28]. In case of multiple objects to manipulate, the order in which they are handled is often of importance. **Figure 2** shows a flowchart of a pick-and-place task, which also applies to many assembly tasks.

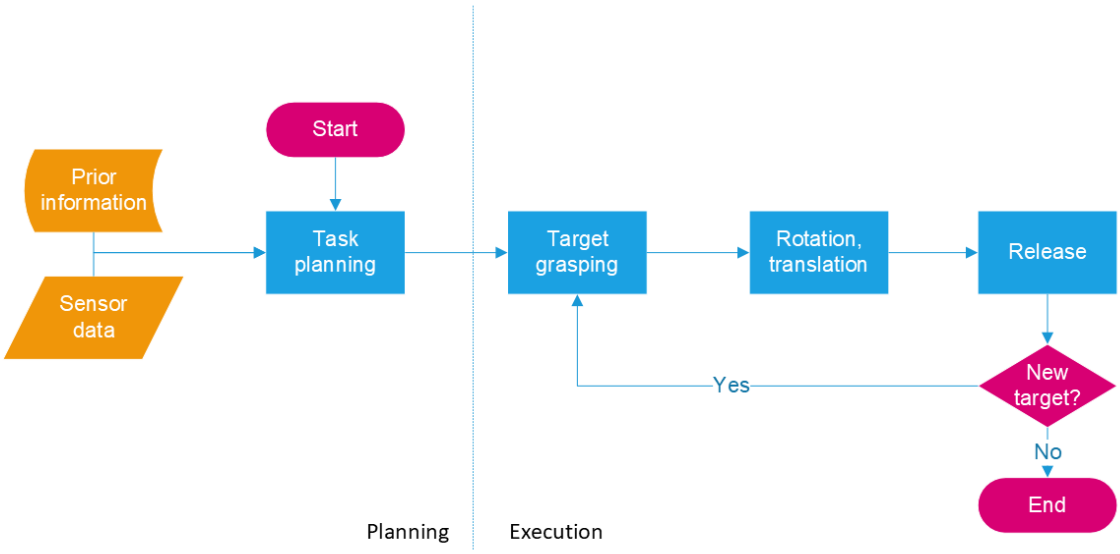

***Figure 2.** Simplified structure of a pick-and-place task.*

To plan a manipulation task, a robot should be aware of its working area. A common approach is depth-image observation, force, tactile, and other sensing modalities can be applied. The robot can also be given prior information about the objects included and the physics involved. Task planning varies between problems. It can include identifying target objects, obstacles, and target locations, as well as planning and optimizing the sequence and paths of actions to be taken. Finally, the execution phase of pick-and-place type problems includes the planned sequence of grasping, moving, and releasing the target objects.

In reality, the process is not always as simple as presented in **Figure 2**. For more robust and successful manipulation, failure recognition and recovery processes, as well as mid-execution observation and planning might be needed. Moreover, it is not directly applicable to other types of manipulation tasks. Nevertheless, the described structure serves as a background for the following discussion on the subtasks included.

3.1 Prior information and observations

The form and amount of information given a priori to the robotic manipulation system varies greatly between studies. In some cases, the entire system, including a physics model, the robot's initial state and properties as well as each object in the working area, can be provided [16], [21], [30], [33], [35], [37] In contrast, many other methods require no prior knowledge of the environment [31], [39]–[50]. While the robot arm's properties are readily available, accurately defining the properties of external objects may be infeasible outside simulation. It is also worth noting that the robot's function may change over time due to physical wear and tear or software modifications [4].

Table 1. *Sensory modalities in the reviewed articles. Papers with state data only are excluded.*

Task	Reference	Visual	Depth	Force / Torque	Tactile	External
6-D Scrabble, cap and peg assembly	[23]	x	x			
8 RLBench tasks	[34]	x	x			
ball throwing game	[51]	x	x			x
cup translation	[52]			x		
deformable object manipulation	[53]	x	x			
garment unfolding	[18]	x	x			
grasp synthesis	[40]	x	x			
grasp synthesis	[41]	x	x			
grasp synthesis	[50]	x	x			
grasp synthesis	[54]	x	x			
grasp synthesis, Ravens-10 benchmark tasks	[31]	x	x			
grasping from clutter, pick-and-place, bottle placement, insertion	[17]	x	x			
grasping, visual servoing	[45]	x		x		
grasp-synthesis and annotation	[46]	x				
imitation of various pick and place actions	[38]	x	x			
in-grasp sliding	[47]				x	
irregular object packing	[36]	x	x			
modular assembly	[26]				x	
motion control	[55]			x		
multi-stage manipulation	[32]	x	x			
needle threading, bolt picking	[22]	x				x
object detection and pose estimation	[56]	x				

object removal by pushing and grasping	[39]	x	x			
object retrieval from clutter	[33]	x	x			
object retrieval, geometry reconstruction	[42]			x		
object sorting	[14]	x	x			
object sorting	[15]	x	x			
object surface wiping	[48]	x		x	x	
obstacle avoidance	[57]		x			
obstacle avoidance, pick-and-place	[29]					x
page flipping	[11]	x			x	
peg-in-hole	[25]			x		
peg-in-hole	[27]	x	x			
peg-in-hole	[28]			x		
pose estimation, surgical tool tracking	[44]	x				
push, grasp, translation, obstacles	[30]	x	x			
pushing, edge following, surface following	[49]				x	
reach, push, reach-push, hammer, sweep, strike	[58]	x				
Soma cube assembly	[24]	x	x			
spade / hammer / scythe manipulation	[59]	x				
upright object placement	[43]	x	x	x		
wall-avoidance, table-sweep, shaft-reach, shaft-insertion	[60]	x				x

Table 1 above shows the sensory modalities used in the reviewed research papers. The popularity of visual and depth data is evident. Some task-dependency can also be seen in the table.

3.1.1 Visual and depth observations

To observe the scene before and during task execution, simulation environments can directly convey accurate state information [16], [21], [25], [26], [30], [33], [35], [59], [61], but in real world settings collecting sensor data is often essential. A common approach is to use an RGB-D sensor to capture pixelwise image and depth data [14], [15], [17], [18], [23], [24], [27], [31], [32], [38]–[41], [43], [50], [51], [53], [54]. In some applications only the depth image is sufficient [36], [57], and some methods do not need depth maps, but use ordinary RGB sensors instead [44]–[46], [48], [58], [59]. Two reviewed studies employ stereo-RGB cameras [22], [56]. Some motion capture systems, such as OptiTrack Flex 13 used in [60], work with grayscale image [62].

Cameras can be mounted on either a fixed position [14], [15], [17], [18], [22], [23], [30], [31], [34], [38]–[40], [44], [46], [48], [50], [51], [53], [57], [58], [60] or on the robot arm's end effector [24], [41], [45], [54], often referred to as eye-in-hand, or both [27], [36]. Fixed mounting points come with the advantage of a consistent view angle and global coordinates. However, occlusion by the robot arm or other objects in the scene is a difficult problem to solve if the camera cannot be moved [30], [48], [53]. Hand- or wrist-mounted cameras allow for active vision, i.e. changing view-angle and distance for more complete information [24], [41], [54].

Camera observation can be used to segment and identify objects in the scene. Segmentation is needed to distinguish objects from the table [24], [28], [41], [56], gripper [43] and each other [32], [56]. Visual data is also useful for tracking moving objects, or the robot arm. This can be done by following keypoints, either autonomously defined [44], or manually set to specific markers [11], [18], [31], [60].

Even though visual data has many uses and advantages, it has one key problem, occlusion. Opaque objects self-occlude the half facing away from the camera, making the shape partially unobservable. This issue can be mitigated by active vision [41] or algorithmic methods, such as filtering [24] or generative adversarial networks to hallucinate how the occluded area might look like [43]. The robot arm can occlude objects in the scene as well [30], [48], [53], and objects can occlude each other in cluttered scenes [5].

Another issue with visual observations is specular reflection, particularly on objects with a smooth and shiny surface. Jayasinghe et al. propose a deep convolutional GAN architecture to remove specular reflections, improving object detection and pose estimation. The model is trained on synthetic images of unicoloured objects and relies on colour-based object detection and segmentation, which limits real-world application. Furthermore, the generalizability of the model is not validated with unseen objects. [56]

3.1.2 Other modalities

In addition to visual and depth information, learning robot systems can utilize data provided by the robot itself. Given a robot arm's physical properties, its orientation and end effector position can be derived from joint angles. Moreover, force and torque measurements can be used to identify contact with the environment [28], [42], [43], [48] or a human [52], [55]. Grippers with tactile sensors can be useful for learning to estimate contact modes between the held object and the gripper [11], [26], [47]–[49]. For training purposes, teleoperation with electronic control signals is used in several studies [22], [23], [29], [60].

Notably, several reviewed studies draw inspiration from humans in their multi-modal approaches. Kim et al. use eye-tracker data to imitate human gaze focusing foveated vision to the task at hand, while learning from teleoperation when to transition from fast to slow approach [22]. Kar et al. combine visual and joint position data with EEG-signal, short for electroencephalography, measuring the brain's electrical activity, to transfer human throwing motion to robot domain, together with motor planning and error detection [51]. Similarly, Batzianoulis et al. propose detecting the expectation of error from EEG-signal in order to customize an obstacle-avoiding path to each person's preference [29]. Saito et al., inspired by human ability to integrate several sensory information, perform a robotic surface wiping task by capturing an initial view of the target object with RGB, but relying on kinematic and tactile data at contact [48].

Overall, it appears that many researchers have followed the multi-modal learning avenue, as listed in Chapter **2.1**, suggested by earlier reviews [3], [4]. The utilization of sensor data beyond vision and depth is still very task-specific, and there is undoubtedly room for further research.

3.2 Task planning

Task planning is essential on many levels of robotic manipulation, ranging from path planning to hierarchies and sequences of subtasks. For the purposes of this thesis, task planning is defined as planning a robot arm's actions beyond immediate control. This section focuses on the role of learning in higher level task planning with multiple phases, whereas planning subtasks, such as grasp synthesis and motion planning, are discussed in the following sections.

Evaluating task planning methods is complicated, and largely depends on the problem at hand. If a task planner is expected to solve a single task or task family on specific hardware and in controlled environment, simple execution metrics such as training, planning, and execution times along with success rate may suffice. However, if the planner is expected to generalize or adapt to new tasks and environments, qualities such as explainability, modularity, and adaptability should be considered.

As the reviewed task planning methods are aimed at different tasks, and different evaluation metrics are prioritized in their design, direct comparison between them is unfeasible. Instead, each approach is described separately, with a purpose of finding common features and ideas that are beneficial for a variety of learned task planning methods.

It is worth noting that in a well-defined and controlled environment, cleverly designed task planning algorithms can be fast and successful without machine learning [24]. However, learning enables planning with less prior information, better generalization, and robustness.

One fully explainable methodology proposed in the studies reviewed for this thesis is the Geometric Task Network (GTN). It is essentially a network of primitive subtasks, from which the most optimal sequence to complete an overall task can be searched. The subtasks themselves, along with their geometric constraints are learned off-policy from exhaustive planners or expert demonstration. The network is limited to the pre-learned skills and structure, and therefore cannot generalize beyond them without additional training, or adapt to unforeseen disturbances such as physical contact during execution. The authors add that adding a level of hierarchy by grouping the learned skills into modules would be beneficial for learning new GTNs. [23]

Chou et al. propose a method where the overall task is expressed as a linear temporal logic formula. The logical structure is derived from demonstrations in terms of atomic propositions and optimized. The learned logic is independent of the environment, and therefore transferable. The model algorithmically generates suboptimal counterexamples for learning purposes, but they also improve explainability. While the proposed method is able to successfully solve multi-step manipulation tasks, computation times, ranging from 5 to 90 minutes, may be too long for some applications. [32]

Khodeir et al. propose improving the efficiency of policy search by expanding the search in prioritized order. The priority of objects and facts is based on a relevance score learned from earlier planning experience. The method manages to solve various block arrangement tasks with a higher success rate than its baseline, although for some tasks as low as 54 %, within a 90 second time limit. The model operates directly on state data, and its real-world application is limited. [37]

To allow a robot to explore actions while performing a task, multiple approaches based on reinforcement learning have been proposed. While reinforcement learning is adaptable by nature, it does not provide an explanation to decisions made. Moreover, the search space tends to grow with task complexity and horizon, resulting in computational problems and lengthy training [5]. Perhaps the simplest approach to alleviate these issues is altering the rewards to guide exploration towards more efficient learning. Tao et al. propose a reward hierarchy that autonomously adapts to task phases, but leave the rewards themselves and the task phase boundaries to be heuristically defined [21].

To limit search space, reducing dimensionality can be considered. Cheong et al. arranged several objects in discretized grids of 18 to 64 cells, and were able to solve a complex rearrangement task within minutes with their proposed deep Q-network [33]. However, operating on a strictly defined coarse grid may not be feasible in other applications. James and Davison deploy a deep Q-learning based attention module to crop

RGB and point cloud input to most relevant locations, which guides the search and improves efficiency, and demonstrate their method on a variety of manipulation tasks [34].

The Q-attention module is trained on demonstrations from the simulation benchmark RLBench, contributed to by the same authors [34], [63]. Other works have utilized demonstrations to guide task planning in different ways. Zuo et al. propose a Gaussian-based graph motion planner able to reduce computation time for repetitive pick-and-place tasks by adaptive sampling from optimal planner demonstrations [35]. Two works propose domain-adaptive meta-learning schemes, designed to adapt to new object sorting tasks with few demonstrations [14], [15]. The latter maps image space into grid space, improving the efficiency of task planning [15].

Several methods use hindsight experience to improve sample efficiency. You et al. propose a deep actor-critic framework that prioritizes experience replays by temporal difference, for pushing and grasping strategy, achieving impressive results in simulation, but with limited real-world applicability [39]. With a similar idea, Beyene and Han apply prioritized experience replay, separated into real and hindsight trajectory buffers, to several RLBench manipulation tasks, managing to speed up task-to-task adaptation [16].

Gieselmann and Pokorny propose a hierarchical actor-critic approach segmenting the task MDP into a graph-searchable hierarchy of learned short-horizon MDPs. Hindsight experience replay, HER for short, is used to improve sample efficiency. Their method achieves promising results in simulation but trains relatively slowly, with interaction steps counted in millions and time in hours. Most failures in their experiments are accounted to the agent getting stuck trying to reach subgoals, possibly due to misidentified state equivalence in close distance. [64]

In another approach combining graph-search with RL and hindsight experience, Bing et al. propose Graph-Curriculum-Guided Hindsight Goal Generation. Their algorithm constructs intermediate tasks from graph representation and samples hindsight goals from a replay buffer based on diversity and proximity metrics. As learning progresses, these metrics are used to adaptively shift from curious exploration to proximal intermediate goal exploitation. The method outperforms some other hindsight-algorithms, but training times are around 10 hours for relatively simple tasks. [30]

Some studies address the complexity of task planning with hand-designed modularity. Huang et al. propose a hierarchical manager-worker Q-learning framework for irregular object packing, consisting of two convolutional neural networks. The manager network learns to plan the packing sequence, while the worker is trained to predict optimal poses for each object. While the method outperforms its baseline especially in harder cases, it utilizes 6 principal view heightmaps of each object and a top view heightmap of the packing box, resulting in relatively high computational cost and extensive prior information requirements. [36]

One useful module for manipulation tasks is an error recovery protocol. Most studies understandably focus on maximising success on first attempt, or gradually improving performance over repeated executions. Kim et al. propose an error recovery module, learning a recovery action network, a recovery classifier, and a recovery step predictor from demonstrations, managing to improve the initial success rate of a needle threading task significantly [22]. Guo and Bürger propose a multi-step error recovery module including skill model assessment, progress re-evaluation, and identifying unrecoverable states, where the system has to be reset [23].

Overall, recent research shows trends towards reinforcement learning, learning from demonstrations, experience replay, modularity, and other structural approaches. In addition, operating on a gridded or cropped space can significantly improve efficiency where applicable.

3.3 Grasping

Finding a successful grasp is an essential prerequisite to many manipulation tasks. Hence, it is no surprise that learning grasp synthesis is a thoroughly studied subtask of robotic manipulation. In [5], over half of the reviewed articles involved grasping, showcasing that deep reinforcement learning is a key methodology extensively studied for grasping purposes. Several studies reviewed in this thesis utilize RL algorithms for grasping [30], [34], [39]–[41]. Other neural network based grasping algorithms have been recently studied as well [17], [31], [45], [46], [50], [54].

Reinforcement learning methods have the advantage of being able to train without the need for preannotated data, although they can pretrain feature extraction on labelled image data such as ImageNet [40]. Similarly, off-policy learning can benefit from pretrained feature extraction [45]. In addition, annotated training, testing, and validation data for the grasp synthesis itself is needed. The dataset can either be self-collected [17] or publicly available.

Multiple annotated image datasets for grasping exist for training, testing, validation, and comparison purposes. Kumra et al. discuss these datasets in detail [31]. The most popular grasping image dataset in studies referenced in this thesis, and overall for benchmarking [31], is the Cornell Grasp Dataset [65], later extended [66], used in [31], [45], [46], [50], [54]. Newer and larger datasets OCID grasp [67], GraspNet-1Billion [68], and

Jacquard [69] are also available and in use [31], [50]. Table 2 presents grasping success accuracies on the Cornell Grasp Dataset (CGD) as reported in the referenced papers, all of which assess grasping success with the same metrics.

Table 2. *Cornell Grasp Dataset accuracies in the reviewed studies*

Authors	Reference	CGD accuracy (%)
Le and Lien	[46]	90.5
Ribeiro et al.	[45]	94.8
Cao et al.	[50]	97.8
Kumra et al.	[31]	98.8

Some studies only consider top-down grasps, where the object is viewed and grasped from above, and have achieved high success rates and low inference times in some circumstances [17], [31], [45], [50]. However, some objects might be difficult to grasp from above, so active vision methods have been developed to view and enable grasping from multiple angles [24], [41], [54]. Natarajan et al. even published a benchmark for active vision [41]. Their research found that heuristic methods are equally or more successful than proposed data-driven methods in some active vision grasping scenarios [41].

Grasping can be performed in scenes with a singular object. Scenes with multiple objects either separated or in clutter, making the task more complicated, have been considered in research as well. Multi-object grasping methods are often limited to scenes where objects are arranged so that they are graspable and without much occlusion [17], [31], [50], [54]. You et al. propose push and grasp actions trained on an actor-critic framework with prioritized experience replay, achieving promising simulation results, with limitations in occlusion, generalization and robustness [39]. Cheong et al. propose a deep Q-learning approach to rearrange objects away from the path to the target object, but their method is limited to a discretized grid-space [33]. For a more extensive review on deep RL methods for grasping in cluttered scenes, refer to [5].

3.4 Translation, rotation, and release

In many tasks and environments motion planning can rely on inverse kinematics [24], [31], [50], [54]. However, there are many conditions that require more sophisticated methods. These conditions include risk of collision, preferred trajectories, deformable objects, and contact-rich manipulation.

Collisions can lead to failed task executions and damage robots, objects or even humans in the working area. Hence, planning movement around obstacles has been studied. Tao et al. propose an RL approach that addresses collisions by penalizing them in the reward function [21]. Bing et al. add collision tolerance to their grid-space trajectory graph planner by virtually increasing the size of obstacles with a safety region [30]. Ando et al. utilize a conditional GAN to generate a collision-free latent space to joint space mapping, where the trajectory can be planned with any path planner, then checked and adjusted at remaining collision points [57]. Batzianoulis et al. consider user's personal preference in by customizing obstacle avoiding paths to the comfort of each user [29]. While some approaches only plan the path of the robot's end effector, it is important to note that different links of the robot arm can collide with obstacles as well. Malik et al. propose a swarm intelligence based approach to overcome this issue, while the end effector follows a predefined path [70].

Besides obstacles, one might want to avoid certain regions for other reasons. For example, learning to move a coffee cup around a laptop rather than above it has been discussed in literature [52], [61]. Bobu et al. propose an inverse reinforcement learning approach, where human users first teach robots features, such as avoiding the space above a laptop, from which task-specific continuous rewards are constructed based on few new demonstrations [61]. Contrastingly, Losey et al. propose an IRL approach that does not require prior demonstrations but approximates reward, from predefined features, based on physical human-robot interaction during execution [52].

When manipulating deformable objects, trajectory planning may require a certain shape instead of a straight line. Triantafyllou et al. propose a method for garment unfolding, learning a trajectory from human demonstrations and adjusting it during execution based on SVM-classification of remaining wrinkles [18]. Similarly, Zheng et al. propose a reinforcement learning method for page flipping, that learns to avoid abrupt changes in tactile sensor data due to warping, resulting in roughly semi-circular trajectories [11].

In addition to the path, movement speed should be considered in some applications. For instance, when throwing a ball the end effectors velocity at release defines the ball´s trajectory and thus success or failure [51]. For a grass cutting task, the speed at which the scythe blade intersects grass must be considered [59].

Sometimes movement cannot be pre-planned but should rather be reactive to the environment. For example, visual servoing, meaning following an object in the scene and moving accordingly, such as moving towards the object for dynamic grasping, may be needed in dynamic environments [45], [54].

Contact-rich manipulation, often encountered in assembly tasks is another area where kinematic solvers might fail. Yang et al. propose an RL-framework that learns skill priors from demonstrations' position- and impedance-spaces to perform contact-rich peg-in-hole tasks [28]. Belousov et al. utilize tactile sensor data in their proposed RL framework to tackle difficulties from contact [26]. For a review on reinforcement learning for contact-rich manipulation, refer to [6].

Two recent studies address stability in learned robot manipulation. Figueroa and Billard propose a locally active globally stable dynamical systems approach, learning to actively return to the preferred trajectory after perturbation, while maintaining the global convergence of dynamical systems [55]. Khader et al. propose a model-free deep RL energy shaping policy that guarantees stability, even in contact-rich tasks, with a fully connected damping network [25].

Although for many pick-and-place tasks the release step is as trivial as dropping the object at or slightly above target location, some applications require more careful consideration before release. In assembly tasks, the target often must be approached at a specific angle and release can only be executed at target. This can be achieved by penalizing object movement after release [26], for example. Newbury et al. consider the upright placement of flat-based objects such as mugs and bottles, and propose an actor-critic like CNN for rotation and its assessment followed by utilizing force-sensor data to identify surface contact [43].

4. SIMULATION

Simulation environments are popular in learned robot manipulation research. The majority of studies reviewed in this thesis utilize simulation, and some of them rely solely on simulation experiments [16], [34], [39], [64]. **Figure 3** shows quantitative analysis of the usage of simulation in the reviewed research papers.

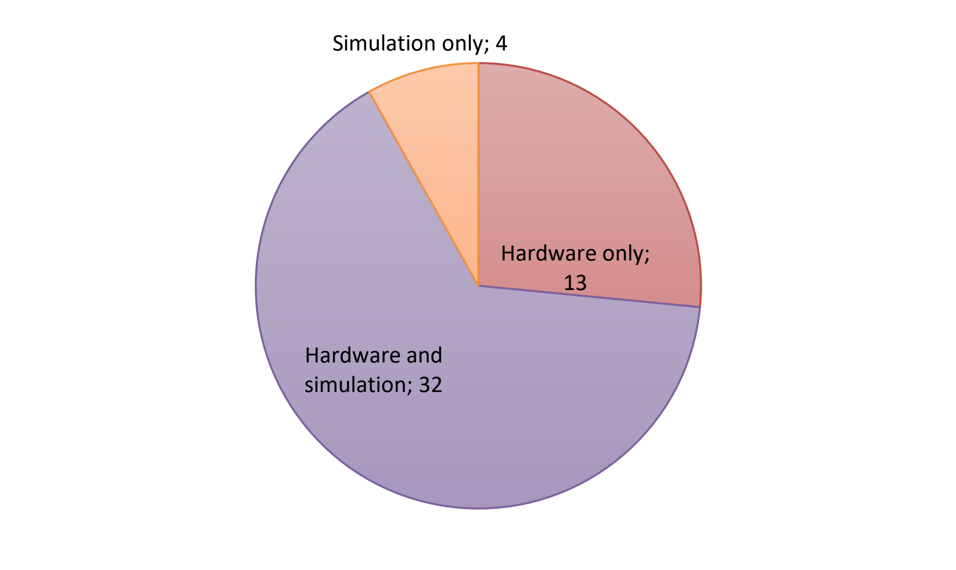

Figure 3. *Usage of simulation in the reviewed studies*

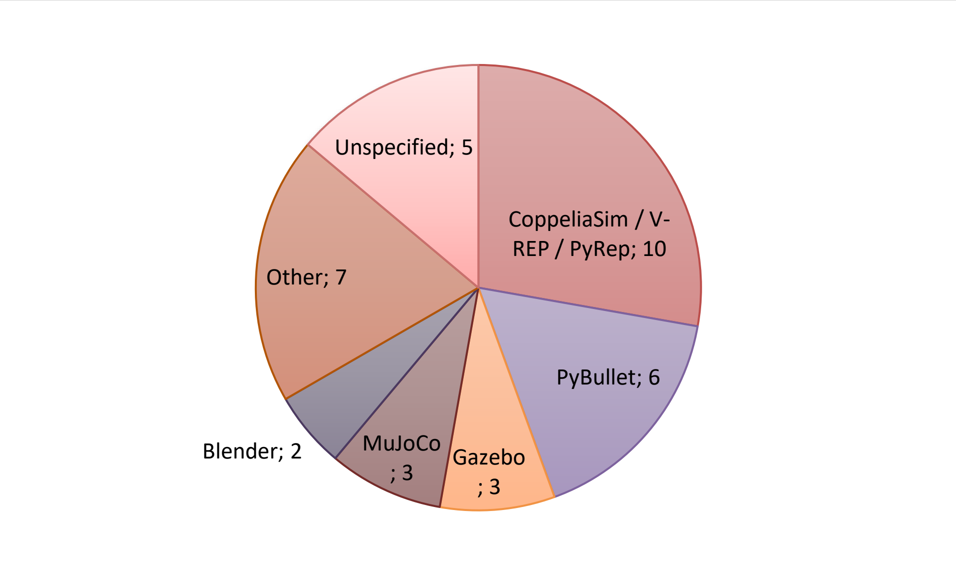

Figure 4. *Simulators used in the reviewed studies*

Figure 4 above shows that in the reviewed research papers, CoppeliaSim, formerly known as V-REP [71], and its Python-toolkit PyRep [72], is the most popular simulation environment [16], [21], [27], [33], [34], [38]–[40], [43], [44], followed PyBullet [73] [26],

[31], [36], [42], [61], [64]. Gazebo [74] gained 3 mentions [18], [41], [57] and MuJoCo [75] was equally popular [25], [30], [58]. The 3D graphics environment Blender was used in two studies [18], [56]. Other simulation platforms were used in only one of the reviewed studies each. While this is not a comprehensive analysis of simulation environments for robotic manipulation, it provides some indication of their popularity.

The 4 most popular platforms have been actively developed, with frequent new releases [76]–[79]. It seems that simulation environments have advanced, as Cui and Trinkle requested [4], but they are still not perfect. Belousov et al. found they could not accurately simulate their chosen optical tactile sensor, DIGIT, in PyBullet and had to resort to training contact identification on hardware [26]. After their study was published, the Tactile Gym simulator extended to version 2.0, simulating the DIGIT sensor among others [49], which showcases the importance of simulator development. The choice of simulator can be task dependent. Chen et al. chose to simulate deformable object dynamics in PlasticineLab [80] due to its differentiable Material Point Method capabilities [53]. Chou et al. decided that grasp simulation was not necessary for their study, so they simulated on Unity VR and allowed the simulated gripper to simply attach to objects at close distance [32].

The reasons behind using simulation for learning robotic manipulation are clear. Training is simpler and faster in simulation, as the scene can be automatically reset and randomized over parameter distributions. Moreover, robot arms can be costly, but many simulation environments are free. A physical robot arm's collision might lead to damage to the robot, environment, or a human nearby, while simulated collisions are safe. Mohammed et al. recommend in their review to utilize simulation to improve comparability between different methods [5].

Several benchmarks and datasets have been created for this purpose. One of them was mentioned in **Chapter 3.2**. RLBench, used in 2 reviewed studies [16], [34], consists of 100 manipulation tasks built on CoppeliaSim and PyRep [63]. Another manipulation task benchmark, Ravens-10, built on PyBullet [81], was used by Kumra et al. for evaluation and comparison to other methods [31]. The PlasticineLab simulator also includes benchmark tasks with reference solutions [80]. Natarajan et al. open-sourced their active vision and grasping environment, 2021ActiveVision, for benchmarking purposes [41]. Real-world benchmarks, such as the Siemens Robot learning challenge, used in [60], or the grasp benchmarking protocol [82], used in [41], exist as well, but simulation benchmarks are more accessible and variable.

Object model datasets provide some degree of comparability, although less strictly than task-specific benchmarks. The Yale-CMU-Berkeley (YCB) object model set [83] is used in several studies [31], [32], [36], [41], one of which also employs object models from the OCRTOC [84] dataset [36]. Image datasets for grasping are discussed in **Chapter 3.3**.

While simulation has many advantages, it may not always be necessary, as demonstrated by the 13 reviewed studies with only real-world deployment. Even the studies carried out in simulation only should be aimed to finally benefit real-world manipulation. The transfer from simulation to real-world (sim2real) is not always simple. Differences in the environment and dynamics might decrease success rate [85]. Moreover, hardware sensors can be noisy and inaccurate, which can also undermine performance [31], [41], [70]. Filtering can reduce the noise and unnecessary background data [43], but noise can also be induced in simulation to aid sim2real [30], [44].

Yang et al. sidestep the reality gap and improve learning efficiency by abstracting the input of their deep Q-network to manipulation affordance masks instead of raw image data. Their method only learns to predict manipulation success in simulation, and the complete system is implemented on a real robot arm. [38]

Chebotar et al. propose a more sophisticated sim2real method, that adapts simulation randomization by alternating between simulation and real-world rollouts, and achieve promising results on complex dynamic systems [85]. The study was published in 2019, and simulated on NVIDIA Flex [85], and it is unclear if state-of-the-art simulators would need such methods for similar results. Lu et al. use a more diverse domain randomization setting, randomizing joint angles, virtual camera poses, lighting, distractor objects, background images, and robot mesh colour, as well as adding white noise, but without adapting on real-world rollouts [44]. Tactile Gym 2.0 provides sim-to-real transfer for optical tactile sensors, by learning translation models for sensor images with a generative adversarial network [49].

5. HUMAN INTERACTION AND LEARNING FROM DEMONSTRATION

Human-robot interaction takes many forms. Collaborative and assistive robots have been a talking point for years. If humans are to enter robot's working area, the robot's action at contact should be carefully designed to ensure safety. The reviewed studies mostly consider physical human-robot contact as means of guidance [52] or demonstration [23], [61], or as perturbance [55]. Robots can react to human contact with full compliance, only compensating for gravity [61], or some level of impedance, acting as a spring-damper system [52], [55]. Behaviour after releasing contact may differ depending on task and the interpretation of human intent. Figueroa and Billard propose a globally stable locally active control scheme that quickly converges to desired path after perturbation, such as human contact [55]. Losey et al. treat human contact as correction, and propose a method for learning a trajectory more desirable for the user [52].

Gravity compensation is useful for kinesthetic teaching, where the human user demonstrates the desired trajectory directly on the robot arm [61]. There are other ways for learning from demonstration, LfD for short, as well. Similarly to kinesthetic teaching, robot arm trajectories can be directly recorded and used for training from teleoperation [22], [60]. Video demonstrations or various types are also common [14], [15], [18], [38], [58], [59]. In simulation, demonstration traits can be selected with a mouse [23] or recorded from keyboard controlled trajectories [27].

There are different ways of utilizing demonstrations for learning manipulation. Perhaps most direct type is behaviour cloning, where the robot aims to reproduce the demonstrated actions accurately. Behaviour cloning from video demonstrations is used as the initial policy of an RL algorithm proposed by Zorina et al [59]. Similarly, Zheng et al. propose fitting initial policy movement parameters to demonstration, and then further optimizing the trajectory with RL [11]. Triantafylloy et al. propose a slightly more elaborate method where the shape of the garment unfolding trajectory is extracted from demonstrations, scaled to fit each fold, and adjusted according to observations during execution [18].

Imitation learning can take less direct forms as well. Kim et al. propose a deep imitation learning policy that imitates human action speed and foveated vision from demonstrations, but not the actual trajectory [22]. Yang et al. divide imitation learning into captioning and manipulation modules. The captioning module employs a recurrent neural network to produce textual commands from visual change maps that are extracted from video demonstrations. The manipulation model predicts manipulation affordances of segmented objects and executes the manipulation actions according to the captioned command. [38]

Oh et al. improve the robustness and flexibility of imitation learning by injecting state-dependent Bayesian disturbance into expert demonstrations and allowing their system to learn multiple policies. While they achieve impressive results on multiple tasks, the key problems with imitation learning remain. New tasks and environments, or even small changes require new demonstrations, environmental uncertainty is unaddressed, and performance is dependent on the quality of demonstrations. [60]

Demonstrations can be utilized in their full length or segmented in various ways. Kar et al. propose converting demonstrations to individual state-action pairs and respective reward values to guide reward learning [51]. For more structured approach, demonstrations can be segmented to primitive skills [23].

Another method deriving structure from demonstrations is proposed by Chou et al. Given the initial configurations of all relevant objects, their algorithm learns a temporal logic structure and its atomic proposition parameters from demonstrations, and optimizes actions accordingly. [32]

Demonstrations can also be used to form reward functions for reinforcement learning in different ways. Pauly et al. extract action vectors from demonstrations, and negative distance to them is used as a reward function [58]. Bobu et al. use demonstrated features as the basis for their reward function [61]. Wang et al. propose an Information Utilization Mechanism, where an inexperienced robot initially relies heavily on demonstration data, but gradually transitions to utilizing more environmental data [27].

An important notion is that the demonstrations for teaching a robot do not have to be human produced. Two referenced meta-learning methods were also demonstrated capable of learning from other robots, enabling policy transfer across robot embodiments. These Domain-Adaptive Meta-Learning methods are designed to adapt to unseen environments with a single new demonstration. [14], [15]

Besides various ways of demonstration, humans can communicate with robots via EEG-signals. Error-related potentials, ErrP for short, have been demonstrated to indicate anticipated collision [29] or failure [51]. Robots can learn accordingly to avoid these situations.

6. CONCLUSIONS

While significant advances in learned robot manipulation have been recently achieved, it is still not as complete as one might wish. Learning methods have shown gradual improvements, but they are still largely task-specific, with varying capability to adapt and generalize. Simple tasks such as grasping singular objects have been solved with high success rates and fast execution, but with growing task and scene complexity failures tend to become more common and computation times drastically increase.

However, looking at the list in **Chapter 2.1**, each challenge outlined by previous reviews has been addressed to some extent. Multi-modal sensing has been incorporated into learning in various ways, often inspired by human perception. Real-time performance has been achieved at least on subtask-level, such as grasp synthesis. Improvements in reinforcement learning methods, such as informed exploration strategies and utilizing experience replay, have led to increased success rates and efficiency, and enabled continuous learning for many tasks. Adaptation and generalization have been addressed with meta-learning schemes and structural learning approaches. Stability and safety of learning robots has gained some attention. Simulator development has demonstrably benefitted manipulation research.

The approaches reviewed in this thesis vary greatly on principle and technical level, even with similar tasks. Hence, defining state-of-the-art methods is difficult, if not impossible, without strictly defining the task, for example to an image or simulation benchmark, evaluation metrics, and maybe even methodology type. Besides benchmarking, literature reviews with a narrower focus on a task family and methodology, such as [5], [6], can provide meaningful comparisons and explore technical details more thoroughly. As researchers keep developing the field, more reviews with a well-defined focus would be useful for summarizing the developments and pointing directions for the future. For the purposes of research, more important than asking what the best method is, is finding areas where significant improvements can be made.

One area that has gained surprisingly little attention is safety and stability. Especially when interacting with humans, safety guarantees should be imposed as a best-practice in future research. Another future challenge is the integration of learning frameworks into more complete and general manipulation systems. Most of the research is understandably task-specific, but modularization of the acquired skills to form extendable skill hierarchies still appears to be in early stages. The complexity of more complete systems is a key challenge to their development, in which further improving efficiency and increasing computational power might help in the future.

REFERENCES

[1] M. T. Mason, ‘Toward Robotic Manipulation’, *Annu. Rev. Control Robot. Auton. Syst.*, vol. 1, no. 1, pp. 1–28, May 2018, doi: 10.1146/annurev-control-060117-104848.

[2] A. Billard and D. Kragic, ‘Trends and challenges in robot manipulation’, *Science (American Association for the Advancement of Science)*, vol. 364, no. 6446, pp. 1149-, 2019, doi: 10.1126/science.aat8414.

[3] O. Kroemer, S. Niekum, and G. Konidaris, ‘A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms’. arXiv, Nov. 06, 2020. Accessed: Oct. 31, 2022. [Online]. Available: http://arxiv.org/abs/1907.03146

[4] J. Cui and J. Trinkle, ‘Toward next-generation learned robot manipulation’, *Sci. Robot.*, vol. 6, no. 54, p. eabd9461, May 2021, doi: 10.1126/scirobotics.abd9461.

[5] M. Q. Mohammed *et al.*, ‘Review of Learning-Based Robotic Manipulation in Cluttered Environments’, *Sensors (Basel, Switzerland)*, vol. 22, no. 20, 2022, doi: 10.3390/s22207938.

[6] Í. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, I. Inziarte-Hidalgo, S. Bøgh, and N. Arana-Arexolaleiba, ‘A review on reinforcement learning for contact-rich robotic manipulation tasks’, *Robotics and Computer-Integrated Manufacturing*, vol. 81, p. 102517, Jun. 2023, doi: 10.1016/j.rcim.2022.102517.

[7] M. Gopal, ‘TOWARDS COGNITIVE MACHINE’, in *Applied Machine Learning*, 1st Edition.New York: McGraw-Hill Education, 2019. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9781260456844/toc-chapter/chapter5/section/section2

[8] Mohit Sewak, Md. Rezaul Karim, and Pradeep Pujari, *Practical Convolutional Neural Networks : Implement Advanced Deep Learning Models Using Python*. Birmingham: Packt Publishing, 2018. Accessed: Apr. 05, 2023. [Online]. Available: http://libproxy.tuni.fi/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,ip,uid&db=e000xww&AN=1728047&site=ehost-live&scope=site

[9] M. Gopal, ‘NETWORK ARCHITECTURES’, in *Applied Machine Learning*, 1st Edition.New York: McGraw-Hill Education, 2019. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9781260456844/toc-chapter/chapter5/section/section11

[10] M. Gopal, ‘Appendix B: REINFORCEMENT LEARNING (RL)’, 1st Edition.New York: McGraw-Hill Education, 2019. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9781260456844/back-matter/appendix2

[11] Y. Zheng, F. F. Veiga, J. Peters, and V. J. Santos, ‘Autonomous Learning of Page Flipping Movements via Tactile Feedback’, *IEEE transactions on robotics*, pp. 1–16, 2022, doi: 10.1109/TRO.2022.3168731.

[12] I. J. Goodfellow *et al.*, 'Generative Adversarial Networks'. arXiv, Jun. 10, 2014. doi: 10.48550/arXiv.1406.2661.

[13] N. Jankowski, W. Duch, and K. Grąbczewski, Eds., *Meta-Learning in Computational Intelligence*, vol. 358. in Studies in Computational Intelligence, vol. 358. Berlin, Heidelberg: Springer, 2011. doi: 10.1007/978-3-642-20980-2.

[14] Z. Hu *et al.*, 'Learning With Dual Demonstration Domains: Random Domain-Adaptive Meta-Learning', *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 3523–3530, 2022, doi: 10.1109/LRA.2022.3145088.

[15] Z. Hu, Z. Gan, W. Li, W. Guo, X. Gao, and J. Zhu, 'Learning From Demonstrations Via Multi-Level and Multi-Attention Domain-Adaptive Meta-Learning', *IEEE robotics and automation letters*, vol. 7, no. 4, pp. 11910-, 2022, doi: 10.1109/LRA.2022.3207558.

[16] S. W. Beyene and J.-H. Han, 'Prioritized Hindsight with Dual Buffer for Meta-Reinforcement Learning', *Electronics*, vol. 11, no. 24, p. 4192, 2022, doi: 10.3390/electronics11244192.

[17] W. Guo, W. Li, Z. Hu, and Z. Gan, 'Few-Shot Instance Grasping of Novel Objects in Clutter', *IEEE robotics and automation letters*, vol. 7, no. 3, pp. 6566–6573, 2022, doi: 10.1109/LRA.2022.3174648.

[18] D. Triantafyllou, P. Koustoumpardis, and N. Aspragathos, 'Type and Fabric Agnostic Methods for Robotic Unfolding of Folded Garments', *Journal of intelligent & robotic systems*, vol. 105, no. 3, 2022, doi: 10.1007/s10846-022-01641-0.

[19] M. Gopal, 'INTRODUCTION', in *Applied Machine Learning*, 1st Edition.New York: McGraw-Hill Education, 2019. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9781260456844/toc-chapter/chapter4/section/section2

[20] M. Gopal, 'LINEAR MAXIMAL MARGIN CLASSIFIER FOR LINEARLY SEPARABLE DATA', in *Applied Machine Learning*, 1st Edition.New York: McGraw-Hill Education, 2019. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9781260456844/toc-chapter/chapter4/section/section6

[21] L. Tao, J. Zhang, and X. Zhang, 'Multi-Phase Multi-Objective Dexterous Manipulation with Adaptive Hierarchical Curriculum', *Journal of intelligent & robotic systems*, vol. 106, no. 1, 2022, doi: 10.1007/s10846-022-01680-7.

[22] H. Kim, Y. Ohmura, and Y. Kuniyoshi, 'Gaze-Based Dual Resolution Deep Imitation Learning for High-Precision Dexterous Robot Manipulation', *IEEE robotics and automation letters*, vol. 6, no. 2, pp. 1630–1637, 2021, doi: 10.1109/LRA.2021.3059619.

[23] M. Guo and M. Bürger, 'Geometric Task Networks: Learning Efficient and Explainable Skill Coordination for Object Manipulation', *IEEE transactions on robotics*, vol. 38, no. 3, pp. 1723–1734, 2022, doi: 10.1109/TRO.2021.3111481.

[24] T. Kang, J.-B. Yi, D. Song, and S.-J. Yi, 'High-Speed Autonomous Robotic Assembly Using In-Hand Manipulation and Re-Grasping', *Applied sciences*, vol. 11, no. 1, pp. 37-, 2020, doi: 10.3390/app11010037.

[25] S. A. Khader, H. Yin, P. Falco, and D. Kragic, ‘Learning Deep Energy Shaping Policies for Stability-Guaranteed Manipulation’, *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8583–8590, 2021, doi: 10.1109/LRA.2021.3111962.

[26] B. Belousov *et al.*, ‘Robotic architectural assembly with tactile skills: Simulation and optimization’, *Automation in construction*, vol. 133, pp. 104006-, 2022, doi: 10.1016/j.autcon.2021.104006.

[27] F. Wang, B. Cui, Y. Liu, and B. Ren, ‘Deep Reinforcement Learning for Peg-in-hole Assembly Task Via Information Utilization Method’, *Journal of intelligent & robotic systems*, vol. 106, no. 1, 2022, doi: 10.1007/s10846-022-01713-1.

[28] Q. Yang, A. Durr, E. A. Topp, J. A. Stork, and T. Stoyanov, ‘Variable Impedance Skill Learning for Contact-Rich Manipulation’, *IEEE robotics and automation letters*, vol. 7, no. 3, pp. 8391–8398, 2022, doi: 10.1109/LRA.2022.3187276.

[29] I. Batzianoulis *et al.*, ‘Customizing skills for assistive robotic manipulators, an inverse reinforcement learning approach with error-related potentials’, *Commun Biol*, vol. 4, no. 1, p. 1406, Dec. 2021, doi: 10.1038/s42003-021-02891-8.

[30] Z. Bing *et al.*, ‘Solving Robotic Manipulation With Sparse Reward Reinforcement Learning Via Graph-Based Diversity and Proximity’, *IEEE Transactions on Industrial Electronics*, vol. 70, no. 3, pp. 2759–2769, Mar. 2023, doi: 10.1109/TIE.2022.3172754.

[31] S. Kumra, S. Joshi, and F. Sahin, ‘GR-ConvNet v2: A Real-Time Multi-Grasp Detection Network for Robotic Grasping’, *Sensors (Basel, Switzerland)*, vol. 22, no. 16, pp. 6208-, 2022, doi: 10.3390/s22166208.

[32] G. Chou, N. Ozay, and D. Berenson, ‘Learning temporal logic formulas from suboptimal demonstrations: theory and experiments’, *Autonomous robots*, vol. 46, no. 1, pp. 149–174, 2022, doi: 10.1007/s10514-021-10004-x.

[33] S. Cheong *et al.*, ‘Obstacle rearrangement for robotic manipulation in clutter using a deep Q-network’, *Intelligent service robotics*, vol. 14, no. 4, pp. 549–561, 2021, doi: 10.1007/s11370-021-00377-4.

[34] S. James and A. J. Davison, ‘Q-Attention: Enabling Efficient Learning for Vision-Based Robotic Manipulation’, *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 1612–1619, 2022, doi: 10.1109/LRA.2022.3140817.

[35] G. Zuo *et al.*, ‘An Efficient Motion Planning Method with a Lazy Demonstration Graph for Repetitive Pick-and-Place’, *Biomimetics*, vol. 7, no. 4, p. 210, 2022, doi: 10.3390/biomimetics7040210.

[36] S. Huang, Z. Wang, J. Zhou, and J. Lu, ‘Planning Irregular Object Packing via Hierarchical Reinforcement Learning’, *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 81–88, Jan. 2023, doi: 10.1109/LRA.2022.3222996.

[37] M. Khodeir, B. Agro, and F. Shkurti, ‘Learning to Search in Task and Motion Planning with Streams’, *IEEE Robot. Autom. Lett.*, vol. 8, no. 4, pp. 1983–1990, Apr. 2023, doi: 10.1109/LRA.2023.3242201.

[38] S. Yang, W. Zhang, R. Song, J. Cheng, H. Wang, and Y. Li, ‘Watch and Act: Learning Robotic Manipulation From Visual Demonstration’, *IEEE Transactions*

on Systems, Man, and Cybernetics: Systems, pp. 1–13, 2023, doi: 10.1109/TSMC.2023.3248324.

[39] T. You, H. Wu, X. Xu, P. B. Petrovic, and A. Rodić, ‘A Proposed Priority Pushing and Grasping Strategy Based on an Improved Actor-Critic Algorithm’, *Electronics*, vol. 11, no. 13, p. 2065, 2022, doi: 10.3390/electronics11132065.

[40] A. Al-Shanoon, H. Lang, Y. Wang, Y. Zhang, and W. Hong, ‘Learn to grasp unknown objects in robotic manipulation’, *Intelligent service robotics*, vol. 14, no. 4, pp. 571–582, 2021, doi: 10.1007/s11370-021-00380-9.

[41] S. Natarajan, G. Brown, and B. Calli, ‘Aiding Grasp Synthesis for Novel Objects Using Heuristic-Based and Data-Driven Active Vision Methods’, *Frontiers in robotics and AI*, vol. 8, pp. 696587–696587, 2021, doi: 10.3389/frobt.2021.696587.

[42] J. Bimbo, A. S. Morgan, and A. M. Dollar, ‘Force-Based Simultaneous Mapping and Object Reconstruction for Robotic Manipulation’, *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 4749–4756, 2022, doi: 10.1109/LRA.2022.3152244.

[43] R. Newbury, K. He, A. Cosgun, and T. Drummond, ‘Learning to Place Objects Onto Flat Surfaces in Upright Orientations’, *IEEE robotics and automation letters*, vol. 6, no. 3, pp. 4377–4384, 2021, doi: 10.1109/LRA.2021.3068122.

[44] J. Lu, F. Richter, and M. C. Yip, ‘Pose Estimation for Robot Manipulators via Keypoint Optimization and Sim-to-Real Transfer’, *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 4622–4629, 2022, doi: 10.1109/LRA.2022.3151981.

[45] E. G. Ribeiro, R. de Queiroz Mendes, and V. Grassi, ‘Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation’, *Robotics and autonomous systems*, vol. 139, pp. 103757-, 2021, doi: 10.1016/j.robot.2021.103757.

[46] M.-T. Le and J.-J. J. Lien, ‘Robot arm grasping using learning-based template matching and self-rotation learning network’, *International journal of advanced manufacturing technology*, vol. 121, no. 3–4, pp. 1915–1926, 2022, doi: 10.1007/s00170-022-09374-y.

[47] Y. Chen, C. Prepscius, D. Lee, and D. D. Lee, ‘Tactile Velocity Estimation for Controlled In-Grasp Sliding’, *IEEE robotics and automation letters*, vol. 6, no. 2, pp. 1614–1621, 2021, doi: 10.1109/LRA.2021.3058931.

[48] N. Saito, T. Shimizu, T. Ogata, and S. Sugano, ‘Utilization of Image/Force/Tactile Sensor Data for Object-Shape-Oriented Manipulation: Wiping Objects With Turning Back Motions and Occlusion’, *IEEE robotics and automation letters*, vol. 7, no. 2, pp. 968–975, 2022, doi: 10.1109/LRA.2021.3136657.

[49] Y. Lin, J. Lloyd, A. Church, and N. F. Lepora, ‘Tactile Gym 2.0: Sim-to-Real Deep Reinforcement Learning for Comparing Low-Cost High-Resolution Robot Touch’, *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10754–10761, Oct. 2022, doi: 10.1109/LRA.2022.3195195.

[50] H. Cao, G. Chen, Z. Li, Q. Feng, J. Lin, and A. Knoll, ‘Efficient Grasp Detection Network With Gaussian-Based Grasp Representation for Robotic Manipulation’,

IEEE/ASME Transactions on Mechatronics, pp. 1–11, 2022, doi: 10.1109/TMECH.2022.3224314.

[51] R. Kar, L. Ghosh, A. Konar, A. Chakraborty, and A. K. Nagar, 'EEG-Induced Autonomous Game-Teaching to a Robot Arm by Human Trainers Using Reinforcement Learning', *IEEE Trans. Games*, pp. 1–1, 2021, doi: 10.1109/TG.2021.3124340.

[52] D. P. Losey, A. Bajcsy, M. K. O'Malley, and A. D. Dragan, 'Physical interaction as communication: Learning robot objectives online from human corrections', *The International journal of robotics research*, vol. 41, no. 1, pp. 20–44, 2022, doi: 10.1177/02783649211050958.

[53] S. Chen, Y. Liu, S. W. Yao, J. Li, T. Fan, and J. Pan, 'DiffSRL: Learning Dynamical State Representation for Deformable Object Manipulation With Differentiable Simulation', *IEEE robotics and automation letters*, vol. 7, no. 4, pp. 9533–9540, 2022, doi: 10.1109/LRA.2022.3192209.

[54] M. H. Sayour, S. E. Kozhaya, and S. S. Saab, 'Autonomous Robotic Manipulation: Real-Time, Deep-Learning Approach for Grasping of Unknown Objects', *Journal of robotics*, vol. 2022, 2022, doi: 10.1155/2022/2585656.

[55] N. Figueroa and A. Billard, 'Locally active globally stable dynamical systems: Theory, learning, and experiments', *The International journal of robotics research*, vol. 41, no. 3, pp. 312–347, 2022, doi: 10.1177/02783649211030952.

[56] D. Jayasinghe *et al.*, 'Minimizing the Effect of Specular Reflection on Object Detection and Pose Estimation of Bin Picking Systems Using Deep Learning', *Machines*, vol. 11, no. 1, p. 91, Jan. 2023, doi: 10.3390/machines11010091.

[57] T. Ando *et al.*, 'Learning-based collision-free planning on arbitrary optimization criteria in the latent space through cGANs', *Advanced Robotics*, vol. 0, no. 0, pp. 1–13, Feb. 2023, doi: 10.1080/01691864.2023.2180327.

[58] L. Pauly, W. C. Agboh, D. C. Hogg, and R. Fuentes, 'O2A: One-Shot Observational Learning with Action Vectors', *Frontiers in robotics and AI*, vol. 8, pp. 686368–686368, 2021, doi: 10.3389/frobt.2021.686368.

[59] K. Zorina, J. Carpentier, J. Sivic, and V. Petrik, 'Learning to Manipulate Tools by Aligning Simulation to Video Demonstration', *IEEE robotics and automation letters*, vol. 7, no. 1, pp. 438–445, 2022, doi: 10.1109/LRA.2021.3127238.

[60] H. Oh, H. Sasaki, B. Michael, and T. Matsubara, 'Bayesian Disturbance Injection: Robust imitation learning of flexible policies for robot manipulation', *Neural Networks*, vol. 158, pp. 42–58, Jan. 2023, doi: 10.1016/j.neunet.2022.11.008.

[61] A. Bobu, M. Wiggert, C. Tomlin, and A. D. Dragan, 'Inducing structure in reward learning by learning features', *The International journal of robotics research*, vol. 41, no. 5, pp. 497–518, 2022, doi: 10.1177/02783649221078031.

[62] 'Flex 13 - Specs', *OptiTrack*. http://optitrack.com/cameras/flex-13/specs.html (accessed May 03, 2023).

[63] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, 'RLBench: The Robot Learning Benchmark & Learning Environment', *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, Apr. 2020, doi: 10.1109/LRA.2020.2974707.

[64] R. Gieselmann and F. T. Pokorny, ‘Planning-Augmented Hierarchical Reinforcement Learning’, *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5097–5104, 2021, doi: 10.1109/LRA.2021.3071062.

[65] Y. Jiang, S. Moseson, and A. Saxena, ‘Efficient grasping from RGBD images: Learning using a new rectangle representation’, in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3304–3311. doi: 10.1109/ICRA.2011.5980145.

[66] I. Lenz, H. Lee, and A. Saxena, ‘Deep learning for detecting robotic grasps’, *The International Journal of Robotics Research*, vol. 34, no. 4–5, pp. 705–724, Apr. 2015, doi: 10.1177/0278364914549607.

[67] S. Ainetter and F. Fraundorfer, ‘End-to-end Trainable Deep Neural Network for Robotic Grasp Detection and Semantic Segmentation from RGB’, in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 13452–13458. doi: 10.1109/ICRA48506.2021.9561398.

[68] H.-S. Fang, C. Wang, M. Gou, and C. Lu, ‘GraspNet-1Billion: A Large-Scale Benchmark for General Object Grasping’, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 11441–11450. doi: 10.1109/CVPR42600.2020.01146.

[69] A. Depierre, E. Dellandréa, and L. Chen, ‘Jacquard: A Large Scale Dataset for Robotic Grasp Detection’, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 3511–3516. doi: 10.1109/IROS.2018.8593950.

[70] A. Malik, T. Henderson, and R. Prazenica, ‘Multi-Objective Swarm Intelligence Trajectory Generation for a 7 Degree of Freedom Robotic Manipulator’, *Robotics (Basel)*, vol. 10, no. 4, pp. 127-, 2021, doi: 10.3390/robotics10040127.

[71] ‘CoppeliaSim - Coppelia Robotics’. https://www.coppeliarobotics.com/coppeliaSim (accessed Feb. 27, 2023).

[72] ‘PyRep’. https://coppeliarobotics.com/helpFiles/en/PyRep.htm (accessed Feb. 27, 2023).

[73] ‘Bullet Real-Time Physics Simulation | Home of Bullet and PyBullet: physics simulation for games, visual effects, robotics and reinforcement learning.’, Mar. 21, 2022. https://pybullet.org/wordpress/ (accessed Feb. 27, 2023).

[74] ‘Gazebo’. https://gazebosim.org/home (accessed Feb. 27, 2023).

[75] ‘MuJoCo — Advanced Physics Simulation’. https://mujoco.org/ (accessed Feb. 27, 2023).

[76] ‘versionInfo’. https://www.coppeliarobotics.com/helpFiles/en/versionInfo.htm (accessed Feb. 27, 2023).

[77] ‘Releases · bulletphysics/bullet3’, *GitHub*. https://github.com/bulletphysics/bullet3/releases (accessed Feb. 27, 2023).

[78] ‘Releases · deepmind/mujoco’, *GitHub*. https://github.com/deepmind/mujoco/releases (accessed Feb. 27, 2023).

[79] 'Gazebo - Docs: Releases'. https://gazebosim.org/docs/garden/releases (accessed Feb. 27, 2023).

[80] Z. Huang *et al.*, 'PlasticineLab: A Soft-Body Manipulation Benchmark with Differentiable Physics'. arXiv, Apr. 07, 2021. Accessed: Mar. 07, 2023. [Online]. Available: http://arxiv.org/abs/2104.03311

[81] A. Zeng *et al.*, 'Transporter Networks: Rearranging the Visual World for Robotic Manipulation'. arXiv, Jan. 05, 2022. Accessed: Mar. 10, 2023. [Online]. Available: http://arxiv.org/abs/2010.14406

[82] Y. Bekiroglu *et al.*, 'Benchmarking Protocol for Grasp Planning Algorithms', *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 315–322, Apr. 2020, doi: 10.1109/LRA.2019.2956411.

[83] B. Calli *et al.*, 'Yale-CMU-Berkeley dataset for robotic manipulation research', *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, Mar. 2017, doi: 10.1177/0278364917700714.

[84] Z. Liu *et al.*, 'OCRTOC: A Cloud-Based Competition and Benchmark for Robotic Grasping and Manipulation', *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 486–493, Jan. 2022, doi: 10.1109/LRA.2021.3129136.

[85] Y. Chebotar *et al.*, 'Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience'. arXiv, Mar. 05, 2019. Accessed: Oct. 15, 2022. [Online]. Available: http://arxiv.org/abs/1810.05687