ARTIFICIAL INTELLIGENCE

Toward next-generation learned robot manipulation

Jinda Cui* and Jeff Trinkle

The ever-changing nature of human environments presents great challenges to robot manipulation. Objects that robots must manipulate vary in shape, weight, and configuration. Important properties of the robot, such as surface friction and motor torque constants, also vary over time. Before robot manipulators can work gracefully in homes and businesses, they must be adaptive to such variations. This survey summarizes types of variations that robots may encounter in human environments and categorizes, compares, and contrasts the ways in which learning has been applied to manipulation problems through the lens of adaptability. Promising avenues for future research are proposed at the end.

INTRODUCTION

"Have we ever built a robot as capable as an ant, at any scale?" asked Mason (*1*) when talking about the variety and refinement of ant manipulation skills in his inspiring overview paper. The brain sizes of insects and animals are often much smaller than humans', yet they can still demonstrate incredible manipulation skills. For example, octopuses can sense live crabs in plugged transparent jars and open them (*2*). They can also escape when trapped in containers closed with screw-on lids (*3*) or carry coconut halves with all tentacles while walking rapidly on the sea floor (*4*, *5*).

As one might infer from the examples above, this review focuses on manipulation tasks that would be most naturally performed through contact between an agent (possibly a robot) and its environment. To be clear, we adopt Mason's definition: "Manipulation refers to an agent's control of its environment through selective contact." (*1*) noting that "agent" refers to a human, animal, or robot. For example, the octopus contacts the jar and its lid, making use of its contacts to rotate the lid relative to the jar.

We have yet to see a robot as dexterous and versatile as ants, octopuses, and many other animals. We can hand engineer robots to perform certain manipulation tasks well in controlled environments, but we have not yet been able to build a general robot that can adapt to substantial variations in task or environment. By contrast, adaptability comes naturally to humans. For example, in pick-and-place tasks, our hands can adapt to novel objects quickly. When handling heavy objects, we can naturally use other body parts or even external supports such as a wall to brace them. Furthermore, our manipulation abilities are not easily disrupted by changes in the environment: We can pick up a pen illuminated with yellow or white light, whether it is on a table or on a shelf, and often we do not even need to see it. All these adaptations seem effortless to us but are still challenging to autonomous robots. As it was seen in the 2015 DARPA Robotics Challenge Finals (*6*–*8*), the uncertain nature of unstructured environments posed great challenges for the robots as they competed to perform tasks that would have been easy for people, such as turning a valve or climbing into a car. Video clips online (*9*) show million-dollar robots tumbling to the ground, because their controllers got overwhelmed with errors resulting from something that humans can easily contend with, such as a near miss in grasping a handle. How long will it be before robots are able to work gracefully and productively in our homes and workplaces?

The current inadequacy of manipulation skills of autonomous robots in unstructured environments is a huge stumbling block for their adoption in businesses and homes. Because the current state of the art has been achieved with the benefit of about 50 years of approaches using traditional engineering modeling and analysis techniques, this review focuses on learning-based methods whose application to manipulation problems is still in their infancy.

Two recent surveys discuss robot learning for manipulation broadly and at a high level. Mason's paper, mentioned above, provides an interesting high-level discussion of many problems faced by robot manipulation researchers and gives intriguing insights from many perspectives. In the end, he suggests that the development of new learning methods designed for learning manipulation tasks is likely to expand the repertoire of manipulation tasks that robots can do. The review of Kroemer *et al.* (*10*) discusses results from over 400 papers, thoroughly covering a broad array of learning techniques and manipulation problems. On the basis of their broad views, they proposed a formal statement of manipulation learning problems. They conclude with a list of specific manipulation challenges and suggest that existing learning methods are not sufficient to solve them. In agreement with Mason, they suggest that new learning methods need to be developed specifically for manipulation problems.

Similar to the Mason and Kroemer papers, ours covers robot learning for manipulation broadly, rather than focusing on a subarea. However, we attempt to contribute a unique perspective to the discussion by focusing on the adaptability of learned manipulation skills. Adaptability is a strength of human beings and other animals that is critical to their survival in the world. Similarly, adaptability will be critical to the long-term survival of personal robots as human companions and helpers in the ever-changing human environment. By summarizing and connecting relevant studies to adaptability, we hope to provide the readers with a unified view of possible research directions to enhance the adaptability of learned manipulation skills.

The rest of this paper is organized as follows: In the second section, we discuss the challenges in robot manipulation to highlight the difficult problems and the variations to which robots must adapt. In the third section, we overview learned robot manipulation skills and identify the research frontiers of adaptability. In the fourth and the fifth sections, we start to review intensively by linking scattered research to the frontiers for achieving adaptability. In the final section, we put the puzzles together to illustrate promising directions for future developments.

Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015, USA.
*Corresponding author. Email: jic519@lehigh.edu

CHALLENGES IN ROBOT MANIPULATION

In general, there are two primary sources of challenges in traditional approaches to robot manipulation: (i) handling of complex contact mechanics and (ii) designing planning and control algorithms that are robust to variations that will be encountered in real-world deployments.

Challenges from contact

Consider a robot, objects to be manipulated, and the environment as a system. In this setting, a task is represented by a set of points representing the start and goal states and the constraints to be imposed on transition states. The robot's manipulation skill can be viewed as its ability to connect the start states to the goal states through consecutive actions. The robot is said to be skilled if it can accomplish a task quickly and reliably in the face of uncertainty.

To accomplish a manipulation task, the robot is required to make and break contacts and possibly use controlled sliding or rolling. Changes in the state of each contact among colliding, sticking, sliding, and separating change the underlying dynamics of the system. This gives mathematical models of manipulation a hybrid structure, with a different dynamic model corresponding to each contact mode, where contact mode is defined as the state of all the contacts. For example, if there are two sticking contacts, then (because collision is impossible) there are three possible future contact states (stick, slip, or separate) for each contact and therefore nine for the pair. One possible mode is (stick, separate) and another is (slip, slip). If there are n existing contacts, then there are 3^n possible contact modes; the number grows exponentially with the number of contacts.

Challenges from variations in human environments

Kemp *et al.* (*11*) summarized the challenges in human environments for robot manipulation. One word that kept reappearing was "variation." In general, human environments are highly unstructured. In contrast to controlled environments such as factories and laboratories, robots in human environments such as homes and businesses face challenges from "variations," because models used and assumptions made during algorithm design or learning differ from reality.

To overcome them, it is important to understand when and where variations may occur. On the "when," in general, it is safe to say variations can happen at any time during task execution in human environments. This implies that the variations can be either static (occur at the beginning of a new task) or dynamic (occur during a task execution). Regarding "where," from a robot's perspective, we can categorize variations into internal or external variations.

Internal variations are the intrinsic changes to a robot after deployment that may affect its capability and functionality:

1) Robot body variations. A robot's physical properties change naturally over time due to wear and tear. Modifications or malfunctions on its parts can also be expected, e.g., a jammed joint motor removes at least one degree of freedom of a robot. In these cases, we may still want the robot to maintain its manipulation skills, at least to a certain level. Big variations may even make a robot to be considered as another class of robot. In fact, different classes of robots, such as industrial robot arms, humanoids, and quadrotors, may all be suitable for some manipulation tasks. Ideally, the manipulation skill, hand-designed or learned, can be transferred to a new embodiment.

2) Robot "brain" variations. Software modifications can also change a robot's behavior, e.g., a change of the gain or frequency of a robot's controller affects its dynamic performance.

3) Robot perception variations. Perception is a crucial interface between the robot and its environment. Variations in sensor modality, capacity, quality, and perspective substantially affect a robot's understanding of the system's state.

External variations are the changes that can also happen in the environment:

1) Object variations. Both objects being manipulated and objects in the background can vary (i) within the same object class and (ii) across object classes. In-class variations may be handled by updating existing models through perception, but cross-class variations may require the construction of new models. The configuration of objects (position and orientation) can also vary and may impose challenges to the robot, e.g., background objects fall over and form a cluster which blocks the target object. These variations often happen dynamically during task executions.

2) Environmental variations. The properties of the environment, such as workspace layout, wall/floor evenness, lighting condition, temperature, humidity, and noise level, are subjected to changes. These variations also may affect the task execution, e.g., changes in humidity may result in changes in friction properties in contact interfaces between a robot's hand and an object.

3) Task variations. With humans in the loop, users may want to tune different aspects of task executions, which may change the task specification, e.g., a user may want the robot to approach the target object faster or slower. What is more, a user may change the task composition by letting the robot perform a new task that may reuse all or part of its existing skills.

Another critical aspect of variations is novelty; some variations can be anticipated, whereas others cannot. Known variations can be taken care of during the development of manipulation skills, e.g., in robot grasping, we can anticipate some object variations and make sure the skill generalizes to them. However, it is always possible to encounter unexpected variations in human environments. Ideally, the robot could acknowledge and adapt to these novel variations and complete the intended task.

As shown in Fig. 1, the space of variations can be represented as four quadrants in the plane. For robots to gracefully work in human environments, they have to be adaptable to the variations in all quadrants. An example is shown in Fig. 2 to illustrate different types of variations that may occur after robot deployment. In the top row, there are four objects on the table, and the table is in the center of a large room. In the bottom row, two new objects are introduced, making the environment more cluttered. In addition, the robot and
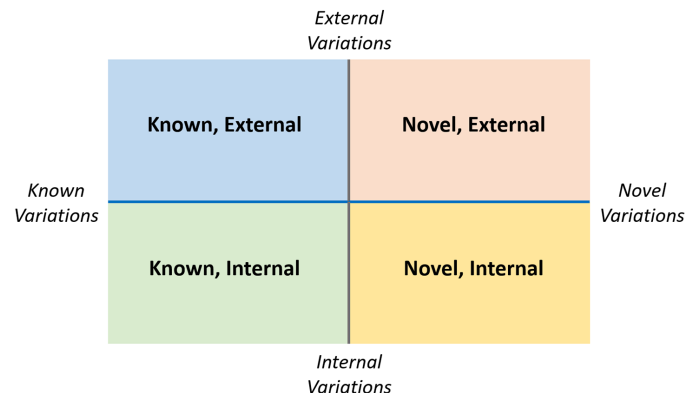

Fig. 1. The variation quadrants for robot manipulation in human environments.
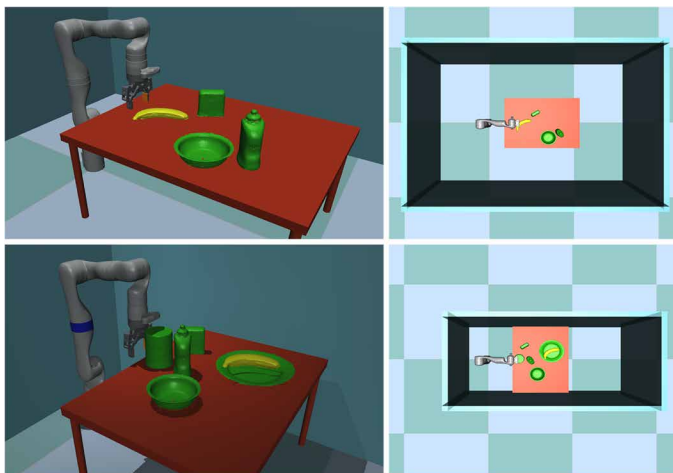
Fig. 2. Example of variations in human environments. (**Top left**) A Kinova GEN3 robot is trying to grasp a banana (the target object is the yellow banana; background objects are in green and red colors). (**Top right**) Top-down view of the scene on the left. (**Bottom left**) Externally, background objects are altered: Objects' configurations are changed, new objects are introduced (cylindrical can, plate), the banana is placed on the plate, and the table is rotated by 90°; an environmental property is also changed: The light now comes from a different direction and casts shadows. Internally and dynamically, the robot's third joint (marked in blue) becomes jammed while it is moving. (**Bottom right**) Top-down view of the scene on the left. The whole setup has been moved to a smaller room.

the table are now moved to a smaller room, so the robot must be careful not to hit a wall while working.

LEARNED ROBOT MANIPULATION AND ADAPTABILITY

Learning approaches have gained in popularity over the past few years. Deep neural networks as universal function approximators (12) along with other powerful tools in machine learning boost the use of learning in robot manipulation. Some successful examples are door opening (13), knot tying (14), and picking up of daily objects (15–17).

If we say that the key to traditional model–based methods is human intelligence, then the key to machine learning is data. Instead of developing models and devising manipulation algorithms through human intelligence, learning approaches shift these loads to computers to automatically find them from data. In learning, the challenges from contact modeling and analysis become implicit: A learner can learn its internal representation of the data and derive its own way of processing them, which alleviates the usually challenging manual designing process. As a result, human efforts will be put into designing and setting up learning processes to acquire manipulation skills.

The challenges from variations in human environments, however, are still prominent. Learned manipulation skills, after robot deployment, still face the aforementioned types of variations. Following the variation quadrants, learned manipulation skills should have the following adaptabilities: (i) the adaptabilities to internal and external variations and (ii) the adaptability to known and novel variations. All of these adaptabilities contribute to the robustness of learned manipulation skills.

It is worth noting that the adaptability to internal variations, in some sense, is equivalent to the learned skill's transferability across robot embodiment: One can think that a change on a robot makes it "another" robot, so that adapting to the change is equal to

transferring the skill to another robot. The two groups of adaptabilities complement each other and are integral to manipulation skills: A skill robust to internal and external variations must be able to handle known (expected) and novel (unexpected) variations.

Although the notion of adaptability is not often mentioned in learning for manipulation literature, they all deal with it to some extent. For example, generalization is fundamental to all learning methods, which depends on the "known" variations conveyed by the training data. In the following sections, we will go through previous research in learning for manipulation to uncover what they did to adapt to known and novel variations and talk about internal and external variations they handle in between.

ADAPTATION VIA GENERALIZATION

A fundamental goal for machine learning is to obtain generalized information (also called knowledge or concepts in this paper), namely, to create systems with the ability to capture abstract information, which generalizes to unseen data, and to do this from a finite amount of training data (18, 19). Mitchell et al. (18) categorized generalization into two forms: (i) similarity-based generalization, which exploits similarities in the training data and relies on the inductive bias to make the search more efficient. The inductive bias (20) is usually a mild guidance, e.g., a feature selection. Still, a learner needs to process a large amount of data, and the resulting generalization lacks context for explanation; (ii) explanation-based generalization, which uses successful examples to learn abstract high-level structures (e.g., a logical structure) to connect together scattered, preacquired domain knowledge. Here, the high-level structure is the generalized information captured by learning. For example, in a task "pick up the banana and place it on the plate," the high-level logic is (object A is picked up) → (object A is on top of object B) → (object A is released) → (object A is picked up and placed on object B). If the robot is equipped with skills to pick up objects, transport objects, place objects, and determine the spatial relationship among objects, then the learned high-level logic can use these skills to perform the task on any objects within the skills' capacity, not only on the banana and the plate. Compared with similarity-based generalization, this form of generalization is much more sample efficient but requires the possession of all related domain knowledge (such as the skills we mentioned in the example).

By using similarity-based methods in conjunction with explanation-based methods, the hybrid approach can leverage the best of both worlds, e.g., individual skills in the previous example can be learned through similarity-based methods. In this case, however, the trade-off between generalizability and explainability needs to be balanced, because the generalization of knowledge learned from similarity-based methods is still unexplained. The notion of explainability will be discussed later in this section.

The term explanation-based generalization was coined back in the 1980s, so it may not be called the same in recent studies. For example, Doumas et al. (21) named it "human-like generalization" in their paper studying predicate-based learning. Explanation-based generalization has seen wide usage in learning from demonstration problems, which will also be discussed.

Capturing generalized information

In similarity-based generalization, inductive bias is used to narrow down the search in the hypothesis space of the learning model to

a good region, which contains a local minimum that generalizes beyond observed data (*20*, *22*). There are many ways to introduce bias into learning, such as cross-validation (*23*), nearest neighbors (*24*), and maximum margin (*25*). One popular approach is particularly useful in robot learning for manipulation: engineering the representation of the data.

Representation learning

One can view the representation as a collection of features extracted from observations (the inputs of the learning model), on which the quality and efficiency of learning often depend. A good representation can focus the learning on aspects of the data that are pertinent to the target knowledge.

Representations can be hand-engineered, but the ease of automatic representation discovery offered by learning techniques is often preferred in unstructured environments. Representations can be learned in probabilistic ways by recovering latent variables describing observations, such as seen in Boltzmann machine and its variants (*26*–*28*). Usually, the learning targets are deterministic numerical feature values, in which case learning a parametric map from observations to representations through computation graphs, e.g., deep neural networks, is easier (*29*). Certain neural network structures are particularly suitable for some representations. For example, the convolutional layers in convolutional neural networks (CNNs) are very good at extracting translation-invariant local features (*30*–*32*); recurrent neural networks (RNNs) and its variants, e.g., long short-term memory, are good at extracting features with temporal patterns (*33*, *34*). Attention mechanism–based methods, e.g., transformer, can relax RNNs' dependency on long-term sequential data and suit data/computation parallelism (*35*). The above structures are often integral parts of larger neural networks and are trained all together, e.g., end-to-end learning of visuomotor skills. Although one can separate learned representations and transfer them to other neural networks (*36*) with careful identification, some architectures can produce representations explicitly, e.g., the autoencoder and its variants (*37*, *38*). Autoencoders are good at dimension reduction (*39*). They are often trained in a self-supervised manner: Observations are propagated through an encoder structure to produce a "code," or a latent representation, and then propagated through a decoder structure. The output from the decoder is then compared with the input, and errors are propagated back to minimize the difference between the input and the output. The encoder can be used as a feature map accordingly. See (*29*) for a detailed review of representation learning.

Learned visual representations have seen wide use in robot manipulation. Vision-based tasks often rely on dimension reductions from learned representations. For example, Levine *et al.* (*40*) proposed a deep neural network to learn visuomotor control policies, in which convolutional layers were used to extract low dimensional feature points from image pixels to better capture objects' spatial information. The network was trained in an end-to-end fashion with a lot of data. Building on a similar architecture, Finn *et al.* (*41*) decomposed training into a two-stage sample-efficient process: First, a CNN-based deep autoencoder is trained to extract position information in a self-supervised manner. The features produced by the encoder are then used as a part of the state observations in the reinforcement learning of visuomotor skills.

Traditionally, nonvisual sensing modalities are also important to robot manipulation (*42*). They have started to gain attention in recent learning for manipulation research. For example, Fazeli *et al.* (*43*) enabled a robot to play Jenga using force sensors along with visual sensors. When interacting with a block, the normal force, the block rotation, and the block extraction were used to estimate the abstract status of the block (no move, small resistance, etc.), which is hence used as inputs of a Bayesian neural network representing the state transition of the block. Cui *et al.* (*44*) used three-dimensional (3D) convolutional layers to interface with time-series measurements of visual and tactile signals. Their method handles the two modalities at a different frequency and fuses them together to produce classifications of grasping status of deformable objects. Hogan *et al.* (*42*) used human intuition to segment primitive manipulation actions based on tactile measurements, but they also pointed out that representations of these primitives can be learned. In a similar vein, Edmonds *et al.* (*45*) learned embodied haptic representations of manipulation actions using tactile and force sensors to identify the same actions performed by different agents.

Learned representations can also be used in analytical methods. For example, Mahler *et al.* (*46*) trained a CNN-based neural network to compute a similarity measure between 3D objects. Combining this with hand-designed features, they can query their Dex-Net 1.0 dataset, which contains object models and corresponding robust grasps, and efficiently select the best grasp for new objects. Learned representations can also go beyond the perception level. Kwiatkowski and Lipson (*47*) used a deep neural network consisting of recurrent, convolutional, and fully connected layers to learn a representation of sequences of state-action pairs, which essentially approximates a robot arm's forward kinematics under joint and self-collision constraints. A review of representations in robot learning for manipulation can be found in (*10*).

Data, simulation, and reality gap

A suitable bias leads to efficient searching, but without appropriate data, it lacks contexts to learn. To obtain manipulation skills robust to future variations, a learner needs to capture generalized information that covers these variations. A straightforward solution is to infuse variations into the training data to encompass as many future variations as possible. Researchers working on learned robot manipulation have tried to enrich training data for better generalization in hardware experiments. For example, in (*48*) and (*49*), objects were placed in various locations in training, so that learned manipulation trajectories can generalize to the variations in object locations. Regarding object shape variations, Yahya *et al.* (*13*) developed a distributed reinforcement learning method and used multiple robots in parallel to learn and complete door opening tasks under door handle variations.

Because of equipment cost and long duration in real-world experiments, it is often more desirable to gather data or learn in simulation. For example, Mahler *et al.* (*50*) upgraded their Dex-Net 1.0 to 2.0 by infusing a large amount of synthetic point clouds to the dataset and successfully learned a grasp robustness function Grasp Quality Convolutional Neural Network (GQ-CNN) from it.

Learning manipulation skills in simulation adds another layer of complication: The reality gap, although we have seen remarkable advances in simulation that produce efficient and life-like physical effects (*51*–*53*), none of them match 100% to real-world physics. Thus, skills learned in virtual environments do not generalize directly to real-world scenarios. Moreover, simulating contact events in robot manipulation can be much more challenging than regular physical events (*54*), which expands the reality gap even further. To overcome the reality gap, Tobin *et al.* (*55*) advocated a straightforward technique called domain randomization. This technique randomly

injects a wide range of variations during training and hopes that these variations (i) capture the differences between simulation and reality and (ii) encourage the learner to learn more generalizable (domain-invariant) skills. In the same light, Chebotar *et al.* (*56*) invoked a massive amount of simulations in parallel to learn cabinet opening and swing-peg-in-hole tasks with simulation parameters randomly sampled from their distributions. The difference between Chebotar *et al.*'s method and a naive domain randomization is that the parameter distributions are not static but are brought closer to reality through distribution updates following a few real-world rollouts of the learned policy.

Sometimes, hardware designs with compliant mechanisms can simplify robot manipulation and reduce the reality gap. A representative design is remote center compliance, developed in the 1970s. It uses fully passive compliant mechanisms to make peg-in-hole assembly robust to lateral and angular misalignment (*57*). More modern adaptive robot hands, which are often soft or underactuated, enable stable grasping of a wide variety of objects (*58–60*). When used in learning, the adaptability of these devices is inherited by the learned skills, making them more generalizable. Manipulation also becomes more straightforward with these designs. For example, with an adaptive hand, grasping can be simplified into correctly posing the hand with respect to the object, rather than figuring out each contact point (*61*). When simulating these tasks, contact dynamics can be replaced by geometric constraints, which markedly reduces simulation difficulty and narrows the reality gap. However, for tasks that require accurate contact physics, e.g., dexterous in-hand manipulation, the reality gap can be even worse because these mechanisms are challenging to simulate accurately due to their deformability and compliance. The reality gap can be viewed as a domain adaptation problem, on which we have more discussions in a later section.

Learned correspondence
In Dex-Net 1.0 (*46*), object observations were mapped into a metric space, in which point-wise distances correspond to the similarity between objects. This is a version of metric learning (*62*). As mentioned in (*10*), such similarity measure can be established at different levels between objects, parts, and points.

Some recent research brought dense correspondence learning into robot manipulation. The word "correspondence" comes from correspondence estimation, which is widely used in computer vision to determine corresponding parts across different images. The word "dense" means such correspondences are determined at the pixel level, i.e., a pixel in one image corresponds to a pixel in another image. Putting together, dense correspondence learning seeks to learn a descriptor in a metric space for each pixel. The distance between two pixels represents their similarity: The closer they are in the metric space, the more similar they are in the images (*63*, *64*).

A dense correspondence descriptor can benefit robot manipulation if pixels are coupled with useful information. For example, a pixel on a mug's handle in an image represents a small surface region on the actual handle, which is tied to its physical properties such as the position in the world frame and contact conditions. Florence *et al.* (*15*) presented a method that provides users great flexibility in choosing grasping points for object pickup. They developed an automated training pipeline to learn dense descriptors for a single object, mixed objects within the same object class, and mixed objects from different classes. When a user selects a point from an object's reference image, the dense descriptor can return a corresponding

point in the freshly taken image. The robot will then grasp and lift the object according to the 3D position of the selected point queried using the accompanying depth image. Similarly, Zakka *et al.* (*65*) leveraged dense descriptors to determine and match positions and orientations of objects in 2.5D kit assembly problems and demonstrated generalization to some unseen kits.

Unlike end-to-end learning, the above examples only used learning for dense correspondence, leaving the majority parts of manipulation to traditional analytical modules. The impressive applications enabled by learned dense correspondence suggest that learning, even at a module level, may make notable improvements to robot manipulation. Although there is no consensus in the community on what part of manipulation should be learned, there are many interesting approaches and trade-offs to consider.

Modularity, transferability, customizability, and explainability
Bengio *et al.* (*29*) observed that learned representations capture underlying knowledge of observations, thus, when shared, can enable multitasking, transfer learning, and domain adaptation. This essentially treats learned representations as modules of "domain knowledge." As discussed at the beginning of this section, the explanation-based generalization learns and generalizes an explainable high-level structure based on scattered domain knowledge. The domain knowledge can be replaced by learned representations to form a hybrid scheme of explanation-based and similarity-based generalization.

To benefit from this, appropriate representations for manipulation tasks must be designed. There are many ways to decompose a task into subtasks. In addition, subtasks may also have internal structures that can be further decomposed. Eventually, a task can be broken down into a task structure of atomic action primitives. However, in robot manipulation, action primitives are robot dependent. As Zech *et al.* (*66*) pointed out, a representation of action in robotics is tied to perception, embodiment, and actuation of the robot. For example, action primitives can be drawn from a robot's mobility, sensing, and control primitives. To achieve better generalization, a task representation must balance the granularity of task decomposition.

For example, a representation most robust to internal variations should be independent of agents' specifics, so that it can generalize across agent classes in a "zero-shot" manner (*67*), e.g., transferring a learned manipulation skill from a human being to a robot without training the robot. Edmonds *et al.* (*45*) proposed a high-level task representation for a medicine bottle opening task, in which the robot learns from human demonstrations to deal with tricky cap locking mechanisms. Each human action, e.g., twisting the cap, is measured haptically, and low-dimensional representations of the measurements are learned using an autoencoder. Another encoder is trained to map haptic measurements from robot actions to the learned representations of the equivalent human actions. This results in agent-agnostic representations such that the learning of the high-level decision graph is separated from low-level actions. In addition, the learned decision graph can command the robot to perform the task directly without training the robot.

Modularity also enables customizability. For example, Araki *et al.* (*68*) proposed a neural network architecture based on linear temporal logic and value iteration network. They learn a policy using a two-level task representation, with a high-level finite-state automaton (FSA), and a low-level Markov decision process. By modifying the

transition matrix of the FSA, they can modify the task composition, e.g., change the robot's task from "pick up and pack the burger first, then the banana" to "pick up and pack the banana first, then the burger."

Another benefit of modularity is explainability. In machine learning, explainability is used to describe the ability to explain the learned model in contrast to "black-box" models such as fully connected neural networks. The Merriam-Webster definitions (69) of the word "explain" as a transitive verb are: 1.a. "to make known"; 1.b. "to make plain or understandable"; 2. "to give the reason for or cause of"; 3. "to show the logical development or relationships of." When asked, "explain the machine learning model," on the basis of the definitions, it refers to expressing the internal mechanisms of the model in human-understandable terms and logic, and this naturally requires modularity in the model.

The two examples above demonstrate certain levels of explainability. The learned decision graph in (45) is explainable because it directs the robot to conduct actions with semantic meanings so that a task execution can be explained in human-understandable terms, e.g., "The robot pushed the cap and rotated it for three times, and the bottle is opened." In the second example (68), the explainability comes from the high-level FSA in which each state is human understandable, e.g., "robot has grasped a banana."

Growing attention has been paid to modularizing control policies for manipulation tasks by introducing policy hierarchies. Starting from primitive actions, Riedmiller et al. (70) learned middle-level manipulation skills named "intentions" and a high-level scheduler to sequence the skills for task completion. In a similar vein, Hausman et al. (71) learned diverse middle-level manipulation skills using an entropy-regularized method in a multitask setting. They created handles of skills in a low-dimensional space to efficiently recomposed learned skills for new tasks. With a maximum-entropy policy, Eysenbach et al. (72) further pushed the diversity of learned middle-level skills. More levels are also possible. For example, Levy et al. (73) learned multiple levels of control policies, with higher-level policies imposing subgoals to lower-level policies. Hindsight (74) transitions for both goals and actions were used to solve the nonstationary state transition problem during off-policy learning. Although these studies have demonstrated the viability of using various levels of abstractions to represent learned robot manipulation, the adaptability that may come along with these representations was not extensively studied, especially on overcoming the internal variations.

ADAPTATION BEYOND GENERALIZATION

Adaptability via generalization is based on the assumption that the knowledge a robot learned in the training environment (source domain) generalizes to the variations in the new environment (target domain). However, it is always possible to encounter novel variations in human environments to which old knowledge will not cover. The adaptation to them demands more than generalization. Specifically, the following two abilities are required: (i) the ability to continually adapt to novel variations and (ii) the ability to remember and build upon previously acquired skills.

The first ability is more fundamental: It is the first step for a robot to accommodate the ever-changing human environment. To achieve it, gathering data about novel variations and learning from them is important. A solution is to use learning methods that can actively explore and gather such data, e.g., reinforcement learning and online learning. Sometimes, exploration can be difficult, e.g., exploring

a space in which objects are sparsely distributed. Thus, sometimes, learning through examples of task executions is desirable.

The second ability augments the first by enabling learning without forgetting. More formally, it addresses a critical problem in continual learning called catastrophic forgetting, or catastrophic interference, in which training on new data may overwrite already learned skills. This problem is primarily studied in the lifelong learning paradigm, which is yet to be applied to robot manipulation at scale. Common approaches to catastrophic forgetting are memory-based augmentation, expansion-based retraining, and regularization-based retraining. All these methods are facing challenges from dimensionality, performance, and training difficulty and are being improved actively by the community. Here, we focus on the first ability. Readers interested in catastrophic forgetting can get further information from (75).

Domain adaptation techniques

In the face of novel variations, a straightforward way of adaptation is to gather more data for extended training sessions, as seen in the work of Kwiatkowski and Lipson (47) dealing with internal variations: When a novel change on the robot body occurred, i.e., a link of the robot is replaced by a longer link with an angle difference, the previously learned neural network can be incrementally trained with a relatively small new training set to restore the functionality.

The amount of new training data can be substantially reduced if the skills being adapted have "learned to learn." Finn et al. (76) introduced model-agnostic meta-learning (MAML) to the field in 2017. The basic idea is similar to domain randomization: By infusing more domain variations during the training phase, the learner may learn more abstract knowledge, which generalizes better to future variations. The difference is the fact that MAML does not immediately update a learner's parameters after training on each variation. Instead, at each update, it gathers losses from all sampled variations in a training-validation style. Intuitively, updates in MAML are "fairer" to all variations, which may force the learner to learn adaptation skills instead of memorizing variations shown in the training data. In their other work, one-shot adaptation (adapt to a new variation using only one new training example) across manipulation task variations was achieved using a variant of MAML (77).

As mentioned before, the modularity of learned skills may contribute to domain adaptation. When the difference between the source domain and the target domain is at the perception level, instead of incremental training, one can try to reduce the difference between the source domain and the target domain, e.g., convert observations in the source domain to match observations in the target domain (78). This approach is particularly useful for crossing the reality gap, where perception differences can be substantial between simulator and reality. For example, Bousmalis et al. (79) trained a generative adversarial network (GraspGAN) to render simulated scenes to visually match the reality in an end-to-end grasp learning problem. They also augmented DANN (80) to learn domain-invariant features, which further enabled the efficient transfer of learned grasping skills from simulation to hardware. For further reading, Tobin et al. (55) have an insightful discussion of domain adaptation in the related works section.

Active learning and exploration

Domain adaptation can be data efficient. However, learning from scratch is sometimes inevitable when the variations are too big. Active learning methods provide nice solutions in such cases.

The term "active learning" was coined by Cohn *et al.* (*81*) in the 1990s to describe learning methods that have control over the inputs they train on. This ability enables autonomous data gathering and learning, which suits the practical scenarios very well: After robot deployment, it is often the case that users will not have the equipment and the ability to gather data and train the robot by themselves.

When learning for manipulation, usually, the goal is a control policy that can command a robot to complete a task through consecutive observations and actions. Closed-loop policies are important because robots need to react to both static and dynamic changes in human environments (*82*, *83*). The policies are often approximated by deep neural networks. To find a good policy, one can directly search in the hypothesis space (or policy space, space spanned by the variables of a neural network modeling the policy). The most naive method is random search, which is the antithesis of active learning: Policies are generated by randomly choosing learning models and assigning their parameters (*84*). A reward function is often used to evaluate policy rollouts for policy selection. Because this method can potentially search the entire hypothesis space, there is a possibility that a globally optimal policy is obtained. However, the chance is usually extremely low, given the large hypothesis space of deep neural networks. More systematic and "active" searching use heuristics such as evolutionary algorithms and use rewards more actively to guide the search (*85*, *86*). Gradient-based optimizations, such as back-propagation with stochastic gradient descent, are often integrated with active learning frameworks, e.g., in most deep reinforcement learning methods. Compared with the aforementioned examples, reinforcement learning usually explores the hypothesis space on a finer scale in terms of policy updates: Whether the specific method is "on-policy" or "off-policy," the policy is usually updated immediately or shortly after each action is taken. A good survey on reinforcement learning in robotics can be found in (*87*).

Active learning methods are subjected to trade-offs between exploration and exploitation. They usually start with exploration to search wildly across the hypothesis landscape to sparsely cover its breadth. Exploitation, on the other hand, refines the search in a subregion so that the policy may lastly converge to a local minimum, regarding a specific reward function. Generally speaking, exploration is difficult even for humans, e.g., it is challenging for us to learn a new subject of study or a new sport from scratch. Specifically, the key challenges in exploration-based methods are (i) how to explore efficiently and effectively (ii) when to transition from exploration to exploitation—if the exploration stops prematurely, e.g., before any contact was made in between of the robot and the target object, exploitation will not make substantial progress toward grasping of the object. On the other hand, exploration should not continue forever without converging to a solution.

A summary for exploration strategies can be found in (*10*). Basically, research in this area aims to address the following questions: (i) how to make sure exploration makes progress, (ii) how to make sure exploration is thorough, and (iii) how to make sure exploration is efficient. A prerequisite for exploration to make progress is continuity: The agent needs to be able to explore continually. OpenAI and coauthors have demonstrated learned in-hand dexterous manipulation of solid cubes (*88*) and Rubik's cubes (*89*). However, their results were under the assumption that the robot hand is palm up. If the hand is palm down, then the object might be dropped frequently during exploration, which breaks the continuity. In the same light, a branch of research studies safe exploration strategies to prevent damage to the robot or irrecoverable failures during exploration. Some directly enforce a set of safety rules to prevent certain actions leading to unsafe states (*90*); some encode task safety specifications in the reward function to encourage agents to explore safely (*91*). An overview for safe exploration in reinforcement learning can be found in (*92*). Although safe exploration reduces exploration failures, which helps maintain exploration continuity, it may limit exploration to a subspace so that the search may not be thorough (*93*). For example, imposing distance thresholds for collision avoidance may prevent the robot from finding narrow passages.

Thoroughness in exploration is also challenging. One can argue that the most thorough exploration is a uniform random search in the space of all possible policy representations and their parameters. This might be probabilistically complete, but it may take forever to find a good solution. Instead of probabilistic completeness, what might be more interesting is the ability to explore without getting trapped in local minima, which is crucial for continual improvements of learned policies. Popular reinforcement learning methods, such as Q-learning variants (*94*, *95*), often use ε-greedy strategies with decay, which start with an exploration stage with more random actions and then transit toward the exploitation stage with more actions from the policy. Often, this process is irreversible: Once it converges, it is stuck in the local minimum. More sophisticated methods use adaptive strategies to balance exploration and exploitation, by allowing reentering of the exploration stage when the agent is uncertain about its decision-making (*96*, *97*). These methods are more stable and may converge to better local minima, but they depend on heuristics, i.e., the uncertainty measure. Policy gradient methods (*98*, *99*) explore by following gradients with noise, which may provide smoother convergence; however, they usually take a long time and still face challenges from local minima.

At the end of the day, the efficiency of the exploration is also important. Can a method yield a reasonably good policy within a relatively short amount of time? Exploration may not be challenging for certain tasks such as simple nonprehensile manipulation (pushing, tilting, etc.) and regular pick and place. However, for tasks involving extensive contact mode switching, a learner may unavoidably explore a large space of contact modes, whose size grows exponentially with the number of contacts. Computing techniques may provide alternative solutions even in the worst case when an exhaustive search is needed: When learning is performed in simulated environments, it can benefit from data parallelism by using a massive number of simulators to gather data at the same time. Successful approaches have been seen in the form of parallel computing (*100*) and distributed systems (*101*, *102*). Although these approaches improve sampling efficiency, more systematic analyses are needed to thoroughly understand the relationship between sampling efficiency and learning performance.

Learning from demonstrations

Active learning is not the only way to acquire new skills. In fact, when people learn, we often leverage external guidance: Written descriptions, verbal instructions, visual demonstrations, hands-on guidance, etc., all can boost the efficiency of learning. Similarly, when new situations require robots to adapt rapidly, instead of exploring tirelessly by themselves, they can query successful examples and learn critical information that might be hard to find through exploration. As seen in the work of Edmonds *et al.* (*45*), to open medicine bottles, the robot may need to turn the cap while pressing it down,

which is hard-to-discover information through exploration. However, this information can be extracted from human demonstrations. Methods that transfer skills to robots through task execution examples are often under one of the following names, which are used interchangeably: programming by demonstration, learning from demonstration, imitation learning, learning from observation, behavior cloning, etc.

Kinesthetic teaching is a popular method that allows robots to experience tasks directly, e.g., dragging the robot around or teleoperating it to complete a task. Often, trajectories of sensor measurements, such as joint encoder readings and force/torque sensor readings, are recorded as training data.

Indirect teaching methods often require robots to observe demonstrations from other agents. Various levels of information can be extracted from such observations: Similar to kinesthetic teaching, one can obtain trajectories from demonstrations, e.g., state-space trajectories of the end effector using a motion capture system; obtain a high-level structure that links task primitives, e.g., a finite state machine with learned transitions as in (68); or learn a reward function that can be used in exploration-based learning, e.g., inverse reinforcement learning (103). For general information in this area, Argall et al. (104) give a comprehensive review regarding demonstration methods and policy derivation; Ravichandar et al. (105) provide a systematic categorization in terms of learning input and outcome for more recent works.

Details aside, demonstration-based learning can be boiled down to this question: What information can be extracted from the examples and transferred to the learner? Generally speaking, successful task completion requires sufficient coverage of all necessary task conditions. For manipulation tasks, these necessary conditions are often object-centric: We only care about target objects' configuration during and after a task. Skill transfer based on robot joint trajectories is the most straightforward, but it relies on the assumption that object state changes are coupled with robot joint state changes (because of contact), which is not guaranteed. For example, Ugur and Girgin (106) use dynamic motion primitives and parametric hidden Markov models to learn joint space trajectories with force coupling for external guidance. In the manipulation tasks they present, either contact is assumed, i.e., the robot and the cabinet handle are attached at the beginning, or the gripper needs to be positioned to the pregrasping location through human guidance. Similarly, in (107), the robot needs human collaboration to make contact with objects in cocktail bottle shaking and painting tasks.

Although trajectory-based learning is efficient for the straightforward transfer of skills, it does not fit complex tasks. For example, Zhang et al. (108) show a trajectory-based, end-to-end visuomotor policy learning for more sophisticated tasks, e.g., pick a ball, place it in a plate, and then push the plate to a target location. However, their method is only suitable for tasks with sequential action composition. To learn/transfer more complex tasks that may have task hierarchy and/or decision-making logic, one must extract the information at a higher level than trajectories.

The research discussed, when talking about modularity, serves as good examples here: In the FSA of Araki et al. (68), state transition probabilities are learned so that the robot can execute actions in a nondeterministic manner similar to humans (recall that in a lunch box packing task, one may pick up and pack the banana first, then the burger, or pack them in the reverse order); Edmonds et al. (45) learned a high-level decision graph, which is a symbolic stochastic

manipulation grammar [see (109) for more details about manipulation grammar] to capture human decision-making under various situations during the opening of medicine bottles, e.g., if pinch open does not work, then try press and twist.

The above examples depend on manual segmentation and annotation of task states and actions, which are possible to be learned. Given joint space trajectory demonstrations, abrupt changes can be used to segment motions, e.g., substantial changes in position and velocity (110). Similarly, to learn manipulation actions, state changes of objects can be used as cues for action identification. Zampogiannis et al. (111) use objects' spatial relationship during manipulation to represent atomic actions for automatic action classification. Based on an object classifier and an action classifier, Yang et al. (112) learn a probabilistic action grammar similar to the work of Araki et al. (68).

Learning from demonstration for robot manipulation is still in its early stage, leaving open research questions. We will discuss this along with opportunities in other areas in the next section.

DISCUSSIONS

In previous sections, we have gone through, broadly, approaches that contribute to the adaptability of learned robot manipulation. Representation learning and data acquisition were discussed for efficient and effective capturing of generalized information. Learned dense correspondence demonstrated the power of learned modules in catalyzing robot manipulation applications. Further discussions were made to show that modularized representations may enable transferability, customizability, and explainability. In the face of novel variations, active learning and learning from demonstration methods provide potential solutions. They both allow robots to learn new manipulation skills, but key problems are still open, e.g., learned dexterous manipulation and continual skill improvement after deployment.

To push forward the capacity and adaptability of learned robot manipulation, the following questions can be asked: (i) Which part of manipulation should be learned, and what software and hardware advancements are needed to support that? (ii) What to do to better capture generalized information? Use special training techniques, innovative architectures, representation engineering, or augmented datasets? (iii) How to enable the transfer of knowledge in the case of external, internal, and novel variations? (iv) How to extend active learning or learning from demonstration methods to enable continual adaptation? (v) What can be done, either on software or hardware, to boost the efficiency of the learning process? To date, these questions were only partially answered, leaving great opportunities for further exploration, as summarized below:

1) Representation learning with more sensing modalities. Most previous studies focus on visual sensing. Indeed, it is arguably the most important sensing modality for robots; however, there is a key limitation that makes vision, by itself, unable to cover all manipulation scenarios: It cannot sense contact if the contact region is visually blocked. In fact, humans use multimodal sensory signals during manipulation (113), which includes but is not limited to tactile, auditory, and temperature signals. Adding representations of these sensing modalities would provide learned robot manipulation more holistic understanding of current system states, thus boost the learning performance.

2) Advanced simulators for manipulation. Before robots and sensors, such as industrial-grade robot arms and high-resolution tactile

sensor arrays, become much cheaper and safer for contact events, physical simulators are crucial for manipulation learning. Ideally, we hope simulators to be as fast and as realistic as possible. By making compromises in physical accuracy, we can enjoy fast (faster than real time in many cases) simulations already, e.g., MuJoCo (*51*). However, realistic physics is still a challenging goal for state-of-the-art simulators, especially when it comes to contact modeling (*54*), which becomes even more challenging when deformable objects and robots come into play. What is more, it is desired to have simulations for more manipulation scenarios (liquid manipulation, cutting/breaking of objects, etc.) and more physical modalities (sound, temperature, etc.).

3) Task/skill customization. As a source of external variations, robot users may change the task composition or task specification. As discussed before, the modularity of manipulation and domain adaptation techniques should be exploited for these customizations.

4) "Portable" task representations. Previous studies primarily focus on the generalization over external variations, leaving the adaptation to internal variations barely touched. As discussed, a key to such adaptation is to identify a proper level of abstraction (representation) of the manipulation task. When the task representation is disentangled from specific embodiment, the learned representation can be transferred across agents. Here, the interesting question is how abstract the representations should be for internal variations that occur at different levels of task decomposition.

5) Informed exploration for manipulation. Active learning methods can find new skills for novel variations. Random sampling-based exploration worked well in motion planning [e.g., Rapidly-exploring Random Tree (*114*)]. However, because of the sparse nature of contact events, it is very inefficient for manipulation tasks. The sparsity of contact events, on the other hand, imposes a strong motivation to adopt informed exploration similar to informed sampling methods seen in motion planning, such as goal-driven and obstacle-aware methods (*115–117*). Usually, reinforcement learning agents obtain such information from reward functions, which can be difficult to hand design. Hindsight experience replay (*74*) demonstrates that, for some simple tasks such as reaching and pushing, skills can be learned even with sparse rewards. However, the theoretical guarantee of convergence and the applicability to more complex manipulation tasks require further study.

6) Continual exploration. As mentioned before, it is challenging for a learned skill to improve continually after robot deployment. A naive way to achieve this is to keep a simulation thread busy learning for novel variations while using the best policy available on the physical robot. However, more sophisticated methodologies must exist and are waiting to be found.

7) Massively distributed/parallel active learning. When learning skills from scratch, efficiency is a critical metric. Most previous research adopts data parallelism to extend single-thread active learning methods. However, the relationship between sampling efficiency and learning performance is unclear, which demands more rigorous studies. What is more, new active learning methods that can benefit from both data and model parallelism are desired to improve efficiency further.

8) Hardware innovation. As discussed, hardware designs with compliant mechanisms (*57–61*) may simplify robot manipulation and increase adaptability, but they are often limited to tasks with simple contact events, e.g., static grasping. Additional studies are needed to simplify more challenging manipulation tasks, e.g., in-hand dexterous

manipulation. Some early designs are the Shadow Dexterous Hand (*118*) and the JamHand (*119*). The former is almost fully actuated (only 4 of 24 joints are underactuated), whereas the latter is not, but, nevertheless, is capable of basic dexterous manipulations. What is more, although these devices' deformability and compliance bring safety and robustness, they are challenging to be accurately and efficiently simulated. For achieving good learning results in simulation, it would be worthwhile to seek simulation-friendly designs and materials. More discussions on hardware design for robot manipulation can be found in the review paper of Billard and Kragic (*120*).

9) Real-time performance. Eventually, learned manipulation skills will be tested in the real world. The latency and frequency of robots' control loops are critical, especially in dynamic scenes. Developing fast learning models and algorithms is essential. For example, Morrison *et al.* (*16*, *83*) proposed a light-weight neural network (Generative Grasping Convolutional Neural Network), which enabled a closed-loop control of 50 Hz (often much slower in previous research) for vision-based grasping. In addition to software speedup, hardware is also critical for further enhancing learned robot manipulation. Fast perception, communication, and actuation are prerequisites for complex dexterous manipulation tasks, which may require low latency control iterations with a frequency as high as 1 kHz or even higher.

Here, although we have aimed to be as comprehensive as possible, it is impossible for us to find and review all remarkable works related to this broad topic. That said, by illustrating the idea of adaptability in learned robot manipulation with reviews of state-of-the-art studies, we hope that we have provided a unique perspective to the manipulation community, which would generate more discussions and ideas leading to a brighter future of robot manipulation.

REFERENCES AND NOTES

1. M. T. Mason, Toward robotic manipulation. *Annu. Rev. Control Robot. Auton. Syst.* **1**, 1–28 (2018).
2. G. Fiorito, G. B. Biederman, V. A. Davey, F. Gherardi, The role of stimulus preexposure in problem solving by *Octopus vulgaris*. *Anim. Cogn.* **1**, 107–112 (1998).
3. Enoshima Aquarium, The terrible ability of the octopus; www.youtube.com/watch?v=HL3M_DcO6wk.
4. J. Finn, Coconut-carrying octopus; www.youtube.com/watch?v=1DoWdHOtlrk&feature=youtu.be&t=58.
5. J. K. Finn, T. Tregenza, M. D. Norman, Defensive tool use in a coconut-carrying octopus. *Curr. Biol.* **19**, R1069–R1070 (2009).
6. E. Ackerman, E. Guizzo, Darpa robotics challenge: Amazing moments, lessons learned, and what's next; https://spectrum.ieee.org/automaton/robotics/humanoids/darpa-robotics-challenge-amazing-moments-lessons-learned-whats-next.
7. DARPA, DRC Final 2015; https://archive.darpa.mil/roboticschallenge/.
8. M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, J. Pratt, Team IHMC'S lessons learned from the DARPA robotics challenge trials. *J. Field Robot.* **32**, 192–208 (2015).
9. I. Spectrum, A compilation of robots falling down at the DARPA robotics challenge; www.youtube.com/watch?v=g0TaYhjpOfo&feature=youtu.be&t=26.
10. O. Kroemer, S. Niekum, G. Konidaris, A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.* **22**, 1–82 (2021).
11. C. C. Kemp, A. Edsinger, E. Torres-Jara, Challenges for robot manipulation in human environments [grand challenges of robotics]. *IEEE Robot. Autom. Mag.* **14**, 20–29 (2007).
12. B. C. Csáji, "Approximation with artificial neural networks," thesis, Eötvös Loránd University, Hungary (2001).
13. A. Yahya, A. Li, M. Kalakrishnan, Y. Chebotar, S. Levine, Collective robot reinforcement learning with distributed asynchronous guided policy search, in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 79–86.

14. J. Schulman, J. Ho, C. Lee, P. Abbeel, Learning from demonstrations through the use of non-rigid registration, in *Robotics Research: The 16th International Symposium ISRR* (Springer, 2016), pp. 339–354.

15. P. R. Florence, L. Manuelli, R. Tedrake, Dense object nets: Learning dense visual object descriptors by and for robotic manipulation, in *Proceedings of the 2nd Conference on Robot Learning* (PMLR, 2018), pp. 373–385.

16. D. Morrison, J. Leitner, P. Corke, Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach, in *Proceedings of the 2018 Robotics: Science and Systems XIV (RSS)* (2018); www.roboticsproceedings.org/rss14/p21.html.

17. J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. De Rose, S. McKinley, K. Goldberg, Learning ambidextrous robot grasping policies. *Sci. Robot.* **4**, eaau4984 (2019).

18. T. M. Mitchell, R. M. Keller, S. T. Kedar-Cabelli, Explanation-based generalization: A unifying view. *Mach. Learn.* **1**, 47–80 (1986).

19. J. Bartlett, E. Holloway, Generalized information: A straightforward method for judging machine learning models. *Commun. Blyth Inst.* **1**, 13–21 (2019).

20. T. M. Mitchell, "The need for biases in learning generalizations" (Technical Report CBM-TR-117, Rutgers Univ., 1980).

21. L. A. A. Doumas, G. Puebla, A. E. Martin, Human-like generalization in a machine through predicate learning. arXiv:1806.01709 [cs.AI] (7 March 2019).

22. J. Baxter, A model of inductive bias learning. *J. Artif. Intell. Res.* **12**, 149–198 (2000).

23. C. Schaffer, Selecting a classification method by cross-validation. *Mach. Learn.* **13**, 135–143 (1993).

24. N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **46**, 175–185 (1992).

25. B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (Association for Computing Machinery, 1992), pp. 144–152.

26. G. E. Hinton, Boltzmann machine. *Scholarpedia* **2**, 1668 (2007).

27. R. Salakhutdinov, G. Hinton, Deep Boltzmann machines, in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics* (PMLR, 2009), pp. 448–455.

28. L. K. Saul, M. I. Jordan, Boltzmann chains and hidden Markov models, in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, T. K. Leen, Eds. (MIT Press, 1995), pp. 435–442.

29. Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013).

30. A. B. Yandex, V. Lempitsky, Aggregating local deep features for image retrieval, in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2015), pp. 1269–1277.

31. Y. LeCun, Learning invariant feature hierarchies, in *Computer Vision - ECCV 2012. Workshops and Demonstrations*, A. Fusiello, V. Murino, R. Cucchiara, Eds. (Springer, 2012), pp. 496–505.

32. J. Laserson, From neural networks to deep learning: Zeroing in on the human brain. *ACM Crossroads* **18**, 29–34 (2011).

33. A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks, in *Proceedings of the 23rd International Conference on Machine Learning* (Association for Computing Machinery, 2006), pp. 369–376.

34. F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, in *Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks* (Institution of Engineering and Technology, 1999), vol. 2, pp. 850–855.

35. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2017), pp. 6000–6010.

36. M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2014), pp. 1717–1724.

37. D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).

38. P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in *Proceedings of the 25th International Conference on Machine Learning (ICML)* (Association for Computing Machinery, 2008), pp. 1096–1103.

39. G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).

40. S. Levine, C. Finn, T. Darrell, P. Abbeel, End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* **17**, 1334–1373 (2016).

41. C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, P. Abbeel, Deep spatial autoencoders for visuomotor learning, in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 512–519.

42. F. R. Hogan, J. Ballester, S. Dong, A. Rodriguez, Tactile dexterity: Manipulation primitives with tactile feedback, in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 8863–8869.

43. N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, A. Rodriguez, See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion. *Sci. Robot.* **4**, eaav3123 (2019).

44. S. Cui, R. Wang, J. Wei, F. Li, S. Wang, Grasp state assessment of deformable objects using visual-tactile fusion perception, in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 538–544.

45. M. Edmonds, F. Gao, H. Liu, X. Xie, S. Qi, B. Rothrock, Y. Zhu, Y. N. Wu, H. Lu, S.-C. Zhu, A tale of two explanations: Enhancing human trust by explaining robot behavior. *Sci. Robot.* **4**, eaay4663 (2019).

46. J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kroger, J. Kuffner, K. Goldberg, 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 *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 1957–1964.

47. R. Kwiatkowski, H. Lipson, Task-agnostic self-modeling machines. *Sci. Robot.* **4**, eaau9354 (2019).

48. B. Reiner, W. Ertel, H. Posenauer, M. Schneider, Lat: A simple learning from demonstration method, in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2014), pp. 4436–4441.

49. M. Schneider, W. Ertel, Robot learning by demonstration with local gaussian process regression, in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2010), pp. 255–260.

50. J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, K. Goldberg, Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics, in *the Proceeding of the 2017 Conference on Robotics: Science and Systems* (2017).

51. E. Todorov, Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco, in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 6054–6061.

52. T. Erez, Y. Tassa, E. Todorov, Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx, in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 4397–4404.

53. F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, S. Cotin, SOFA: A multi-model framework for interactive physical simulation, in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, Y. Payan, Ed. (Springer, 2012), vol. 11.

54. P. C. Horak, J. C. Trinkle, On the similarities and differences among contact models in robot simulation. *IEEE Robot. Autom. Lett.* **4**, 493–499 (2019).

55. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 23–30.

56. Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, D. Fox, Closing the sim-to-real loop: Adapting simulation randomization with real world experience, in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019), pp. 8973–8979.

57. D. E. Whitney, J. M. Rourke, Mechanical behavior and design equations for elastomer shear pad remote center compliances. *J. Dyn. Sys. Meas. Control* **108**, 223–232 (1986).

58. E. Brown, N. Rodenberg, J. Amend, A. Mozeika, E. Steltz, M. R. Zakin, H. Lipson, H. M. Jaeger, Universal robotic gripper based on the jamming of granular material. *Proc. Natl. Acad. Sci. U.S.A.* **107**, 18809–18814 (2010).

59. R. Ma, A. Dollar, Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption. *IEEE Robot. Autom. Mag.* **24**, 32–40 (2017).

60. M. G. Catalano, G. Grioli, A. Serio, E. Farnioli, C. Piazza, A. Bicchi, Adaptive synergies for a humanoid robot hand, in *Proceedings of the 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE, 2012), pp. 7–14.

61. M. Bianchi, G. Averta, E. Battaglia, C. Rosales, M. Bonilla, A. Tondo, M. Poggiani, G. Santaera, S. Ciotti, M. G. Catalano, A. Bicchi, Touch-based grasp primitives for soft hands: Applications to human-to-robot handover tasks and beyond, in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 7794–7801.

62. B. Kulis, Metric learning: A survey. *Found. Trends Mach. Learn.* **5**, 287–364 (2013).

63. C. B. Choy, J. Gwak, S. Savarese, M. Chandraker, Universal correspondence network, in *Proceedings of the 2016 Conference on Advances in Neural Information Processing System* (Curran Associates Inc., 2016), pp. 2414–2422.

64. T. Schmidt, R. Newcombe, D. Fox, Self-supervised visual descriptor learning for dense correspondence. *IEEE Robot. Autom. Lett.* **2**, 420–427 (2017).

65. K. Zakka, A. Zeng, J. Lee, S. Song, Form2fit: Learning shape priors for generalizable assembly from disassembly. arXiv:1910.13675 [cs.RO] (25 February 2020).

66. P. Zech, E. Renaudo, S. Haller, X. Zhang, J. Piater, Action representations in robotics: A taxonomy and systematic classification. *Int. J. Robot. Res.* **38**, 518–562 (2019).

67. M. Lázaro-Gredilla, D. Lin, J. S. Guntupalli, D. George, Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs. *Sci. Robot.* **4**, eaav3150 (2019).

68. B. Araki, K. Vodrahalli, T. Leech, C.-I. Vasile, M. Donahue, D. Rus, Learning to plan with logical automata, in *Proceedings of the 2019 Robotics: Science and Systems XV (RSS)* (2019); www.roboticsproceedings.org/rss15/p64.html.

69. Merriam-Webster.com Dictionary, Explain; www.merriam-webster.com/dictionary/explain.

70. M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Van de Wiele, V. Mnih, N. Heess, J. T. Springenberg, Learning by playing solving sparse reward tasks from scratch, in *Proceedings of the 35th International Conference on Machine Learning (ICML)* (PMLR, 2018), pp. 4344–4353.

71. K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, M. Riedmiller, Learning an embedding space for transferable robot skills, in *International Conference on Learning Representations* (ICLR, 2018); https://openreview.net/forum?id=rk07ZXZRb.

72. B. Eysenbach, A. Gupta, J. Ibarz, S. Levine, Diversity is all you need: Learning skills without a reward function, in *International Conference on Learning Representations* (ICLR, 2019); https://openreview.net/forum?id=SJx63jRqFm.

73. A. Levy, G. Konidaris, R. Platt, K. Saenko, Learning multi-level hierarchies with hindsight, in *International Conference on Learning Representations* (ICLR, 2019); https://openreview.net/forum?id=ryzECoAcY7.

74. M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. M. Grew, J. Tobin, P. Abbeel, W. Zaremba, Hindsight experience replay, in *Proceedings of the 2017 Advances in Neural Information Processing Systems* (Curran Associates, 2017), pp. 5048–5058.

75. G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: A review. *Neural Netw.* **113**, 54–71 (2019).

76. C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in *Proceedings of the 34th International Conference on Machine Learning (ICML)* (PMLR, 2017), vol. 70, pp. 1126–1135.

77. C. Finn, T. Yu, T. Zhang, P. Abbeel, S. Levine, One-shot visual imitation learning via meta-learning, in *Proceedings of the 1st Conference on Robot Learning* (PMLR, 2017), vol. 78, pp. 357–368.

78. K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), pp. 3722–3731.

79. K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, V. Vanhoucke, Using simulation and domain adaptation to improve efficiency of deep robotic grasping, in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 4243–4250.

80. Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**, 2096–2030 (2016).

81. D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning. *Mach. Learn.* **15**, 201–221 (1994).

82. D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, S. Levine, Scalable deep reinforcement learning for vision-based robotic manipulation, in *Proceedings of the 2nd Conference on Robot Learning* (PMLR, 2018), pp. 651–673.

83. D. Morrison, P. Corke, J. Leitner, Learning robust, real-time, reactive robotic grasping. *Int. J. Robot. Res.* **39**, 183–201 (2020).

84. L. Li, A. Talwalkar, Random search and reproducibility for neural architecture search. arXiv:1902.07638 [cs.LG] (30 July 2019).

85. D. Floreano, P. Dürr, C. Mattiussi, Neuroevolution: From architectures to learning. *Evol. Intell.* **1**, 47–62 (2008).

86. A. Aly, G. Guadagni, J. B. Dugan, Derivative-free optimization of neural networks using local search, in *Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)* (IEEE, 2019), pp. 0293–0299.

87. J. Kober, J. A. Bagnell, J. Peters, Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* **32**, 1238–1274 (2013).

88. O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. M. Grew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, W. Zaremba, Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **39**, 3–20 (2020).

89. OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. M. Grew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, L. Zhang, Solving rubik's cube with a robot hand. arXiv:1910.07113 [cs.LG] (16 October 2019).

90. A. Hans, D. Schneegaß, A. M. Schäfer, S. Udluft, Safe exploration for reinforcement learning, in *Proceedings of the 16th European Symposium on Artificial Neural Networks* (d-side publishing, 2008), pp. 143–148.

91. X. Li, C. Vasile, C. Belta, Reinforcement learning with temporal logic rewards, in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 3834–3839.

92. M. Pecka, T. Svoboda, Safe exploration techniques for reinforcement learning - an overview, in *Modelling and Simulation for Autonomous Systems*, J. Hodicky, Ed. (Springer, Cham, 2014), pp. 357–375.

93. T. M. Moldovan, P. Abbeel, Safe exploration in Markov decision processes, in *Proceedings of the 29th International Conference on Machine Learning (ICML)* (Omnipress, 2012), pp. 1451–1458.

94. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).

95. C. J. C. H. Watkins, P. Dayan, Q-learning. *Mach. Learn.* **8**, 279–292 (1992).

96. M. Tokic, Adaptive E-greedy exploration in reinforcement learning based on value differences, in *KI 2010: Advances in Artificial Intelligence*, R. Dillmann, J. Beyerer, U. D. Hanebeck, T. Schultz, Eds. (Springer, 2010), pp. 203–210.

97. M. Tokic, G. Palm, Value-difference based exploration: Adaptive control between epsilon-greedy and softmax, in *KI 2011: Advances in Artificial Intelligence*, J. Bach, S. Edelkamp, Eds. (Springer, 2011), pp. 335–346.

98. R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992).

99. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms. arXiv:1707.06347 [cs.LG] (28 August 2017).

100. A. Stooke, P. Abbeel, Accelerated methods for deep reinforcement learning. arXiv:1803.02811 [cs.LG] (10 January 2019).

101. A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen, S. Legg, V. Mnih, K. Kavukcuoglu, D. Silver, Massively parallel methods for deep reinforcement learning, arXiv:1507.04296 [cs.LG] (16 July 2015).

102. D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, D. Silver, Distributed prioritized experience replay. arXiv:1803.00933 [cs.LG] (2 March 2018).

103. M. Kalakrishnan, P. Pastor, L. Righetti, S. Schaal, Learning objective functions for manipulation, in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2013), pp. 1331–1336.

104. B. D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration. *Robot. Auton. Syst.* **57**, 469–483 (2009).

105. H. Ravichandar, A. S. Polydoros, S. Chernova, A. Billard, Recent advances in robot learning from demonstration. *Annu. Rev. Control Robot. Auton. Syst.* **3**, 297–330 (2020).

106. E. Ugur, H. Girgin, Compliant parametric dynamic movement primitives. *Robotica* **38**, 457–474 (2020).

107. J. Silvério, Y. Huang, L. Rozo, S. Calinon, D. G. Caldwell, Probabilistic learning of torque controllers from kinematic and force constraints, in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 1–8.

108. T. Zhang, Z. M. Carthy, O. Jow, D. Lee, X. Chen, K. Goldberg, P. Abbeel, Deep imitation learning for complex manipulation tasks from virtual reality teleoperation, in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 5628–5635.

109. Y. Yang, Y. Aloimonos, C. Fermüller, E. E. Aksoy, Learning the semantics of manipulation action, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (Association for Computational Linguistics, 2015), vol. 1, pp. 676–686.

110. A. Vakanski, I. Mantegh, A. Irish, F. Janabi-Sharifi, Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping. *IEEE Trans. Syst. Man Cybern. B Cybern.* **42**, 1039–1052 (2012).

111. K. Zampogiannis, Y. Yang, C. Fermüller, Y. Aloimonos, Learning the spatial semantics of manipulation actions through preposition grounding, in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 1389–1396.

112. Y. Yang, Y. Li, C. Fermuller, Y. Aloimonos, Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (AAAI Press, 2015), pp. 3686–3692.

113. R. Johansson, J. Flanagan, Tactile Sensory Control of Object Manipulation in Humans, in *The Senses, A Comprehensive Reference*, R. H. Masland, T. D. Albright, T. D. Albright, R. H. Masland, P. Dallos, D. Oertel, S. Firestein, G. K. Beauchamp, M. C. Bushnell, A. I. Basbaum, J. H. Kaas, E. P. Gardner, Eds. (Academic Press, 2008), vol. 6, chap. 5, pp. 67–86.

114. S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning" (Technical Report 98-11, Iowa State Univ., 1998).

115. J. D. Gammell, S. S. Srinivasa, T. D. Barfoot, Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs, in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2015), pp. 3067–3074.

116. B. Burns, O. Brock, Sampling-based motion planning using predictive models, in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2005), pp. 3120–3125.

117. N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, D. Vallejo, Obprm: An obstacle-based PRM for 3d workspaces, in *Robotics: The Algorithmic Perspective*, P. K. Agarval, L. E. Kavraki, M. T. Mason, Eds. (A. K. Peters, 1998), pp. 155–168.

118. ShadowRobot, Shadow dexterous hand; www.shadowrobot.com/dexterous-hand-series/.

119. J. Amend, H. Lipson, The Jamhand: Dexterous manipulation with minimal actuation. *Soft Robot.* **4**, 70–80 (2017).

120. A. Billard, D. Kragic, Trends and challenges in robot manipulation. *Science* **364**, eaat8414 (2019).

Citation: J. Cui, J. Trinkle, Toward next-generation learned robot manipulation. *Sci. Robot.* **6**, eabd9461 (2021).

Science Robotics

Toward next-generation learned robot manipulation

Jinda Cui and Jeff Trinkle

ARTICLE TOOLS	http://robotics.sciencemag.org/content/6/54/eabd9461
RELATED CONTENT	http://robotics.sciencemag.org/content/robotics/6/54/eabd8803.full http://robotics.sciencemag.org/content/robotics/4/26/eaav3123.full http://robotics.sciencemag.org/content/robotics/4/37/eaay4663.full
REFERENCES	This article cites 45 articles, 3 of which you can access for free http://robotics.sciencemag.org/content/6/54/eabd9461#BIBL
PERMISSIONS	http://www.sciencemag.org/help/reprints-and-permissions

Use of this article is subject to the Terms of Service