ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Recent Advances in Robot Learning from Demonstration

Harish Ravichandar,[1,*] Athanasios S. Polydoros,[2,*] Sonia Chernova,[1,†] and Aude Billard[2,†]

[1]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, Georgia 30332, USA; email: harish.ravichandar@gatech.edu, chernova@gatech.edu

[2]Learning Algorithms and Systems Laboratory, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland; email: athanasios.polydoros@epfl.ch, aude.billard@epfl.ch

Annu. Rev. Control Robot. Auton. Syst. 2020. 3:297–330

First published as a Review in Advance on December 6, 2019

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-100819-063206

*These authors contributed equally to this article

†These authors contributed equally to this article

Keywords

learning from demonstration, imitation learning, programming by demonstration, robot learning

Abstract

In the context of robotics and automation, learning from demonstration (LfD) is the paradigm in which robots acquire new skills by learning to imitate an expert. The choice of LfD over other robot learning methods is compelling when ideal behavior can be neither easily scripted (as is done in traditional robot programming) nor easily defined as an optimization problem, but can be demonstrated. While there have been multiple surveys of this field in the past, there is a need for a new one given the considerable growth in the number of publications in recent years. This review aims to provide an overview of the collection of machine-learning methods used to enable a robot to learn from and imitate a teacher. We focus on recent advancements in the field and present an updated taxonomy and characterization of existing methods. We also discuss mature and emerging application areas for LfD and highlight the significant challenges that remain to be overcome both in theory and in practice.

1. INTRODUCTION

In the context of robotics and automation, learning from demonstration (LfD) is the paradigm in which robots acquire new skills by learning to imitate an expert (1–4). In this article, we review recent advances in LfD and their implications for robot learning.

The development of novel robot tasks via traditional robot programming methods requires expertise in coding and a significant time investment. Furthermore, traditional methods require users to explicitly specify the sequence of actions or movements a robot must execute in order to accomplish the task at hand. Methods that utilize motion planning (5, 6) aim to overcome some of the burdens of traditional robot programming by eliminating the need to specify the entire sequence of low-level actions, such as trajectories. However, motion-planning methods still require the user to specify higher-level actions, such as goal locations and sequences of via points. Such specifications are also not robust to changes in the environment and require respecification or programming.

An attractive aspect of LfD is its ability to facilitate nonexpert robot programming. LfD accomplishes this by implicitly learning task constraints and requirements from demonstrations, which can enable adaptive behavior. Put another way, LfD enables robots to move away from repeating simple prespecified behaviors in constrained environments and toward learning to take optimal actions in unstructured environments without placing a significant burden on the user. As a result, LfD approaches have the potential to significantly benefit a variety of industries, such as manufacturing (7) and health care (8), in which it can enable subject-matter experts with limited robotics knowledge to efficiently and easily program and adapt robot behaviors.

Research interest in teaching robots by example has been steadily increasing over the past decade. Indeed, as seen in **Figure 1**, the field has seen considerable growth in the number of publications in recent years. The field remains diverse in terms of both its algorithms (see Sections 2

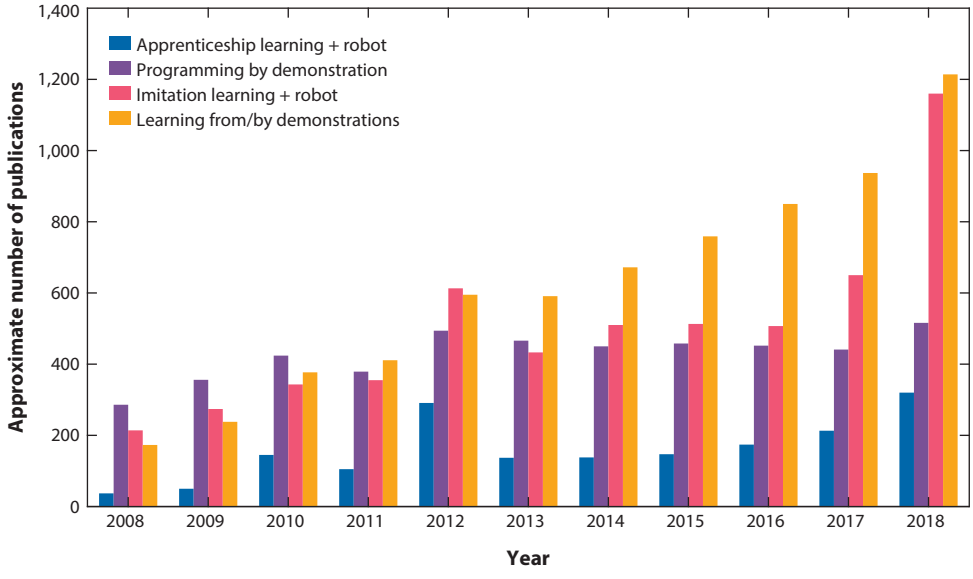

Figure 1

Consistent growth in the number of publications concerning learning from demonstration over the past decade, as reflected by the trend in the number of search results on Google Scholar that contain key related phrases.

and 3) and its terminology (see **Figure 1**). Imitation learning, programming by demonstration, and behavioral cloning are other popular phrases used to describe the process of learning from demonstrations. In this review, we use the term LfD to encompass the field as a whole.

Different flavors of learning—supervised, reinforcement, and unsupervised—have been utilized to solve a plethora of problems in robot learning. The choice among the different flavors is not trivial and is guided by the requirements and restrictions associated with the problem of interest. LfD in particular can be viewed as a supervised-learning problem since it attempts to acquire new skills from external teachers (available demonstrations). The choice of LfD over other robot learning methods is particularly compelling when ideal behavior can be neither scripted (as is done in traditional robot programming) nor easily defined as optimizing a known reward function (as is done in reinforcement learning), but can be demonstrated. Learning only from demonstrations does limit the performance of LfD techniques to the abilities of the teacher; to tackle this problem, LfD methods can be combined with exploration-based methods.

As with any learning paradigm, LfD presents its share of challenges and limitations. The underlying machine-learning methods have a significant impact on the types of skills that can be learned through LfD, and therefore many of the challenges in LfD follow directly from challenges faced by machine-learning techniques. Such challenges include the curse of dimensionality, learning from very large or very sparse data sets, incremental learning, and learning from noisy data. Besides these challenges, when LfD is applied to control a physical robotic system, it also inherits challenges from control theory, such as the predictability of the response of the system under external disturbances, ensuring stability when in contact, and convergence guarantees. Finally, and perhaps most importantly, as LfD relies on demonstrations by an external agent (usually a human), it must overcome a variety of challenges well known in human–robot interaction, such as finding an adequate interface, variability in human performance, and variability in knowledge across human subjects. And while humans may differ from one another, they differ less significantly (at least physically) than robots do. Hence, LfD is not only sensitive to who teaches the robot but also quite dependent on the platform (robot plus interface) used.

Multiple surveys of LfD, which focus on different subsets of the field, have been published over the past two decades (1–4, 7, 9–11); these surveys are representative of the evolution of the field. Schaal (1) presented the first survey of LfD, focusing on imitation and trajectory-based skills. Osa et al. (11) presented a more recent account of the same topic that focused on a more algorithmic perspective. Billard et al. (2, 10) presented broad syntheses of LfD that incorporated elements of human–robot interaction and framed the inquiry from the perspective of four core questions for the field: how, what, when, and whom to imitate. Argall et al. (3) and Chernova & Thomaz (4) presented taxonomies of LfD, characterizing types of demonstration inputs and variations on learning methods. Bohg et al. (12) presented a detailed survey on the general topic of grasp synthesis that included a taxonomy of LfD methods used in that particular area. Finally, Zhu & Hu (7) discussed how the need for adaptable manufacturing robotic system has led to the application of LfD methods in industrial assembly tasks.

While there have been many surveys of the field in the past, there is a need for a new one given the steady growth of the domain. This review therefore offers an overview of the collection of machine-learning methods used to enable a robot to learn from and imitate a teacher. We focus on recent advancements in the field and present an updated taxonomy and characterization of existing methods. We also touch on mature and emerging application areas for LfD and seek to underline the significant challenges that remain to be overcome both in theory and in applications.

The organization of the article is as follows. Section 2 categorizes the LfD literature based on how demonstrations are acquired, and Section 3 categorizes it based on what is learned. Section 4 then identifies the various application areas, and Section 5 presents the strengths and limitations

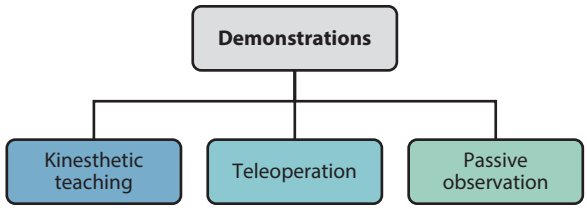

Figure 2

Categorization of learning-from-demonstration methods based on the demonstrations they utilize.

of the various flavors of LfD. Section 6 discusses open problems and challenges. Finally, Section 7 provides some concluding remarks.

2. CATEGORIZATION BASED ON DEMONSTRATIONS

One of the first decisions to be made when designing an LfD paradigm is the technique by which demonstrations will be performed. Although this choice may appear straightforward, it depends on multiple factors and has a wide range of possible consequences. Most generally, as shown in **Figure 2**, demonstration approaches fall into three categories: kinesthetic teaching, teleoperation, and passive observation. **Table 1** summarizes the key similarities and differences among these categories in terms of ease of demonstration, ability to handle a large number of degrees of freedom, and whether it is easy to map the demonstrations on the configuration or operational space of the robot. Below, we discuss each demonstration approach in detail.

2.1. Kinesthetic Teaching

Kinesthetic teaching, which is applied primarily to manipulation platforms (13–18), enables the user to demonstrate by physically moving the robot through the desired motions (19) (**Figure 3*a***). The state of the robot during the interaction is recorded through its onboard sensors (e.g., joint angles and torques), resulting in training data for the machine-learning model. Kinesthetic teaching is popular for manipulators, including lightweight industrial robots, due to its intuitive approach and minimal user training requirements. Additionally, kinesthetic teaching requires only the development and maintenance of the robot hardware and does not rely on additional sensors, interfaces, or inputs. Finally, recording demonstrations directly on the robot using its integrated sensors eliminates the correspondence problem (3, 20), thereby simplifying the machine-learning process.

Kinesthetic teaching does have several limitations. The quality of the demonstrations depends on the dexterity and smoothness of the human user, and even with experts, data obtained through this method often require smoothing or other postprocessing techniques. In addition, kinesthetic teaching is most effective for manipulators due to their relatively intuitive form factor; its

Table 1 Characteristics of learning-from-demonstration methods categorized based on the demonstrations they utilize

Demonstration	Ease of demonstration	High DOFs	Ease of mapping
Kinesthetic teaching	✓		✓
Teleoperation		✓	✓
Passive observation	✓	✓	

Abbreviation: DOF, degree of freedom.

Figure 3

Examples of the three categories of robot demonstrations.

applicability is limited on other platforms, such as legged robots or robotic hands, where demonstrations are more challenging to perform.

2.2. Teleoperation

Teleoperation is another widely used demonstration input and has been applied to trajectory learning (21), task learning (22), grasping (23), and high-level tasks (24). It requires an external input to the robot through a joystick, graphical user interface, or other means (**Figure 3*b***). A wide range of interfaces have been explored, including haptic devices (25, 26) and virtual-reality interfaces (27–29). Unlike kinesthetic teaching, teleoperation does not require the user to be copresent with the robot, allowing LfD techniques to be applied in remote settings (22). Additionally, access to remote demonstrators opens the opportunity for crowdsourcing demonstrations at a large scale (30–33).

Limitations of teleoperation include the additional effort required to develop the chosen input interface, in some cases a more lengthy user training process, and the availability of input hardware (e.g., a virtual-reality headset) when required. However, as a result of these efforts, teleoperation can be applied to more complex systems, including robotic hands (34), humanoids (28), and underwater robots (35). Teleoperation can also be easily coupled with simulation to further facilitate data collection and experimentation at scale, as is often required within reinforcement-learning frameworks.

2.3. Passive Observation

The third demonstration approach is for the robot to learn from passive observation of the user (36–38). In this approach, the user performs the task using their own body, sometimes instrumented by additional sensors to facilitate tracking (**Figure 3*c***). The robot takes no part in the execution of the task and acts only as a passive observer. This type of learning, often referred to as imitation learning (3, 20), is particularly easy for the demonstrator, requiring almost no training to perform. It is also highly suitable for application to robots with many degrees of freedom and nonanthropomorphic robots, where kinesthetic teaching is difficult. However, the machine-learning problem is complicated by the need to either encode or learn a mapping from the human's actions to those executable by the robot. Occlusions, rapid movement, and sensor noise in the observations of human actions present additional challenges for this type of task demonstration. Despite the challenges, learning from passive observation has been successfully applied to various tasks, such as collaborative furniture assembly (39), autonomous driving (40), table-top actions (41, 42), and knot tying (43). In some cases, the human user is not observed directly, and only the objects in the scene are tracked (31, 44).

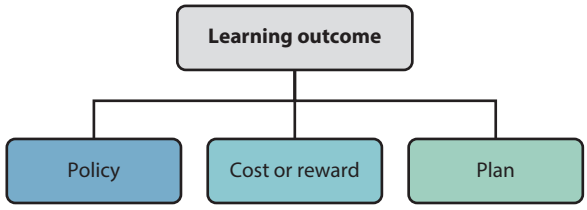

Figure 4

Categorization of learning-from-demonstration methods based on learning outcome.

2.4. Active and Interactive Demonstrations

Once the demonstration method has been selected, there remain the choices of what to demonstrate and whether demonstrations should be requested by the robot or initiated by the human. Chernova & Thomaz (4) covered techniques for managing the interaction, such as active learning (39, 45–48) and corrective demonstrations (49, 50), in greater detail. Amershi et al. (51) presented several case studies that illustrate the importance of studying the users of intelligent systems in order to improve both user experience and robot performance. While more general than the field of LfD, a comprehensive notion of interactive task learning was introduced by Laird et al. (52), who emphasized the importance of intelligent agents taking a more active role in the learning process and attempting to reason about the instruction from which they learn. Further examples of work in this area include modeling and use of social cues during learning (53), reasoning about the availability of human demonstrators and how to behave in their absence (54), how to ask for help (55), and techniques for human-aided feature selection during task learning (56).

3. CATEGORIZATION BASED ON LEARNING OUTCOME

An important categorization of LfD methods can be achieved by answering a fundamental question: What is learned? The learning outcome of LfD methods depends on the level of abstraction that is appropriate, and thus chosen, for the problem of interest. For instance, while one task might require learning the low-level behavior of the robot, another might require extracting the sequence dynamics of a set of basic actions and/or their interdependence. Specifically, as shown in **Figure 4**, learning methods can be divided into three broad categories, each with a different learning outcome: policy, cost or reward, and plan. Choosing which learning outcome to pursue is not trivial and depends on the task and the associated constraints. **Table 2** summarizes the key similarities and differences among these choices in terms of their ability to learn low-level policies, handle continuous action spaces, compactly represent the learned skill, plan over long time horizons, and learn complex tasks composed of several subtasks and sequencing constraints. In the sections below, we discuss each learning outcome and its underlying assumptions, providing subcategorizations where appropriate.

Table 2 Characteristics of learning-from-demonstration methods categorized based on learning outcome

Learning outcome	Low-level control	Action space continuity	Compact representation	Long-horizon planning	Multistep tasks
Policy	✓	✓	✓		
Cost or reward	✓	✓		✓	
Plan			✓	✓	✓

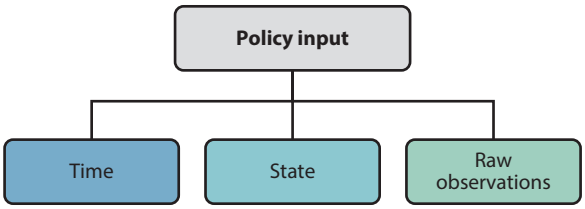

Figure 5

Categorization of policy-learning methods based on the policy's input space.

3.1. Learning Policies from Demonstrations

Policy-learning methods assume that there exists a direct and learnable function (i.e., the policy) that generates desired behavior. We define a policy as a function that maps available information onto an appropriate action space. A policy can be mathematically represented as $\pi : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ is the domain (the input space of the policy) and $\mathcal{Y}$ is its codomain (action space). The goal of policy-learning methods is to learn a policy $\pi(\cdot)$ that generates state trajectories $\{x(t)\}$ that are similar to those demonstrated by the expert.

Recently, the connection between adversarial learning (57) and inverse reinforcement learning (IRL) (see Section 3.2.2) was leveraged to propose generative adversarial imitation learning (GAIL) (58). Though closely related, GAIL cannot be classified as an IRL algorithm because it does not learn a reward function. Instead, GAIL can be considered a policy-learning algorithm because it learns policies directly from demonstrations.

Policy-learning methods can be further categorized into different types based on various considerations. In Sections 3.1.1–3.1.3, we present three such subcategorizations, each based on an important design choice, and discuss specific properties of the categorized methods.

3.1.1. Policy input. A key design choice for policy-learning methods involves identifying the appropriate input to the policy. This choice must sufficiently capture the information that is necessary for generating optimal actions. Each policy-learning method can be viewed as having one of three choices for input: time, state, and raw observations (see **Figure 5**). **Table 3** summarizes the key characteristics associated with each of these choices in terms of ease of policy design, whether performance guarantees can be provided, robustness to spatiotemporal perturbations during execution, the diversity of tasks that can be learned, and computational efficiency.

3.1.1.1. Time. The first class of methods in this categorization utilizes time as the primary input to the policy (13, 19, 59–65). The policy learned by these methods can be denoted by a function $\pi : (\mathcal{X} = \mathbb{R}^+) \rightarrow \mathcal{Y}$ that maps time onto an appropriate action space. Demonstrations for such methods consist of time–action pairs. The underlying assumption in these methods is that it is possible to take optimal actions based primarily on initial conditions and the current

Table 3 Characteristics of policy-learning methods categorized based on the policy's input space

Policy input	Ease of design	Performance guarantees	Robustness to perturbations	Task variety	Algorithmic efficiency
Time	✓	✓			✓
State		✓	✓		✓
Raw observations	✓		✓	✓	

time without relying on additional feedback. Therefore, time-based policies are analogous to an open-loop controller since they do not depend on feedback from the policy's output or state.

Time-based models are capable of identifying and capturing important features that are anchored in time (19, 60, 63). With time as the primary input, important temporal constraints, such as when to precisely follow a trajectory, can be identified by utilizing heteroscedastic stochastic processes (59, 65) or geometric structures (13). Furthermore, learning time-based trajectory distributions is helpful in generalizing to new scenarios involving different initial, final, and via points (62). Time has also been utilized to encode the correlations between multiple modalities (61).

A notable limitation of time-driven policies is the lack of robustness to perturbations. Since the policy depends primarily on time as the input, any changes in the environment or perturbations are not taken into account. Even when such changes are detected, it is not trivial to warp the time or phase variable in order to adapt to perturbations. This limits the application of the method to cases where actions are driven only by time and the system will not be subject to unexpected perturbations.

3.1.1.2. State. A popular category of policy-learning methods assumes direct access to the state and utilizes it as the input to the policy (15, 16, 26, 66–83). Existing literature in LfD has explored a wide variety of choices for what is considered to be the state, such as end-effector position (81), velocity (70), orientation (82), force (66, 76), joint angles (75), and torques (79). An underlying idea in all these methods is that the state serves as feedback about the robot, and sometimes the task, at any given moment. Therefore, from the control-theory perspective, state-based policies correspond to closed-loop controllers because each action affects the next state and thus the input of the policy. A state-driven policy can be represented as $\pi : (\mathcal{X} = \mathcal{S}) \rightarrow \mathcal{Y}$, where $\mathcal{S}$ is the relevant state space. These methods require demonstrations consisting of state–action pairs, which are used to learn the mapping that specifies the optimal action. Note that the state information can include time either implicitly (e.g., 14, 84) or explicitly (e.g., 75, 85).

What constitutes the state is often manually specified. By contrast, when an appropriate and tractable state space is not known a priori, some methods attempt to learn the most appropriate state space. Learning the appropriate state or feature space can be accomplished in either an unsupervised or supervised manner. Unsupervised approaches (86–88) rely on techniques such as clustering and dimensionality reduction. Supervised approaches, on the other hand, utilize tools such as hierarchical task networks (24) and neural networks (89, 90).

State-based policies enable the robot to take into account the current state of the task and thus allow it to be reactive. Furthermore, state-based policies offer a compact representation for a variety of skills by directly mapping the state space onto an appropriate action space. Despite their many advantages, state-based policies can depend on a high-dimensional input space, which creates a more challenging machine-learning problem compared with time-based representations. Another challenge of state-based policies is that theoretical guarantees, such as stability (70, 82), are more challenging to prove and realize practically than they are for time-based systems.

3.1.1.3. Raw observations. Unlike the two categories of methods discussed above, a third category does not rely on a succinct input representation. Methods in this category learn to map raw observations to actions and are often referred to as end-to-end methods. End-to-end policies can be denoted by $\pi : (\mathcal{X} = \mathcal{O}) \rightarrow \mathcal{Y}$, where $\mathcal{O}$ is the space of raw observations. Such methods require functions that can approximate complex relationships, along with significant computational resources. Thanks to the recent developments in deep learning and the availability of impressive computational power, a class of end-to-end methods has recently been introduced (41, 43, 91–93). End-to-end LfD methods determine the appropriate action directly based on high-dimensional

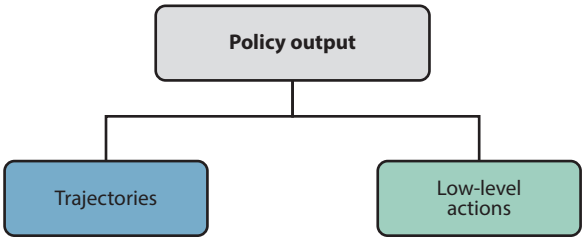

Figure 6

Categorization of policy-learning methods based on policy's output space.

raw observations from sensors, such as cameras. This approach is particularly useful when learning tasks in which a succinct input representation either is unknown or does not exist.

End-to-end policies inherit the limitations of deep-learning approaches, which require a large amount of data and computational resources for training. Furthermore, due to the large number of nonlinear transformations present in the models, the derivation of theoretical guarantees is very challenging. These limitations inhibit the use of end-to-end approaches in safety-critical applications and in scenarios where labeled data are hard to acquire.

3.1.2. Policy output. Policy-learning methods can also be categorized based on the space onto which the policy maps its inputs. As shown in **Figure 6**, the two primary categories of policy outputs are trajectories and low-level actions. **Table 4** summarizes the key similarities and differences among methods with different policy outputs in terms of their ability to operate without knowledge of the robot model, learn platform-independent policies, learn policies for underactuated robots, and learn end-to-end policies.

3.1.2.1. Learning trajectories. Trajectory-learning methods learn policies that map onto the trajectory space. These methods encode trajectories of certain variables of interest by extracting patterns from the available demonstrations. Examples of such variables include end-effector position (13, 17, 64, 68, 70, 80, 81, 94–96), end-effector pose (67, 82, 97), end-effector force (61, 72, 74, 77), and joint state (62, 98). When reproducing the skill, these methods generate trajectories based on the initial state and (in some cases) the current state. The policy of a trajectory-learning method can be represented as $\pi : \mathcal{X} \rightarrow (\mathcal{Y} = \mathcal{T})$, where $\mathcal{T}$ is the space of trajectories.

Several approaches to trajectory learning, such as dynamical systems (70, 82, 96, 97, 99, 100) and probabilistic inference (13, 62, 81), have been studied in the literature. When trajectories are learned using dynamical systems as models, the demonstrated trajectories are assumed to be solutions of the dynamical systems. In probabilistic-inference-based methods, the demonstrated trajectories are assumed to represent samples of an underlying stochastic process, and thus reproductions are obtained by sampling the learned distribution over trajectories after potentially conditioning on initial, via, and final points.

Table 4 Characteristics of policy-learning methods categorized based on the output

Policy output	Model free	Platform independence	Underactuated robots	End to end
Trajectories	✓	✓		
Low-level actions			✓	✓

Trajectory-learning methods rely on low-level controllers to execute the generated reference trajectories. They have proven to be particularly well suited for overactuated systems, such as redundant manipulators, for which kinematic feasibility is relatively easier to achieve. Furthermore, trajectory-learning methods do not require knowledge of robot dynamics or repeated data collection. As a result, trajectory-learning methods are one of the most popular classes of LfD methods.

Learning trajectories can be achieved in two different spaces: the joint space and the operational space. Learning in each of those spaces has its own limitations. In the joint space, the learned policy depends on the kinematic chain of the robot and thus cannot be directly transferred to another robotic system. On the other hand, learning in the operational space does not guarantee that the generated motions are feasible or that singularities are avoided (81).

3.1.2.2. Learning low-level actions. Methods in this category learn policies that directly generate appropriate low-level control signals, such as joint torques, given the current state of the task. Such policies are mathematically represented as $\pi : \mathcal{X} \rightarrow (\mathcal{Y} = \mathcal{A})$, where $\mathcal{A}$ is the robot's low-level action space.

A common approach for methods that map to the robot action space is the derivation of velocities or accelerations from the LfD policy. Those are propagated to the low-level controller of the robot, which converts them—through the inverse dynamics model—to commands such as joint torques. An alternative approach is to directly learn the necessary joint torques or forces at each state, which also allows the tuning of impedance parameters, resulting in compliant control. In the context of human–robot interaction, research has focused on learning the required torques (26, 79, 83, 101, 102) and stiffness parameters (69, 103) for producing compliant motions. Nevertheless, pursuing low-level learning outcomes brings challenges, such as obtaining accurate force and torque demonstrations, determining compliant axes, and estimating the robot's physical properties. Research has also explored learning the required stiffness variations from physical human–robot interaction (63, 72, 104). In these approaches, demonstrations are used to learn the axes on which the system is allowed to be compliant.

Pursuing low-level control inputs as the learning outcome, while well suited for underactuated systems, has limitations that are associated with the collection of demonstrations and knowledge of the robot's physical properties. For instance, the applicability of low-level torque control is limited by the need for the robot's inverse dynamics model. Such models depend on the physical properties of the robot and are likely to change due to wear and tear or changes in the task of interest (105).

3.1.3. Policy class. Another dimension of categorization can be achieved by examining the class of mathematical functions to which the policy belongs (see **Figure 7**). The appropriate policy class for a given problem is an important design choice that has significant implications. **Table 5** summarizes such implications associated with this choice in terms of the learned policy's ability to learn skills that depend on temporal context, handle temporal perturbations, consistently and reliably repeat policy rollouts, and encode multimodal behavior.

3.1.3.1. Deterministic versus stochastic policies. The primary categories of the policy class are deterministic and stochastic. The choice between the two policy classes is made by considering a fundamental question: Given a particular context, does a demonstration represent the singular or absolute ideal behavior, or a sample from a distribution of ideal behaviors?

Deterministic policies assume that a singular optimal action exists for every situation and attempts to extract all such optimal actions from the demonstrations. Mathematically, deterministic policies are given by $\pi : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ are the policy's input and output

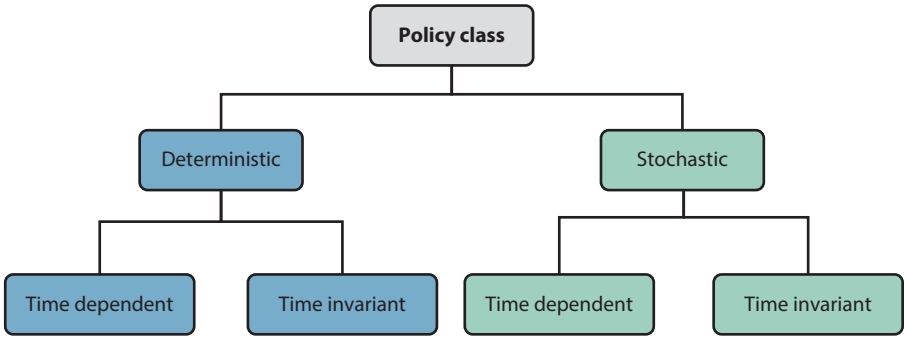

Figure 7

Categorization of policy-learning methods based on the mathematical class of the policy.

spaces, respectively. As a result, these policies generate predictable or repeatable behaviors. This property has helped methods with deterministic policies provide strong theoretical guarantees on the performance, such as global asymptotic convergence (70, 78, 82).

Stochastic policies sample a behavior from a learned distribution of behaviors during each execution (62, 100, 106). They can be mathematically represented as $\pi(x) \sim \mathcal{P}(y|x)$, where $x \in \mathcal{X} \subseteq \mathbb{R}^m$ is the input to the policy, $y \in \mathcal{Y} \subseteq \mathbb{R}^n$ is the policy's output, and $\mathcal{P}$ is a conditional probability distribution. An advantage of stochastic policies is their ability to capture inherent uncertainty. For instance, while traversing around an obstacle, it might be equally optimal to stay to the right or the left of the obstacle. While stochastic policies capable of encoding multimodal distributions can effectively capture this uncertainty, deterministic policies cannot and need to resolve the seemingly conflicting paths. Furthermore, a deterministic policy might result in unsafe behaviors such as traversing the average path, which will lead to collisions. While not as prevalent as in deterministic algorithms, theoretical guarantees for stochastic policies, such as probabilistic convergence, have recently been proposed (100).

3.1.3.2. Time-dependent versus time-invariant policies. Irrespective of stochasticity, the policy class can be subcategorized based on the stationarity of the policy. Specifically, policies can be either time dependent or time invariant (see **Figure 7**).

As the name would suggest, time-dependent policies rely on time, potentially in addition to any other form of feedback (59, 60, 62, 65). Indeed, they are potentially more expressive than their time-invariant counterparts and are capable of capturing strategies that vary with time and involve time-based requirements. For instance, time-dependent policies provide a straightforward mechanism to ensure that the reproduced behavior aligns with the demonstration in terms of speed and duration (64).

Table 5 Characteristics of policy-learning methods categorized based on the mathematical class of the policy

Policy class	Temporal context	Robustness to temporal perturbations	Repeatability	Multimodal behavior
Deterministic and time dependent	✓		✓	
Deterministic and time invariant		✓	✓	
Stochastic and time dependent	✓			✓
Stochastic and time invariant		✓		✓

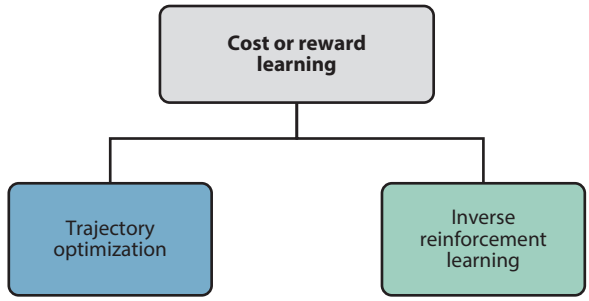

Figure 8

Categorization of cost- and reward-learning methods.

Time-invariant policies, on the other hand, are capable of capturing general strategies that are independent of time and are more robust to intermittent spatiotemporal perturbations during execution. For instance, when reproducing a reaching motion, time-invariant methods are more robust to interruptions in their motions because they do not have to rely on an internal clock and can depend on state feedback (e.g., 70, 82). However, note that certain time-invariant policies, such as dynamical movement primitives (99), have an implicit dependence on time. This implicit dependence provides a mechanism to capture sequential aspects of a behavior using phase variables, which are not required to be synchronized with the wall-clock time.

3.2. Learning Cost and Reward Functions from Demonstrations

Methods that fall into this category assume that ideal behavior results from the optimization of a hidden function, known as a cost function or reward function. The goal of such methods is then to extract the hidden function from the available demonstrations. Subsequently, the robot reproduces the learned behavior by optimizing an identified function. Indeed, learning cost or reward functions requires making certain assumptions about the task and the environment. Below, we discuss two classes of methods that make two different sets of assumptions regarding the task and the environment: trajectory optimization and IRL (**Figure 8**).

3.2.1. Trajectory optimization. Traditional trajectory-optimization methods (5, 107, 108) were originally introduced to generate smooth and efficient trajectories for robots to move between any two points in space. In such methods, the cost function is prespecified in order to produce trajectories with desired properties. However, when attempting to learn skills from demonstrations, no cost functions are provided. To circumvent this issue, LfD has been successfully applied to trajectory-optimization-based methods by assuming that the expert minimizes a hidden cost function when demonstrating a skill (95, 109–111). Demonstrations are thus viewed as optimal solutions and are used to infer this underlying cost function. In order to enable this inference, it is common practice to assume that the hidden cost function takes a certain parametric form, and its parameters are to be learned from demonstrations. The choice of the form is based on what is assumed to be relevant to the task. For instance, demonstrations and reproductions can be viewed as minimizing a Hilbert norm, the parameters of which are to estimated (110).

3.2.2. Inverse reinforcement learning. IRL is a popular class of methods that learn a cost or reward function (112). These methods typically assume that the demonstrator optimizes an unknown reward function. Demonstrations are thus utilized to learn the hidden reward function, after which classical reinforcement-learning approaches can be used to compute optimal actions.

In the continuous case, the methods become very similar to inverse optimal control, where a hidden objective function for optimal control is estimated from demonstrations (113–116), and these terms are often used equivalently in the literature. Depending on the complexity of the problem of interest, IRL methods assume that the reward function is either linear (114, 117–119) or nonlinear (58, 116, 120).

Since there might be multiple reward functions that optimally explain the available demonstrations, IRL is referred to as an ill-posed problem. To arrive at a unique reward function, IRL methods consider different additional optimization goals, such as maximum margin (112, 117, 121, 122) and maximum entropy (58, 118, 123, 124). In maximum-margin-based IRL, the reward function is identified by maximizing the difference between the best policy and all other policies. On the other hand, maximum-entropy-based IRL identifies a distribution that maximizes the entropy subject to constraints regarding feature expectation matching that ensure that the reproductions are similar to the demonstrations.

Robot learning methods that perform IRL can be further categorized into model-based and model-free methods depending on what prior knowledge is assumed. Model-based IRL approaches (117, 119, 123, 124) exploit the knowledge of the transition probabilities (i.e., the system model) to ensure that the reward function and policy are accurately updated. However, the system model might not always be available. To circumvent this challenge, recent methods have explored model-free IRL that utilizes sampling-based methods to recover the reward function. Specific techniques that have been studied include minimizing the relative entropy between the state–action trajectory distributions of a baseline policy and the learned policy (118), considering a direct policy search (115), and alternating between reward and policy optimization (116). Recent work comparing model-based and model-free learning from human data has suggested that model-based approaches have numerous computational advantages (125).

3.2.3. Limitations. Similar to the policy-learning approaches discussed in Section 3.1, cost- or reward-learning approaches present their own set of challenges. For instance, learning cost or reward functions could be sensitive to suboptimal demonstrations. The choice of structure for the cost or reward function is not trivial and can significantly influence performance. Furthermore, such choices tend to depend on the task of interest and thus cannot be applied to other applications without modification. Compared with policy-learning approaches, cost- and reward-learning methods introduce more structure but are less compact representations of a task since they rely on the structure to derive the final policy. Finally, since IRL optimizes the identified reward function via reinforcement learning, it inherits the limitations of reinforcement learning, such as the need for a large number of episodes to converge (126) and the existence of transition models that can be hard to derive (105).

3.3. Learning Plans from Demonstrations

This category includes methods that learn at the highest levels of task abstraction. The task of interest is assumed to be performed according to a structured plan made up of several subtasks or primitive actions (i.e., a task plan) (127–135). Structured task plans typically encode the patterns and constraints in the subtasks or primitives and take the robot from an initial state to the goal state. Given the current state of the task, a task plan provides the most appropriate subtask to execute next among a finite set of subtasks.

Complex tasks, such as assembling a device or packing a lunch, are often demonstrated in continuous sessions. Such demonstrations contain several subtasks that exhibit specific ordering constraints and interdependence. Thus, segmentation plays a crucial role in task plan learning.

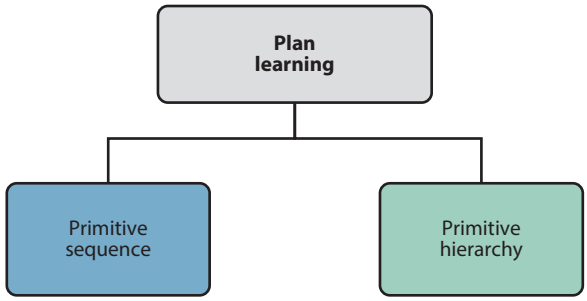

Figure 9

Categorization of learning-from-demonstration methods that learn task plans.

Several methods for automated segmentation of demonstrations into subtasks, based either on the similarity of component subtasks (136–141) or on the occurrence of indicative events (25, 142, 143), have been explored.

Task plans typically include preconditions and postconditions for each subtask. Preconditions specify the conditions that need to be satisfied before a subtask or a step begins. Similarly, postconditions provide the conditions that are expected to be satisfied following a successful execution of a subtask. The extracted task plan, including pre- and postconditions, is subsequently used to generate actions for each subtask.

Methods that learn task plans from demonstration can learn either a primitive sequence (131, 132, 140, 144) or a primitive hierarchy (127, 145–148) (see **Figure 9**). Primitive sequences represent simple ordering and the associated constraints of the steps involved in a task. Primitive hierarchies, on the other hand, incorporate high-level structured instructions and provide a plan that can capture variable sequencing and nondeterminism. For instance, hierarchies can be used to capture the fact that certain subtasks could be carried out in any order (24). LfD has also been applied to learning a plan schedule from expert demonstrations (149).

3.4. Pursuing Multiple Learning Outcomes Simultaneously

As opposed to the majority of the methods presented in Sections 3.1–3.3, which focus on one type of learning outcome, it is possible to learn complex behaviors at multiple levels of abstraction by pursuing multiple learning outcomes simultaneously. A number of recent approaches attempt to learn from demonstrations at different levels of abstraction (129, 131, 135, 148, 150–154).

The above-mentioned methods utilize demonstrations to combine trajectory learning and task plan learning. For instance, Niekum et al. (152) used unstructured demonstrations and interactive corrections to learn finite state machines that are made of several trajectory models. Pastor et al. (135) introduced the notion of associated skill memories, which capture sensory associations in addition to the primitives and utilize such associations to effectively sequence the learned primitives in order to achieve manipulation goals. Kroemer et al. (148) utilized demonstrations to learn a stochastic model of the phases and transitions involved in manipulation skills and then used reinforcement learning to learn how to transition between the previously identified learned phases. Mohseni-Kabir et al. (129) had human demonstrators provide narrations as they demonstrated the task, then used the demonstrated trajectories to learn trajectory-level primitives and used the associated narrations to infer the boundaries between primitives and recover the task plan. Finally, Krishnan et al. (155) approximated tasks with long time horizons as a sequence of local reward functions and subtask transition conditions, which are learned via IRL.

4. APPLICATIONS OF LEARNING FROM DEMONSTRATION

The various LfD approaches introduced above have been successfully applied to numerous applications in robotics. It is important to note that, while several LfD approaches are evaluated on specific platforms or application areas, the underlying algorithms are not necessarily limited to those platforms or application areas. In this section, we discuss widely used application platforms and areas and provide relevant examples.

4.1. Manipulators

Manipulators are perhaps the most popular application platform for LfD methods. Below, we discuss specific application areas where LfD for manipulators has proven effective.

4.1.1. Manufacturing. Training of manipulators from demonstrations is common for manufacturing applications due to the need for production adaptability and transferability. Indeed, it is more profitable to employ robots that can learn from a few examples than to use ones that require significant reprogramming efforts and result in downtime. LfD has been studied to teach manipulators a variety of skills related to manufacturing since the 1980s (156, 157). Popular examples include pick and place (79), peg insertion (116), polishing (158), grasping (33, 34, 159), and assembly operations (7, 38, 39). The most common approach for introducing demonstrations in manufacturing applications is through kinesthetic teaching. In cases where the learning objective is a low-level trajectory, policy-learning approaches are used. A plan is learned from demonstrations in cases where the system has to learn high-level action sequences, such as the steps involved in an assembly task. The input of the policy depends on whether perturbations are expected during the operation of the robot; therefore, state-based policies are used when perturbations are expected, and time-based policies are used when tasks involve time-sensitive constraints.

4.1.2. Assisting and health-care robotics. Another popular area of application for manipulators is assisting and health-care robotics. LfD has proven to be an effective way to teach manipulators important movement skills necessary when assisting people and providing health-care services. Specifically, LfD methods have helped teach manipulators a variety of skills, such as feeding (160), physical rehabilitation (161), robotic surgery (85, 162), assisting children with cerebral palsy (163), supporting daily activities (164), hand rehabilitation (165), handing over objects (166), and motion planning for rehabilitation (8). Manipulators that are expected to provide assistance and health care are also usually taught by kinesthetic demonstrations, similar to the manufacturing applications. In addition, assistive robots are expected to operate in closer proximity to humans compared with manufacturing cases. This increases the need for safe operation, which can be satisfied by providing convergence and stability guarantees for the learned policy.

4.1.3. Human–robot interaction. In addition to learning to autonomously perform tasks in different areas, LfD has been explored to provide manipulators with the ability to collaborate with humans in close proximity (14, 63, 66, 75, 84, 102, 133). Effective collaboration requires the generation of the desired robotic movements that are complementary to those of the human. To this end, observations of human–human interaction are used to teach manipulators how to cooperate with a human partner so as to enhance the fluency and safety of the collaboration. Therefore, such applications require LfD methods that are able to compensate for perturbations during the operation of the robot. The most common approach to achieve this is by learning a policy function whose inputs are the states of the robot. Human–robot interaction applications also present the need for compliant manipulators, which is typically met by learning the appropriate joints' torques (79) and stiffness and damping parameters (103).

4.2. Mobile Robots

In addition to manipulators, LfD has enjoyed considerable success in a variety of mobile robots. In the sections below, we discuss specific mobile platforms and applications in which LfD-based algorithms have demonstrated their suitability for use in mobile robots.

4.2.1. Ground vehicles. Autonomous navigation has numerous applications, including autonomous cars, warehouse automation, and autonomous defense vehicles. In fact, one of the earliest applications of LfD, introduced in 1991, involved the autonomous navigation of a car (167). This seminal work described a neural-network-based algorithm that can learn the mapping between inputs and a discrete action space from human driving data. Since this early success, LfD-based autonomous navigation for ground vehicles has been attempted using IRL (117, 121, 168), interactive learning (169), active learning (170), adversarial learning (171), and end-to-end learning (40, 172). Demonstrations in such platforms are usually provided through teleoperation by joystick for small vehicles. Nevertheless, kinesthetic teaching can be applied in large vehicles, such as cars, where a human takes the driver's seat and demonstrates a desired behavior by driving (173).

4.2.2. Aerial vehicles. LfD has also been applied successfully to the problem of autonomous aerial navigation. A popular example involved teaching a helicopter to perform complicated maneuvers, such as flips, rolls, and tic-tocs (21). LfD has been demonstrated to be effective in teaching aerial vehicles to navigate in cluttered environments (174). Furthermore, recent advances in deep learning have fueled the development of end-to-end LfD methods for aerial vehicles (175, 176). Demonstrations for training flying robots are usually done through teleoperation. Hence, training pertains primarily to teaching a desired trajectory, while the stability of the robot is handled by traditional control approaches.

4.2.3. Bipedal and quadrupedal robots. In bipedal and quadrupedal robots, LfD has been used primarily for locomotion. LfD approaches have been successfully used to enable bipedal robots to learn to walk (177, 178) and optimize their gaits (179). LfD has also enjoyed some successes with quadrupedal locomotion (122, 180, 181). The demonstrations for training bipedal robots can be introduced either by teleoperation or by observation, where the gait of a human demonstrator can be captured by appropriate sensors and transferred to the robot by deriving a correspondence mapping (182).

4.2.4. Underwater vehicles. Finally, LfD has also been demonstrated to be useful to underwater robots. LfD algorithms have been shown to be effective in facilitating underwater applications, such as underwater valve turning (183), underwater robot teleoperation (35, 184, 185), and marine data collection (186). Similarly to what is done with flying and humanoid robots, applications of LfD to train mobile robots use teleoperation in combination with control methods to ensure stability.

5. STRENGTHS AND LIMITATIONS OF LEARNING FROM DEMONSTRATION

Sections 2 and 3 discussed a rich variety of techniques and approaches utilized in LfD. In this section, we discuss the strengths and limitations inherent in the choice of LfD.

5.1. Strengths of Learning from Demonstration

The field of LfD offers several advantages. Indeed, different types of LfD algorithms provide different benefits, making them suitable for different scenarios and problems (see **Tables 2–5**). Below, we identify particular strengths of LfD methods in general.

5.1.1. Nonexpert robot programming. In general, LfD has been successful in solving problems for which optimal behavior can be demonstrated but not necessarily succinctly specified in mathematical form (e.g., a reward function). For instance, while it is straightforward to demonstrate how to drive a car, it is very challenging to describe an all-encompassing reward function for optimal driving. This observation explains one of the most attractive aspects of LfD: It enables easier programming of robots. Specifically, LfD reduces the barriers to entry into robot programming by empowering nonexperts to teach a variety of tasks to robots, without the need for significant software development or subject-matter expertise.

5.1.2. Data efficiency. Several LfD methods typically learn from a small number of expert demonstrations. For instance, trajectory-learning methods typically utilize fewer than 10 demonstrations to learn new skills (62, 70, 95), and high-level task learning is feasible even with just a single demonstration (24). Reinforcement-learning-based approaches, on the other hand, typically optimize a specified reward function instead of demonstrations to learn new skills. Since reinforcement-learning methods employ a trial-and-error approach to discover optimal policies, they tend to be significantly less efficient than LfD approaches that utilize expert demonstrations (187). This property of LfD lends itself to solving problems in high-dimensional state spaces and is considered effective in addressing the so-called curse of dimensionality. In an effort to leverage the benefits of LfD's data efficiency, researchers have demonstrated that LfD can be combined with reinforcement learning to improve sample efficiency (188–195).

5.1.3. Safe learning. Since LfD utilizes expert demonstrations, the robot can be better incentivized to stay within safe or relevant regions of the state space, especially when compared with techniques that require significant exploration, such as reinforcement learning. This is because demonstrations provide a way to assess the safety or risk associated with regions of the state space (e.g., 196–198). Furthermore, several LfD methods provide and utilize measures of uncertainty associated with different parts of the state space (e.g., 62, 81, 100), enabling communication of the system's confidence to the user. This property is particularly relevant in safety-critical applications, such as those involving close-proximity interactions with humans. Indeed, several LfD approaches have been proposed to learn how to interact with a human user (62, 63, 69, 72, 75, 79, 104). Admittedly, not all LfD methods can guarantee that the robot will stay within safe or known regions, and some encounter unknown regions due to compounding factors (169). However, recent advances provide ways to recognize and handle such scenarios (for further discussion, see Section 6.1).

5.1.4. Performance guarantees. One of the significant factors that enable the widespread adoption of technology is reliability. Within the context of LfD, reliability could be achieved by providing theoretical guarantees on an algorithm's ability to consistently and successfully perform the task. Over the past decade, several LfD approaches have proven to be capable of providing such guarantees. For instance, numerous dynamical-systems-based trajectory-learning methods provide strong convergence guarantees (70, 78, 80, 82, 99).

5.1.5. Platform independence. As identified in Section 4, LfD has been successfully applied not only to multiple domains, but also to a variety of platforms, such as manipulators, mobile robots, underwater vehicles, and aerial vehicles. A particular reason for this diversity in application platform is LfD's ability to acquire and exploit expert demonstrations and to learn policies, cost or reward functions, and plans that are platform independent. With the selection of a suitable common representation for the task, several LfD methods have proven to be applicable to a wide range of platforms while requiring only minimal modification and the availability of low-level controllers. For instance, the dynamical movement primitives (99) algorithm has been utilized in a variety of platforms, including manipulators (152), robotic hands (67, 199), humanoids (200), and aerial vehicles (201).

5.2. Limitations of Learning from Demonstration

In addition to the strengths identified above, the choice of LfD over other robot learning approaches is accompanied by a few limitations that are inherent to the field. Below, we discuss such limitations, which stem from LfD's core assumptions and approaches. Note that we differentiate between the inherent deficiencies of LfD identified below and the exciting challenges and directions for future research discussed in Section 6.

5.2.1. Demonstrating complex behaviors. LfD necessitates an interface through which an expert can demonstrate the behavior. The choice of such an interface directly affects several factors, including the demonstrator comfort, the applicability to specific robotic platforms, and the correspondence between the operational spaces of the demonstrator and the robotic system. For instance, kinesthetic demonstrations are usually limited to robotic manipulators and are unsuitable for platforms such as a humanoid robot, because it is very challenging to physically manipulate robotic platforms with joints, which belong to different kinematic chains and must be operated simultaneously to achieve the desired behavior. On the other hand, visual demonstrations, while being the most intuitive type for the user, suffer from the correspondence problem, as the demonstrator's actuation space differs from that of the robotic system. Overcoming this problem requires a mapping between those spaces, which may be hard to provide due to the differences in motion constraints and dimensionality between the two systems. Furthermore, learning from observation requires the existence of a perception system for capturing demonstrations and thus inherits limitations relevant to computer vision, such as occlusions, pose estimation, and noise. In the case of teleoperation, capturing demonstrations requires a mechanical or software interface, which can be challenging to design.

5.2.2. Reliance on labeled data. As mentioned above, LfD relies on supervised-learning-based techniques that extract information from labeled data. This reliance limits the ability of LfD to acquire new skills when a sufficient amount of labeled data is unavailable. Indeed, as pointed out above, LfD is known to be data efficient and has proven to be capable of learning a wide variety of skills with limited data. However, such data efficiency comes from systematic design choices in state, feature, and action spaces; imitation goals; and so on. In scenarios where such prior or expert knowledge is unavailable, LfD would indeed require a considerable amount of demonstrated data. This issue is exacerbated by the fact that acquiring a large number of demonstrations for robot LfD requires a significant investment of time and resources.

5.2.3. Suboptimal and inappropriate demonstrators. It is common for LfD methods to assume that the available demonstrations are optimal and are provided by an expert user. This assumption is carried over from supervised-learning techniques that rely on accurately labeled data

to achieve good performance, and it implicitly answers the important question of whom to imitate. However, it might not hold in a variety of scenarios, such as when learning from novices, crowdsourcing demonstrations, and utilizing noisy sensors. Existing solutions to suboptimal data are limited mostly to filtering suboptimal demonstrations (202) or identifying suboptimal demonstrations when the majority of the demonstrations are optimal (203). When most or all of the demonstrations are suboptimal, it might not be feasible to utilize LfD methods without other sources of information that reveal the quality of the demonstrations. This limitation is thus not likely to be overcome by LfD alone. However, the use of other learning approaches, such as reinforcement learning, in conjunction with LfD can provide the robot with the necessary tools to also learn from experience when examples are insufficient.

6. CHALLENGES AND FUTURE DIRECTIONS

The field of LfD has generated innumerable insights into the science and art of teaching robots to perform a variety of tasks across multiple domains. However, several challenges remain to be addressed if we aspire to enable robots to fluently and efficiently learn from humans and operate in challenging environments. In this section, we identify and discuss some of the most prominent hurdles and the promising directions for future research that might overcome them.

6.1. Generalization

Cognitive psychology defines generalization in learning as the ability of an organism to effectively respond to new stimuli that are similar to those previously encountered (204). Indeed, generalization is seen as one of the central properties of animal cognition that helps in dealing with novelty and variability (205).

Taking inspiration from the natural world, researchers in the area of machine learning have studied generalization in artificial systems extensively. Indeed, generalization is at the core of machine learning—the ability to generalize differentiates systems that learn from those that memorize the training data. Machine-learning algorithms tackle generalization by making certain assumptions about the problem. However, some of those assumptions do not hold in several robotic systems and applications. Below, we provide specific examples and discuss the challenges involved in teaching robots to generalize.

Supervised-learning algorithms assume that the training and testing data are independent and identically distributed. However, since demonstrations seldom cover all parts of the problem space, the robot is likely to discover scenarios where the input distributions are different from those of the demonstrations (206). This results in a phenomenon known as the covariate shift or compounding of error.

One solution to avoid the compounding of error leverages interactions with the user to acquire corrective demonstrations as the robot veers off the training distribution and encounters new states while executing a learned policy (169). However, this solution assumes the extended and continuous availability of the demonstrator to provide corrections at appropriate times. Furthermore, executions of suboptimal policies in the initial stages of learning might not be suitable for physical systems with safety-critical constraints.

A distinction can be made between intratask and intertask generalization. The former refers to an algorithm's ability to generalize to novel conditions within a particular task, such as new initial and goal locations (70, 82, 99), new via points (62), and new object locations (15, 207). Intertask generalization, on the other hand, refers to the ability to generalize the learned skill to new but similar tasks, which is known as skill transfer.

Recently, meta-learning or multitask learning algorithms (e.g., 153, 208) have been introduced to learn meta-policies that can be quickly fine-tuned within a few iterations of training with data from the new task. Multitask learning, however, assumes access to demonstrations from multiple tasks.

We need learning methods that can extrapolate acquired information to novel scenarios and, more importantly, estimate the suitability of the learned policy to new scenarios. Put differently, the robot must identify when to extrapolate and when to request user intervention.

Another crucial challenge related to generalization involves the selection of the hypothesis class (the set of all possible functions that we consider when learning). Indeed, the choice of hypothesis class has profound impacts on the performance of the algorithm. It is still unclear how to systematically choose the hypothesis class for a given skill or a set of skills such that it would help effectively resolve the bias–variance trade-off.

6.2. Hyperparameter Selection

The challenge of hyperparameter selection originates from the machine-learning method used to learn the mapping between the representation and the action. The vast majority of machine-learning methods face this challenge, but research in the field of LfD needs to overcome this issue by providing methodologies that can automatically choose the hyperparameters. Since one of the motives for applying LfD is to enable nonexperts to program a robotic system, the need to manually tune the model significantly decreases the ability of nonexperts to use such a methodology.

Hyperparameters can be found in many policy representations. For instance, end-to-end representations are usually modeled by neural networks. In those cases, the hyperparameters are the numbers of hidden layers and neurons per layer. Modeling highly nonlinear relationships may require a large number of hidden units, while less complex relationships require fewer units. In the case of time-based representations, methods based on dynamical movement primitives are widely used, and an important hyperparameter of this model is the number of radial basis functions. Highly nonlinear motions require modeling more radial basis functions. Nevertheless, if the demonstrated motion is not complex and a relatively large number of radial basis functions are used, then the model will also capture noise introduced from demonstrations, resulting in overfitting. State–action representations are usually modeled as a Gaussian mixture model. In these models, the choice for the number of Gaussian components affects the complexity of the estimated functions, similarly to the above-mentioned cases. For learning a cost or reward function through IRL, hyperparameters, such as the type of function, must be set alongside the number of desired features. In the case of learning high-level task plans, the hyperparameters are the number of actions, sequences, and states. Similarly to policy-learning methods, setting inappropriate choices for hyperparameters can result in a bad fit.

Automated hyperparameter selection could be achieved using machine-learning methods with learning rules that provide a trade-off between model fitting and complexity such as Gaussian processes. Nevertheless, their usability as policy functions is limited due to their computational complexity and difficulty of guaranteeing system stability. Regarding the use of Gaussian mixture models as policy functions, a more intuitive way to determine the optimal number of components is the use of Bayesian variational inference (209). Moreover, determining the hyperparameters when learning high-level plans could be achieved automatically through clustering methods that would be capable of separating different sequences and actions without user intervention.

6.3. Evaluation and Benchmarking

Evaluating the performance of an LfD method is challenging due to the multiple factors that must be taken into consideration. First of all, the learning method should be data and computationally efficient, and the outcome of the method must be smooth, without sharp changes, in order to minimize the risk of damage. Moreover, the method must provide stability guarantees, be able to generalize efficiently, and be able to solve the desired task with high repeatability. The majority of these criteria are usually included when evaluating LfD methods; nevertheless, some of them are not easily quantifiable, resulting in qualitative evaluations.

The smoothness of the provided plan is an important criterion (especially for low-level control) that is usually evaluated qualitatively. This evaluation usually includes plots of the generated trajectories, and the smoothness is visually examined. This type of analysis does not provide any quantitative information, which makes comparison across multiple LfD methods difficult. Another criterion that is usually evaluated qualitatively is the generalization ability. In this case, ground truth does not exist for the states that are not included in the demonstrations. Thus, the evaluation of generalization is performed either qualitatively, by plotting the plans generated from various initial states, or quantitatively, by measuring the ability of the method to successfully perform the task from unknown initial states.

By contrast, criteria such as repeatability and learning efficiency are easily quantifiable. Repeatability highlights how reliable the proposed method is for performing the desired task. Its evaluation is straightforward and usually involves a success or failure rate of task completions. The learning efficiency is a measurement of how much data and computational time the machine-learning method requires to converge to a solution. A learning method that does not require much data and thus can converge with a small number of demonstrations has a significant advantage in real-life applications. One approach to evaluate the convergence is by a data–error plot, which plots the number of demonstrations with respect to the error of the objective function. The convergence point corresponds to the number of demonstrations that do not cause significant changes to the objective function. Thus, this point can be determined by the elbow method (210). The computational complexity of the learning method is usually reported since it is important for time-sensitive applications.

The plethora of LfD approaches alongside the multiple evaluation criteria makes comparisons across methods extremely challenging. This fact underscores the need for benchmarks that can highlight the strengths and weaknesses of proposed approaches and provide insights into when each method should be utilized. Benchmarking LfD methods will require the design of a standard, which should include evaluation criteria, metrics, and tasks. The evaluation criteria should provide a comprehensive overview of the methods' strengths and weaknesses, and therefore they should include generalization ability, convergence, stability, and repeatability. To quantify generalization, a set of demonstrations can be divided into training and testing sets, where the methods are trained with the training demonstrations and used to predict the testing demonstrations. Thus, the prediction error on the testing set can be used as a representative metric of the algorithm's generalization ability. The convergence of the learning method can be evaluated by plots that illustrate the amount of training data in relation to the prediction error. The ability of the LfD method to stabilize the system should ideally be mathematically proven, but this may be extremely challenging or impossible for certain models. In those cases, stability could be empirically demonstrated by letting the system execute actions from a large number of states within the operational space and report the ratio of successful convergences. Regarding repeatability, a ratio of successful task completions can be used.

6.4. Other Challenges

In addition to the challenges identified above, several other challenges need to be addressed to realize efficient, practical, and scalable LfD algorithms. We identify a few of these challenges below.

6.4.1. Appropriate distance metric to minimize during reproduction. One of the primary goals of policy-learning methods is to imitate the demonstrator's behavior. To this end, all methods, in one way or another, aim to minimize the distance between the behavior of the robot and that of the demonstrator. Thus, the choice of the metric used to compute distances becomes vital. The most widely used measure of distance is the Euclidean distance. However, other generalized definitions of distance might be more appropriate for a given task (94, 110). This warrants further exploration into the use of different distance measures when computing the deviation between demonstrations and reproductions.

6.4.2. Simultaneous learning of low- and high-level behaviors. To learn complex tasks, it is important to learn both the high-level task plans and the low-level primitives and to effectively capture their interdependence. However, methods designed to learn high-level task plans typically assume that the low-level primitives are known a priori and are fixed, and methods for learning primitives ignore the existence of high-level task goals. Thus, a trivial combination of techniques at each level might not be effective owing to their incompatible and potentially conflicting goals. Indeed, over the past decade, methods that can simultaneously learn both low- and high-level behaviors have been explored (see Section 3.4). However, further research is needed to develop efficient methods for multilevel LfD that are generalizable and can apply to a wide variety of tasks.

6.4.3. Learning from multimodal demonstrations. Research in cognitive psychology suggests that utilizing multisensory stimuli enhances human perceptual learning (e.g., 211). Indeed, when we learn from others, we utilize a variety of multimodal information, including verbal and nonverbal cues, to make sense of what is being taught. Current methods for multimodal LfD are limited to learning from a small number of prespecified modalities (e.g., 26, 212). To effectively learn a wide variety of complex skills, we need methods that reason over demonstrations in multiple modalities, identify the most relevant demonstrations, and learn from them. Another challenging aspect of utilizing multimodal demonstrations is user comfort and accessibility. It is not clear how to acquire highly multimodal demonstrations without burdening the user by placing an overwhelming number of sensors. Effectively collecting multimodal demonstrations from remote users also remains challenging.

6.4.4. Learning from multiple demonstrators and cloud robotics. Most algorithms in LfD assume that there is a single optimal function (policy, cost/reward, or plan) to be learned. While this assumption is valid when there is a single demonstrator, it does not hold true in scenarios with multiple demonstrators. Different demonstrators tend to have different priorities and notions of optimal behavior. In these scenarios, it is important to disentangle the vital components of the skill from the demonstrators' idiosyncrasies. This challenge is especially important in the context of cloud and crowdsourced robotics, wherein multiple demonstrators can provide demonstrations of the same skill. Another aspect of learning from multiple demonstrators is the potential to learn from people with varying levels of expertise. In the context of IRL, learning from demonstrations that reflect different levels of expertise exposes a richer reward function and structure compared with learning from a single demonstrator or demonstrators with similar expertise (213).

7. CONCLUSION

This survey has provided an overview of methodologies, categorizations, applications, strengths, limitations, challenges, and future directions for research associated with the field of LfD. We have surveyed several aspects of the LfD pipeline. First, we identified the different approaches for acquiring demonstrations—kinesthetic teaching, teleoperation, and passive observations—and their associated characteristics, benefits, and limitations. Next, we introduced a comprehensive categorization of the abundance of learning algorithms based on an important design choice—the learning outcome. Specifically, we identified three learning outcomes—policy learning, cost or reward learning, and plan learning—and discussed the relative benefits of these choices. Furthermore, for each category of methods, we introduced detailed subcategorizations and identified their core assumptions and utility.

LfD methods have been successfully applied in various industries and to the vast majority of existing robotic platforms. This fact highlights the research field's potential to impact a wide variety of platforms, extending from manipulators in the manufacturing and health-care industries to mobile robots such as autonomous vehicles and legged robots. This widespread application is explained by the strengths of LfD methods, which include enabling nonexpert robot programming, supporting sample-efficient learning from a small number of demonstrations, and providing measures of confidence and theoretical guarantees on performance. In addition to these merits, we discussed the limitations that either are inherent to choosing LfD over other robot learning methods or are specific to and originate from the identified design choices.

As we continue to push the boundaries of what robots can learn from humans, the field faces important challenges and exciting avenues for further research. In this survey, we have identified open problems and challenges that span all categories of LfD. There is a need for LfD methods that can generalize efficiently across variations in several dimensions while remaining cognizant of the limits of the available knowledge. We also discussed the particular challenges associated with the automated tuning of hyperparameters, which have a significant impact on performance. To ensure that the algorithms we develop continue to push boundaries, we require consistent and objective evaluation and benchmarking protocols to compare and highlight the relative merits of different LfD approaches. Finally, if we are to realize effective ways of teaching robots to navigate the unstructured world that we live in, we must provide them with the capability to simultaneously learn at multiple levels of abstraction, learn from multimodal information, and learn from users with varying levels of expertise, both near and remote.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

This work was supported in part by the US Army Research Laboratory [grant W911NF-17-2-0181 from the Distributed and Collaborative Intelligent Systems and Technology Collaborative Research Alliance (DCIST CRA)], the US National Science Foundation Division of Information and Intelligent Systems (grant 1564080), the US Office of Naval Research (grant N000141612835), the NASA Space Technology Research Grants Program (Early Career Faculty grant), and the European Union projects Cognitive Interaction in Motion (CogiMon) (grant H2020-ICT-23-2014) and SecondHands (grant H2020-ICT-643950).

LITERATURE CITED

1. Schaal S. 1999. Is imitation learning the route to humanoid robots? *Trends Cogn. Sci.* 3:233–42
2. Billard A, Calinon S, Dillmann R, Schaal S. 2008. Robot programming by demonstration. In *Springer Handbook of Robotics*, ed. B Siciliano, O Khatib, pp. 1371–94. Berlin: Springer
3. Argall BD, Chernova S, Veloso M, Browning B. 2009. A survey of robot learning from demonstration. *Robot. Auton. Syst.* 57:469–83
4. Chernova S, Thomaz AL. 2014. *Robot Learning from Human Teachers*. San Rafael, CA: Morgan & Claypool
5. Schulman J, Ho J, Lee A, Awwal I, Bradlow H, Abbeel P. 2013. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems IX*, ed. P Newman, D Fox, D Hsu, pap. 31. N.p.: Robot. Sci. Syst. Found.
6. Zucker M, Ratliff N, Dragan AD, Pivtoraiko M, Klingensmith M, et al. 2013. CHOMP: covariant Hamiltonian optimization for motion planning. *Int. J. Robot. Res.* 32:1164–93
7. Zhu Z, Hu H. 2018. Robot learning from demonstration in robotic assembly: a survey. *Robotics* 7:17
8. Lauretti C, Cordella F, Guglielmelli E, Zollo L. 2017. Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics. *IEEE Robot. Autom. Lett.* 2:1375–82
9. Friedrich H, Kaiser M, Dillmann R. 1996. What can robots learn from humans?. *Annu. Rev. Control* 20:167–72
10. Billard AG, Calinon S, Dillmann R. 2016. Learning from humans. In *Springer Handbook of Robotics*, ed. B Siciliano, O Khatib, pp. 1995–2014. Berlin: Springer. 2nd ed.
11. Osa T, Pajarinen J, Neumann G, Bagnell JA, Abbeel P, Peters J. 2018. An algorithmic perspective on imitation learning. *Found. Trends Robot.* 7:1–179
12. Bohg J, Morales A, Asfour T, Kragic D. 2014. Data-driven grasp synthesis—a survey. *IEEE Trans. Robot.* 30:289–309
13. Ahmadzadeh SR, Kaushik R, Chernova S. 2016. Trajectory learning from demonstration with canal surfaces: a parameter-free approach. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots*, pp. 544–49. Piscataway, NJ: IEEE
14. Maeda GJ, Neumann G, Ewerton M, Lioutikov R, Kroemer O, Peters J. 2017. Probabilistic movement primitives for coordination of multiple humanrobot collaborative tasks. *Auton. Robots* 41:593–612
15. Pervez A, Lee D. 2018. Learning task-parameterized dynamic movement primitives using mixture of GMMs. *Intell. Serv. Robot.* 11:61–78
16. Shavit Y, Figueroa N, Salehian SSM, Billard A. 2018. Learning augmented joint-space task-oriented dynamical systems: a linear parameter varying and synergetic control approach. *IEEE Robot. Autom. Lett.* 3:2718–25
17. Elliott S, Xu Z, Cakmak M. 2017. Learning generalizable surface cleaning actions from demonstration. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 993–99. Piscataway, NJ: IEEE
18. Chu V, Fitzgerald T, Thomaz AL. 2016. Learning object affordances by leveraging the combination of human-guidance and self-exploration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction*, pp. 221–28. Piscataway, NJ: IEEE
19. Calinon S, Guenter F, Billard A. 2007. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans. Syst. Man Cybernet. B* 37:286–98
20. Nehaniv CL, Dautenhahn K. 2002. The correspondence problem. In *Imitation in Animals and Artifacts*, ed. K Dautenhahn, CL Nehaniv, pp. 41–61. Cambridge, MA: MIT Press
21. Abbeel P, Coates A, Ng AY. 2010. Autonomous helicopter aerobatics through apprenticeship learning. *Int. J. Robot. Res.* 29:1608–39
22. Peters RA, Campbell CL, Bluethmann WJ, Huber E. 2003. Robonaut task learning through teleoperation. In *2003 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 2806–11. Piscataway, NJ: IEEE
23. Whitney D, Rosen E, Phillips E, Konidaris G, Tellex S. 2020. Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality. In *Robotics Research: The 18th International Symposium ISRR*, ed. NM Amato, G Hager, S Thomas, M Torres-Torriti, pp. 335–50. Cham, Switz.: Springer

24. Mohseni-Kabir A, Rich C, Chernova S, Sidner CL, Miller D. 2015. Interactive hierarchical task learning from a single demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pp. 205–12. New York: ACM
25. Su Z, Kroemer O, Loeb GE, Sukhatme GS, Schaal S. 2016. Learning to switch between sensorimotor primitives using multimodal haptic signals. In *International Conference on Simulation of Adaptive Behavior*. pp. 170–82. Cham, Switz.: Springer
26. Kormushev P, Calinon S, Caldwell DG. 2011. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Adv. Robot.* 25:581–603
27. Rosen E, Whitney D, Phillips E, Ullman D, Tellex S. 2018. *Testing robot teleoperation using a virtual reality interface with ROS reality*. Paper presented at the 1st International Workshop on Virtual, Augmented, and Mixed Reality for Human-Robot Interaction, Chicago, Mar. 5
28. Zhang T, McCarthy Z, Jow O, Lee D, Chen X, et al. 2018. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation*, pp. 5628–35. Piscataway, NJ: IEEE
29. Spranger J, Buzatoiu R, Polydoros A, Nalpantidis L, Boukas E. 2018. Human-machine interface for remote training of robot tasks. In *2018 IEEE International Conference on Imaging Systems and Techniques*. Piscataway, NJ: IEEE. **https://doi.org/10.1109/IST.2018.8577081**
30. Whitney D, Rosen E, Tellex S. 2018. *Learning from crowdsourced virtual reality demonstrations*. Paper presented at the 1st International Workshop on Virtual, Augmented, and Mixed Reality for Human-Robot Interaction, Chicago, Mar. 5
31. Toris R, Kent D, Chernova S. 2015. Unsupervised learning of multi-hypothesized pick-and- place task templates via crowdsourcing. In *2015 IEEE International Conference on Robotics and Automation*, pp. 4504–10. Piscataway, NJ: IEEE
32. Mandlekar A, Zhu Y, Garg A, Booher J, Spero M, et al. 2018. RoboTurk: a crowdsourcing platform for robotic skill learning through imitation. In *Proceedings of the 2nd Conference on Robot Learning*, ed. A Billared, A Dragan, J Peters, J Morimoto, pp. 879–93. Proc. Mach. Learn. Res. Vol. 87. N.p.: PMLR
33. Kent D, Behrooz M, Chernova S. 2016. Construction of a 3D object recognition and manipulation database from grasp demonstrations. *Auton. Robots* 40:175–92
34. Aleotti J, Caselli S. 2011. Part-based robot grasp planning from human demonstration. In *2011 IEEE International Conference on Robotics and Automation*, pp. 4554–60. Piscataway, NJ: IEEE
35. Havoutis I, Calinon S. 2018. Learning from demonstration for semi-autonomous teleoperation. *Auton. Robots* 43:713–26
36. Kaiser J, Melbaum S, Tieck JCV, Roennau A, Butz MV, Dillmann R. 2018. Learning to reproduce visually similar movements by minimizing event-based prediction error. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics*, pp. 260–67. Piscataway, NJ: IEEE
37. Dillmann R. 2004. Teaching and learning of robot tasks via observation of human performance. *Robot. Auton. Syst.* 47:109–16
38. Vogt D, Stepputtis S, Grehl S, Jung B, Amor HB. 2017. A system for learning continuous human-robot interactions from human-human demonstrations. In *2017 IEEE International Conference on Robotics and Automation*, pp. 2882–89. Piscataway, NJ: IEEE
39. Hayes B, Scassellati B. 2014. Discovering task constraints through observation and active learning. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4442–49. Piscataway, NJ: IEEE
40. Codevilla F, Miiller M, López A, Koltun V, Dosovitskiy A. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation*, pp. 4693–700. Piscataway, NJ: IEEE
41. Pervez A, Mao Y, Lee D. 2017. Learning deep movement primitives using convolutional neural networks. In *2017 IEEE-RAS International Conference on Humanoid Robots*, pp. 191–97. Piscataway, NJ: IEEE
42. Liu Y, Gupta A, Abbeel P, Levine S. 2018. Imitation from observation: learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation*, pp. 1118–25. Piscataway, NJ: IEEE

43. Schulman J, Ho J, Lee C, Abbeel P. 2016. Learning from demonstrations through the use of non-rigid registration. In *Robotics Research*, pp. 339–54. Cham, Switz.: Springer
44. Fitzgerald T, McGreggor K, Akgun B, Thomaz A, Goel A. 2015. Visual case retrieval for interpreting skill demonstrations. In *International Conference on Case-Based Reasoning*, pp. 119–33. Cham, Switz.: Springer
45. Cakmak M, Thomaz AL. 2012. Designing robot learners that ask good questions. In *Proceedings of the Seventh Annual ACM/IEEE International Conference On Human-Robot Interaction*, pp. 17–24. New York: ACM
46. Cakmak M, Chao C, Thomaz AL. 2010. Designing interactions for robot active learners. *IEEE Trans. Auton. Mental Dev.* 2:108–18
47. Bullard K, Schroecker Y, Chernova S. 2019. Active learning within constrained environments through imitation of an expert questioner. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, ed. S Kraus, pp. 2045–52. Calif.: IJCAI
48. Bullard K, Thomaz AL, Chernova S. 2018. Towards intelligent arbitration of diverse active learning queries. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6049–56. Piscataway, NJ: IEEE
49. Gutierrez RA, Short ES, Niekum S, Thomaz AL. 2019. Learning from corrective demonstrations. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction*, pp. 712–14. Piscataway, NJ: IEEE
50. Bajcsy A, Losey DP, O'Malley MK, Dragan AD. 2018. Learning from physical human corrections, one feature at a time. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 141–49. New York: ACM
51. Amershi S, Cakmak M, Knox WB, Kulesza T. 2014. Power to the people: the role of humans in interactive machine learning. *AI Mag.* 35(4):105–20
52. Laird JE, Gluck K, Anderson J, Forbus KD, Jenkins OC, et al. 2017. Interactive task learning. *IEEE Intell. Syst.* 32:6–21
53. Saran A, Short ES, Thomaz A, Niekum S. 2019. Enhancing robot learning with human social cues. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction*, pp. 745–47. Piscataway, NJ: IEEE
54. Kessler Faulkner T, Gutierrez RA, Short ES, Hoffman G, Thomaz AL. 2019. Active attention-modified policy shaping: socially interactive agents track. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pp. 728–36. Richland, SC: Int. Found. Auton. Agents Multiagent Syst.
55. Kessler Faulkner T, Niekum S, Thomaz A. 2018. Asking for help effectively via modeling of human beliefs. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 149–50. New York: ACM
56. Bullard K, Chernova S, Thomaz AL. 2018. Human-driven feature selection for a robotic agent learning classification tasks from demonstration. In *2018 IEEE International Conference on Robotics and Automation*, pp. 6923–30. Piscataway, NJ: IEEE
57. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, et al. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, ed. Z Ghahramani, M Welling, C Cortes, ND Lawrence, KQ Weinberger, pp. 2672–80. Red Hook, NY: Curran
58. Ho J, Ermon S. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*, ed. DD Lee, M Sugiyama, UV Luxburg, I Guyon, R Garnett, pp. 4565–73. Red Hook, NY: Curran
59. Schneider M, Ertel W. 2010. Robot learning by demonstration with local Gaussian process regression. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 255–60. Piscataway, NJ: IEEE
60. Akgun B, Cakmak M, Jiang K, Thomaz AL. 2012. Keyframe-based learning from demonstration. *Int. J. Soc. Robot.* 4:343–55
61. Lin Y, Ren S, Clevenger M, Sun Y. 2012. Learning grasping force from demonstration. In *2012 IEEE International Conference on Robotics and Automation*, pp. 1526–31. Piscataway, NJ: IEEE

62. Paraschos A, Daniel C, Peters J, Neumann G. 2013. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems 26*, ed. CJC Burges, L Bottou, M Welling, Z Ghahramani, KQ Weinberger, pp. 2616–24. Red Hook, NY: Curran
63. Rozo L, Calinon S, Caldwell D, Jimenez P, Torras C, Jiménez P. 2013. Learning collaborative impedance-based robot behaviors. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 1422–28. Palo Alto, CA: AAAI Press
64. Osa T, Harada K, Sugita N, Mitsuishi M. 2014. Trajectory planning under different initial conditions for surgical task automation by learning from demonstration. In *2014 IEEE International Conference on Robotics and Automation*, pp. 6507–13. Piscataway, NJ: IEEE
65. Reiner B, Ertel W, Posenauer H, Schneider M. 2014. LAT: a simple learning from demonstration method. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4436–41. Piscataway, NJ: IEEE
66. Calinon S, Evrard P. 2009. Learning collaborative manipulation tasks by demonstration using a haptic interface. In *2009 International Conference on Advanced Robotics*. Piscataway, NJ: IEEE. **https://ieeexplore.ieee.org/document/5174740**
67. Pastor P, Hoffmann H, Asfour T, Schaal S. 2009. Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation*, pp. 763–68. Piscataway, NJ: IEEE
68. Calinon S, Sardellitti I, Caldwell DG. 2010. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 249–54. Piscataway, NJ: IEEE
69. Calinon S, Florent D, Sauser EL, Caldwell DG, Billard AG. 2010. Learning and reproduction of gestures by imitation: an approach based on hidden Markov model and Gaussian mixture regression. *IEEE Robot. Autom. Mag.* 17(2):44–45
70. Khansari-Zadeh SM, Billard A, Mohammad Khansari-Zadeh S, Billard A, Khansari-Zadeh SM, Billard A. 2011. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Trans. Robot.* 27:943–57
71. Rozo L, Jiménez P, Torras C. 2011. Robot learning from demonstration of force-based tasks with multiple solution trajectories. *IEEE 15th International Conference on Advanced Robotics*, pp. 124–29. Piscataway, NJ: IEEE
72. Kronander K, Billard A. 2012. Online learning of varying stiffness through physical human-robot interaction. In *2012 IEEE International Conference on Robotics and Automation*, pp. 1842–49. Piscataway, NJ: IEEE
73. Herzog A, Pastor P, Kalakrishnan M, Righetti L, Bohg J, et al. 2014. Learning of grasp selection based on shape-templates. *Auton. Robots* 36:51–65
74. Li M, Yin H, Tahara K, Billard A. 2014. Learning object-level impedance control for robust grasping and dexterous manipulation. In *2014 IEEE International Conference on Robotics and Automation*, pp. 6784–91. Piscataway, NJ: IEEE
75. Amor HB, Neumann G, Kamthe S, Kroemer O, Peters J. 2014. Interaction primitives for human-robot cooperation tasks. In *2014 IEEE International Conference on Robotics and Automation*, pp. 2831–37. Piscataway, NJ: IEEE
76. Kober J, Gienger M, Steil JJ. 2015. Learning movement primitives for force interaction tasks. In *2015 IEEE International Conference on Robotics and Automation*, pp. 3192–99. Piscataway, NJ: IEEE
77. Lee AX, Lu H, Gupta A, Levine S, Abbeel P. 2015. Learning force-based manipulation of deformable objects from multiple demonstrations. In *2015 IEEE International Conference on Robotics and Automation*, pp. 177–84. Piscataway, NJ: IEEE
78. Neumann K, Steil JJ. 2015. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robot. Auton. Syst.* 70:1–15
79. Denisa M, Gams A, Ude A, Petric T. 2016. Learning compliant movement primitives through demonstration and statistical generalization. *IEEE/ASME Trans. Mechatron.* 21:2581–94
80. Perrin N, Schlehuber-Caissier P. 2016. Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Syst. Control Lett.* 96:51–59

81. Rana MA, Mukadam M, Ahmadzadeh SR, Chernova S, Boots B. 2017. Towards robust skill generalization: unifying learning from demonstration and motion planning. In *Proceedings of the 1st Annual Conference on Robot Learning*, ed. S Levine, V Vanhoucke, K Goldberg, pp. 109–18. Proc. Mach. Learn. Res. Vol. 78. N.p.: PMLR
82. Ravichandar H, Dani A. 2018. Learning position and orientation dynamics from demonstrations via contraction analysis. *Auton. Robots* 43:897–912
83. Silverio J, Huang Y, Rozo L, Calinon S, Caldwell DG. 2018. Probabilistic learning of torque controllers from kinematic and force constraints. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 6552–59. Piscataway, NJ: IEEE
84. Maeda G, Ewerton M, Lioutikov R, Ben Amor H, Peters J, Neumann G. 2015. Learning interaction for collaborative tasks with probabilistic movement primitives. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pp. 527–34. Piscataway, NJ: IEEE
85. van den Berg J, Miller S, Duckworth D, Hu H, Wan A, et al. 2010. Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *2010 IEEE International Conference on Robotics and Automation*, pp. 2074–81. Piscataway, NJ: IEEE
86. Ciocarlie M, Goldfeder C, Allen PK. 2007. Dimensionality reduction for hand-independent dexterous robotic grasping. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3270–75. Piscataway, NJ: IEEE
87. Jonschkowski R, Brock O. 2015. Learning state representations with robotic priors. *Auton. Robots* 39:407–28
88. Ugur E, Piater J. 2015. Bottom-up learning of object categories, action effects and logical rules: from continuous manipulative exploration to symbolic planning. In *2015 IEEE International Conference on Robotics and Automation*, pp. 2627–33. Piscataway, NJ: IEEE
89. Byravan A, Fox D. 2017. SE3-Nets: learning rigid body motion using deep neural networks. In *2017 IEEE International Conference on Robotics and Automation*, pp. 173–80. Piscataway, NJ: IEEE
90. Finn C, Levine S. 2017. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation*, pp. 2786–93. Piscataway, NJ: IEEE
91. Mayer H, Gomez F, Wierstra D, Nagy I, Knoll A, Schmidhuber J. 2008. A system for robotic heart surgery that learns to tie knots using recurrent neural networks. *Adv. Robot.* 22:1521–37
92. Polydoros AS, Boukas E, Nalpantidis L. 2017. Online multi-target learning of inverse dynamics models for computed-torque control of compliant manipulators. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4716–22. Piscataway, NJ: IEEE
93. Rahmatizadeh R, Abolghasemi P, Boloni L, Levine S. 2018. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *2018 IEEE International Conference on Robotics and Automation*, pp. 3758–65. Piscataway, NJ: IEEE
94. Ravichandar H, Salehi I, Dani AP. 2017. Learning partially contracting dynamical systems from demonstrations. In *Proceedings of the 1st Annual Conference on Robot Learning*, ed. S Levine, V Vanhoucke, K Goldberg, pp. 369–78. Proc. Mach. Learn. Res. Vol. 78. N.p.: PMLR
95. Ravichandar H, Ahmadzadeh SR, Rana MA, Chernova S. 2019. Skill acquisition via automated multi-coordinate cost balancing. In *2019 IEEE International Conference on Robotics and Automation*, pp. 7776–82. Piscataway, NJ: IEEE
96. Manschitz S, Gienger M, Kober J, Peters J. 2018. Mixture of attractors: a novel movement primitive representation for learning motor skills from demonstrations. *IEEE Robot. Autom. Lett.* 3:926–33
97. Silvério J, Rozo L, Calinon S, Caldwell DG. 2015. Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 464–70. Piscataway, NJ: IEEE
98. Chatzis SP, Korkinof D, Demiris Y. 2012. A nonparametric Bayesian approach toward robot learning by demonstration. *Robot. Auton. Syst.* 60:789–802
99. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S. 2013. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput.* 25:328–73
100. Umlauft J, Hirche S. 2017. Learning stable stochastic nonlinear dynamical systems. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 3502–10. Proc. Mach. Learn. Res. Vol. 70. N.p.: PMLR

101. Petrič T, Gams A, Colasanto L, Ijspeert AJ, Ude A. 2018. Accelerated sensorimotor learning of compliant movement primitives. *IEEE Trans. Robot.* 34:1636–42
102. Ravichandar H, Trombetta D, Dani A. 2019. Human intention-driven learning control for trajectory synchronization in human-robot collaborative tasks. *IFAC-PapersOnLine* 51(34):1–7
103. Peternel L, Petrič T, Babič J. 2018. Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation. *Auton. Robots* 42:1–17
104. Suomalainen M, Kyrki V. 2016. Learning compliant assembly motions from demonstration. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 871–76. Piscataway, NJ: IEEE
105. Polydoros AS, Nalpantidis L. 2017. Survey of model-based reinforcement learning: applications on robotics. *J. Intell. Robot. Syst.* 86:153–73
106. Englert P, Paraschos A, Deisenroth MP, Peters J. 2013. Probabilistic model-based imitation learning. *Adapt. Behav.* 21:388–403
107. Ratliff N, Zucker M, Bagnell JA, Srinivasa S. 2009. CHOMP: gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pp. 489–94. Piscataway, NJ: IEEE
108. Kalakrishnan M, Chitta S, Theodorou E, Pastor P, Schaal S. 2011. STOMP: stochastic trajectory optimization for motion planning. In *2011 IEEE International Conference on Robotics and Automation*, pp. 4569–74. Piscataway, NJ: IEEE
109. Kalakrishnan M, Pastor P, Righetti L, Schaal S. 2013. Learning objective functions for manipulation. In *2013 IEEE International Conference on Robotics and Automation*, pp. 1331–36. Piscataway, NJ: IEEE
110. Dragan AD, Muelling K, Bagnell JA, Srinivasa SS. 2015. Movement primitives via optimization. In *2015 IEEE International Conference on Robotics and Automation*, pp. 2339–46. Piscataway, NJ: IEEE
111. Bajcsy A, Losey DP, O'Malley MK, Dragan AD. 2017. Learning robot objectives from physical human interaction. In *Proceedings of the 1st Annual Conference on Robot Learning*, ed. S Levine, V Vanhoucke, K Goldberg, pp. 217–26. Proc. Mach. Learn. Res. Vol. 78. N.p.: PMLR
112. Ng AY, Russell SJ. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 663–70. San Francisco: Morgan Kaufmann
113. Dvijotham K, Todorov E. 2010. Inverse optimal control with linearly-solvable MDPs. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pp. 335–42. Madison, WI: Omnipress
114. Levine S, Koltun V. 2012. Continuous inverse optimal control with locally optimal examples. arXiv:1206.4617 [cs.LG]
115. Doerr A, Ratliff ND, Bohg J, Toussaint M, Schaal S. 2015. Direct loss minimization inverse optimal control. In *Robotics: Science and Systems XI*, ed. LE Kavraki, D Hsu, J Buchli, pap. 13. N.p.: Robot. Sci. Syst. Found.
116. Finn C, Levine S, Abbeel P. 2016. Guided cost learning: deep inverse optimal control via policy optimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, ed. MF Balcan, KQ Weinberger, pp. 49–58. Proc. Mach. Learn. Res. Vol. 48. N.p.: PMLR
117. Silver D, Bagnell JA, Stentz A. 2010. Learning from demonstration for autonomous navigation in complex unstructured terrain. *Int. J. Robot. Res.* 29:1565–92
118. Boularias A, Kober J, Peters J. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ed. G Gordon, D Dunson, M Dudik, pp. 182–89. Proc. Mach. Learn. Res. Vol. 15. N.p.: PMLR
119. Hadfield-Menell D, Russell SJ, Abbeel P, Dragan A. 2016. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems 29*, ed. DD Lee, M Sugiyama, UV Luxburg, I Guyon, R Garnett, pp. 3909–17. Red Hook, NY: Curran
120. Ratliff ND, Silver D, Bagnell JA. 2009. Learning to search: functional gradient techniques for imitation learning. *Auton. Robots* 27:25–53
121. Abbeel P, Ng AY. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pap. 1. New York: ACM
122. Zucker M, Ratliff N, Stolle M, Chestnutt J, Bagnell JA, et al. 2011. Optimization and learning for rough terrain legged locomotion. *Int. J. Robot. Res.* 30:175–91

123. Ziebart BD. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD Thesis, Carnegie Mellon Univ., Pittsburgh, PA
124. Choi J, Kim KE. 2011. Map inference for Bayesian inverse reinforcement learning. In *Advances in Neural Information Processing Systems 24*, ed. J Shawe-Taylor, RS Zemel, PL Bartlett, F Pereira, KQ Weinberger, pp. 1989–97. Red Hook, NY: Curran
125. Choudhury R, Swamy G, Hadfield-Menell D, Dragan AD. 2019. On the utility of model learning in HRI. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction*, pp. 317–25. Piscataway, NJ: IEEE
126. Kober J, Bagnell JA, Peters J. 2013. Reinforcement learning in robotics: a survey. *Int. J. Robot. Res.* 32:1238–74
127. Ekvall S, Kragic D. 2008. Robot learning from demonstration: a task-level planning approach. *Int. J. Adv. Robot. Syst.* 5:223–34
128. Grollman DH, Jenkins OC. 2010. Incremental learning of subtasks from unsegmented demonstration. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 261–66. Piscataway, NJ: IEEE
129. Mohseni-Kabir A, Li C, Wu V, Miller D, Hylak B, et al. 2018. Simultaneous learning of hierarchy and primitives for complex robot tasks. *Auton. Robots* 43:859–74
130. Yin H, Melo FS, Paiva A, Billard A. 2018. An ensemble inverse optimal control approach for robotic task learning and adaptation. *Auton. Robots* 43:875–96
131. Konidaris G, Kuindersma S, Grupen R, Barto A. 2012. Robot learning from demonstration by constructing skill trees. *Int. J. Robot. Res.* 31:360–75
132. Manschitz S, Kober J, Gienger M, Peters J. 2014. Learning to sequence movement primitives from demonstrations. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4414–21. Piscataway, NJ: IEEE
133. Hayes B, Scassellati B. 2016. Autonomously constructing hierarchical task networks for planning and human-robot collaboration. In *2016 IEEE International Conference on Robotics and Automation*, pp. 5469–76. Piscataway, NJ: IEEE
134. Lioutikov R, Maeda G, Veiga F, Kersting K, Peters J. 2018. Inducing probabilistic context-free grammars for the sequencing of movement primitives. In *2018 IEEE International Conference on Robotics and Automation*, pp. 5651–58. Piscataway, NJ: IEEE
135. Pastor P, Kalakrishnan M, Righetti L, Schaal S. 2012. Towards associative skill memories. In *2012 12th IEEE-RAS International Conference on Humanoid Robots*, pp. 309–15. Piscataway, NJ: IEEE
136. Dang H, Allen PK. 2010. Robot learning of everyday object manipulations via human demonstration. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1284–89. Piscataway, NJ: IEEE
137. Meier F, Theodorou E, Stulp F, Schaal S. 2011. Movement segmentation using a primitive library. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3407–12. Piscataway, NJ: IEEE
138. Kulić D, Ott C, Lee D, Ishikawa J, Nakamura Y. 2012. Incremental learning of full body motion primitives and their sequencing through human motion observation. *Int. J. Robot. Res.* 31:330–45
139. Niekum S, Osentoski S, Konidaris G, Barto AG. 2012. Learning and generalization of complex tasks from unstructured demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5239–46. Piscataway, NJ: IEEE
140. Kroemer O, Van Hoof H, Neumann G, Peters J. 2014. Learning to predict phases of manipulation tasks as hidden states. In *2014 IEEE International Conference on Robotics and Automation*, pp. 4009–14. Piscataway, NJ: IEEE
141. Lioutikov R, Neumann G, Maeda G, Peters J. 2015. Probabilistic segmentation applied to an assembly task. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots*, pp. 533–40. Piscataway, NJ: IEEE
142. Su Z, Kroemer O, Loeb GE, Sukhatme GS, Schaal S. 2018. Learning manipulation graphs from demonstrations using multimodal sensory signals. In *2018 IEEE International Conference on Robotics and Automation*, pp. 2758–65. Piscataway, NJ: IEEE

143. Baisero A, Mollard Y, Lopes M, Toussaint M, Lütkebohle I. 2015. Temporal segmentation of pair-wise interaction phases in sequential manipulation demonstrations. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 478–84. Piscataway, NJ: IEEE
144. Takano W, Nakamura Y. 2015. Statistical mutual conversion between whole body motion primitives and linguistic sentences for human motions. *Int. J. Robot. Res.* 34:1314–28
145. Nicolescu MN, Mataric MJ. 2003. Natural methods for robot task learning: instructive demonstrations, generalization and practice. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 241–48. New York: ACM
146. Pardowitz M, Knoop S, Dillmann R, Zollner RD. 2007. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Trans. Syst. Man Cybernet. B* 37:322–32
147. Jäkel R, Schmidt-Rohr SR, Rühl SW, Kasper A, Xue Z, Dillmann R. 2012. Learning of planning models for dexterous manipulation based on human demonstrations. *Int. J. Soc. Robot.* 4:437–48
148. Kroemer O, Daniel C, Neumann G, Van Hoof H, Peters J. 2015. Towards learning hierarchical skills for multi-phase manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation*, pp. 1503–10. Piscataway, NJ: IEEE
149. Gombolay M, Jensen R, Stigile J, Golen T, Shah N, et al. 2018. Human-machine collaborative optimization via apprenticeship scheduling. *J. Artif. Intell. Res.* 63:1–49
150. Schmidt-Rohr SR, Lösch M, Jäkel R, Dillmann R. 2010. Programming by demonstration of probabilistic decision making on a multi-modal service robot. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 784–89. Piscataway, NJ: IEEE
151. Butterfield J, Osentoski S, Jay G, Jenkins OC. 2010. Learning from demonstration using a multi-valued function regressor for time-series data. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp. 328–33. Piscataway, NJ: IEEE
152. Niekum S, Osentoski S, Konidaris G, Chitta S, Marthi B, Barto AG. 2015. Learning grounded finite-state representations from unstructured demonstrations. *Int. J. Robot. Res.* 34:131–57
153. Hausman K, Chebotar Y, Schaal S, Sukhatme G, Lim JJ. 2017. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In *Advances in Neural Information Processing Systems 30*, ed. I Guyon, UV Luxburg, S Bengio, H Wallach, R Fergus, et al., pp. 1235–45. Red Hook, NY: Curran
154. Krishnan S, Garg A, Liaw R, Thananjeyan B, Miller L, et al. 2019. SWIRL: a sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. *Int. J. Robot. Res.* 38:126–45
155. Krishnan S, Garg A, Liaw R, Miller L, Pokorny FT, Goldberg K. 2016. HIRL: hierarchical inverse reinforcement learning for long-horizon tasks with delayed rewards. arXiv:1604.06508 [cs.RO]
156. Hirzinger G, Heindl J. 1983. Sensor programming, a new way for teaching a robot paths and forces torques simultaneously. In *Proceedings of the 3rd International Conference on Robot Vision and Sensory Control*, ed. B Rooks, pp. 549–58. Oxford, UK: Cotswold
157. Asada H, Izumi H. 1987. Direct teaching and automatic program generation for the hybrid control of robot manipulators. In *1987 IEEE International Conference on Robotics and Automation*, Vol. 4, 1401–6. Piscataway, NJ: IEEE
158. Kronander K, Khansari M, Billard A. 2015. Incremental motion learning with locally modulated dynamical systems. *Robot. Auton. Syst.* 70:52–62
159. Kent D, Chernova S. 2014. Construction of an object manipulation database from grasp demonstrations. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3347–52. Piscataway, NJ: IEEE
160. Bhattacharjee T, Lee G, Song H, Srinivasa SS. 2019. Towards robotic feeding: role of haptics in fork-based food manipulation. *IEEE Robot. Autom. Lett.* 4:1485–92
161. Fong J, Tavakoli M. 2018. Kinesthetic teaching of a therapist's behavior to a rehabilitation robot. In *2018 International Symposium on Medical Robotics*. Piscataway, NJ: IEEE. **https://doi.org/10.1109/ISMR.2018.8333285**
162. Wang H, Chen J, Lau HYK, Ren H. 2016. Motion planning based on learning from demonstration for multiple-segment flexible soft robots actuated by electroactive polymers. *IEEE Robot. Autom. Lett.* 1:391–98

163. Najafi M, Sharifi M, Adams K, Tavakoli M. 2017. Robotic assistance for children with cerebral palsy based on learning from tele-cooperative demonstration. *Int. J. Intell. Robot. Appl.* 1:43–54
164. Moro C, Nejat G, Mihailidis A. 2018. Learning and personalizing socially assistive robot behaviors to aid with activities of daily living. *ACM Trans. Human-Robot Interact.* 7:15
165. Ma Z, Ben-Tzvi P, Danoff J. 2015. Hand rehabilitation learning system with an exoskeleton robotic glove. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24:1323–32
166. Strabala K, Lee MK, Dragan A, Forlizzi J, Srinivasa SS, et al. 2013. Toward seamless human-robot handovers. *J. Hum.-Robot Interact.* 2:112–32
167. Pomerleau DA. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural Comput.* 3:88–97
168. Boularias A, Krömer O, Peters J. 2012. Structured apprenticeship learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 227–42. Berlin: Springer
169. Ross S, Gordon G, Bagnell D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ed. G Gordon, D Dunson, M Dudik, pp. 627–35. Proc. Mach. Learn. Res. Vol. 15. N.p.: PMLR
170. Silver D, Bagnell JA, Stentz A. 2012. Active learning from demonstration for robust autonomous navigation. In *2012 IEEE International Conference on Robotics and Automation*, pp. 200–7. Piscataway, NJ: IEEE
171. Li Y, Song J, Ermon S. 2017. InfoGAIL: interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems 30*, ed. I Guyon, UV Luxburg, S Bengio, H Wallach, R Fergus, et al., pp. 3812–22. Red Hook, NY: Curran
172. Pan Y, Cheng CA, Saigol K, Lee K, Yan X, et al. 2018. Agile autonomous driving using end-to-end deep imitation learning. In *Robotics: Science and Systems XIV*, ed. H Kress-Gazit, S Srinivasa, T Howard, N Atanasov, pap. 56. N.p.: Robot. Sci. Syst. Found.
173. Kuderer M, Gulati S, Burgard W. 2015. Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE International Conference on Robotics and Automation*, pp. 2641–46. Piscataway, NJ: IEEE
174. Ross S, Melik-Barkhudarov N, Shankar KS, Wendel A, Dey D, et al. 2013. Learning monocular reactive UAV control in cluttered natural environments. In *2013 IEEE International Conference on Robotics and Automation*, pp. 1765–72. Piscataway, NJ: IEEE
175. Kaufmann E, Loquercio A, Ranftl R, Dosovitskiy A, Koltun V, Scaramuzza D. 2018. Deep drone racing: learning agile flight in dynamic environments. In *Proceedings of the 2nd Conference on Robot Learning*, ed. A Billard, A Dragan, J Peters, J Morimoto, pp. 133–45. Proc. Mach. Learn. Res. Vol. 87. N.p.: PMLR
176. Loquercio A, Maqueda AI, Del-Blanco CR, Scaramuzza D. 2018. DroNet: learning to fly by driving. *IEEE Robot. Autom. Lett.* 3:1088–95
177. Farchy A, Barrett S, MacAlpine P, Stone P. 2013. Humanoid robots learning to walk faster: from the real world to simulation and back. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multiagent Systems*, pp. 39–46. Richland, SC: Int. Found. Auton. Agents Multiagent Syst.
178. Meriçliç, Veloso M. 2010. Biped walk learning through playback and corrective demonstration. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 1594–99. Palo Alto, CA: AAAI Press
179. Calandra R, Gopalan N, Seyfarth A, Peters J, Deisenroth MP. 2014. Bayesian gait optimization for bipedal locomotion. In *International Conference on Learning and Intelligent Optimization*, pp. 274–90. Cham, Switz.: Springer
180. Kolter JZ, Abbeel P, Ng AY. 2008. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Advances in Neural Information Processing Systems 20*, ed. JC Platt, D Koller, Y Singer, ST Roweis, pp. 769–76. Red Hook, NY: Curran
181. Kalakrishnan M, Buchli J, Pastor P, Mistry M, Schaal S. 2011. Learning, planning, and control for quadruped locomotion over challenging terrain. *Int. J. Robot. Res.* 30:236–58
182. Nakanishi J, Morimoto J, Endo G, Cheng G, Schaal S, Kawato M. 2004. Learning from demonstration and adaptation of biped locomotion. *Robot. Auton. Syst.* 47:79–91

183. Carrera A, Palomeras N, Ribas D, Kormushev P, Carreras M. 2014. An intervention-AUV learns how to perform an underwater valve turning. In *OCEANS 2014 - TAIPEI*. Piscataway, NJ: IEEE. **https://doi.org/10.1109/OCEANS-TAIPEI.2014.6964483**
184. Havoutis I, Calinon S. 2017. Supervisory teleoperation with online learning and optimal control. In *2017 IEEE International Conference on Robotics and Automation*, pp. 1534–40. Piscataway, NJ: IEEE
185. Birk A, Doernbach T, Mueller C, Łuczynski T, Chavez AG, et al. 2018. Dexterous underwater manipulation from onshore locations: streamlining efficiencies for remotely operated underwater vehicles. *IEEE Robot. Autom. Mag.* 25(4):24–33
186. Somers T, Hollinger GA. 2016. Human–robot planning and learning for marine data collection. *Auton. Robots* 40:1123–37
187. Sun W, Venkatraman A, Gordon GJ, Boots B, Bagnell JA. 2017. Deeply AggreVaTeD: differentiable imitation learning for sequential prediction. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 3309–18. Proc. Mach. Learn. Res. Vol. 70. N.p.: PMLR
188. Kober J, Peters JR. 2009. Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems 21*, ed. D Koller, D Schuurmans, Y Bengio, L Bottou, pp. 849–56. Red Hook, NY: Curran
189. Taylor ME, Suay HB, Chernova S. 2011. Integrating reinforcement learning with human demonstrations of varying ability. In *The 10th International Conference on Autonomous Agents and Multiagent Systems*, Vol. 2, pp. 617–24. Richland, SC: Int. Found. Auton. Agents Multiagent Syst.
190. Pastor P, Kalakrishnan M, Chitta S, Theodorou E, Schaal S. 2011. Skill learning and task outcome prediction for manipulation. In *2011 IEEE International Conference on Robotics and Automation*, pp. 3828–34. Piscataway, NJ: IEEE
191. Kim B, Farahmand A, Pineau J, Precup D. 2013. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems 26*, ed. CJC Burges, L Bottou, M Welling, Z Ghahramani, KQ Weinberger, pp. 2859–67. Red Hook, NY: Curran
192. Vecerik M, Hester T, Scholz J, Wang F, Pietquin O, et al. 2017. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. arXiv:1707.08817 [cs.AI]
193. Vecerik M, Sushkov O, Barker D, Rothörl T, Hester T, Scholz J. 2019. A practical approach to insertion with variable socket position using deep reinforcement learning. In *2019 International Conference on Robotics and Automation*, pp. 754–60. Piscataway, NJ: IEEE
194. Nair A, McGrew B, Andrychowicz M, Zaremba W, Abbeel P. 2018. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation*, pp. 6292–99. Piscataway, NJ: IEEE
195. Sermanet P, Lynch C, Chebotar Y, Hsu J, Jang E, et al. 2018. Time-contrastive networks: self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation*, pp. 1134–41. Piscataway, NJ: IEEE
196. Brown DS, Niekum S. 2017. *Toward probabilistic safety bounds for robot learning from demonstration*. Tech. Rep. FS-17-01, Assoc. Adv. Artif. Intell., Palo Alto, CA
197. Laskey M, Staszak S, Hsieh WYS, Mahler J, Pokorny FT, et al. 2016. SHIV: reducing supervisor burden in dagger using support vectors for efficient learning from demonstrations in high dimensional state spaces. In *2016 IEEE International Conference on Robotics and Automation*, pp. 462–69. Piscataway, NJ: IEEE
198. Zhou W, Li W. 2018. Safety-aware apprenticeship learning. In *Computer Aided Verification: 30th International Conference, CAV 2018*, ed. H Chockler, G Weissenbacher, pp. 662–80. Cham, Switz.: Springer
199. Gupta A, Eppner C, Levine S, Abbeel P. 2016. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3786–93. Piscataway, NJ: IEEE
200. Ogrinc M, Gams A, Petrič T, Sugimoto N, Ude A, et al. 2013. Motion capture and reinforcement learning of dynamically stable humanoid movement primitives. In *2013 IEEE International Conference on Robotics and Automation*, pp. 5284–90. Piscataway, NJ: IEEE
201. Lee H, Kim H, Kim HJ. 2016. Planning and control for collision-free cooperative aerial transportation. *IEEE Trans. Autom. Sci. Eng.* 15:189–201

202. Coates A, Abbeel P, Ng AY. 2008. Learning for control from multiple demonstrations. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 144–51. New York: ACM
203. Choi S, Lee K, Oh S. 2016. Robust learning from demonstration using leveraged Gaussian processes and sparse-constrained optimization. In *2016 IEEE International Conference on Robotics and Automation*, pp. 470–75. Piscataway, NJ: IEEE
204. Shepard RN. 1987. Toward a universal law of generalization for psychological science. *Science* 237:1317–23
205. Ghirlanda S, Enquist M. 2003. A century of generalization. *Anim. Behav.* 66:15–36
206. Bagnell JA. 2015. *An invitation to imitation*. Tech. Rep. CMU-RI-TR-15-08, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA
207. Calinon S, Bruno D, Caldwell DG. 2014. A task-parameterized probabilistic model with minimal intervention control. In *2014 IEEE International Conference on Robotics and Automation*, pp. 3339–44. Piscataway, NJ: IEEE
208. Finn C, Yu T, Zhang T, Abbeel P, Levine S. 2017. One-shot visual imitation learning via meta-learning. arXiv:1709.04905 [cs.LG]
209. Corduneanu A, Bishop CM. 2001. Variational Bayesian model selection for mixture distributions. In *Artificial Intelligence and Statistics 2001*, ed. T Jaakkola, T Richardson, pp. 27–34. San Francisco: Morgan Kaufmann
210. Ketchen DJ, Shook CL. 1996. The application of cluster analysis in strategic management research: an analysis and critique. *Strateg. Manag. J.* 17:441–58
211. Shams L, Seitz AR. 2008. Benefits of multisensory learning. *Trends Cogn. Sci.* 12:411–17
212. Sung J, Jin SH, Saxena A. 2018. Robobarista: object part based transfer of manipulation trajectories from crowd-sourcing in 3D pointclouds. In *Robotics Research*, ed. A Bicchi, W Burgard, pp. 701–20. Cham, Switz.: Springer
213. Castro PS, Li S, Zhang D. 2019. Inverse reinforcement learning with multiple ranked experts. arXiv:1907.13411 [cs.LG]