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; email: {harish.ravichandar, chernova}@gatech.edu
[2] Learning Algorithms and Systems Laboratory, École Polytechnique Fédérale de Lausanne; email: {athanasios.polydoros, aude.billard}@epfl.ch
[†‡] Equal contribution by junior[†] and senior[‡] authors.

Posted with permission from the Annual Review of Control, Robotics, and Autonomous Systems, Volume 3 2020 by Annual Reviews, http://www.annualreviews.org/.

Annual Review of Control, Robotics, and Autonomous Systems 2020. 3:1–33

Keywords

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

Abstract

In the context of robotics and automation, learning from demonstrations (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 neither be easily scripted, as done in traditional robot programming, nor be easily defined as an optimization problem, but can be demonstrated. While there have been multiple surveys of the field in the past, there is a need for a new survey given the considerable growth in the number of publications in recent years. This survey aims at offering 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, as well as 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 practice.

1. INTRODUCTION

In the context of robotics and automation, learning from demonstrations (LfD) is the paradigm in which robots acquire new skills by learning to imitate an expert (1, 2, 3, 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. Further, 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 of the environment and require re-specification or programming.

An attractive aspect of LfD is its ability to facilitate non-expert robot programming. LfD accomplishes this by *implicitly* learning task constrains and requirements from demonstrations which can enable adaptive behavior. Put another way, LfD enables robots to move away from repeating simple pre-specified behaviors in constrained environments and towards 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 healthcare (8), wherein it can empower 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 Fig. 1, the field has seen considerable growth in the number of publications in recent years. The field remains diverse both in terms of its algorithms (see Sections 2 and 3) and its terminology (see Fig. 1). *Imitation learning*, *programming by demonstration*, and *behavioral cloning* are other popular phrases used to describe the process of learning from demonstrations. In

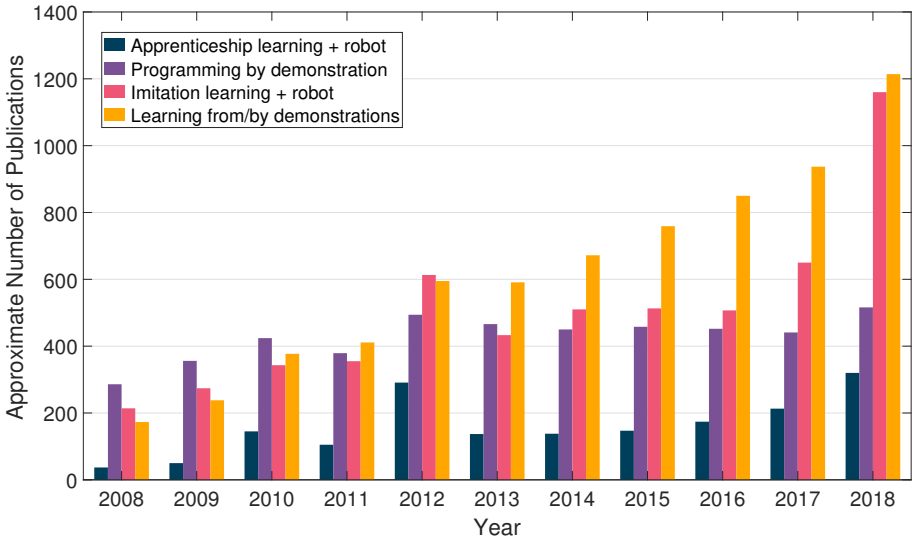

Figure 1

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

this review, we will use the term *Learning from Demonstration* 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 between 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 neither be scripted (as done in traditional robot programming) nor be easily defined as optimizing a known a reward function (as 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 type of skills that can be learned through LfD, 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 datasets, incremental learning, and learning from noisy data. Besides these challenges, when LfD is applied to control a real robotic physical system, it also inherits challenges from control theory such as 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 getting demonstrations from an external agent, usually a human, it must overcome a variety of challenges well known in human-robot interaction, such as finding the 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 from each other (at least physically) than robots. Hence, LfD is not only sensitive to who teaches the robot, but it is also still quite dependent on the platform (robot + interface) used.

Multiple surveys of LfD, which focus on different subsets of the field, have been published over the past two decades (1, 2, 3, 4, 7, 9, 10, 11); these surveys are representative of the evolution of the field. In (1), the author presented the first survey of LfD, focusing on imitation and trajectory-based skills. A more recent account of the same topic, nearly 20 years later and focusing on a more algorithmic perspective, is presented in (11). A broad synthesis of LfD which incorporates elements of human-robot interaction, is presented in (2) and later in (10) which frames the inquiry from the perspective of four core questions for the field: *how*, *what*, *when*, and *whom* to imitate. A taxonomy of LfD, characterizing types of demonstration inputs and variations on learning methods was presented in (3), and later in (4). A detailed survey on the general topic of grasp synthesis is presented in (12), where the authors also include a taxonomy of LfD methods used in the particular area. Finally, the need of adaptable manufacturing robotic system has led to the application of LfD methods on industrial assembly tasks, as presented in (7).

While there have been many surveys of the field in the past, there is a need for a new survey given the steady growth of the domain. This survey hence aims at offering 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, as well as present an updated taxonomy and characterization of existing methods. This survey also touches on mature and emerging application areas for LfD, and seeks to underline the significant challenges that remain to be overcome both in theory and applications.

The organization of the survey is as follows. We first categorize the LfD literature based on how demonstrations are acquired in Section 2, followed by a categorization based on the what is learned in Section 3. The various application areas are identified in Section 4. The strengths and limitations of the various flavors of LfD are presented in Section 5, followed by a discussion of open

problems and challenges in Section 6. Finally, we provide some concluding remarks in Section 7.

2. CATEGORIZATION BASED ON DEMONSTRATIONS

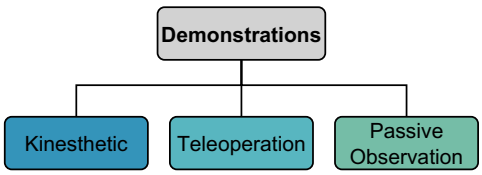

Figure 2

Categorization of LfD methods based on the demonstrations they utilize.

One of the first decisions to be made when designing a 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, demonstration approaches fall into three categories – *kinesthetic*, *teleoperation*, and *passive observation*. Table 1 provides a summary of the key similarities and differences between these categories in terms of *i)* ease of demonstration, *ii)* ability to handle high degrees-of-freedom (DOF), and *iii)* 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, primarily applied to manipulation platforms (13, 14, 15, 16, 17, 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 on-board sensors (e.g., joint angles, 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 only requires the development and maintenance of the robot hardware and does not rely no 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 a number of 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 requires smoothing or other post-processing techniques. Finally, kinesthetic teaching is most effective for manipulators due to their relatively intuitive form fac-

Demonstration	Ease of Demonstration	High DOF	Ease of Mapping
Kinesthetic	✓		✓
Teleoperation		✓	✓
Observation	✓	✓	

Table 1 Characteristics of LfD methods categorized based on demonstrations.

(a) Kinesthetic (b) Teleoperation (c) Observation

Figure 3

Examples of robot demonstrations.

tor; its 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 which has been applied to trajectory learning (21), task learning (22), grasping (23), and high level tasks (24). Teleoperation 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, 28, 29). Unlike kinesthetic teaching, teleoperation does not require the user to be co-present 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, 31, 32, 33).

Limitations of teleoperation include additional effort to develop the chosen input interface, in some cases a more lengthy user training process, and the availability of input hardware (e.g., VR 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). Finally, teleoperation can be easily coupled with simulation to further facilitate data collection and experimentation at scale, as often required within reinforcement learning (RL) frameworks.

2.3. Passive Observations

The third demonstration approach is for the robot to learn from passive observations of the user (36, 37, 38). In this approach, the users 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, acting 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 high-DOF or non-anthropomorphic 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 demonstrations. 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, an only the objects in the scene are tracked (31, 44).

2.4. Active and Interactive Demonstrations

Once the choice of the demonstration method has been made, there remains the choice of *what* to demonstrate, and whether demonstrations should be requested by the robot or initiated by the human. Techniques for managing the interaction, such as through active learning (39, 45, 46, 47, 48) and corrective demonstrations (49, 50), are covered in greater detail in (4). In (51), several case studies are presented which illustrate the importance of studying the users of intelligent systems in order improve both user experience and robot performance. While more general than the field of LfD, a comprehensive notion of *interactive task learning* is introduced in (52), which emphasizes 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 Fig. 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 provides a summary of the key similarities and differences between these choices in terms of their ability to *i)* learn low-level policies, *ii)* handle continuous action spaces, *iii)* compactly represent the learned skill, *iv)* plan over long time horizons, and *v)* learn complex tasks composed of several sub-tasks and sequencing constraints. In the sections below, we discuss each learning outcome and their underlying assumptions, providing sub-categorizations where appropriate.

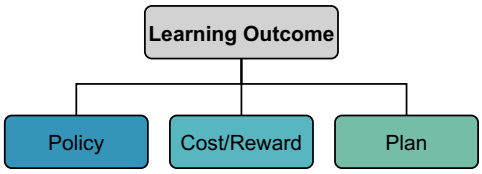

Figure 4

Categorization of LfD methods based on what is learned.

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) $\mathcal{Y}$ is its co-domain (action space). The goal of policy learning methods is to learn a policy $\pi(\cdot)$ that generates state trajectories $\{x(t)\}$ that are

Learning Outcome	Low-Level Control	Action Space Continuity	Compact Representation	Long-Horizon Planning	Multi-Step Tasks
Policy	✓	✓	✓		
Cost/Reward	✓	✓		✓	
Plan			✓	✓	✓

Table 2 Characteristics of LfD methods based on learning outcome.

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 as it does not learn a reward function. Instead, GAIL can be considered as a policy learning algorithm as 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 sub-categorizations, 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 Fig. 5). Table 3 provides a summary of the key characteristics associated with each of these choices in terms of *i)* ease of policy design, *ii)* whether performance guarantees can be provided, *iii)* robustness to spatio-temporal perturbations during execution, *iv)* the diversity of tasks that can be learned, and *v)* computational efficiency.

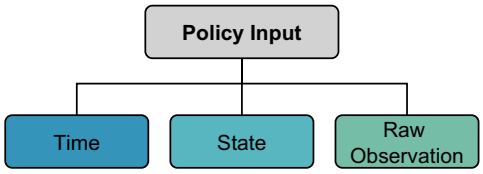

Figure 5

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

Time

The first class of methods in this categorization utilize time as the primary input to the policy (13, 19, 59, 60, 61, 62, 63, 64, 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 primarily based on initial conditions and the current 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 have been shown to be capable of identifying and capturing important features that are anchored in time (19, 60, 63). With time as the primary input, important temporal

Policy Input	Ease of Design	Performance Guarantees	Robustness to Perturbations	Task Variety	Algorithmic Efficiency
Raw Observations	✓		✓	✓	
Time	✓	✓			✓
State		✓	✓		✓

Table 3 Characteristics of policy learning methods categorized based the input.

constraints, such as when to precisely follow a trajectory, can be identified by utilizing heteroscedastic stochastic processes (59, 65) or geometric structures (13). Further, learning time-based trajectory distributions has been shown to be 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 primarily depends 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.

State

A popular category of policy learning methods assume direct access to the state and utilize it as the input to the policy (15, 16, 26, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84). Existing literature in LfD has explored a wide variety of choices for what is considered to be the state, such as end-effector position (82), velocity (70), orientation (83), force (66, 76), joint angles (75), and torques (80). 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 as 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, 85)) or explicitly (e.g., (75, 86)).

What constitutes the state is often manually specified. In 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 either be accomplished in an unsupervised or supervised manner. Unsupervised approaches (87, 88, 89) 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 (90, 91).

State-based policies enable the robot to take into account the current state of task and thus allows them to be reactive. Further, 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 to time-based representations. Another challenge of state based policies is that theoretical guarantees, such as stability (70, 83), are more challenging to prove and realize practically than for time-based systems.

Raw observations

Unlike the two categories of methods discussed above, a third category does not rely on a succinct

input representation. Methods of 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, and require 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 have recently been introduced (41, 43, 92, 93, 94). End-to-end LfD methods determine the appropriate action directly based on high-dimensional raw observations from sensors, such as cameras. This approach is particularly useful when learning tasks in which a succinct input representation is either 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 in order to be trained. Further, due to the large amount of non-linear 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 labelled 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 seen in Fig. 6, the two primary categories of policy outputs *trajectories* and *low-level actions*. Table 4 provides a summary of the key similarities and differences between methods with different policy outputs in terms of their ability to *i)* operate without the knowledge of the robot model, *ii)* learn platform-independent policies, *iii)* learn policies for under-actuated robots, *iv)* learn end-to-end policies.

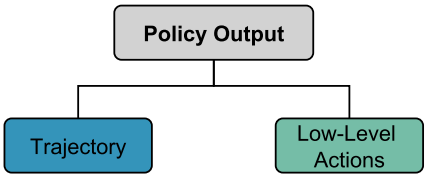

Figure 6

Categorization of policy learning methods based on the output of the policy.

Policy Output	Model-Free	Platform Independence	Under-Actuated Robots	End-to-End
Trajectory	✓	✓		
Low-Level Action			✓	✓

Table 4 Characteristics of policy learning methods categorized based the output.

Trajectory learning

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, 81, 82, 95, 96, 97), end-effector pose (67, 83, 98), end-effector force (61, 72, 74, 78), and joint state (62, 99). When reproducing the skill, these methods generate trajectories based on the initial 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, 83, 97, 98, 100, 101) and probabilistic inference (13, 62, 82), 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.

Trajectory learning methods rely on low-level controllers to execute the generated reference trajectories. They have shown to be particularly well-suited for over-actuated systems, such as redundant manipulators, for which kinematic feasibility is relatively easier to achieve. Further, trajectory learning methods do not require the knowledge of robot dynamics and do not necessitate 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 joints' space and the operational space. Learning in each of those space 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 an other robotic system. On the other hand, learning in the operational space does not guarantee that the generated motions are feasible and that singularities are avoided (82).

Learning low-level actions

Methods of 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/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/forces at each state which also allows tuning of impedance parameters resulting to compliant control. In the context of human-robot interaction, research has focused on learning the required torques (26, 80, 84, 102, 103) and stiffness parameters (69, 104) for producing compliant motions. Nevertheless, pursuing low-level learning outcomes brings about 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, 105). 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 under-actuated systems, has limitations that are associated with the collection of demonstrations, and knowledge of the robots physical properties. For instance, the applicability of low-level torque control is limited by the 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 (106).

3.1.3. Policy class. Another dimension of categorization could be achieved by examining the class of mathematical functions to which the policy belongs (see Fig. 7). The appropriate policy class for a given problem is an important design choice that has significant implications. Table 5 provides a summary of such implications associated with this choice in terms of the learned policy's ability to *i)* learn skills that depend on temporal context, *ii)* handle temporal perturbations, *iii)* consistently and reliably repeat policy roll-outs, and *iv)* encode multi-modal behavior.

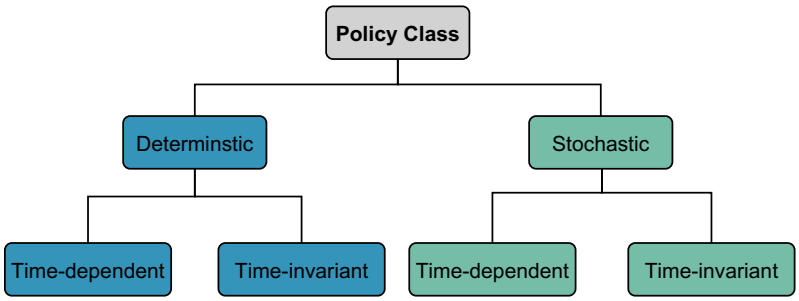

Figure 7

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

Deterministic vs stochastic policies

The primary categories of the policy class are the *deterministic* and *stochastic* polices. The choice between the two policy classes is made by considering a fundamental question: *Given a particular context, does a demonstration represent either i) the singular or absolute ideal behavior, or ii) 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 $\pi : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ are the policy's input and output 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, 79, 83).

Stochastic policies sample a behavior from a learned distribution of behaviors during each execution (62, 101, 107). 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 multi-modal distributions can effectively capture this uncertainty, deterministic policies cannot and have to resolve the seemingly conflicting paths. Further, 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

Policy Class	Temporal context	Robustness to temporal perturbations	Repeatability	Mutli-modal behavior
Deterministic & time-invariant		✓	✓	
Deterministic & time-dependent	✓		✓	
Stochastic & time-invariant		✓		✓
Stochastic & time-dependent	✓			✓

Table 5 Characteristics of policy learning methods categorized based the policy class.

convergence have recently been proposed (101).

Time-dependent vs invariant policies

Irrespective of stochasticity, the policy class can be sub-categorized based on the stationarity of the policy. Specifically, policies can either be *time-dependent* or *time-invariant* (see Fig. 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 straight-forward mechanism to ensure that the reproduced behavior aligns with the demonstration in terms of speed and duration (64).

Time-invariant policies, on the other hand, are capable of capturing general strategies that are independent of time and are more robust to intermittent spatio-temporal perturbations during executions. For instance, when reproducing a reaching motion, time-invariant methods are more robust to interruptions in their motions as they do not have to rely on an internal "clock" and can depend on state feedback (e.g., (70, 83)). However, note that certain time-invariant policies, such as DMPs (100), 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* or a *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 identified function. Indeed, learning cost or reward functions requires certain assumptions to be made about the task and the environment. Below, we discuss two classes of methods that make two different set of assumptions regarding the task and the environment.

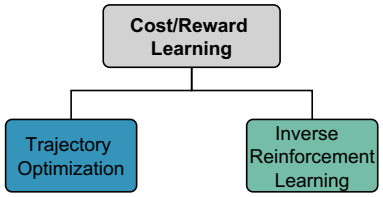

Figure 8

Categorization of cost and reward learning methods.

3.2.1. Trajectory optimization. Traditional trajectory optimization methods (5, 108, 109) were originally introduced to generate smooth and efficient trajectories for robots to move between any two given points in space. In such methods, the cost function is pre-specified 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 (96, 110, 111, 112). 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 (111).

3.2.2. Inverse reinforcement learning. A popular class of methods, that learn a cost or reward function, is *inverse reinforcement learning* (IRL) (113). In IRL, it is typically assumed that the demonstrator optimizes an unknown reward function. Demonstrations are thus utilized to learn the hidden reward function, following which classical reinforcement learning approaches can be used to compute optimal actions. In the continuous case, the methods become very akin to *inverse optimal control* (IOC) where a hidden objective function for optimal control is estimated from demonstrations (114, 115, 116, 117), and both terms are often used equivalently in the literature. Depending on the complexity of the problem of interest, IRL methods either assume that the reward function is linear (115, 118, 119, 120) or nonlinear (58, 117, 121).

Since there might be multiple reward functions that optimally explain the available demonstrations, IRL is referred to as an "ill-posed" problem. In order to arrive at a unique reward function, IRL methods consider different additional optimization goals, such as *maximum margin* (113, 118, 122, 123) and *maximum entropy* (58, 119, 124, 125). 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 which 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 what prior knowledge is assumed. Model-based IRL approaches (118, 120, 124, 125) exploit the knowledge of the transition probabilities (i.e., system model) to ensure that the reward function and the 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 utilize 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 (119), considering a direct policy search (116), and alternating between reward and policy optimization (117). Recent work comparing model-based and model-free learning from human data has suggested that model-based approaches have numerous computational advantages (126).

3.2.3. Limitations. Similar to 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 sub-optimal demonstrations. The choice of structure for the cost or reward function is not trivial and can significantly influence performance. Further, such choices tend to depend on the task of interest and thus cannot be applied without modifications to other applications. Compared to policy learning approaches, while cost or reward learning methods introduce more structure, they are the 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 its limitations such as the need for a large amount of episodes to converge (127) and the existence of transition models which can be hard to derive (106).

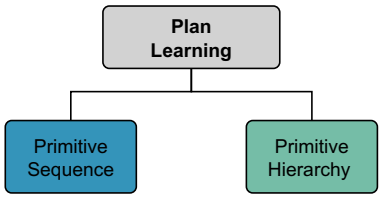

Figure 9

Categorization of LfD methods that learn task plans.

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 that is made up of several sub-tasks or primitive actions (i.e. a *task plan*) (128, 129, 130, 131, 132, 133, 134, 135, 136). Structured task plans typically encode the patterns and constraints in the sub-tasks 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 sub-task to execute next, among a finite set of sub-tasks.

Complex tasks, such as assembling a device and packing lunch, are often demonstrated in continuous sessions. Such demonstrations contain several sub-tasks which exhibit specific ordering constraints and inter-dependence. Thus, segmentation plays a crucial role in task plan learning. Several methods for automated segmentation of demonstrations into sub-tasks, either based on the similarity of component sub-tasks (137, 138, 139, 140, 141, 142) or based on the occurrence of indicative events (25, 143, 144), have been explored.

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

Methods that learn task plans from demonstration can either learn a *primitive sequence* (132, 133, 141, 145) or a *primitive hierarchy* (128, 146, 147, 148, 149) (see Fig. 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 non-determinism. For instance, hierarchies can be used to capture the fact that certain sub-tasks could be carried out in any order (24). LfD has also been applied to learning a plan schedule from expert demonstrations (150).

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 (130, 132, 136, 149, 151, 152, 153, 154, 155).

The above motioned methods utilize demonstrations to combine trajectory learning and task plan learning. For instance, in (153), unstructured demonstrations and interactive corrections are used to learn finite state machines that are made of several trajectory models. In (136), the notion

of associated skill memories is introduced, which captures sensory associations in addition to the primitives and utilize such associations to effectively sequence the learned primitives in order to achieve manipulation goals. In (149), demonstrations are utilized to learn a stochastic model of the phases and transitions involved in manipulation skills. Then, reinforcement learning is utilized to learn how to transition between the previously identified learned. In (130), human demonstrators are requested to provide narrations as demonstrate the task. While the demonstrated trajectories are utilized to learn trajectory-level primitives, the associated narrations are used to infer the boundaries between primitives and to recover the task plan. Finally, in (156), long time horizon tasks are approximated as a sequence of local reward functions and subtask transition conditions, which are learned via inverse reinforcement learning.

4. APPLICATIONS OF LFD

The various LfD approaches introduced thus far 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 been shown to be effective.

Manufacturing Training of manipulators from demonstrations is very usual 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 compared to 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 80's (157, 158). Popular examples include pick-and-place (80), peg insertion (117), polishing (159), grasping (33, 34, 160) 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 when tasks involve time-sensitive constraints.

Assisting and Healthcare Robotics Another popular area of application for manipulators is assisting and healthcare robotics. LfD has proven to be an effective way to teach manipulators important movements skill necessary when provide healthcare services and assisting people. Specifically, LfD methods have helped teach manipulators a variety of skills, such as feeding (161), physical rehabilitation (162), robotic surgery (86, 163), assisting children with cerebral palsy (164), supporting daily activities (165), hand rehabilitation (166), handover objects (167), and motion planning for rehabilitation (8). Manipulators which are expected to provide assistance and healthcare, are also usually taught by kinesthetic demonstrations similar to the manufacturing applications. In addition, assistive robots are expected to operate in closer-proximity with humans compared to manufacturing cases. This rises the need for safe operation which can be satisfied by providing convergence and stability guarantees of the learned policy.

Human-Robot Interaction (HRI) In addition to learning to autonomously perform tasks in different areas, LfD has also been explored to provide manipulators with the ability to collaborate with humans in close proximity (14, 63, 66, 75, 85, 103, 134). Effective collaboration requires the generation of the desired robotic movements that are complementary to the 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 which are able to compensate for perturbations that will happen during the operation of the robot. The most common approach to achieve that is by learning a policy function whose inputs are the states of the robot. HRI applications also present the need for compliant manipulators, which is typically met by learning the appropriate joints' torques (80) and stiffness and damping parameters (104).

4.2. Mobile Robots

In addition to manipulators, LfD has also 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 to mobile robots.

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 involved the autonomous navigation of a car and was introduced in 1991 (168). In this seminal work, a neural network-based algorithm is described that can learn the mapping between from inputs and a discrete action space from human driving data. Since the early success noted in (168), LfD-based autonomous navigation for ground vehicles has been attempted using inverse reinforcement learning (118, 122, 169), interactive learning (170), active learning (171), adversarial learning (172), end-to-end learning (40, 173). Demonstrations in such platforms are usually provided through teleoperation by joystick for small vehicles. Nevertheless, kinesthetic teaching can be applied on large vehicles, such as cars, where a human takes the driver's seat ad demonstrate a desired behaviour by driving (174).

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 to aerial vehicles navigate in cluttered environments (175). Further, recent advances in deep learning have fueled the development of end-to-end LfD methods for aerial vehicles (176, 177). Demonstrations for training flying robots are usually done through teleoperation. Hence, training pertain primarily to teach a desired trajectory, while stability of the robot is handled by traditional control approaches.

Bipedal and Quadrupedal Robots In these platforms, LfD has been primarily used for the purpose of locomotion. LfD approaches have been successfully used in bipedal robots for learning to walk (178, 179) and gait optimization (180). Additionally, learning from demonstrations has also enjoyed some successes with quadrupedal locomotion (123, 181, 182). 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 (183).

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 (184), underwater robot teleoperation (35, 185, 186) and marine data collection (187). Similarly to what is done with flying and humanoid robots, applications of LfD to train mobile robots uses teleoperation in combination with control methods for ensuring stability.

5. STRENGTHS AND LIMITATIONS OF LFD

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

5.1. Strengths of LfD

The field of learning from demonstrations offers a number of advantages. Indeed, different types of LfD algorithms provide different benefits, making them suitable for different scenarios and problems (see Tables 2 to 5). Below, we identify particular strengths of LfD methods at large.

Non-expert robot programming In general, LfD has been successful in solving problems for which optimal behavior can be demonstrated, but not necessarily be succinctly specified in mathematical form (e.g., a reward function). For instance, while it is straight-forward 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 non-experts to teach a variety of tasks to robots, without the need for significant software development or subject-matter expertise.

Data efficiency A number of 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, 96), and high-level task learning has been shown to be feasible with as few as one demonstration (24). Reinforcement learning (RL)-based approaches, on the other hand, typically optimize a specified reward function instead of demonstrations to learn new skills. Since RL 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 (188). 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 in order to improve sample efficiency (189, 190, 191, 192, 193, 194, 195, 196).

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 to techniques, such as reinforcement learning, that require significant exploration. This is due to the fact that demonstrations provide a way to assess the safety or risk associated with regions of state-space (e.g., (197, 198, 199)). Further, several LfD methods provide and utilize measures of uncertainty associated with different parts of the state-space (e.g., (62, 82, 101)), 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, 80, 105). 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 (170). However, recent advances provide ways to recognize and handle such scenarios (see Section 6.1 for further discussion).

Performance guarantees One of the significant factors that enable the wide-spread 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 shown to be capable of providing such guarantees. For instance, numerous dynamical systems-based trajectory learning methods provide strong convergence guarantees (70, 79, 81, 83, 100).

Platform independence As identified in Section 4, LfD has been successfully applied not only to a number of 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 learn policies, cost functions, and plans that are platform independent. By choosing a suitable common representation for the task, a number of LfD methods have been shown to be applicable to a wide range of platforms while only requiring minimal modifications and the availability of low-level controllers. For instance, the dynamical movement primitives (DMP) (100) algorithm has been utilized in a variety of platforms including manipulators (153), robotic hands (67, 200), humanoids (201), and aerial vehicles (202).

5.2. Limitations of LfD

In addition to the strengths identified above, the choice of LfD over other robot learning approaches is also 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 as identified below and its exciting challenges and directions for future research as discussed in Section 6.

Demonstrating complex behaviors Learning from demonstration necessitates an interface through which an expert can demonstrate the behavior. The choice of such interface directly impacts 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. This is due to the fact that it is very challenging to physically manipulate robotic platforms with joints, that belong to different kinematic chains and have to be operated simultaneously in order to achieve the desired behaviour. On the other hand, visual demonstrations, while being the most intuitive for the user, suffer from the correspondence problem as the demonstrator's actuation space differs from that of the robotic system. This requires the existence of a mapping between those spaces, which may be hard to provide due to the differences between the two systems in terms of motion constraints and dimensionality. 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 the existence of a mechanical or software interface that can be challenging to design.

Reliance on labeled data LfD, as mentioned earlier, relies on supervised learning-based techniques that extract information from labeled data. This reliance limits the ability of LfD to acquire new skills when sufficient amount of labeled data is unavailable. Indeed, as pointed out earlier, LfD is known to be data efficient and has been 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, etc. 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.

Sub-optimal 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 in order achieve good performance. This assumption implicitly answers the important question of *whom to imitate?*. However, this assumption might not hold in a variety of scenarios, such as when learning

from novices, crowd-sourcing demonstrations, and utilizing noisy sensors. Existing solutions to sub-optimal data are mostly limited to filtering sub-optimal demonstrations (203) or identifying sub-optimal demonstrations when the majority of the demonstrations are optimal (204). When most or all of the demonstrations are sub-optimal, it might not be feasible to utilize LfD methods without other sources of information revealing 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, a number of challenges still remain to be addressed if we aspire to enable robots that can 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 of 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 unseen, yet similar, stimuli (205). Indeed, generalization is seen as one of the central properties of animal cognition that helps in effectively dealing with novelty and variability (206).

Taking inspiration from the natural world, machine learning has studied generalization in artificial systems extensively. Indeed, generalization is at the core of machine learning - the ability to generalize differentiates systems that learn and 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 make the assumption that the training and testing data are independent and identically distributed (i.i.d). 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 (207). 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 it "veers" off the training distribution and encounters new states while executing a learned policy (170). However, this solution assumes extended and continuous availability of the demonstrator to provide corrections at appropriate times. Further, executions of sub-optimal policies in the initial stages of learning might not be suitable for physical systems with safety-critical constraints.

A distinction can be made between intra-task and inter-task generalization. The former refers to an algorithm's ability to generalize to novel conditions within a particular task (e.g., new initial and goal locations (70, 83, 100), new via points (62), and new object locations (15, 208)). Inter-task generalization, on the other hand, refers to the ability to generalize the learned skill to new, yet similar tasks. This is referred to as *skill transfer*.

Recently, meta or multi-task learning algorithms (e.g., (154, 209)) have been introduced to learn meta polices that can be quickly "fine-tuned" within a few iterations of training with data from the new task. Multi-task learning however assumes the 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 yet 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. Hyper-parameter selection

The challenge of hyper-parameter 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 suffer from this challenge, but the research in the field of LfD has to overcome this issue by providing methodologies which are capable of automatically choosing the hyper-parameters. Since one of the motives for applying LfD is to enable non-experts to program a robotic system, the need for manually tuning the model significantly decreases the ability of non-experts to use such methodology.

Hyperparameters can be found in many policy representations. For instance, end-to-end representations are usually modelled by neural networks. In those cases, the hyper-parameters are the number 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, DMP-based methods are widely used and an important hyper-parameter of this model is the number of radial basis functions (RBF). Highly nonlinear motions require more RBFs to be modelled. Nevertheless, if the demonstrated motion is not complex and a relatively large number of RBFs are used, then the model will also capture noise introduced from demonstrations, resulting in over-fitting. State-action representations are usually modelled as a Gaussian mixture model (GMM). In GMMs, the choice for the number of the Gaussian components affects the complexity of the estimated functions similarly to the aforementioned cases. For learning a cost/reward function through IRL, hyper-parameters, such as the type of function, have to be set alongside the number of desired features. In the case of learning high-level task plans, the hyper-parameters 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 hyper-parameter 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 (GPs). 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 determinine the optimal number of components is the use of Bayesian variational inference (210). Moreover, determining the hyper-parameters when learning high-level plans could be achieved automatically through clustering methods which would be capable of separating different sequences and actions without user intervention.

6.3. Evaluation and Benchmarking

Evaluating the performance of a LfD method is a challenging task due to the multiple factors that have to be taken into consideration. First of all, the learning method should be data and computationally efficient, the outcome of the method has to be smooth without sharp changes in order to minimize the risk of damage. Moreover, stability guarantees have to be provided, the method should be able to generalize efficiently and it should be able to solve the desired task with high repeatability. The majority of the aforementioned criteria are usually included for 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 – which 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 the comparison across multiple LfD methods difficult. Another criterion which is usually evaluated qualitatively is the generalization ability. In this case, ground-truth does not exist for the states which are not included in the demonstrations. Thus, the evaluation of generalization is either performed 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.

On the other hand, 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 straight forward and usually involves a success or failure rate of task completions. The learning efficiency is a measurement of how much data and computational time are required by the machine learning method in order to converge to a solution. A learning method which does not require much data and thus it can converge with a small amount of demonstrations significantly benefits its applicability to real-life application. An approach to evaluate the convergence is by a data-error plot where the number of demonstrations are illustrated w.r.t the error of the objective function. The convergence point corresponds to the amount of data that do not cause significant changes to the objective function. Thus this point can be determined by the elbow method (211). The computational complexity of the learning method is usually be reported since it is important for time-sensitive applications.

The plethora of LfD approaches alongside with the multiple evaluation criteria makes the comparison across methods extremely challenging. This fact underscores the need for benchmarks which will highlight the strengths and weaknesses of proposed approaches and provide insights regarding the cases in which each method should be utilized. In order to benchmark LfD methods, the design of a standard is needed 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 they are used to predict the testing demonstrations. Thus, the prediction error on the testing set can be a used as a representative metric of the algorithms generalization ability. Convergence of the learning method can be evaluated by plots which illustrate the amount of training data in relation with the prediction error. The ability of the LfD method to stabilize the system should ideally be mathematically proven, nevertheless this may be extremely challenging or impossible for certain model. In those cases, stability could be empirically demonstrated by letting the system execute actions from a large amount 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, there are number of other challenges that need to be addressed in order 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 the other, 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, there might exist other generalized definitions of distance that might be more appropriate for a given task (95, 111). 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. In order to learn complex tasks, it is important to learn both the high-level task plans and the low-level primitives, and 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. On the other hand, primitive learning methods ignore the existence of high-level task goals. Thus, a trivial combination of techniques at each level might not be effective owing to incompatible and potentially conflicting goals. Indeed, over the past decade, methods that can learn both low- and high-level behaviors simultaneously have been explored (see Section 3.4). However, further research is required in order to develop efficient methods for multi-level LfD that are generalizable and can apply to a wide variety of tasks.

6.4.3. Learning from multi-modal demonstrations. Research in cognitive psychology suggests that utilizing multi-sensory stimuli enhance human perceptual learning (e.g., (212)). Indeed, when we learn from others, we utilize a variety of multi-modal information, including verbal and non-verbal cues, to make sense of what is being taught. Current methods to multi-modal LfD are limited to learning from a small number of pre-specified modalities (e.g., (26, 213)). In order to effectively learn a wide variety of complex skills, we need methods that reason over demonstrations in multiple modalities, identify the most relevant and learn from them. Another challenging aspect of utilizing multi-modal demonstrations is user comfort and accessibility. It is not clear how to acquire highly multi-modal demonstrations without burdening the user by placing an overwhelming number of sensors. It also remains a challenge to effectively collect multi-modal demonstrations from remote users.

6.4.4. Learning from multiple demonstrators and cloud robotics. Most algorithms in LfD assume that there is a single optimal function (policy, cost, or plan) to be learned. While this assumptions 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 crowd-sourced 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, it has been shown that learning from demonstrations that reflect different levels of expertise exposes a richer reward function and structure, when compared to learning from a single demonstrator or demonstrators with similar expertise (214).

7. CONCLUSION

This survey provided an overview of methodologies, categorizations, applications, strengths, limitations, challenges, and future directions of research associated with the field of learning from demonstrations (LfD). Several aspects involved in the LfD pipeline were surveyed. First, we iden-

tified 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. Further, for each category of methods, we introduced detailed sub-categorizations and identified their core assumptions and utility.

LfD methods have been successfully applied in various industries and to the vast majority of the existing robotic platforms. This fact highlights the research field's potential to impact a wide variety of platforms, which extends from manipulators in the manufacturing and healthcare industries to mobile robots, such as autonomous vehicles and legged robots. This wide-spread application is explained by the strengths of LfD methods, which include enabling non-expert robot programming, 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 are either inherent to choosing LfD over other robot learning methods, or 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 is faced with important challenges and exciting avenues for further research. In this survey, we identified open problems and challenges which span across 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 hyper-parameters which have significant impact on performance. To ensure that the algorithms we develop remain to push boundaries, we require consistent and objective evaluation and benchmarking protocols which would compare and highlight the relative merits of 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 multi-modal 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.

ACKNOWLEDGEMENTS

This work is supported in part by the U.S. Army Research Lab Grant W911NF-17-2-0181 (DCIST CRA), NSF IIS 1564080, ONR N000141612835, an Early Career Faculty grant from NASA's Space Technology Research Grants Program, the EU projects Cogimon H2020-ICT-232014, and SecondHands H2020-ICT-643950.

LITERATURE CITED

1. Schaal S. 1999. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences* 3:233–242
2. Billard A, Calinon S, Dillmann R, Schaal S. 2008. Robot Programing by Demonstration. *Springer Handbook of Robotics* :1371–1394
3. Argall BD, Chernova S, Veloso M, Browning B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57:469–483

4. Chernova S, Thomaz AL. 2014. Robot learning from human teachers, vol. 8. Morgan & Claypool Publishers
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*
6. Zucker M, Ratliff N, Dragan AD, Pivtoraiko M, Klingensmith M, et al. 2013. Chomp: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research* 32:1164–1193
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 Robotics and Automation Letters* 2:1375–1382
9. Friedrich H, Kaiser M, Dillmann R. 1996. What can robots learn from humans? *Annual Reviews in Control* 20:167–172
10. Billard AG, Calinon S, Dillmann R. 2016. Learning from humans. In *Springer handbook of robotics*. Springer, 1995–2014
11. Osa T, Pajarinen J, Neumann G, Bagnell JA, Abbeel P, Peters J. 2018. An Algorithmic Perspective on Imitation Learning. *Foundations and Trends in Robotics* 7:1–179
12. Bohg J, Morales A, Asfour T, Kragic D. 2014. Data-driven grasp synthesis-A survey. *IEEE Transactions on Robotics* 30:289–309
13. Ahmadzadeh SR, Kaushik R, Chernova S. 2016. Trajectory learning from demonstration with canal surfaces: A parameter-free approach. *IEEE-RAS International Conference on Humanoid Robots* :544–549
14. Maeda GJ, Neumann G, Ewerton M, Lioutikov R, Kroemer O, Peters J. 2017. Probabilistic movement primitives for coordination of multiple humanrobot collaborative tasks. *Autonomous Robots* 41:593–612
15. Pervez A, Lee D. 2018. Learning task-parameterized dynamic movement primitives using mixture of GMMs. *Intelligent Service Robotics* 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 Robotics and Automation Letters* 3:2718–2725
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 (RO-MAN)*. 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 (HRI)*. IEEE
19. Calinon S, Guenter F, Billard A. 2007. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37:286–298
20. Nehaniv CL, Dautenhahn K, Dautenhahn K. 2002. Imitation in Animals and Artifacts (Chapter 2). MIT press
21. Abbeel P, Coates A, Ng AY. 2010. Autonomous Helicopter Aerobatics through Apprenticeship Learning. *The International Journal of Robotics Research* 29:1608–1639
22. Peters RA, Campbell CL, Bluethmann WJ, Huber E. 2003. Robonaut task learning through teleoperation. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE
23. Whitney D, Rosen E, Phillips E, Konidaris G, Tellex S. 2017. Comparing robot grasping teleoperation across desktop and virtual reality with ros reality. In *Proceedings of the International Symposium on Robotics Research*
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*. 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. Springer
26. Kormushev P, Calinon S, Caldwell DG. 2011. Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input. *Advanced Robotics* 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. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*
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 (ICRA)*. 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 (IST)*. IEEE
30. Whitney D, Rosen E, Tellex S. 2018. Learning from crowdsourced virtual reality demonstrations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*
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 (ICRA)*. 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 *Conference on Robot Learning*
33. Kent D, Behrooz M, Chernova S. 2016. Construction of a 3D object recognition and manipulation database from grasp demonstrations. *Autonomous Robots* 40:175–192
34. Aleotti J, Caselli S. 2011. Part-based robot grasp planning from human demonstration. In *2011 IEEE International Conference on Robotics and Automation*. IEEE
35. Havoutis I, Calinon S. 2018. Learning from demonstration for semi-autonomous teleoperation. *Autonomous Robots* 43:1–14
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 (Biorob)*. IEEE
37. Dillmann R. 2004. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems* 47:109–116
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 (ICRA)*. 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*. 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 (ICRA)*. IEEE
41. Pervez A, Mao Y, Lee D. 2017. Learning deep movement primitives using convolutional neural networks. In *IEEE-RAS International Conference on Humanoid Robots*
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 (ICRA)*. IEEE
43. Schulman J, Ho J, Lee C, Abbeel P. 2016. Learning from demonstrations through the use of non-rigid registration. In *Robotics Research*. Springer, 339–354
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*. 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*. ACM
46. Cakmak M, Chao C, Thomaz AL. 2010. Designing interactions for robot active learners. *IEEE Transactions on Autonomous Mental Development* 2:108–118
47. Bullard K, Schroecker Y, Chernova S. 2019. Active Learning within Constrained Environments through

Imitation of an Expert Questioner. In *2019 International Joint Conference on Artificial Intelligence (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 (IROS)*. 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 (HRI)*. 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*. ACM
51. Amershi S, Cakmak M, Knox WB, Kulesza T. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35:105–120
52. Laird JE, Gluck K, Anderson J, Forbus KD, Jenkins OC, et al. 2017. Interactive task learning. *IEEE Intelligent Systems* 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 (HRI)*. 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*. International Foundation for Autonomous Agents and Multiagent Systems
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*. 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 (ICRA)*. 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*
58. Ho J, Ermon S. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*
59. Schneider M, Ertel W. 2010. Robot learning by demonstration with local gaussian process regression. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings* :255–260
60. Akgun B, Cakmak M, Jiang K, Thomaz AL. 2012. Keyframe-based learning from demonstration. *International Journal of Social Robotics* 4:343–355
61. Lin Y, Ren S, Clevenger M, Sun Y. 2012. Learning grasping force from demonstration. In *Proceedings - IEEE International Conference on Robotics and Automation*
62. Paraschos A, Daniel C, Peters J, Neumann G. 2013. Probabilistic Movement Primitives. *Neural Information Processing Systems* :1–9
63. Rozo L, Calinon S, Caldwell D, Jimenez P, Torras C, Jiménez P. 2013. Learning Collaborative Impedance-based Robot Behaviors. In *AAAI Conference on Artificial Intelligence*
64. Osa T, Harada K, Sugita N, Mitsuishi M. 2014. Trajectory planning under different initial conditions for surgical task automation by learning from demonstration. *Proceedings - IEEE International Conference on Robotics and Automation* :6507–6513
65. Reiner B, Ertel W, Posenauer H, Schneider M. 2014. LAT: A simple Learning from Demonstration method. In *IEEE International Conference on Intelligent Robots and Systems*
66. Calinon S, Evrard P. 2009. Learning collaborative manipulation tasks by demonstration using a haptic interface. *International Conference on Advanced Robotics* :1–6
67. Pastor P, Hoffmann H, Asfour T, Schaal S. 2009. Learning and Generalization of Motor Skills by Learning from Demonstration. *Proceedings of the 2009 IEEE International Conference on Robotics and Automation* :1293–1298
68. Calinon S, Sardellitti I, Caldwell DG. 2010. Learning-based control strategy for safe human-robot

interaction exploiting task and robot redundancies. In *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*
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 Robotics and Automation Magazine* 17:44–45
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 Transactions on Robotics* 27:943–957
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: New Boundaries for Robotics, ICAR 2011* :124–129
72. Kronander K, Billard A. 2012. Online learning of varying stiffness through physical human-robot interaction. In *Proceedings - IEEE International Conference on Robotics and Automation*
73. Herzog A, Pastor P, Kalakrishnan M, Righetti L, Bohg J, et al. 2014. Learning of grasp selection based on shape-templates. *Autonomous 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 *Proceedings - IEEE International Conference on Robotics and Automation*
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 (ICRA)*. IEEE
76. Kober J, Gienger M, Steil JJ. 2015. Learning movement primitives for force interaction tasks. In *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June
77. Pervez A, Lee D. 2015. Task parameterized Dynamic Movement Primitives by using mixture of GMMs
78. 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 (ICRA)*. IEEE
79. Neumann K, Steil JJ. 2015. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robotics and Autonomous Systems* 70:1–15
80. Denisa M, Gams A, Ude A, Petric T. 2016. Learning Compliant Movement Primitives Through Demonstration and Statistical Generalization. *IEEE/ASME Transactions on Mechatronics* 21:2581–2594
81. Perrin N, Schlehuber-Caissier P. 2016. Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Systems and Control Letters* 96:51–59
82. Rana MA, Mukadam M, Ahmadzadeh SR, Chernova S, Boots B. 2017. Towards Robust Skill Generalization: Unifying Learning from Demonstration and Motion Planning. *Conference on Robot Learning* :109–118
83. Ravichandar H, Dani A. 2018. Learning position and orientation dynamics from demonstrations via contraction analysis. *Autonomous Robots* 43:1–16
84. Silverio J, Huang Y, Rozo L, Calinon S, Caldwell DG. 2018. Probabilistic Learning of Torque Controllers from Kinematic and Force Constraints. In *IEEE International Conference on Intelligent Robots and Systems*
85. Maeda G, Ewerton M, Lioutikov R, Ben Amor H, Peters J, Neumann G. 2015. Learning interaction for collaborative tasks with probabilistic movement primitives. In *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Febru
86. 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*. IEEE, IEEE
87. Ciocarlie M, Goldfeder C, Allen PK. 2007. Dimensionality reduction for hand-independent dexterous robotic grasping
88. Jonschkowski R, Brock O. 2015. Learning state representations with robotic priors. *Autonomous Robots* 39:407–428
89. 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 (ICRA). IEEE
90. Byravan A, Fox D. 2017. Se3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE
91. Finn C, Levine S. 2017. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE
92. 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. *Advanced Robotics* 22:1521–1537
93. Polydoros AS, Boukas E, Nalpantidis L. 2017. Online multi-target learning of inverse dynamics models for computed-torque control of compliant manipulators. In *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe. IEEE
94. 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 *Proceedings - IEEE International Conference on Robotics and Automation*
95. Ravichandar H, Salehi I, Dani AP. 2017. Learning Partially Contracting Dynamical Systems from Demonstrations. In *Conference on Robot Learning*
96. Ravichandar H, Ahmadzadeh SR, Rana MA, Chernova S. 2019. Skill Acquisition via Automated Multi-Coordinate Cost Balancing. In *IEEE International Conference on Robotics and Automation*
97. Manschitz S, Gienger M, Kober J, Peters J. 2018. Mixture of attractors: A novel movement primitive representation for learning motor skills from demonstrations. *IEEE Robotics and Automation Letters* 3:926–933
98. 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 (IROS)*. IEEE
99. Chatzis SP, Korkinof D, Demiris Y. 2012. A nonparametric Bayesian approach toward robot learning by demonstration. *Robotics and Autonomous Systems* 60:789–802
100. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S. 2013. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation* 25:328–373
101. Umlauft J, Hirche S. 2017. Learning stable stochastic nonlinear dynamical systems. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org
102. Petrič T, Gams A, Colasanto L, Ijspeert AJ, Ude A. 2018. Accelerated sensorimotor learning of compliant movement primitives. *IEEE Transactions on Robotics* 34:1636–1642
103. Ravichandar H, Trombetta D, Dani A. 2019. Human Intention-Driven Learning Control for Trajectory Synchronization in Human-Robot Collaborative Tasks. *IFAC-PapersOnLine* 51
104. Peternel L, Petrič T, Babič J. 2018. Robotic assembly solution by human-in-the-loop teaching method based on real-time stiffness modulation. *Autonomous Robots* 42:1–17
105. Suomalainen M, Kyrki V. 2016. Learning compliant assembly motions from demonstration. In *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem
106. Polydoros AS, Nalpantidis L. 2017. Survey of Model-Based Reinforcement Learning: Applications on Robotics. *Journal of Intelligent & Robotic Systems* 86:153–173
107. Englert P, Paraschos A, Deisenroth MP, Peters J. 2013. Probabilistic model-based imitation learning. *Adaptive Behavior* 21:388–403
108. Ratliff N, Zucker M, Bagnell JA, Srinivasa S. 2009. CHOMP: Gradient optimization techniques for efficient motion planning
109. 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*. IEEE
110. Kalakrishnan M, Pastor P, Righetti L, Schaal S. 2013. Learning objective functions for manipulation. In *2013 IEEE International Conference on Robotics and Automation*. IEEE
111. Dragan AD, Muelling K, Bagnell JA, Srinivasa SS. 2015. Movement primitives via optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE
112. Bajcsy A, Losey DP, OMalley MK, Dragan AD. 2017. Learning robot objectives from physical human interaction. *Proceedings of Machine Learning Research* 78:217–226

113. Ng AY, Russell SJ, others. 2000. Algorithms for inverse reinforcement learning. In *Icml*, vol. 1
114. Dvijotham K, Todorov E. 2010. Inverse optimal control with linearly-solvable MDPs. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*
115. Levine S, Koltun V. 2012. Continuous inverse optimal control with locally optimal examples. *arXiv preprint arXiv:1206.4617*
116. Doerr A, Ratliff ND, Bohg J, Toussaint M, Schaal S. 2015. Direct Loss Minimization Inverse Optimal Control. In *Robotics: Science and Systems*
117. Finn C, Levine S, Abbeel P. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*
118. Silver D, Bagnell JA, Stentz A. 2010. Learning from demonstration for autonomous navigation in complex unstructured terrain. In *International Journal of Robotics Research*, vol. 29
119. Boularias A, Kober J, Peters J. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*
120. Hadfield-Menell D, Russell SJ, Abbeel P, Dragan A. 2016. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems*
121. Ratliff ND, Silver D, Bagnell JA. 2009. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots* 27:25–53
122. Abbeel P, Ng AY. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*. ACM
123. Zucker M, Ratliff N, Stolle M, Chestnutt J, Bagnell JA, et al. 2011. Optimization and learning for rough terrain legged locomotion. *The International Journal of Robotics Research* 30:175–191
124. Ziebart BD. 2010. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. Ph.D. thesis, figshare
125. Choi J, Kim KE. 2011. Map inference for bayesian inverse reinforcement learning. In *Advances in Neural Information Processing Systems*
126. 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 (HRI)*. IEEE
127. Kober J, Bagnell JA, Peters J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32:1238–1274
128. Ekvall S, Kragic D. 2008. Robot learning from demonstration: A task-level planning approach. *International Journal of Advanced Robotic Systems* 5:223–234
129. Grollman DH, Jenkins OC. 2010. Incremental learning of subtasks from unsegmented demonstration. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings* :261–266
130. Mohseni-Kabir A, Li C, Wu V, Miller D, Hylak B, et al. 2018. Simultaneous learning of hierarchy and primitives for complex robot tasks. *Autonomous Robots* 43:1–16
131. Yin H, Melo FS, Paiva A, Billard A. 2018. An ensemble inverse optimal control approach for robotic task learning and adaptation. *Autonomous Robots* 43:1–22
132. Konidaris G, Kuindersma S, Grupen R, Barto A. 2012. Robot learning from demonstration by constructing skill trees. *International Journal of Robotics Research* 31:360–375
133. Manschitz S, Kober J, Gienger M, Peters J. 2014. Learning to sequence movement primitives from demonstrations. In *IEEE International Conference on Intelligent Robots and Systems*
134. 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 (ICRA)*. IEEE
135. 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 (ICRA)*. IEEE
136. Pastor P, Kalakrishnan M, Righetti L, Schaal S. 2012. Towards associative skill memories. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE
137. 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*. IEEE

138. 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*. IEEE
139. 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. *The International Journal of Robotics Research* 31:330–345
140. 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*. IEEE
141. 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 (ICRA)*. IEEE
142. 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 (Humanoids)*. IEEE
143. 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 (ICRA)*. IEEE
144. Baisero A, Mollard Y, Lopes M, Toussaint M, Lütkebohle I. 2015. Temporal segmentation of pairwise interaction phases in sequential manipulation demonstrations. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE
145. Takano W, Nakamura Y. 2015. Statistical mutual conversion between whole body motion primitives and linguistic sentences for human motions. *The International Journal of Robotics Research* 34:1314–1328
146. 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*. ACM
147. Pardowitz M, Knoop S, Dillmann R, Zollner RD. 2007. Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 37:322–332
148. 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. *International Journal of Social Robotics* 4:437–448
149. 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 (ICRA)*. IEEE
150. Gombolay M, Jensen R, Stigile J, Golen T, Shah N, et al. 2018. Human-machine collaborative optimization via apprenticeship scheduling. *Journal of Artificial Intelligence Research* 63:1–49
151. 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*. IEEE
152. Butterfield J, Osentoski S, Jay G, Jenkins OC. 2010. Learning from demonstration using a multivalued function regressor for time-series data. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE
153. Niekum S, Osentoski S, Konidaris G, Chitta S, Marthi B, Barto AG. 2015. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research* 34:131–157
154. 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*
155. 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. *The International Journal of Robotics Research* 38:126–145
156. 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. *CoRR* abs/1604.0

157. Hirzinger G, Heindl J. 1983. Sensor programming, a new way for teaching a robot paths and forces torques simultaneously. In *Intelligent Robots: Conference on Robot Vision and Sensory Controls, Cambridge, Massachusetts/USA*
158. Asada H, Izumi H. 1987. Direct teaching and automatic program generation for the hybrid control of robot manipulators. In *IEEE International Conference on Robotics and Automation*, vol. 4. IEEE
159. Kronander K, Khansari M, Billard A. 2015. Incremental motion learning with locally modulated dynamical systems. *Robotics and Autonomous Systems* 70:52–62
160. 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*. IEEE
161. Bhattacharjee T, Lee G, Song H, Srinivasa SS. 2019. Towards robotic feeding: Role of haptics in fork-based food manipulation. *IEEE Robotics and Automation Letters* 4:1485–1492
162. Fong J, Tavakoli M. 2018. Kinesthetic teaching of a therapist's behavior to a rehabilitation robot. In *2018 International Symposium on Medical Robotics (ISMR)*. IEEE
163. 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 Robotics and Automation Letters* 1:391–398
164. Najafi M, Sharifi M, Adams K, Tavakoli M. 2017. Robotic assistance for children with cerebral palsy based on learning from tele-cooperative demonstration. *International Journal of Intelligent Robotics and Applications* 1:43–54
165. Moro C, Nejat G, Mihailidis A. 2018. Learning and Personalizing Socially Assistive Robot Behaviors to Aid with Activities of Daily Living. *ACM Transactions on Human-Robot Interaction (THRI)* 7:15
166. Ma Z, Ben-Tzvi P, Danoff J. 2015. Hand rehabilitation learning system with an exoskeleton robotic glove. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24:1323–1332
167. Strabala K, Lee MK, Dragan A, Forlizzi J, Srinivasa SS, et al. 2013. Toward seamless human-robot handovers. *Journal of Human-Robot Interaction* 2:112–132
168. Pomerleau DA. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation* 3:88–97
169. Boularias A, Krömer O, Peters J. 2012. Structured apprenticeship learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer
170. 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*
171. Silver D, Bagnell JA, Stentz A. 2012. Active learning from demonstration for robust autonomous navigation. In *2012 IEEE International Conference on Robotics and Automation*. IEEE
172. Li Y, Song J, Ermon S. 2017. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*
173. 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*
174. Kuderer M, Gulati S, Burgard W. 2015. Learning driving styles for autonomous vehicles from demonstration. In *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June
175. 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*. IEEE
176. Kaufmann E, Loquercio A, Ranftl R, Dosovitskiy A, Koltun V, Scaramuzza D. 2018. Deep Drone Racing: Learning Agile Flight in Dynamic Environments. In *Conference on Robot Learning*
177. Loquercio A, Maqueda AI, Del-Blanco CR, Scaramuzza D. 2018. Dronet: Learning to fly by driving. *IEEE Robotics and Automation Letters* 3:1088–1095
178. 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 multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems
179. Meriçli , Veloso M. 2010. Biped walk learning through playback and corrective demonstration. In

Twenty-Fourth AAAI Conference on Artificial Intelligence
180. 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*. Springer
181. Kolter JZ, Abbeel P, Ng AY. 2008. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Advances in Neural Information Processing Systems*
182. Kalakrishnan M, Buchli J, Pastor P, Mistry M, Schaal S. 2011. Learning, planning, and control for quadruped locomotion over challenging terrain. *The International Journal of Robotics Research* 30:236–258
183. Nakanishi J, Morimoto J, Endo G, Cheng G, Schaal S, Kawato M. 2004. Learning from demonstration and adaptation of biped locomotion. In *Robotics and Autonomous Systems*, vol. 47
184. 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*. IEEE
185. Havoutis I, Calinon S. 2017. Supervisory teleoperation with online learning and optimal control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE
186. 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 Robotics & Automation Magazine* 25:24–33
187. Somers T, Hollinger GA. 2016. Human–robot planning and learning for marine data collection. *Autonomous Robots* 40:1123–1137
188. 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-Volume 70*. JMLR. org
189. Kober J, Peters JR. 2009. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*
190. 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 - Volume 2*, AAMAS '11. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems
191. 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*. IEEE
192. Kim B, Farahmand Am, Pineau J, Precup D. 2013. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems*
193. Večer\'\ik 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 preprint arXiv:1707.08817*
194. 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 (ICRA)*. IEEE
195. 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 (ICRA)*. IEEE
196. 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 (ICRA)*. IEEE
197. Brown DS, Niekum S. 2017. Toward probabilistic safety bounds for robot learning from demonstration. In *2017 AAAI Fall Symposium Series*
198. 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 (ICRA)*. IEEE
199. Zhou W, Li W. 2018. Safety-aware apprenticeship learning. In *International Conference on Computer Aided Verification*. Springer
200. 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 (IROS)*. IEEE

201. 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*. IEEE
202. Lee H, Kim H, Kim HJ. 2016. Planning and control for collision-free cooperative aerial transportation. *IEEE Transactions on Automation Science and Engineering* 15:189–201
203. Coates A, Abbeel P, Ng AY. 2008. Learning for control from multiple demonstrations. In *International Conference on Machine learning*. ACM
204. Choi S, Lee K, Oh S. 2016. Robust learning from demonstration using leveraged Gaussian processes and sparse-constrained optimization. In *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June
205. Shepard RN. 1987. Toward a universal law of generalization for psychological science. *Science* 237:1317–1323
206. Ghirlanda S, Enquist M. 2003. A century of generalization. *Animal Behaviour* 66:15–36
207. Bagnell JA. 2015. An invitation to imitation. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST
208. Calinon S, Bruno D, Caldwell DG. 2014. A task-parameterized probabilistic model with minimal intervention control. In *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE
209. Finn C, Yu T, Zhang T, Abbeel P, Levine S. 2017. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*
210. Corduneanu A, Bishop CM. 2001. Variational Bayesian Model Selection for Mixture Distributions. *In Artificial Intelligence and Statistics* :27–34
211. Ketchen DJ, Shook CL. 1996. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal* 17:441–458
212. Shams L, Seitz AR. 2008. Benefits of multisensory learning. *Trends in cognitive sciences* 12:411–417
213. Sung J, Jin SH, Saxena A. 2018. Robobarista: Object part based transfer of manipulation trajectories from crowd-sourcing in 3d pointclouds. In *Robotics Research*. Springer, 701–720
214. Castro PS, Li S, Zhang D. 2019. Inverse Reinforcement Learning with Multiple Ranked Experts