This chapter surveys the main approaches developed to date to endow robots with the ability to learn from human guidance. The field is best known as robot programming by demonstration, robot learning from/by demonstration, apprenticeship learning and imitation learning. We start with a brief historical overview of the field. We then summarize the various approaches taken to solve four main questions: when, what, who and when to imitate. We emphasize the importance of choosing well the interface and the channels used to convey the demonstrations, with an eye on interfaces providing force control and force feedback. We then review algorithmic approaches to model skills individually and as a compound and algorithms that combine learning from human guidance with reinforcement learning. We close with a look on the use of language to guide teaching and a list of open issues.

74.1 Learning of Robots

Robot learning from humans relates to situations in which the robot learns from interacting with a human. This must be contrasted to the vast body of work on robot learning where the robot learns on its own, that is, through trial and error and without external guidance. In this chapter, we cover works that combine reinforcement learning (RL) with techniques that use human guidance, e.g., to bootstrap the search in RL. However, we exclude from this survey all works that use purely reinforcement learning, even though one could argue that providing a reward is one form of human guidance. We consider that providing a reward function is akin to providing an objective function and hence refer the reader to the companion chapter on Machine Learning for robotics. We also exclude works where the robot learns implicitly from being in presence of a human, while the human is not actively coaching the robot, as these works are covered in the companion chapter on Social Robotics. We hence focus our survey to all works where the human is actively teaching the robot, by providing demonstrations of how to perform the task.

Various terminologies have been used to refer to this body of work. These include programming by demonstration (PbD), learning from human demonstration (LfD), imitation learning, and apprenticeship learning. All of these refer to a general paradigm for enabling robots to autonomously perform new tasks from ob-
serving and learning, therefore, from the observation of humans performing these tasks.

74.1.1 Principle

Rather than requiring users to analytically decompose and manually program a desired behavior, work in LfD-PbD takes the view that an appropriate robot controller can be derived from observations of a human’s own performance thereof. The aim is for robot capabilities to be more easily extended and adapted to novel situations, even by users without programming ability:

*The main principle of robot learning from demonstration is that end-users can teach robots new tasks without programming.*

Consider a household robot capable of performing manipulation tasks. One task that an end-user may desire the robot to perform is to prepare a meal, such as preparing an orange juice for breakfast (Fig. 74.1 and [VIDEO 29](#)). Doing so may involve multiple sub-tasks, such as juicing the orange, throwing the rest of the orange in the trash, and pouring the liquid into a cup. Further, every time this meal is prepared, the robot will need to adapt its motion to the fact that the location and type object (cup, juicer) may change.

In a traditional programming scenario, a human programmer would have to code a robot controller that is capable of responding to any situation the robot may face. The overall task may need to be broken down into tens or hundreds of smaller steps, and each one of these steps should be tested for robustness prior to the robot leaving the factory. If and when failures occur in the field, highly-skilled technicians would need to be dispatched to update the system for the new circumstances. Instead, LfD allows the end-user to *program* the robot simply by showing it how to perform the task – no coding is required. Then, when failures occur, the end-user only needs to provide more demonstrations, rather than calling for professional help. LfD hence seeks to endow robots with the ability to learn what it means to perform a task by generalizing from several observations (Fig. 74.1 and [VIDEO 29](#)).

*LfD is not a record and play technique. LfD implies learning, henceforth, generalization.*

Next, we give a brief historical overview of the way the field evolved over the years. This is followed, in Sect. 74.2, by an introduction to the issues at the core of LfD. In Sect. 74.3, we discuss the crucial role that the interface used for LfD plays in the success of the teaching, emphasizing how the choice of interface determines the type of information that can be conveyed to the robot. Finally, in Sect. 74.4, we give a generic view of the main approaches to solving LfD and conclude with an outlook on open issues.

74.1.2 Brief History

Robot learning from demonstration started in the 1980s. Then, and still to a large extent now, robots had to be explicitly and tediously hand programmed for each task they had to perform. PbD sought to minimize, or even eliminate, this difficult step.

The rationale for moving from purely preprogrammed robots to very flexible user-based interfaces for training the robot to perform a task is threefold. First and foremost, PbD is a powerful mechanism for reducing the complexity of search spaces for learning. When observing either good or bad examples, one can reduce the search for a possible solution, by either starting the search from the observed good solution (local optima), or conversely, by eliminating from the search space what is known as a bad solution. Imitation learning is, thus, a powerful tool for enhancing and accelerating learning in both animals and artifacts.

![Using invariants in relative positions](#)

Fig. 74.1 (a) The teacher does several demonstrations of the task of juicing an orange, by changing the location of each item to allow the robot to generalize correctly. That is, the robot should be able to infer, by comparing the demonstrations, that only the relative locations matter, as opposed to the exact locations as recorded from a global coordinate system. (b) The robot can then reproduce the task even when the objects are located in positions not seen in the demonstrations [VIDEO 29](#)
Second, imitation learning offers an implicit means of training a machine, such that explicit and tedious programming of a task by a human user can be minimized or eliminated. Imitation learning is thus a natural means of interacting with a machine that would be accessible to lay people.

Third, studying and modeling the coupling of perception and action, which is at the core of imitation learning, helps us to understand the mechanisms by which the self-organization of perception and action could arise during development. The reciprocal interaction of perception and action could explain how competence in motor control can be grounded in the rich structure of perceptual variables, and vice versa, how the processes of perception can develop as means to create successful actions.

PbD promises were thus multiple. On the one hand, one hoped that it would make the learning faster, in contrast to trial-and-error methods trying to learn the skill tabula rasa. On the other hand, one expected that being user-friendly, the methods would enhance the application of robots in human daily environments.

At the beginning of the 1980s, LfD, known then as programming by demonstration (PbD), started attracting attention in manufacturing robotics. PbD appeared as a promising route to automate the tedious manual programming of robots, reducing the costs involved in the development and maintenance of robots in the factory.

As a first approach in PbD, symbolic reasoning was commonly adopted in robotics [74.1−5], with processes referred to as teach-in, guiding, or playback methods. In these works, PbD was performed through manual (teleoperated) control. The position of the end-effector and the forces applied on the object manipulated were stored throughout the demonstrations together with the positions and orientations of the obstacles and of the target. This sensorimotor information was then segmented into discrete subgoals (key points along the trajectory) and into appropriate pre-defined actions to attain these subgoals. Actions were commonly chosen to be simple point-to-point movements that industrial robots employed at this time. Examples of subgoals would be, e.g., the robot’s gripper orientation and position in relation to the goal [74.3]. Consequently, the demonstrated task was segmented into a sequence of state-action-state transitions.

To take into account the variability of human motion and the noise inherent to the sensors capturing the movements, it appeared necessary to develop a method that would consolidate all demonstrated movements. For this purpose, the state-action-state sequence was converted into symbolic if-then rules, describing the states and the actions according to symbolic relationships, such as in contact, close-to, move-to, grasp-object, move-above, etc. Appropriate numerical definitions of these symbols (i.e., when would an object be considered as close-to or far-from) were given as prior knowledge to the system. A complete demonstration was thus encoded in a graph-based representation, where each state constituted a graph node and each action a directed link between two nodes. Symbolic reasoning could then unify different graphical representations for the same task by merging and deleting nodes [74.2].

Munch et al. [74.6] suggested the use of machine learning (ML) techniques to recognize elementary operators (EOs), thus defining a discrete set of basic motor skills, with industrial robotics applications in mind. In this early work, the authors already established several key issues of PbD in robotics. These include questions such as how to generalize a task, how to reproduce a skill in a completely novel situation, how to evaluate a reproduction attempt, and how to better define the role of the user during learning. Munch et al. [74.6] admitted that generalizing over a sequence of discrete actions was only one part of the problem since the controller of the robot also required the learning of continuous trajectories to control the actuators. They proposed to overcome the missing parts of the learning process by leveraging them to the user, who took an active role in the teaching process.

These early works highlighted the importance of providing a set of examples that are usable by the robot: (1) by constraining the demonstrations to modalities that the robot can understand; and (2) by providing a sufficient number of examples to achieve a desired generality. They noted the importance of providing an adaptive controller to reproduce the task in new situations, that is, how to adjust an already acquired program. The evaluation of a reproduction attempt was also leveraged to the user by letting him/her provide additional examples of the skill in the regions of the learning space that had not been covered yet. In this way, the teacher/expert could control the generalization capabilities of the robot.

With the increasing development of mobile and humanoid robots, the field went on adopting an interdisciplinary approach, taking into account evidence of specific neural mechanisms for visuomotor imitation in primates [74.7−9] and of developmental stages of imitation capacities in children [74.10, 11]. The latter promotes the introduction of socially driven behavior in the robot to sustain interaction and improve teaching [74.12, 13] and of an interactive teaching process, in which the robot takes a more active role and may ask the user for additional sources of information, when
needed [74.14, 15]. Eventually, the notion of robot programming by demonstration was replaced by the more biological labeling of imitation learning. In essence, a large part of current works in PbD follow a conceptual approach very similar to that followed by these prior works.

Recent progress affected mostly the interfaces at the basis of the teaching. Traditional ways of guiding/teleoperating the robot have been progressively replaced by more user-friendly interfaces, such as vision [74.16, 17], speech command [74.18], data gloves [74.19], the laser range finder [74.20] or kinesthetic teaching (i.e., by manually guiding the robot’s arms through the motion) [74.21–23].

The field progressively moved from simply copying the demonstrated movements to generalizing across sets of demonstrations. As machine learning progressed, PbD started incorporating more of those tools to tackle both the perception issue, i.e., how to generalize across demonstrations, and the production issue, i.e., how to generalize the movement to new situations. Initially, tools such as artificial neural networks (ANNs) [74.24, 25], radial-basis function networks (RBFs) [74.26], and fuzzy logic [74.27] were quite popular. These have lately been replaced by hidden Markov models (HMMs) [74.28–33] and various non-linear regression techniques [74.21, 34, 35], as we will discuss in more detail in Sect. 74.4.

New learning challenges were, thus, set forth. Robots were expected to show a high degree of flexibility and versatility both in their learning system and in their control system in order to be able to interact naturally with human users and demonstrate similar skills (e.g., by moving in the same rooms and manipulating the same tools as humans). Robots were more and more expected to act human-like to enhance the interaction and so that their behavior would be more predictable and, hence, more acceptable.

### 74.2 Key Issues When Learning from Human Demonstrations

As mentioned in the beginning, learning from demonstration (LfD) has at core to develop algorithms that are generic in their representation of the skills and in the way they generate the skills.

The field has identified a number of key problems that need to be solved for ensuring such a generic approach to transferring skills across various agents and situations [74.36, 37]. These have been formulated as a set of generic questions, namely what to imitate, how to imitate, when to imitate, and who to imitate. These questions were formulated in response to the large body of diverse work in robotics LfD [74.18, 26, 38–41] that could not easily be unified under a small number of coherent operating principles. The above four questions and their solutions aim at being generic in the sense of making no assumptions on the type of skills that may be transmitted.

#### 74.2.1 When and Whom to Imitate

Whom and when to imitate has been largely unexplored so far, and hence to date, only the first two questions have really been addressed. Figure 74.2 and [Fig. 74.2](#) illustrate how these two problems can be solved in a principled manner through statistical observation of the demonstrations.

**How to Determine the Evaluation Metric**

What to imitate relates to the problem of determining which aspects of the demonstration should be imitated. For a given task, certain observable or affectable properties may be irrelevant and safely ignored. For instance, if the demonstrator always approaches a location from the north, is it necessary for the robot to do the same? The answer to this question strongly in-
fluences whether or not a derived robot controller is a successful imitation – a robot that approaches from the south is appropriately trained if direction is not important, but needs further education if it is. This issue is related to questions of signal versus noise and is answered by determining the metric by which the resulting behavior is evaluated. Different ways can be taken to address this issue. The simplest approach is to take a statistical perspective and deem as relevant the parts (dimension, region of input space) of the data that are consistently measured across all demonstration instances [74.21]. If the dimension of the data is too high, such an approach may require too many demonstrations to gather enough statistics. An alternative is then to have the teacher help the robot determine what is relevant by pointing out the parts of the task that are most important.

In summary, what to imitate removes consideration of details that, while perceptible/performable, do not matter for the task. It participates in determining the metric by which the reproduction of the robot can be measured. In continuous control tasks, what to imitate relates to the problem of defining automatically the feature space for learning, as well the constraints and the cost function. In discrete control tasks, such as those treated by reinforcement learning and symbolic reasoning, what to imitate relates to the problem of how to define the state and action space and of how to automatically learn the pre/post conditions in an autonomous decision system.

74.2.2 How to Imitate and How to Solve the Correspondence Problem

How to imitate consists in determining how the robot will actually perform the learned behaviors to maximize the metric found when solving the what to imitate problem. Often, a robot cannot act exactly the same way as a human does, due to differences in physical embodiment. For example, if the demonstrator uses a foot to move an object, is it acceptable for a wheeled robot to bump it, or should it use a gripper instead? If the metric does not have appendage-specific terms, it may not matter.

This issue is closely related to that of the correspondence problem [74.36]. Robots and humans, while inhabiting the same space and interacting with the same objects, and perhaps even superficially similar, still perceive and interact with the world in fundamentally different ways. To evaluate the similarity between human behavior and that of robots, we must first deal with the fact that humans and robots may occupy different state spaces, of perhaps different dimensions. We identify two different ways in which states of demonstrator and imitator can be said to correspond, and give brief examples:

- **Perceptual equivalence:** Due to differences between human and robot sensory capabilities, the same scene may appear to be very different. For instance, while a human may identify humans and...
gestures from color and intensity, a robot may use depth measurements to observe the same scene (Fig. 74.3a). Another point of comparison is tactile sensing. Most tactile sensors allow robots to perceive contact, but do not offer information about temperature, in contrast to the human skin. Moreover, the low resolution of the robots’ tactile sensors does not allow robots to discriminate across the variety of existing textures, while human skin does. As the same data may, therefore, not be available to both humans and robots, successfully teaching a robot may require a good understanding of the robot’s sensors and their limitations. LfD explores the limits of these perceptual equivalences, by building interfaces that either automatically correct or make explicit these differences.

- **Physical equivalence**: Due to differences between human and robot embodiments, humans and robots may perform different actions to accomplish the same physical effect. For instance, even when performing the same task (soccer), humans and robots may interact with the environment in different ways (Fig. 74.3b). Humans run and kick, while robots roll and bump. Solving this discrepancy in motor capabilities is akin to solving the *how to imitate* problem to achieve the same effect. LfD develops ways to solve this problem. Typically, the robot may compute a path (in Cartesian space) for its end-effector that is close to the path followed by the human hand, while relying on inverse kinematics to find the appropriate joint displacements. In the football example above, this would require the robot to determine a path for its center of mass which corresponds to the path followed by the human’s right foot when projected on the ground. Clearly, this equivalence is very task dependent. Recent solutions to this problem for hand motion and body motion can be found in [74.45, 46].

We can think of perceptual equivalence as dealing with the manner in which the agents perceive the world. Perceptual equivalence requires to make sure that the information necessary to perform the task is available to both humans and robots. Physical equivalence deals with the manner in which agents affect and interact with the world, so that the task is performable by both agents.

### 74.3 Interfaces for Demonstration

The interface used to provide demonstration plays a key role in the way the information is gathered and transmitted. We distinguish three major trends:

1. One may directly record human motions. If one is interested solely in the kinematic of the motion, one may use any of the various existing motion tracking systems, whether these are based on vision, exoskeleton, or other types of wearable motion sensors. The left-hand side of Fig. 74.4b and [VIDEO 98] show an example of full body motion tracking during walking using vision. The motion of the human body is first extracted from the background using a model of human body. This model is subsequently mapped to an avatar and then to the humanoid robot DB at ATR, Kyoto, Japan.

These external means of tracking human motion return precise measurement of the angular displacement of the limbs and joints. They have been used in various works for LfD of full body motion [74.33, 47–49]. These methods are advantageous in that they allow the human to move freely. However, they require solutions to the correspondence problem, i.e., the problem of how to transfer motion from human to robot when both differ in the kinematic and dynamics of their body or, in other words, if the configuration space is of different dimension and size. This is typically done when mapping the motion of the joints that are tracked visually to a model of the human body that matches closely that of the robot. Such mapping would be particularly difficult to perform when the walking machine (e.g., a hexapod) differs importantly from the human body. The problem of mapping actions across two dissimilar bodies was already evoked earlier on and refers to the correspondence problem.

2. Second, there are techniques such as kinesthetic teaching, where the robot is physically guided through the task by the human. This approach simplifies the correspondence problem by letting the user demonstrate the skill in the robot’s environment with the robot’s own capabilities. It also provides a natural teaching interface to correct a skill reproduced by the robot. Recent advances in skin technology offer the possibility to teach robots how to exploit tactile contact on an object (Fig. 74.4 middle and [VIDEO 104]). By exploiting the compliance of the iCub robot’s fingers, the teacher can teach the robot how to adapt the posture of the fingers in response to a change in tactile sensing as measured at the robot’s finger tips [74.50]).
One main drawback of kinesthetic teaching is that the human must often use more degrees of freedom to move the robot than the number of degrees of freedom moved on the robot. This is visible in Fig. 74.4. To move the fingers of one hand of the robot, the teacher must use both hands. This limits the type of tasks that can be taught through kinesthetic teaching. Typically tasks that would require moving both hands simultaneously could not be taught this way. One could either proceed incrementally, teaching first the task for the right hand and then, while the robot replays the motion with its right hand, teach the motion of the left hand. However, this may prove to be cumbersome. The use of external trackers as reviewed above are more amenable to teaching coordinated motion between several limbs.

3. Third, there are immersive teleoperation scenarios, where a human operator is limited to using the robot’s own sensors and effectors to perform the task. Teleoperation may be done using simple joysticks or other remote control devices, including haptic devices (Fig. 74.4 bottom and VIDEO 101). The later have the advantage that they can allow the teacher to teach tasks that require precise control of forces, while joysticks would only provide kinematic information (position, speed).

Teleoperation is advantageous compared to external motion tracking systems, as this solves the correspondence problem entirely, since the system directly records the perception and action from the robot’s configuration space. It is also advantageous compared to kinesthetic training, as it allows training the robots from a distance and is, hence, particularly suited for teaching navigation and locomotion patterns. The teacher no longer needs to share the same space with the robot. Teleoperation is, usually, used to transmit the kinematics of motion. For instance, in [74.51], the acrobatic trajectories of a helicopter are learned by recording the motion of the helicopter when teleoperated by an expert pilot. In [74.52], a robot dog is taught to play soccer by a human guiding it via a joystick. However, in recent work, teleoperation has been used successfully to teach a humanoid robot balancing techniques [74.53]. Learning to react to perturbations is done through a haptic interface attached to the torso of the demonstrator, which measures the interaction forces when the human is pushed around. The kinematics of motion of the demonstrator are directly transmitted to the robot through teleoperation and are combined with haptic information to train a model of motion conditioned on perceived forces.

The disadvantage of teleoperation techniques is that the teacher often needs training to learn to use the remote control device. Teleoperation using a simple joystick allows guiding only a subset of degrees of freedom. To control for all degrees of freedom, very complex, exoskeleton type of devices must be used, which can be cumbersome. Moreover, teleoperation prevents the teacher from observing all sensorial information required to perform the task. For instance, teleoperation, even when using haptic device, poorly renders the contacts perceived at the robot’s end-effector. To palliate to this, one may provide the teacher with visualization interfaces to simulate the interaction forces.

4. Lastly, one can use explicit information, such as that conveyed by speech, to provide additional advice and comments to the demonstration [74.18, 57, 58] and VIDEO 103. Speech is a very natural means of communication among humans and, hence, is viewed as an easy way to allow the end-user to communicate with robots. However, it necessitates that vocabulary that is understandable to the robot and
grounded in the actions and perceptions of the robot be defined beforehand. While this restricts teaching to discrete state–action pairs, it is particularly useful for symbolic reasoning.

Each teaching interface has its pros and cons. It is thus interesting to investigate how these interfaces could be used in conjunction to exploit complementary information provided by each modality [74.50].

74.4 Algorithms to Learn from Humans

Current approaches to encoding skills through LfD can be broadly divided into two trends: a low-level representation of the skill, taking the form of a non-linear mapping between sensory and motor information, and, a high-level representation of the skill that decomposes the skill into a sequence of action-perception units.

While the majority of work in LfD uses solely the demonstrations for learning, a growing number of works develops methods by which LfD can be combined with other learning techniques. One group of work investigates how to combine imitation learning with reinforcement learning, a method by which the robot learns through trial and error to maximize a given reward. Other works take inspiration in the way humans teach each other and introduce interactive and bidirectional teaching scenarios whereby the robot becomes an active partner during the teaching phase. We briefly review the main principles underlying each of these areas below:

74.4.1 Learning Individual Motions

Individual motions/actions (e.g., juicing an orange, trashing it, and pouring liquid into the cup in the example shown in 74.1) could be taught separately instead of simultaneously, as shown in this previous example. The human teacher would then provide one or more examples of each submotion. If learning proceeds from the observation of a single instance of the motion/action, one calls this one-shot learning [74.60]. Examples of learning locomotion patterns can be found in [74.61]. To make sure that this is not akin to simple record and play, the controller is provided with prior knowledge in the form of primitive motion patterns. Learning then consists of instantiating the parameters modulating these motion patterns.

Teaching can also proceed in batch mode after recording several demonstrations, or incrementally by adding recursively more information trial by trial [74.12, 50, 62]. When learning in batch mode, learning considers all examples and draws inference by comparing the individual demonstrations. Inference is usually based on a statistical analysis, where the demonstration signals are modeled via a probability density function, exploiting various non-linear regression techniques stemming from machine learning. Popular methods these days include Gaussian processes, Gaussian mixture Models, and support vector machines.

Choosing properly the variables to encode a particular movement is crucial, as it already implies part of the solution to the problem of defining what is important to imitate. Work in LfD encodes human movements in either joint space, task space, or torque space [74.63–65]. The encoding may be specific to cyclic motion [74.22], discrete motion [74.21], or to a combination of both [74.61].

Fig. 74.5 Probabilistic encoding of motion in a subspace of reduced dimensionality (after [74.59] and [74.59])
Encoding often encompasses the use of dimensionality reduction techniques that project the recorded signals into a latent space of motion of reduced dimensionality. These techniques may either perform a local linear transformations [74.66–68] or exploit global non-linear methods [74.59, 69, 70] (Fig. 74.5). Additionally, task-specific rating functions [74.71] and simulation-based optimization [74.72] are investigated to identify relevant learning features.

Teaching Force-Control Tasks
While most LfD to date worked focused on learning the kinematics of motions by recording the position of the end-effector and/or the position of the robot’s joints, more recently, some works have investigated transmission of force-based signals through human demonstration [74.56, 73–76]. See |VIDEO 478| and |VIDEO 479| for examples of kinesthetic teaching of compliant motion. Transmitting information about force is difficult for humans and for robots alike. Force can be sensed only when performing the task ourselves. Current efforts, hence, seek to develop methods by which one may embody the robot. This allows human and robot to simultaneously perceive the forces applied when performing the task. A new exciting line of research, hence, leverages on recent advances in the design of haptic devices and tactile sensing, and on the development of torque and variable impedance actuated systems to teach force-control tasks through human demonstration.

74.4.2 Learning Compound Actions

Learning complex tasks, composed of a combination and juxtaposition of individual motions, is the ultimate goal of LfD. There are two major ways to proceed to learning of such complex tasks:

1. One may first learn models of all individual motions, using demonstrations of each of these actions individually. In a second stage, one may learn the right sequence and combination of these actions by observing a human performing the whole task. This approach, however, assumes that one can list all necessary individual actions, so-called primitive actions. To date, there does not exist a database of such primitive actions and one may wonder whether the variability of human motion may really be reduced to a finite list of possible motions. A common approach is to first learn models of all of the individual motions, using demonstrations of each of these actions individually [74.77, 78], and then learn the right sequencing/combination in a second stage either by observing a human performing the whole task [74.79, 80] or through reinforcement learning [74.81]. However, this approach assumes that there is a known set of all necessary primitive actions. For specific tasks this may be true, but to date there does not exist a database of general purpose primitive actions, and it is unclear whether the variability of human motion may really be reduced to a finite list.

2. The alternative is to observe the human performing the complete task and to automatically segment the task to extract the primitive actions, which may then become task-dependent, see e.g., [74.82, 83]. This has the advantage of learning, in one swipe, both the primitive actions and the way they should be combined. One issue that arises is that the number of primitive tasks is often unknown, and there could be multiple possible segmentations that must be considered [74.52].

Other examples include learning how to sequence known behaviors to enable complex navigation tasks through the imitation of a more knowledgeable robots or humans [74.9, 84, 85] and learning how to sequence primitive motions for full body motion in humanoid robots [74.25, 33, 86].

A large body of these works uses a symbolic representation of both the learning and the encoding of the task [74.6, 30, 85, 87–91]. This symbolic way of encoding skills may take several forms. One common way is to segment and encode the task according to sequences of predefined actions, described symbolically. Encoding and regenerating the sequences of these actions can, however, be done using classical machine learning techniques, such as HMM, [74.30].

Often, these actions are encoded in a hierarchical manner. In [74.85], a graph-based approach is used to generalize an object moving skill, using a wheeled mobile robot. In this model, each node in the graph represents a complete behavior and generalization takes place at the level of the topological representation of the graph. The latter is updated incrementally.

References [74.88, 89] follow a similar hierarchical and incremental approach to encode various household tasks (such as setting the table and putting dishes in a dishwasher) (Fig. 74.6 and |VIDEO 103|). There, learning consists in identifying a sequence of predefined, elementary actions, which is further combined into a hierarchical task network. By analyzing multiple demonstrations, the ordering of elementary actions is learned, resulting in a precedence graph. The precedence graph defines a partial ordering on the set of learned elementary actions, which can be exploited to execute elementary actions in parallel, extracting symbolic rules that manage the way each object must be
handled. In [74.92], the approach was extended to learning subsymbolic goal and constraint descriptions for each elementary action. In the execution phase, the robot applies motion planning to generate a motion to reach the goals while obeying the constraints. The resulting task description mimics the strategy that humans follow when performing the task. Based on the subsymbolic goal and constraint descriptions, the robot can reason to adapt the strategy to changes in object location, obstacle occurrence, and varying start configurations.

The approaches reviewed above assume a deterministic world, where actions unfold uniquely from perception of the current state of the world. However, robots operating in real environments will observe the world using imperfect sensors and the effects of their actions may be stochastic. To account for the stochasticity of the robot’s perceptions and actions, Schmidt-Rohr et al. [74.93] use a model of the task with partially observable Markov decision processes (POMDP). At runtime, an optimal (in a maximum likelihood sense) decision is then taken.

Reference [74.90] exploits also a hierarchical approach to encoding a skill in terms of pre-defined behaviors. The skill consists in moving through a maze where a wheeled robot must avoid several kinds of obstacles and reach a set of specific subgoals. The particularity of this approach lies in the use of symbolic representations of the skill, which are applied to explore the role of the teacher in guiding incremental learning of the robot.

Finally, [74.91] took a symbolic approach to encoding human motions as sets of pre-defined postures, positions, or configurations, considering different levels of granularity for the symbolic representation of the motion. This a priori knowledge is then used to explore the correspondence problem through several simulated setups, including motion in joint space of arm links and displacements of objects on a two-dimensional (2-D) plane.

The main advantage of these symbolic approaches is that high-level skills (consisting of sequences of symbolic cues) can be learned efficiently through an interactive process. However, because of the symbolic nature of their encoding, the methods rely on a large amount of prior knowledge to predefine the important cues and to segment those efficiently.

### 74.4.3 Incremental Teaching Methods

The statistical approach described previously is an interesting way to extract autonomously the important features of the task, and, thus, to avoid putting too much prior knowledge in the system. However, it requires a large number of demonstrations to draw statistically valid inference. It is not reasonable to assume that a layuser will perform many demonstrations of the same task. Hence, for LfD to be amenable to lay users, learning should require as few demonstrations as possible. Ideally, one would like the robot to be bootstrapped with some initial knowledge, so that the robot can start
Learning from Humans

Incremental learning approaches that gradually refine task knowledge as more examples become available pave the way towards LfD systems suitable for such continuous and long-life robot learning. Figure 74.7 and VIDEO 104 shows an example of such incremental teaching of a simple skill.

These incremental learning methods use various forms of deixis, as well as verbal and non-verbal interactions, to guide the robot’s attention to the important parts of the demonstration or to particular mistakes produced by the robot during the reproduction of the task. Such incremental and guided learning is often referred to as scaffolding or molding of the robot’s knowledge, and is key to teaching robots tasks of increasing complexity [74.90, 94].

Research on the use of incremental learning techniques for robot LfD has contributed to the development of methods for learning complex tasks within the household domain from as few demonstrations as possible. Moreover, it has contributed to the development and application of machine learning that allow a continuous and incremental refinement of the task model. Such systems have sometimes been referred to as background knowledge-based or EM deductive LfD-systems, as presented in [74.95, 96]. They usually require very few or even only a single user demonstration to generate executable task descriptions. The main objective of this line of research is to build a meta-representation of the knowledge that the robot has acquired on the task and to apply reasoning methods on this knowledge database (Fig. 74.6). In this scenario, reasoning involves recognizing, learning, and representing repetitive tasks.

Pardowitz et al. [74.97] discuss how different forms of knowledge can be balanced in an incremental learning system. The system relies on building task precedence graphs. Task precedence graphs encode hypotheses that the system makes on the sequential structure of a task. Learning of the task precedence graphs allows the system to schedule its operations most flexibly, while still meeting the goals of the task ([74.98] for details). Task precedence graphs are directed, acyclic graphs that contain a temporal precedence relation that can be learned incrementally. Incremental learning of task precedence graphs leads to a more general and flexible representation of the task knowledge (Fig. 74.6 and VIDEO 105).

74.4.4 Combining Learning from Humans with Other Learning Techniques

To recall, a main argument for the development of LfD methods was that they would speed up learning by providing examples of good solutions. This assumption, however, is realistic only if the context for the reproduction is sufficiently similar to that of the demonstration. We saw previously that the use of dynamical systems-based representation at the trajectory level allows the robot to depart to some extent from a learned trajectory to reach the target, even when both the object and the hand of the robot have moved from the location shown during the demonstration. There are, however, situations in which such an approach would fail, such as, for instance, when placing a large obstacle in the robot’s pathway (Fig. 74.8). Besides, robots and humans may differ significantly in their kinematics and dynamics of motion and, although there are varieties of ways to bypass the so-called correspondence problem, relearning a new model may still be required in special cases.

To allow the robot to relearn to perform a task in any new situation, it appeared important to combine LfD methods with other motor learning techniques. Reinforcement learning (RL) appeared particularly suitable for type of problem. Indeed, imitation learning is limiting in that it requires the robot to learn only from what has been demonstrated. Reinforcement learning, in contrast, allows the robot to discover new control policies through free exploration of the state-action space. Approaches that combine imitation learning and reinforcement learning aim at exploiting the strength of both algorithms to overcome their respective drawbacks. Demonstrations are used to guide the exploration
in reinforcement learning (RL). This, hence, reduces
the time it takes for RL algorithms to find an adequate
control policy, while allowing the robot to depart from
the demonstrated behavior. Figures 74.8 show two ex-
amples of techniques that use reinforcement learning
in conjunction with LfD to improve the robot’s perfor-
mance beyond that of a demonstrator.

Early work on LfD using RL started in the 1990s
with learning to swing up and control an inverse pen-
dulum [74.100] and learning industrial tasks like peg-
in-hole with a robot arm [74.26]. More recent efforts
include [74.101–103], who tackled robust control of the
upper body of humanoid robots in various manipulation
tasks, learning an archery skill [74.104], and learning
how to hit a snooker ball [74.105].

Demonstrations can be used in different ways to
bootstrap RL. They may be used as initial roll-outs
from which an initial estimate of the policy is com-
puted [74.106–108], or to generate an initial set of
primitives [74.81, 103, 107]. In the latter case, RL is
then used to learn how to select across these primitives.
Demonstrations can also be used to limit the search
space covered by RL [74.101, 109], or to estimate the
reward function [74.110, 111]. Finally, RL and imita-
tion learning can be used in conjunction at run time,
by letting the demonstrator take over part of the control
during one trial [74.112].

Another way to enable the robot to learn a control
strategy through a combination of self experimentation
and learning from watching others is to evolve popu-
lation of agents that mimic each other. Such an evo-
olutionary approach using genetic algorithms has been
investigated by a number of authors, e.g., for learn-
ing of manipulation skills [74.113], navigation strate-
gies [74.114], or sharing a common vocabulary to name
sensorperception and actions [74.115].

**Variants on Reinforcement Learning**

While most of the works that combine imitation learn-
ing with reinforcement learning assume the reward to
be known, inverse reinforcement learning (IRL) offers
a framework to determine automatically the reward and
the optimal control policy [74.116]. When using human
demonstrations to guide learning, IRL solves jointly
the what to imitate and how to imitate problems. Other
approaches to estimating the reward or cost function au-
tomatically have been proposed, see, for instance, the
maximum margin planning technique [74.117] and the
automatic extraction of constraints [74.118].

Underlying all IRL works is the assumption of
a consistent reward function. When demonstrations are
provided by multiple experts, this assumes that all ex-
erts optimize the same objectives. This is constraining
and does not exploit the variability of ways in which
humans may solve the same task. Recent IRL works
consider multiple experts and identify multiple different
reward functions [74.119, 120]. This allows the robots
to learn multiple (albeit suboptimal) ways to perform
the same task. The hope is that this multiplicity of
policies will make the controller more robust, offering
alternative ways to complete the task, when the context
no longer allows the robot to perform the task in the
optimal way.

The vast majority of work on LfD relies on suc-
cessful demonstrations of the desired task by the hu-
man. It hence assumes that all the demonstrations are
good demonstrations and discards those that are poor
proxy of what would be deemed as a good demonstra-
tion. Recent work has also investigated the possibility
that demonstrations may instead be failed attempts at
performing the task [74.121, 122]. Learning then pro-
ceeds from observing solely incorrect demonstrations
(VID 476 and VID 477). Note that demonstra-
tions are never completely incorrect. Learning from
failed demonstration then attempts to discover which
parts of the demonstrations were correct and which
were incorrect, so as to improve solely the incorrect
parts. In this context, LfD addresses the questions of
what to and what not to imitate. It offers an interesting
alternative to approaches that combine imitation learn-
ing and reinforcement learning, in that no reward needs
to be explicitly determined.

**74.4.5 Learning from Humans, a Form
of Human–Robot Interaction**

Another perspective adopted by LfD to make the trans-
fer of skill more efficient is to focus on the interaction

Fig. 74.8 Illustration of the use of
reinforcement learning in policy
parameter space to refine a skill
initially learned from demonstration
(after [74.99] and VID 105)
aspect of the transfer process. As this transfer problem is complex and involves a combination of social mechanisms, several insights from human–robot interaction (HRI) were explored to make efficient use of the teaching capabilities of the human user, [74.123–125] for surveys. Next, we briefly survey some of these works.

The development of algorithms for detecting social cues given implicitly or explicitly by the teacher during training and the integration of those as part of other generic mechanisms for LfD has become the focus of a large body of work in LfD. Such social cues can be viewed as a way to introduce priors in a statistical learning system, and, by so doing, to speed up learning. Indeed, several hints can be used to transfer a skill not only by demonstrating the task multiple times but also by highlighting the important components of the skill. This can be achieved by various means, using different modalities.

A large body of work explored the use of pointing and gazing (Fig. 74.9 left and [VIDEO 106]) as a way of conveying the intention of the user [74.79, 126–132]. Vocal deixis, using a standard speech recognition engine, has also been explored widely [74.79, 133]. In [74.88], the user makes vocal comments to highlight the steps of the teaching that are deemed as being the most important. In [74.134, 135], only the prosody of the speech pattern is looked at, rather than the exact content of the speech, as a way to infer some information on the user’s communicative intent.

In [74.136], these social cues are learned through an imitative game, whereby the user imitates the robot. This allows the robot to build a user-specific model of these social pointers, and, hence be more robust to detecting those.

Recent lines of research in interactive LfD seeks to give a more active role to the teacher in a bidirectional teaching process [74.15, 137, 138]. Robots become more active partners and can indicate which portion of the demonstration was unclear. Teachers may in turn refine the robot’s knowledge by providing complementary information where the robot is performing poorly. This supplementary information may consist of additional rounds of demonstrations of the complete task [74.139], or may be limited to subparts of the task [74.140, 141]. The information can be conveyed through specific task’s features, such as a list of waypoints [74.142]. The robot is then left free to interpolate a trajectory using these key points.

The design of such incremental teaching methods calls for machine learning techniques that enable the incorporation of new data in a robust manner. It also opens the door to the design of other human–robot interfacing systems, including the use of speech, which leads to meaningful dialogs between humans and robots. An example of such bidirectional teaching is given on the right-hand side of Fig. 74.10. The robot asks for help during or after teaching, verifying that its understanding of the task is correct [74.14]. This teaching interaction is tailored to let the user become an active participant in the learning process (and not only a model of expert behavior).

By taking inspiration from the human tutelage paradigm, [74.15] shows that a socially guided approach can improve both the human–robot interaction and the machine learning process by taking into account human benevolence. That work highlights the role of the teacher in organizing the skill into manageable steps and maintaining an accurate mental model.
of the learner’s understanding. Reference [74.138] use a similar teaching paradigm and extend the concept to the learning of continuous motion trajectories and of actions on objects, and propose experiments where a humanoid robot learns new manipulation skills by first observing a human demonstrator (through motion sensors) and then gradually refining its skill through teacher support. In this application, the user provides scaffolds to the robot for the reproduction of the skill by moving kinesthetically a subset of the motors. Through the supervision of the user who progressively disassembles the scaffolds after each reproduction attempt, the robot can finally reproduce the skill on its own. Reference [74.143] highlights the importance of an active participation of the teacher not only to demonstrate a model of expert behavior but also to refine the acquired motion through spoken feedback.

Reference [74.90] provides experiments where a wheeled robot is teleoperated through a screen interface to simulate a molding process, that is, by letting the robot experience sensory information when exploring its environment through the teacher’s support. Their model uses a memory-based approach in which the user provides labels for the different components of the task to teach hierarchically high-level behaviors.

Finally, a core idea of the HRI approach to LfD is that imitation is goal directed, that is, actions are meant to fulfill a specific purpose and to convey the intention of the actor [74.144]. While a longstanding trend in LfD approached the problem from the standpoint of trajectory following [74.84, 145, 146] and joint motion replication, [74.147–150], recent works, inspired by the above rationale, start from the assumption that imitation is not just about observing and replicating the motion, but rather about understanding the goals of a given action (see the above survey of approaches to determining automatically the reward or what to imitate).

Determining the way humans learn to both extract the goals of a set of observed actions and give these goals a hierarchy of preference is fundamental to our understanding of the underlying decisional process to imitation. While we have surveyed recent work in that area, it is important to recall other approaches to tackling these issues that have previously followed a probabilistic approach to explain the derivation and sequential application of goals and apply this to enable learning of manipulatory tasks requiring sequencing of subsets of goals [74.97, 145, 151, 152].

Understanding the goal of the task is still only half of the picture, as there may be several ways of achieving the goal of the task. Moreover, what is feasible (or optimal) for the demonstrator may not necessarily be appropriate for the imitator [74.36]. Thus, different models, modes and communication channels, should be used in conjunction to find a solution that is optimal both from the point of view of the imitator and that achieves what the demonstrator seeks to teach the robot.

This concludes our survey. As the reader can see, the issues of what and how to imitate are tightly connected and to a large extent remain only partly solved.

### 74.5 Conclusions and Open Issues in Robot LfD

Research in LfD or programming by demonstration (PbD) is progressing rapidly, pushing back limits and posing new questions all the time. As such, any list of limitations and open questions is bound to be incomplete and out of date. However, there are a few long-standing limitations and open questions that bear further attention.

Generally, work in LfD assumes a fixed, given form for the robot’s control policy, and learns appropriate parameters. To date, there are several different forms of policies in common usage, and there is no clear correct (or dominant) technique. Furthermore, it is possible that a system could be provided with multiple possible representations of controllers and select which is most appropriate.

The combination of reinforcement learning and imitation learning has been shown to be effective in addressing the acquisition of skills that require fine tuning of the robot’s dynamics. Likewise, more interactive learning techniques have proven successful in allowing for collaborative improvement of the learnt policy by switching between human-guided and robot-initiated learning. However, there do not yet exist protocols to determine when it is best to switch between the various learning modes available. The answer may, in fact, be task dependent.

In work to date, teaching is usually done by a single teacher, or teachers with an explicit concept of the task to teach. More work needs to be done to address issues related to conflicting demonstrations across teachers with different styles. Similarly, teachers are usually human beings, but could instead be an arbitrary expert agent. This agent could be a more knowledgeable robot or a computer simulation. Finally, another relatively little explored question relates to the problem of how to transfer skills across multiple agents, including multiple robots (i.e., teaching is done from a teacher robot to various learner robots). Early work in this direction was
done in the 1990s [74.115, 153, 154]. This work, how-
ever, has so far been reduced to transfer of navigation or
communication skills across swarms of simple mobile
robots.

Experiments in LfD have mostly focused on a sin-
gle task (or set of closely related tasks), and each
experiment starts with a tabula rasa. As learning of
complex tasks progresses, means to store and reuse
prior knowledge at a large scale will have to be devised.
Learning stages, akin perhaps to those found in child
development, may be required. There will need to be
a formalism to allow the robot to select information, to
reduce redundant information, select features, and store
new data efficiently.

References

74.1 T. Lozano-Pérez: Robot programming, Proceed-
ings IEEE 71(7), 821–841 (1983)
74.2 B. Dufay, J.-C. Latombe: An approach to au-
tomatic robot programming based on inductive
74.3 A. Levas, M. Selfridge: A user–friendly high–level
robot teaching system, IEEE Int. Conf. Robotics,
Atlanta (1994) pp. 413–416
74.4 A.B. Segre, G. Delong: Explanation–based manip-
ulator learning: Acquisition of planning ability
St. Louis (1985) pp. 555–560
74.5 A.M. Segre: Machine Learning of Robot Assembly
Plans (Kluwer, Boston 1988)
74.6 S. Muench, J. Kreuziger, M. Kaiser, R. Dillmann:
Robot programming by demonstration (RPD) – Us-
ing machine learning and user interaction meth-
ods for the development of easy and comfortable
robot programming systems, Proc. Int. Symp. In-
74.7 A. Billard: Imitation: A review. In: The Handbook
of Brain Theory and Neural Network, 2nd edn.,
pp. 566–569


74.36 C. Nehaniv, K. Dautenhahn: Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications.


74.68 N. Rambhatla: Local Models and Gaussian Mixture Models for Statistical Data Processing, PhD Thesis
Robotics and Humans


74.111 P. Abbeel, A. Coates, A. Ng: Autonomous helicopter aerobatics through apprenticeship learning, Int. J. Robotics Res. 29(13), 1608–1639 (2010)


74.131 C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, B. Blumberg: Learning from and about others:


