REINFORCEMENT LEARNING WITH HUMAN ADVICE. A SURVEY.

A PREPRINT

Anis Najar
Laboratoire de Neurosciences Cognitives Computationnelles (LNC2)
INSERM U960, Paris, France
anis.najar@ens.fr

Mohamed Chetouani
Institute for Intelligent Systems and Robotics,
Sorbonne Université, CNRS UMR 7222, Paris, France

May 25, 2020

ABSTRACT

In this paper, we provide an overview of the existing methods for integrating human advice into a Reinforcement Learning process. We propose a taxonomy of different types of teaching signals, and present them according to three main aspects: how they can be provided to the learning agent, how they can be integrated into the learning process, and how they can be interpreted by the agent if their meaning is not determined beforehand. Finally, we compare the benefits and limitations of using each type of teaching signals, and propose a unified view of interactive learning methods.

Teaching a machine through natural interaction is an old idea dating back to the foundations of AI, as it was already stated by Alan Turing in 1950: *"It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. That process could follow the normal teaching of a child. Things would be pointed out and named, etc."* [124]. Since then, many efforts have been made for endowing robots and artificial agents with the capacity to learn from humans in a natural and unconstrained manner [23]. However, designing human-like learning robots still raises several challenges regarding their capacity to adapt to different teaching strategies and their ability to take advantage of the variety of teaching signals that can be produced by humans [127]. The interactive learning literature references a plethora of teaching signals such as instructions [99], demonstrations [7] and feedback [55, 89]. These signals can be categorized in several ways depending on what, when, and how they are produced. For example, a common taxonomy is to divide interactive learning methods into three groups: learning from advice (telling), learning from evaluative feedback (criticizing), and learning from demonstration (showing) [55, 51, 58]. While this taxonomy is commonly used in the literature, it is not infallible as these categories can overlap.

The definition of advice in the literature is relatively vague and there is no specific constraint about what type of input can be provided to the learning agent. In [45], advice is defined as *"concept definitions, behavioral constraints, and performance heuristics"*. In [100], it refers to *"any external input to the control algorithm that could be used by the agent to take decisions about and modify the progress of its exploration or strengthen its belief in a policy"*. Advice can represent state preferences [125], action preferences [77], constraints on action values [78, 122], instructions [27, 79, 65, 102], feedback [128, 51, 21], explanations [64], and even demonstrations [68, 128]. Demonstrations are sometimes referred to as advice, whenever they are not executed by the teacher but rather communicated to the robot [68, 128]. Evaluative feedback is considered as advice in that it constitutes an information that is communicated to the robot [128, 51, 38]. In some papers, the term feedback is used as a shortcut for evaluative feedback [116, 66, 38, 61, 71]. However, the same term is sometimes used to refer to corrective feedback [6]. While these two types of feedback, evaluative and corrective, are sometimes designated by the same label, they are basically different. The lack of consensus about the terminology in the literature makes all these concepts difficult to disentangle, and represents

an obstacle towards establishing a systematic understanding of how these teaching signals relate to each other from a computational point of view.

The goal of this survey is two-fold. First, we clarify some of the terminology used in the interactive learning literature by proposing a taxonomy of the existing teaching signals. Second, we review how these signals can be represented, interpreted, and operationalized from a computational point of view. We mainly focus on advice, *i.e.* teaching signals that rely only on communication, as opposed to demonstrations which require the execution of the task by the teacher. However, we still follow the standard taxonomy "Advice - Evaluative feedback - Demonstration" in the organisation of the paper for two main reasons. First, evaluative feedback has been extensively covered in the literature, so we present it separately to emphasize its characteristics with respect to other forms of advice. Second, even though demonstrations do not fit, strictly speaking, into the definition of advice, we briefly cover them in order to highlight some common aspects between both categories. Readers who are interested in this topic can refer to the existing literature [7, 23].

Although the methods we cover belong to various mathematical frameworks, we mainly focus our analysis from a Reinforcement Learning (RL) perspective [110]. So here advice denotes any information that can be communicated by a human teacher to an RL agent in order to modify its behaviour. We equivalently use the terms of "agent", "robot" and "system", to make abstraction of the support over which the Reinforcement Learning algorithm is implemented.

Throughout this paper, we use the term "shaping" to refer to the mechanism by which teaching signals are integrated into the learning process. This concept has been mainly used within the RL literature, as a method for accelerating the learning process, by providing the learning agent with intermediate rewards [41, 105, 35, 55, 50, 18]. However, the general meaning of shaping is equivalent to training, which is to make an agent's *"behavior converge to a predefined target behavior"* [35]. More specifically, it is based on the idea that *"learning to solve complex problems can be facilitated by first learning to solve related simpler problems"* [41]. In this survey, the term shaping is employed in its general meaning as influencing a learning agent towards a desired behaviour. Technically, it qualifies the method used for integrating human teaching signals into an agent's policy.

The paper is organized as follows. In the next section, we provide an overview of the literature dealing with advice. Section 2 is dedicated to evaluative feedback. In Section 3, we briefly cover demonstrations. These three sections follow the same structure based on three main aspects: how advice can be provided to the agent, how it can be integrated into the learning process, and how it can be interpreted by the agent if their meaning is not determined beforehand. In Section 4, we compare the benefits and limitations of using different types of teaching signals, and we propose a unified view of interactive learning methods. Finally, Section 5 concludes this survey.

1 Advice

In one of the first papers of Artificial Intelligence, John McCarthy described an *"Advice Taker"* system that could learn by being told [83]. This idea was then elaborated in [44, 45], where a general framework for learning from advice was proposed. This framework can be summarized in the following five steps [28]:
1. Requesting or receiving the advice.
2. Converting the advice into an internal representation (Interpretation).
3. Converting the advice into a usable form (Operationalization).
4. Integrating the reformulated advice into the agent's knowledge base.
5. Judging the value of the advice.

The first step describes how advice can be provided to the system. Step 2 is related to interpreting advice. Steps 3, 4 and 5 describe how advice can be integrated into the learning process. In what follows, we review advice-taking systems based on these three aspects. As interpreting advice is a relatively recent research question, and most of existing methods predefine the meaning of teaching signals, we cover this aspect at the end of each section.

1.1 Providing advice

Advice can be provided in two different forms: *general* and *contextual* (Fig. 1, Table1). We define *general advice* as general information about the task, such as concept definitions, behavioral constraints, and performance heuristics, that do not depend on the context in which they are provided. They are self-sufficient in that they include all the required information for being converted into a usable form (operationalization). They can take the form of *if-then* rules that inform the agent about the optimal behaviour, either directly by specifying which actions should be performed in different situations [79, 65], or indirectly by defining task constraints [77, 78, 122]. As *general advice* is not state-dependent, it can be communicated to the system at any moment of the task, even prior to the learning process (off-line).

Contextual advice, on the other hand, is state-dependent, in that the communicated information depends on the current state of the task. So, unlike *general advice*, it must be provided interactively along the task. Typical examples include guidance [116, 107], instructions [27, 102, 98, 90] and feedback [128, 51, 21, 90].

The means by which advice is communicated by the human teacher to the system vary. The most natural and challenging way is to provide advice through natural language [65, 31, 97]. Alternative solutions include specifying *general advice* via hand-written rules [79, 78, 77, 122] and delivering *contextual advice* either via artificial interfaces such as keyboards and mouse clicks [107, 61] or through gestures [90].

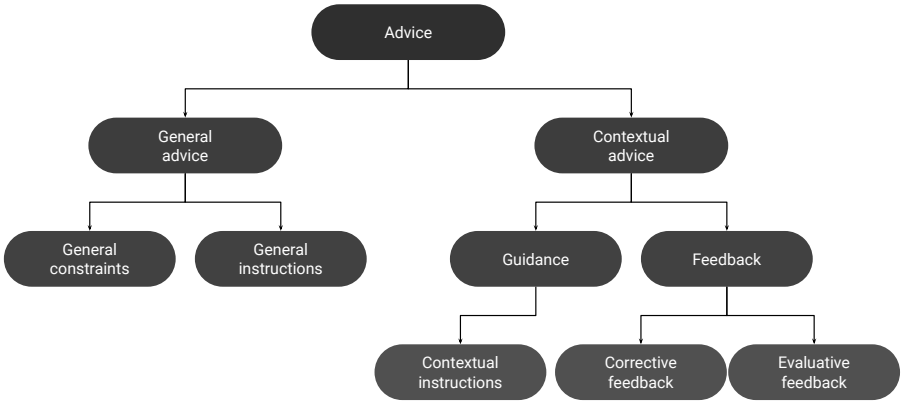

Figure 1: Taxonomy of advice.

Category	References
General constraints	[45, 80, 65, 77, 78, 122]
General instructions	[79, 65, 12, 126, 13]
Guidance	[115, 120, 107, 109, 26]
Contextual instructions	[125, 27, 95, 102, 119, 103, 114, 13, 99, 40, 74, 31, 81, 90]
Corrective feedback	[95, 25, 6, 21]
Evaluative feedback	[35, 29, 49, 52, 121, 54, 55, 51, 114, 72, 56, 57, 60, 59, 40] [38, 39, 70, 46, 71, 81, 75, 89, 90]

Table 1: Papers dealing with each type of advice teaching signals.

1.2 Learning from advice

The way advice is used for learning depends on which kind of information is communicated.

Learning from general constraints: The first ever implemented advice-taking system relied on general constraints that were written as LISP expressions, and converted into plans using a predefined set of transformations [45]. General constraints were defined as domain concepts, behavioural constraints and performance heuristics. When the executed advice lead to unexpected or unfavorable consequences, learning was triggered by correcting the advice and refining the knowledge base through predefined learning rules.

Knowledge-Based Kernel Regression (KBKR) is a method that allows incorporating advice, given in the form of *if-then* rules, into a kernel-based regression model [80]. This method was used for providing advice to an RL agent with Support Vector Regression as value function approximation [78]. In this case, advice was provided in the form of constraints on action values (e.g. *if* condition *then* $Q(s, a) \geq 1$), and incorporated into the value function through the KBKR method. This approach was extended in [77], by proposing a new way of defining constraints on action values. In the new method, pref-KBKR (preference KBKR), the constraints were expressed in terms of action preferences (e.g. *if* condition *then* prefer action a to action b). This method was also used in [122].

Learning from general instructions: General instructions inform more directly about the optimal behaviour, compared to general constraints, by explicitly specifying what to do in different situations. Like general constraints, they can be provided in the form of *if-then* rules. For example, RATLE was an advice taking RL system that built on the 5-steps advice-taking process, and used a Q-learning agent with a neural network for value function approximation

[79]. Advice was written, using an expert programming language with a predefined set of domain specific terms, in the form of *if-then* rules and *while-repeat* loops. It was then incorporated into the Q-function, by using an extension of the Knowledge-Based Neural Network method (KBANN), that allows incorporating knowledge expressed in the form of rules into a neural network [123].

In [65], a SARSA(λ) agent using linear tile-coding function approximation was augmented with an Advice Unit that computed additional action values. Advice was expressed in a specific formal language in the form of *if-then* rules. Each time a rule was activated in a given step, the value of the corresponding action in the Advice Unit was increased or decreased by a constant, depending on whether the rule advised for or against the action. These values were added to those generated by the function approximator, and presented to the learning algorithm.

Besides *if-then* rules, general instructions can also be provided in the form of detailed plan descriptions, or recipes, containing the sequence of actions that should be performed [12]. These batch instructions are themselves composed of a sequence of contextual instructions (cf. next paragraph). They can be considered as a special form of communicated demonstrations [68, 129]. However, unlike in demonstrations or *if-then* rules, state information in general instructions can be implicit. For example, they can be implicitly formulated within the expression of the action (e.g. *"Click start, point to search, and then click for files or folders."*) [12].

Learning from contextual instructions: In contrast to general instructions, a contextual instruction depends on the state in which it is provided. To use the terms of the advice-taking process, a part of the information that is required for operationalization is implicit. More specifically, the condition part of each instruction is not explicitly communicated by the teacher, but must be inferred by the learner from the current context. Consequently, contextual instructions must be progressively provided to the learning agent along the task. Contextual instructions can be either low-level or high-level [13]. Low-level instructions indicate the next action to be performed [40], while high-level instructions indicate a more extended goal without explicitly specifying the sequence of actions that should be executed [74].

A mathematical formulation of contextual instructions was proposed in [99]. Particularly, the authors distinguished two types of instructions: $\pi-$instructions and $\phi-$ instructions. $\pi-$instructions modify the agent's policy towards a specific action by setting its selection probability to 1. For example, pointing to an object makes the agent perform a predefined action on it. $\phi-$instructions, on the other hand, reduce the complexity of the current state by projecting its representation into a subspace of features. For example, pointing to an object makes the agent consider only its features and ignore all other aspects.

We identify three ways of using contextual instructions (Fig. 2, Table 2). First, the communicated action can be simply executed. For example, verbal instructions can be used for guiding a robot along the task [119, 114, 31]. In [95] and [103], a Learning-from-Demonstrations (LfD) system was augmented with verbal instructions, in order to make the robot perform some actions during the demonstrations. This way of using instructions can be referred to as guidance, which is the term used in [119] and [31]. However, the term guidance can have other meanings that we detail later.

The second way of using instructions is to integrate the information about the action within the model of the task. In [125], the authors presented a State Preference method (SP), where a teacher interactively informed a Temporal Difference (TD) agent about the next preferred state. This information could be provided by telling the agent what action to perform. State preferences were transformed into linear inequalities, which were integrated into the TD algorithm in order to accelerate the learning process. In [27], instructions were integrated into an RL algorithm by positively reinforcing the proposed action. In [102], the authors presented an Actor-Critic architecture that used instructions for both decision-making and learning. For decision-making, the robot executed a composite real-valued action that was computed as a linear combination of the *actor*'s decision and the supervisor's instruction. Then, the error between the instruction and the *actor*'s decision was used as an additional parameter to the TD error for updating the *actor*'s policy.

A third approach consists in using the provided instructions for building an instruction model besides the task model. Both models are then combined for decision-making. For example, in [99], the RL agent arbitrates between the action proposed by its Q-learning policy and the one proposed by the instruction model, based on a confidence criterion. The same arbitration criterion was used in [90], to decide between the outputs of a Task Model and an Instruction Model.

Note that in Figure 2 and Table 2, we use the term shaping to describe the way instructions are integrated into the learning process, even though this term was never employed in the cited papers. This is justified by our general definition of shaping as the modification of an agent's behaviour through the use of human teaching signals, and the similarity at the computational level of the methods used for instructions and feedback (cf. Section 2). Using the same terminology is important for unifying feedback and instructions under the same general view (cf. Section 4).

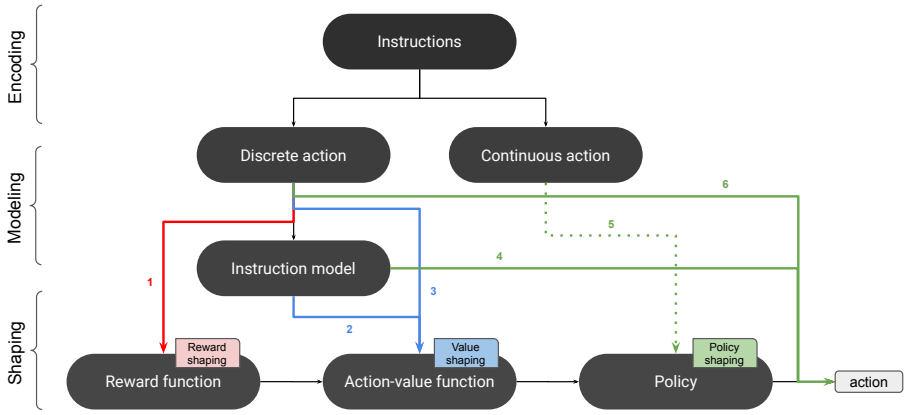

Figure 2: Learning from instructions. Two ways of encoding instructions: as discrete or continuous actions. Instructions can be used either directly for shaping or via an instruction model. Three possible shaping methods: combining evaluative feedback with the reward function, with action values, or with the policy.

Shaping method	References
Model-free reward shaping (1)	[27]
Model-based value shaping (2)	[88]
Model-free value shaping (3)	[125, 79, 65, 78, 77, 122]
Model-based policy shaping (4)	[40, 99, 90]
Model-free policy shaping (5&6)	[119, 114, 31, 95, 103, 102, 107, 116, 21]

Table 2: Papers about learning from instructions. Numbers in parentheses correspond to the arrows of Fig. 2

Learning from guidance: Guidance is a term that is encountered in many papers and has been made popular by the work of Thomaz et al. [115] about Socially Guided Machine Learning . In the broad sense, guidance represents the general idea of guiding the learning process of an agent. In this sense, all interactive learning methods such as demonstrations and feedback can be considered as a form of guidance.

A bit more specific definition of guidance is when human inputs are provided in order to bias the exploration strategy [120]. For instance, in [109], demonstrations were provided in order to teach the agent how to explore interesting regions of the state space. In [26], kinesthetic teaching was used for guiding the exploration process for learning object affordances.

In the most specific sense, guidance constitutes a form of advice that consists in suggesting a limited set of actions from all the possible ones [107, 116]. In this sense, it is a generalization of contextual instructions where the teacher proposes more than one single action at a time. For example, in [31], the authors used both terms of advice and guidance for referring to instructions. It is to be noted that in these works, the use of guidance was limited to the execution of the suggested action, so it was not directly integrated into the agent's policy.

Learning from feedback: We distinguish two main forms of feedback: evaluative and corrective. Evaluative feedback, also called critique, consists in evaluating the quality of the agent's actions. Corrective feedback, also called instructive feedback, implicitly implies that the performed action is wrong. However, it goes beyond simply criticizing the performed action, by informing the agent about the correct one. Both forms of feedback can be provided either interactively after each performed action, or a posteriori to the task execution, in a batch fashion. Examples of interactive feedback can be found in [55] for evaluative feedback, and in [21] for corrective feedback. Examples of batch feedback can be found in [51] for evaluative feedback, and in [6] for corrective feedback. While corrective feedback is used in several works, much more emphasis has been put on evaluative feedback, especially as a standalone training method. So, in this paragraph, we focus on corrective feedback; and we dedicate the next section to evaluative feedback.

Corrective feedback generally takes the form of either a contextual instruction [25] or a demonstration [95]. In the latter case, we also talk about corrective demonstrations. The only difference with instructions (resp. demonstrations) is that they are provided after an action (resp. a sequence of actions) is executed by the robot, not before. So, operationalization is made with respect to the previous state, not to the current one.

So far, corrective feedback has been mainly used for augmenting LfD systems [95, 25, 6]. For example, in [25], while the robot is reproducing the provided demonstrations, the teacher could interactively correct any incorrect action. In [95], corrective feedback took the form of a shadowed demonstration. The corrective demonstration was delimited by two predefined verbal commands that were pronounced by the teacher. In [6], the authors presented a framework based on advice-operators, allowing a teacher to correct entire segments of demonstrations through a visual interface. Advice-operators were defined as numerical operations that can be performed on state-action pairs. The teacher could choose an operator from a predefined set, and apply it to the segment to be corrected.

In [21], the authors took inspiration from advice-operators to propose learning from corrective feedback as a standalone method, contrasting with other methods for learning from evaluative feedback such as TAMER [55].

1.3 Interpreting Advice

The second step of the advice-taking process stipulates that advice needs to be converted into an internal representation. This step corresponds to interpreting the perceived advice. Predefining the meaning of contextual advice, for example by hand-coding the mapping between instructions and their corresponding actions, has been widely used in the literature [27, 95, 102, 119, 103, 25, 114, 69, 98, 31, 21]. However, this solution has many drawbacks. First, it limits the possibility for the teacher to use its own preferred signals. Second, it becomes even more inconvenient when considering general instructions, which often require expert programming skills [79, 80, 65, 77, 78, 122].

The ultimate goal of advice-taking methods is to allow robots to take advantage of human advice in a natural and unconstrained manner. However, natural language understanding still raises many challenges. To address this question, different approaches can be taken. Some methods relied on pre-trained parsers using supervised learning methods [53, 132, 82]. For example, in [65], the system was able to convert domain-specific advice expressed in a constrained natural language into a formal advice, by using a parser trained with annotated data.

More recent approaches take inspiration from the *grounded language acquisition* literature [84], to learn a model that grounds the meaning of instructions into concepts from the real world. For example, natural language instructions can be paired with demonstrations of the corresponding tasks to learn the mapping between instructions and actions [22, 113, 37].

In [74], the authors proposed a model for grounding high-level instructions into reward functions from user demonstrations. The agent had access to a set of hypotheses about possible tasks, in addition to command-to-demonstration pairings. Generative models of tasks, language, and behaviours were then inferred using Expectation Maximization (EM) [33]. The authors extended their model in [76], to ground command meanings in reward functions, using evaluative feedback instead of demonstrations. In addition to having a set of hypotheses about possible reward functions, the agent was also endowed with planning abilities that allowed it to infer a policy according to the most likely task. In a similar work [40], a generative model was used for inferring a task from unlabeled low-level contextual instructions. The robot inferred a task while learning the meaning of the interactively provided verbal instructions. As in [76], the robot had access to a set of hypotheses about possible tasks, in addition to a planning algorithm.

A different approach for interpreting instructions relies on Reinforcement Learning [12, 13, 126, 81, 90]. In [12], the authors used a policy-gradient RL algorithm with a predefined reward function, to map textual low-level instructions into actions in a GUI application. Contextual low-level instructions were provided a priori as a general instruction, detailing the step-by-step sequence of actions that must be performed. This model was extended in [13], to allow for the interpretation of high-level instructions, by including a model of the environment as in model-based RL [110]. In [126], the authors followed the same idea for interpreting navigational instructions, in a path-following task, using the SARSA algorithm with value-function approximation. The rewards were computed according to the deviation from a reference path. In [90], human instructions were interpreted by a robot and used simultaneously for learning a task. Even hough their meaning was not determined beforehand, the use of unlabeled instructions allowed the robot to learn the task faster, while reducing the number of interactions with the human teacher. The authors proposed an interpretation method allowing for sporadic instructions, as opposed to the standard RL-based interpretation methods used in [12, 13, 126, 81]. An extended comparison between different interpretation methods can be found in [86].

2 Evaluative feedback

Training a robot by evaluating its actions can be an alternative solution to the standard Reinforcement Learning approach, whenever the implementation of a proper reward function turns out to be challenging [62]. It can also be effective in situations where it is difficult for the teacher to execute demonstrations, and where instructions would require a sophisticated communication channel.

2.1 Providing evaluative feedback

In the literature, there exist different views about how to represent evaluative feedback. It can be represented as a scalar value $f \in [-1, 1]$ [55], a binary value $f \in \{-1, 1\}$ [121, 90], a positive reinforcer $f \in \{"Good!", "Bravo!"\}$ [52], or a categorical information $f \in \{Correct, Wrong\}$ [71].

Traditionally, evaluative feedback has been largely considered as a reward shaping technique that consists in providing the robot with intermediate rewards to speed-up the learning process [49, 121, 114, 81]. In these works, evaluative feedback was considered in the same way as the feedback provided by the agent's environment in RL; so intermediate rewards are homogeneous to MDP rewards.

Other works pointed out the difference that exists between immediate and delayed rewards [35, 29, 59]. Particularly, they considered evaluative feedback as an immediate information about the value of an action [47]. For example, in [35], the authors did not address temporal credit assignment, and the generated rewards constituted *"immediate reinforcements in response to the actions of the learning agent"*, which comes to consider rewards as equivalent to action values. In the TAMER framework [55], human-generated rewards were used for computing a regression model $\hat{H}$, called the "Human Reinforcement Function, to predict the amount of provided rewards $\hat{H}(s, a)$ for each state-action pair (s, a). In [59], different discount factors for $\hat{H}$ were compared, and it was shown that setting the discount factor to zero was better suited, which came to consider $\hat{H}$ more as an action value function than as a reward function [1]. This *myopic discounting* strategy was also employed in [89], where the head nods and shakes of a human teacher were converted into binary values and used for updating a robot's action values.

In a different approach, evaluative feedback is not converted into a numerical value, but treated as a categorical piece of information that is directly used for deriving a policy, within a Bayesian framework [72, 38, 71]. This approach is similar to the latter in that evaluative feedback only informs about the last performed action [46].

2.2 Learning from evaluative feedback

There exist different methods for deriving a policy from evaluative feedback that mostly depend on the adopted representation of the feedback (e.g., numerical values vs. categorical information). In the literature, we find different terminologies for qualifying the policy derivation methods, such as reward shaping [114], interactive shaping [55] and policy shaping [38, 18]. In some works, the term shaping is not even adopted [71]. In this survey we consider the term shaping in its general meaning as influencing a learning system towards a desired behaviour. In this sense, all methods deriving a policy from evaluative feedback are considered as shaping methods. Here we propose a categorization of policy derivation methods under the scope of shaping.

To do so, we need to distinguish two cases. In the first case, evaluative feedback is combined with a model of the task that is derived from another source of information, such as a reward function. According to how it is represented, it can be used for biasing either the reward function [121], the value function [60], or the policy [38]. So, here the term shaping can be used for qualifying the combination method. In the second case, evaluative feedback constitutes the only available source of information [70]. However, we can consider this as a special case of the former situation in which the other source (e.g. reward function) provides no information. For example, we can consider a null reward function, a null value function, or a uniform policy. So, even when evaluative feedback is not combined with another source of information, we can still talk about a shaping method.

Thus, we can divide the literature treating about evaluative feedback into three groups: reward shaping, value shaping and policy shaping methods. Figure 3 and Table 3 summarize the different possibilities for shaping with evaluative feedback, depending on the adopted representation.

Reward shaping: After converting evaluative feedback into a numerical value, it can be considered as a delayed reward, just like MDP rewards, and used for computing a value function through temporal credit assignment [49, 121, 114, 81]. This means that the effect of the provided feedback extends beyond the last performed action. When the robot has also access to a predefined reward function R, a new reward function R' is computed by summing both forms of reward: $R' = R + R^h$, where R^h is the human delivered reward.

[1] The authors proposed another mechanism for handling temporal credit assignment, in order to alleviate the effect of highly dynamical tasks [55]. In their system, human-generated rewards were distributed backward to previously performed actions within a fixed time window.

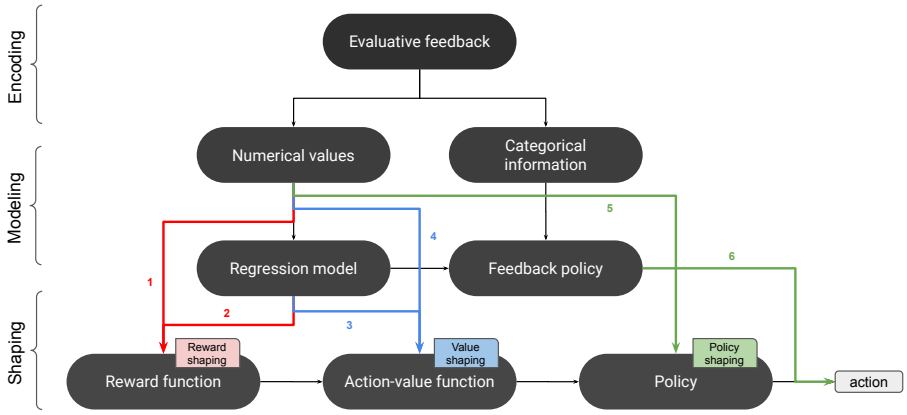

Figure 3: Shaping with evaluative feedback. Two ways of encoding evaluative feedback: as numerical values or as categorical information. Feedback can be used either directly for shaping or via a feedback model. Three possible shaping methods: combining evaluative feedback with the reward function, with action values, or with the policy.

Shaping method	References
Model-free reward shaping (1)	[49, 121, 114, 81]
Model-based reward shaping (2)	[56, 57, 60]
Model-based value shaping (3)	[56, 57, 60]
Model-free value shaping (4)	[35, 29, 89]
Model-free policy shaping (5)	[46, 75, 90]
Model-based policy shaping (6)	[56, 57, 60, 72, 38, 71]

Table 3: Papers about learning from evaluative feedback. Numbers in parentheses correspond to the arrows of Fig. 3

Knox and Stone [56, 57, 60] proposed eight different shaping methods for combining the human reinforcement function $\hat{H}$ with a predefined MDP reward function R. One of them, Reward Shaping, generalizes the reward shaping method by introducing a decaying weight factor β that controls the contribution of $\hat{H}$ over R:

$$R'(s,a) = R(s,a) + \beta * \hat{H}(s,a). \tag{1}$$

Although reward shaping has been effective in many domains [49, 121, 114, 81], this way of providing intermediate rewards does not fit into the definition of potential-based reward shaping [92, 131]. Consequently, it has been shown that it can cause sub-optimal behaviours such as positive circuits [59, 46].

Value shaping: Value shaping consists in considering evaluative feedback as an action-preference function. The numerical representation of evaluative feedback is used for modifying the Q-function rather than the reward function. Q-Augmentation [56, 57, 60] uses the human reinforcement function $\hat{H}$ for augmenting the MDP Q-function using:

$$Q'(s,a) = Q(s,a) + \beta * \hat{H}(s,a). \tag{2}$$

When comparing different shaping methods, Knox and Stone observed that *"the more a technique directly affects action selection, the better it does, and the more it affects the update to the Q function for each transition experience, the worse it does"* [60]. In fact, this can be explained by the specificity of the Q-function with respect to other preference functions. Unlike others preference function (e.g. Advantage function [42]), a Q-function also informs about the proximity to the goal. Evaluative feedback, however, informs about local preferences without necessarily including such information [46]. So, augmenting a Q-function with evaluative feedback may lead to convergence problems.

Policy shaping: In policy shaping, evaluative feedback is used for biasing the MDP policy, without interfering with the value function. Action Biasing [56, 57, 60] uses the same equation as Q-Augmentation but only in decision-making, so that the agent's Q-function is not modified:

$$a^* = argmax_a[Q(s,a) + \beta * \hat{H}(s,a)]. \tag{3}$$

Control Sharing [56, 57, 60] arbitrates between the decisions of both evaluation sources based on a probability criterion. A parameter β is used as a threshold for determining the probability of selecting the decision according to $\hat{H}$:

$$Pr(a = argmax_a[\hat{H}(s,a)]) = min(\beta, 1). \tag{4}$$

Otherwise, the decision is made according to the MDP policy.

Other policy shaping methods do not convert evaluative feedback into a scalar but into a categorical information [71]. The distribution of provided feedback is used within a Bayesian framework in order to derive a policy. Griffith et al. [38] proposed a policy shaping method that outperformed Action Biasing, Control Sharing and Reward Shaping. After inferring the teacher's policy from the feedback distribution, it computed the Bayes optimal combination with the MDP policy by multiplying both probability distributions.

It should be noted that in the aforementioned policy shaping methods, evaluative feedback was used only for biasing the MDP policy at decision-time, while reward and value shaping methods modified the task model. More recent policy shaping methods take a hybrid approach, where policy shaping is performed by modifying the agent's policy. In [75] and [90], evaluative feedback was used for updating the actor of an Actor-Critic architecture, without interfering with the value function. In [75] the update term was scaled by the gradient of the policy; whereas in [90] the authors did not consider a multiplying factor for evaluative feedback.

Overall, policy shaping methods show better performance compared to other shaping methods [60, 38, 46]. In addition to performance, another advantage of policy shaping is that it is applicable to a wider range of methods that directly derive a policy, without computing a value function or even using rewards.

2.3 Interpreting evaluative feedback

As with instructions, some works proposed to interpret the meaning of evaluative feedback signals, in order to give more possibilities to the teacher for employing her own preferred signals.

In [52], a robot had the capacity to learn new stimuli as secondary reinforcers, by associating them to primary reinforcers through the *clicker training* method. In [54], a binary classification of prosodic features was performed offline, before using it as a reward signal for task learning. In [72], a predefined set of known feedback signals, both evaluative and corrective, were used for interpreting additional signals within an Inverse Reinforcement Learning framework [93]. In [40], a robot learned to interpret evaluative feedback, while inferring the task using an EM algorithm. The robot knew the set of possible tasks, and was endowed with a planning algorithm allowing it to derive a policy for each possible task. This model was used for interpreting EEG-based evaluative feedback signals [39]. Finally, some papers addressed the question of interpreting the teacher's silence, which was referred to as implicit feedback [71].

3 Demonstration

Learning from Demonstration (LfD), also known as Programming by Demonstration (PbD) or Imitation Learning, has appeared in the 80's as a method for programming industrial robots [73]. It has, since then, given rise to numerous works in robotics aiming at developing intuitive teaching methods for non-expert users [11, 7, 23]. The main idea of LfD is to teach by example. The human teacher provides a set of examples of task executions, from which the robot must infer a model of the task. Each example is encoded as a sequence of state-action pairs.

3.1 Providing demonstrations

There exist different ways for a human teacher to demonstrate a task to a robot. The most natural way is to execute the task by herself, while the robot is observing. Demonstrations can be observed either through external sensors like a camera [8], or by attaching sensors to the teacher's body [17]. This mode of demonstration, called imitation setting, is challenging as it requires to map the perceived examples from the teacher into an internal representation that is directly exploitable by the robot. This issue is commonly referred to as the correspondence problem [91].

Practical solutions to circumvent this problem exist. Early works about LfD in industrial settings used teleoperation, where the demonstrator executed the task by directly controlling the robot's actuators [73]. This way, states and actions that need to be reproduced were directly experienced by the robot. More recent works use sophisticated teleoperation devices, including joysticks [1], data gloves [34] and teleoperation suits [67], along with other methods for controlling the robot's joints, such as kinesthetic teaching [4]. Shadowing provides an alternative approach to teleoperation that consists in controlling the robot indirectly through preprogrammed behaviours. For example, the robot can be endowed with a following behaviour, so it can be guided by the human while directly experiencing world state transitions [43, 94, 103].

3.2 Learning from demonstrations

To be able to execute the task, the robot needs to build a model that generalizes over all provided task executions. The main challenge is to exactly identify what should be kept from the provided examples. This question involves many aspects, such as feature selection and dealing with sub-optimal demonstrations [85].

Another important aspect is to identify sub-goals or important keyframes from the demonstrated trajectories [4]. For instance, we can distinguish two types of imitation: mimicry and goal emulation. Mimicry consists in replicating the trajectories of the demonstrator. Goal emulation, on the other hand, consists in reproducing the effects of the demonstrations by one's own means. The difference between the two imitation modes has been defined in terms of granularity of the task [91]. Different degrees of granularity can be defined, in order to take into account intermediate effects or sub-goals.

Mimicry has been used since early works in industrial contexts, where the goal was to *"play-back"* recorded trajectories. Earliest methods were based on symbolic reasoning, where each demonstration was discretized then transformed into a first order logic representation [36]. More recent approaches rely on supervised learning methods that learn a mapping from states to actions [104, 25, 48].

Goal emulation techniques infer the teacher's intention from the provided demonstrations. For example, Inverse Reinforcement Learning (IRL) consists in inferring the goal of the demonstrated task in the form of a reward function [93]. The inferred reward function then enables the robot to derive a policy, through planning [2, 1].

3.3 Interpreting demonstrations

In one respect, we can consider goal emulation techniques, such as IRL, as interpretation methods in that they infer the teacher's intention from the observed behaviour. However, for these methods to work, they need appropriate state-action labels, expressed within the robot's own referential. In other words, they need to overcome the correspondence problem. So, there is another level of interpretation of demonstrations that concerns the understanding of the observed states and actions, i.e. the mapping between the teacher's states and actions and the robot's states and actions [7].

The resolution of the correspondence problem is still an open research question. However, the neuroscience literature has provided us with many insights about the mechanisms involved in recognizing other's actions [101]. For instance, it has been established that imitation is triggered by automatic activation of existing motor representations [14].

These ideas inspired some solutions for the correspondence problem in robotics. For example, Alissandrakis et al. [5] proposed a generic framework, ALICE, that builds a correspondence library, by comparing observations to generated behaviours through a predefined similarity measure. In a similar approach, Demiris and Hayes [32] used a combination of inverse and forward models for both control and action recognition. Notably, these solutions reflect the importance of subjective experience in the interpretation process, that is the comparison between one's own behaviour and the observed one.

4 Discussion

In this section, we first compare the benefits and the limitations of the mentioned interactive learning methods. Then, we highlight their similarities.

4.1 Comparing different learning methods

When designing an interactive learning method, one may ask which one is better suited [108]. The methods we presented so far rely on a wide variety of teaching signals and interaction modalities with the learning system.

In this survey, we categorized interactive learning methods according to the characteristics of teaching signals. These signals differ in how they are represented and in how they are integrated into the learning process. Particularly, each teaching signal requires a different level of involvement from the human teacher and provides a different level of control over the learning process. Some of them provide poor information about the policy, so the learning process relies mostly on autonomous exploration. Others are more informative regarding the policy, so the learning process mainly depends on the human teacher.

This aspect has been described by Breazeal and Thomaz as the guidance-exploration spectrum [15]. In Section 1, we presented guidance as a special type of advice. So, in order to avoid confusion about the term guidance, we will use the term exploration-control spectrum instead of guidance-exploration (Fig. 4).

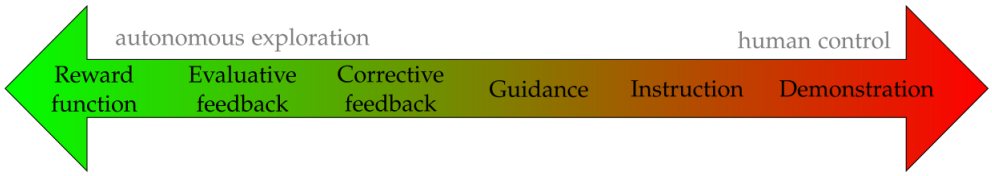

Figure 4: Exploration-control spectrum. As we move to the right, teaching signals inform more directly about the optimal policy and provide more control to the human over the learning process.

Autonomous learning: At one end of the exploration-control spectrum, we have autonomous learning methods that consist in endowing the learning agent with the capacity to evaluate its behaviour through predefined performance criteria. This can be done by implementing, for example, a reward function or a fitness function. These functions constitute evaluation sources that allow the agent to optimize its behaviour by trial-and-error, relying only on autonomous exploration, without requiring the help of a supervisor. However, a common issue in autonomous exploration is to find a suitable trade-off between exploration and exploitation. In fact, at no point of the learning process does the agent know whether its behaviour is the optimal one, or if it can still improve it. Consequently, it faces the dilemma of keeping the behaviour that it has already acquired, or exploring new ones. Systematic exploitation may lead the agent to sub-optimal behaviours, while exploration may be problematic in real-world applications.

Evaluative feedback: Evaluative feedback constitutes another evaluation criterion that has many advantages over reward and fitness functions. First, like all interactive learning methods, it alleviates the limitations of autonomous learning, i.e. slow convergence and unsafe exploration. Whether it is represented as categorical information [38] or as immediate rewards [35], it provides a more straightforward evaluation of the policy, as it directly informs about the optimality of the performed action [46].

Second, from an engineering point of view, evaluative feedback is generally easier to implement than a reward or a fitness function. For instance it is often hard, especially in complex environments, to design *a priori* an evaluation function that could anticipate all aspects of a task, and to take into account several criteria at once, such as risk and performance [62]. By contrast, evaluative feedback generally takes the form of binary values that can be easily implemented [61].

However, the informativeness of evaluative feedback is still limited, as it is only a reaction to the agent's actions and does not communicate the optimal one. So, the agent still needs to explore different actions, with trial-and-error, as in the autonomous learning setting. The main difference is that exploration is not required any more once the robot tries the optimal action and gets a positive feedback. So, the trade-off between exploration and exploitation is less tricky to address than in autonomous learning.

The limitation in the informativeness of evaluative feedback can lead to poor performance. In fact, when it is the only available communicative channel, people tend to use it also as a form of guidance, in order to inform the agent about future actions [121]. This violates the assumption about how evaluative feedback should be used, which affects learning performance. Performance significantly improves when teachers are provided with an additional communicative channel for guidance [116]. This reflects the limitations of evaluative feedback and demonstrates that human teachers also need to provide guidance.

Corrective feedback: One possibility for improving the feedback channel is to allow for corrections and refinements [118]. Corrective feedback improves the informativeness of evaluative feedback, by allowing the teacher to inform

the robot about the optimal action [21]. But being also reactive to the robot's actions, it still requires exploration. However, it prevents from waiting until the robot tries the correct action by its own.

On the other hand, corrective feedback requires more engineering efforts than evaluative feedback, as it is generally more than a binary information. As it operates over the action space, it requires to encode the mapping between feedback signals and their corresponding actions. In this aspect, it is homogeneous to contextual instructions as both operate on the same space.

An even more informative form of corrective feedback is provided by corrective demonstrations, which extend beyond correcting one single action to correcting a whole sequence of actions [25]. Corrective demonstrations operate on the same space as demonstrations, which require more engineering than contextual instructions and also provide more control over the learning process.

Guidance: The experiments of Thomaz and Breazeal have shown that human teachers want to provide guidance [116]. In contrast to feedback, guidance allows the agent to be informed about future aspects of the task, such as the next action to perform (instruction) [31], an interesting region to explore (demonstration) [109] or a set of interesting actions to try [116].

However, the control over the learning process is exerted indirectly. By performing the communicated guidance, the robot does not directly integrate this information as being the optimal behaviour. Instead, it will be able to learn only through the experienced effects, for example by receiving a reward.

Instructions: With respect to guidance, instructions inform more directly about the optimal policy, in two main aspects. First, instructions are a special case of guidance where the teacher communicates only the optimal action. Second, the information about the optimal action can be integrated into the learning process more directly than with pure guidance. The difference can be better explained in terms of operationalization. When the teacher tells the robot to perform an action a in state s as an instruction, learning can be done by integrating into the policy the information that *"in state s the optimal action is a"*. However, with guidance, operationalisation still requires the evaluation of the performed action. So, guidance is only about limiting exploration, without providing full control over the learning process.

In Section 1, we presented two ways for providing instructions: providing general instructions in the form of *if-then* rules, or interactively providing contextual (low-level or high-level) instructions as the agent progresses in the task. The advantage of general instructions is that they do not depend on the dynamics of the task, so they can be provided at any time. This puts less interactive load on the teacher in that he/she is not required to stay concentrated in order to provide the correct information at the right moment. However, they present some drawbacks. First, they can be difficult to formulate. The teacher needs to gain insight about the task and the environment dynamics, in order to take into account different situations in advance and to formulate relevant rules [65]. Furthermore, they require to know about the robot's sensors and effectors in order to correctly express conditions and actions. So, formulating rules requires expertise about the task, the environment and the robot. Second, general instructions are difficult to communicate. They require either expert programming skills or sophisticated natural language communication.

Contextual instructions, on the other hand, communicate a less sophisticated message at a time, which makes them easier to formulate and to provide. They only inform about next the action to perform, without expressing the condition, which can be inferred by the agent from the current task state. However, this makes them more prone to ambiguity. For instance, writing general instructions by hand allows the teacher to specify the features that are relevant to the application of each rule, i.e., to control generalization. With contextual instructions, however, generalization has to be inferred by the agent from the context.

Finally, interactively providing instructions makes it easy for the teacher to adapt to changes in the environment's dynamics. However, this can be difficult to do in highly dynamical tasks, as the teacher needs a lapse of time to communicate each instruction.

Demonstrations: Demonstrations are on the control end of the spectrum. They provide more control to the teacher over the learning process. In contrast with instructions, they inform about more than one single action, by communicating sequences of state-action pairs. However, providing such control requires to overcome the correspondence problem. This is generally addressed through teleoperation or kinesthetic teaching. These solutions overcome the correspondence problem in two ways. First, the state mapping is avoided as the robot experiences its own states. Second, the mapping of actions is made by controlling the robot joints, either through an interface, or by exerting forces on the robot's body. This can be seen as sending a continuous stream of instructions: the commands sent via the joystick or

the forces exerted on the robot's kinesthetic device. So, we can consider demonstrations as a sequence of contextual instructions[2].

However, there still exists some difference between demonstrations and instructions. First, teleoperated or kinesthetic demonstrations provide more control, not only over the learning process, but also over task execution. When providing demonstrations, the teacher controls the robot joints, so the communicated instruction streams are systematically executed. With instructions, however, the robot is in control of it's own actions. The teacher only communicates the action to perform, and the robot can decide whether to execute it or not.

Second, demonstrations involve more human load than instructions. Demonstrations require from the teacher to be active in executing the task, while instructions involve only communication. This aspect confers some advantages to instructions in that they offer more possibilities in terms of interaction. Instructions can be provided with different modalities such as speech or gesture, and by using a wider variety of words or signals. Demonstrations, however, are constrained by the control interface. Moreover, demonstrations require continuous focus in providing complete trajectories, while instructions can be sporadic.

Therefore, instructions can be better suited in situations where demonstrations can be difficult to provide. For example, people with limited autonomy may be unable to demonstrate a task by themselves, or to control a robot's joints. In these situations, communication is more convenient.

On the other hand, demonstrations are more adapted for highly dynamical tasks and continuous environments, since instructions require some time to be communicated.

4.2 Toward a unified view

Overall, all interactive learning methods overcome the limitations of autonomous learning, by providing more control over the agent's policy. However, more control means more interaction load. So, the autonomy of the learning process is important for minimizing the burden on the human teacher. A central question in the interactive learning literature is how to combine different learning modalities in order take advantage of each one of them. So, often, different methods are combined within a single framework.

For example, RL can be augmented with evaluative feedback [51, 106, 60], corrective feedback [20], instructions [79, 65, 102, 99], instructions and evaluative feedback [90], demonstrations [112, 109], demonstrations and evaluative feedback [66], or demonstrations, evaluative feedback and instructions [114]. Demonstrations can be augmented with corrective feedback [25, 6], instructions [103], instructions and feedback, both evaluative and corrective [95], or with prior Reinforcement Learning [111].

Integrating different teaching signals into one single and unified formalism remains an active research question. In this survey, we extracted several aspects that were shared across different approaches. We can see that the same overall process applies regardless of which specific teaching signals are in use (Fig. 5). For instance, whether we deal with contextual instructions or evaluative feedback, we need to go through the same overall process and ask the same questions about the computational implementation of these teaching signals. First, we need to think about how these signals must be encoded and whether or not their meaning will be hand-coded or interpreted by the learning agent. Second, we need to decide whether we should aggregate teaching signals into a "Teacher Model", or directly use them for influencing the learning process (model-based vs. model-free interactive learning). Finally, we need to choose a specific computational mechanism through which teaching signals (or their aggregated model) should influence the learning process, as various shaping strategies can be considered: reward shaping, value shaping or policy shaping.

From this perspective, all shaping methods that were specifically designed for evaluative feedback could be used for instructions, and *vice-versa*. For example, all the methods proposed by Knox and Stone for learning from evaluative feedback [56, 57, 60], can be recycled for learning from instructions. Similarly, the confidence criterion used in [99] for learning from instructions can serve as another Control Sharing mechanism, similar to the one proposed in [56, 57, 60] for learning from evaluative feedback. Finally, we also note that the policy shaping method proposed in [38] is mathematically equivalent to the Boltzmann Multiplication reported in [130] as an ensemble method for combining multiple policies. Although ensemble methods have not been proposed for this purpose, they could also be used for policy shaping with both feedback and instructions [86].

Common aspects between advice and demonstrations: Until recently, advice and demonstrations have been mainly considered as two complementary but distinct approaches, i.e., communication vs. action. However, these two approaches share many common aspects that are illustrated by the 5-steps advice-taking process [44, 45]. The first step, requesting or receiving the advice deals with transparency and active learning issues that are common to

[2]This does not hold for the imitation setting, where the mapping between observed and experienced states is not given.

both advice and demonstration settings [125, 117, 24, 16]. The second step, converting the advice into an internal representation also applies for demonstrations. In this case, it refers to the correspondence problem. With advice, we also have a correspondence problem that consists in interpreting the teaching signals, whether feedback or instructions. So, we can consider a more general correspondence problem, that is not proper to learning from demonstrations, and that consists in interpreting the perceived teaching signals, independently from their nature. The third, fourth and fifth steps, operationalization, integration and refinement, correspond to policy derivation. These three steps can sometimes be confounded, and we can regroup them into one more general step called shaping, i.e., using the interpreted teaching signal for biasing the behaviour, whether the teaching signal is aggregated into a separate model or directly integrated into the agent's policy. These steps are also common for both advice and demonstrations.

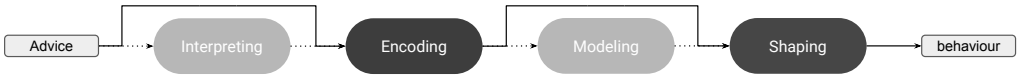

Figure 5: Shaping with advice, a unified view. When advice is provided to the learning agent, it has first to be encoded into an appropriate representation. If the mapping between teaching signals and their corresponding internal representation is not predetermined, then advice has to be interpreted by the agent. Then advice can be integrated into the learning process (shaping), either in a model-free or a model-based fashion. Optional steps, interpretation and modeling, are sketched in light grey.

General correspondence problem: So far, the correspondence problem has been mainly addressed within the community of learning by imitation. Imitation is a special type of social learning in which the robot reproduces what it perceives. So, there is an assumption about the fact that what is seen has to be reproduced. Advice is different from imitation in that the robot has to reproduce what is communicated by the advice and not what is perceived. For instance, saying "turn left", requires from the robot to perform the action of turning left, not to reproduce the sentence "turn left".

However, evidence from neuroscience gave rise to a new understanding of the emergence of human language as a sophistication of imitation throughout evolution [3]. In this view, language is grounded in action, just like imitation [30]. For example, there is evidence that the mirror neurons of monkeys also fire to the sounds of certain actions, such as the tearing of paper or the cracking of nuts [63], and that spoken phrases about movements of the foot and the hand activate the corresponding mirror-neuron regions of the pre-motor cortex in humans [9].

So, one challenging question it whether we could unify the problem of interpreting any kind of teaching signal under the scope of one general correspondence problem. This is a relatively new research question, and few attempts have been made in this direction. For example, Cederborg and Oudeyer [19] proposed a unified theoretical framework for learning from different sources of information. Their main idea is to relax the assumptions about the meaning of teaching signals, by taking advantage of the coherence between the different information sources.

Finally, when comparing demonstrations with instructions, we mentioned that demonstrations could be considered as a way of providing continuous streams of instructions, with the subtle difference that demonstrations are systematically executed by the robot. Considering this analogy, the growing literature about interpreting instructions [13, 126, 40, 90] could provide insights for designing new ways of solving the correspondence problem in imitation.

5 Conclusion

In this paper, we provided an overview of the existing methods for integrating advice into a Reinforcement Learning process. We proposed a taxonomy of different types of teaching signals, and described them according to three main aspects: how they can be provided to the learning agent, how they can be integrated into the learning process, and how they can be interpreted by the agent if their meaning is not determined beforehand. Finally, we compared the benefits and limitations of using each type of teaching signals and proposed a unified view of interactive learning methods.

The computational questions covered in this survey extend beyond the boundaries of Artificial Intelligence, as similar research questions regarding the computational implementation of social learning strategies are also raised in the field of Cognitive Neuroscience [10, 87, 96]. Thus we think this survey can be of interest for both communities.

Acknowledgments

This work was supported by the Romeo2 project.

References

[1] P. Abbeel, A. Coates, and A. Y. Ng. Autonomous Helicopter Aerobatics Through Apprenticeship Learning. *Int. J. Rob. Res.*, 29(13):1608–1639, Nov. 2010.

[2] P. Abbeel and A. Y. Ng. Apprenticeship Learning via Inverse Reinforcement Learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 1–, New York, NY, USA, 2004. ACM.

[3] I. Adornetti and F. Ferretti. The pragmatic foundations of communication: An action-oriented model of the origin of language. *Theoria et Historia Scientiarum*, 11(0):63–80, 2015.

[4] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz. Trajectories and Keyframes for Kinesthetic Teaching: A Human-robot Interaction Perspective. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '12, pages 391–398, New York, NY, USA, 2012. ACM.

[5] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn. Solving the Correspondence Problem between Dissimilarly Embodied Robotic Arms Using the ALICE Imitation Mechanism. In *In Proceedings of the second international symposium on imitation in animals & artifacts*, pages 79–92, 2003.

[6] B. D. Argall, B. Browning, and M. M. Veloso. Teacher Feedback to Scaffold and Refine Demonstrated Motion Primitives on a Mobile Robot. *Robot. Auton. Syst.*, 59(3-4):243–255, Mar. 2011.

[7] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A Survey of Robot Learning from Demonstration. *Robot. Auton. Syst.*, 57(5):469–483, May 2009.

[8] C. G. Atkeson and S. Schaal. Learning tasks from a single demonstration. In *Proceedings of International Conference on Robotics and Automation*, volume 2, pages 1706–1712 vol.2, Apr. 1997.

[9] L. Aziz-Zadeh, S. M. Wilson, G. Rizzolatti, and M. Iacoboni. Congruent Embodied Representations for Visually Presented Actions and Linguistic Phrases Describing Actions. *Current Biology*, 16(18):1818 – 1823, 2006.

[10] G. Biele, J. Rieskamp, L. K. Krugel, and H. R. Heekeren. The neural basis of following advice. *PLoS biology*, 9(6):e1001089, 2011.

[11] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot Programming by Demonstration. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1371–1394. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. DOI: 10.1007/978-3-540-30301-5_60.

[12] S. R. K. Branavan, H. Chen, L. S. Zettlemoyer, and R. Barzilay. Reinforcement Learning for Mapping Instructions to Actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 82–90, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[13] S. R. K. Branavan, L. S. Zettlemoyer, and R. Barzilay. Reading Between the Lines: Learning to Map High-level Instructions to Commands. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1268–1277, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[14] M. Brass and C. Heyes. Imitation: is cognitive neuroscience solving the correspondence problem? *Trends in Cognitive Sciences*, 9(10):489 – 495, 2005.

[15] C. Breazeal and A. L. Thomaz. Learning from human teachers with Socially Guided Exploration. In *2008 IEEE International Conference on Robotics and Automation*, pages 3539–3544, May 2008.

[16] J. Broekens and M. Chetouani. Towards transparent robot learning through tdrl-based emotional expressions. *IEEE Transactions on Affective Computing*, pages 1–1, 2019.

[17] S. Calinon and A. Billard. Incremental Learning of Gestures by Imitation in a Humanoid Robot. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, HRI '07, pages 255–262, New York, NY, USA, 2007. ACM.

[18] T. Cederborg, I. Grover, C. L. Isbell, and A. L. Thomaz. Policy Shaping with Human Teachers. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3366–3372, Buenos Aires, Argentina, 2015. AAAI Press.

[19] T. Cederborg and P.-Y. Oudeyer. A social learning formalism for learners trying to figure out what a teacher wants them to do. *Paladyn Journal of Behavioral Robotics*, 5:64–99, Oct. 2014.

[20] C. Celemin, G. Maeda, J. R. del Solar, J. Peters, and J. Kober. Reinforcement learning of motor skills using policy search and human corrective advice. *The International Journal of Robotics Research*, 38(14):1560–1580, 2019.

[21] C. Celemin and J. Ruiz-Del-Solar. An interactive framework for learning continuous actions policies based on corrective feedback. *J. Intell. Robotics Syst.*, 95(1):7797, July 2019.

[22] D. L. Chen and R. J. Mooney. Learning to Interpret Natural Language Navigation Instructions from Observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 859–865, San Francisco, California, 2011. AAAI Press.

[23] S. Chernova and A. L. Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

[24] S. Chernova and M. Veloso. Confidence-based Policy Learning from Demonstration Using Gaussian Mixture Models. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, pages 233:1–233:8, New York, NY, USA, 2007. ACM.

[25] S. Chernova and M. Veloso. Interactive Policy Learning Through Confidence-based Autonomy. *J. Artif. Int. Res.*, 34(1):1–25, Jan. 2009.

[26] V. Chu, T. Fitzgerald, and A. L. Thomaz. Learning Object Affordances by Leveraging the Combination of Human-Guidance and Self-Exploration. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, HRI '16, pages 221–228, Piscataway, NJ, USA, 2016. IEEE Press.

[27] J. A. Clouse and P. E. Utgoff. A Teaching Method for Reinforcement Learning. In *Proceedings of the Ninth International Workshop on Machine Learning*, ML '92, pages 92–110, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[28] P. Cohen and E. A. Feigenbaum. *The Handbook of Artificial Intelligence Vol. 3*. William Kaufmann & HeurisTech Press, 1982.

[29] M. Colombetti, M. Dorigo, and G. Borghi. Behavior analysis and training-a methodology for behavior engineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(3):365–380, June 1996.

[30] M. C. Corballis. Mirror neurons and the evolution of language. *Brain and Language*, 112(1):25 – 35, 2010.

[31] F. Cruz, J. Twiefel, S. Magg, C. Weber, and S. Wermter. Interactive reinforcement learning through speech guidance in a domestic scenario. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2015.

[32] J. Demiris and G. M. Hayes. Imitation As a Dual-route Process Featuring Predictive and Learning Components: A Biologically Plausible Computational Model. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 327–361. MIT Press, Cambridge, MA, USA, 2002.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[34] R. Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(23):109 – 116, 2004.

[35] M. Dorigo and M. Colombetti. Robot shaping: developing autonomous agents through learning. *Artificial Intelligence*, 71(2):321 – 370, 1994.

[36] B. Dufay and J.-C. Latombe. An Approach to Automatic Robot Programming Based on Inductive Learning. *The International Journal of Robotics Research*, 3(4):3–20, 1984.

[37] F. Duvallet, T. Kollar, and A. Stentz. Imitation learning for natural language direction following through unknown environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1047–1053, May 2013.

[38] S. Griffith, K. Subramanian, J. Scholz, C. L. Isbell, and A. Thomaz. Policy Shaping: Integrating Human Feedback with Reinforcement Learning. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 2625–2633, USA, 2013. Curran Associates Inc.

[39] J. Grizou, I. Iturrate, L. Montesano, P.-Y. Oudeyer, and M. Lopes. Interactive Learning from Unlabeled Instructions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pages 290–299, Arlington, Virginia, United States, 2014. AUAI Press.

[40] J. Grizou, M. Lopes, and P. Y. Oudeyer. Robot learning simultaneously a task and how to interpret human instructions. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–8, Aug. 2013.

[41] V. Gullapalli and A. G. Barto. Shaping as a method for accelerating reinforcement learning. In *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*, pages 554–559, Aug. 1992.

[42] M. E. Harmon, L. C. Baird, and A. H. Klopf. Advantage updating applied to a differential game. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS94, page 353360, Cambridge, MA, USA, 1994. MIT Press.

[43] G. M. Hayes and J. Demiris. A robot controller using learning by imitation. In *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, Grenoble, France, July 1994.

[44] F. Hayes-Roth, P. Klahr, and D. J. Mostow. *Knowledge Acquisition, Knowledge Programming, and Knowledge Refinement.* Rand Corporation, 1980.

[45] F. Hayes-Roth, P. Klahr, and D. J. Mostow. Advice Taking and Knowledge Refinement: An Iterative View of Skill. *Cognitive skills and their acquisition*, page 231, 1981.

[46] M. K. Ho, M. L. Littman, F. Cushman, and J. L. Austerweil. Teaching with Rewards and Punishments: Reinforcement or Communication? In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, July 2015.

[47] M. K. Ho, J. MacGlashan, M. L. Littman, and F. Cushman. Social is special: A normative framework for teaching with and learning from evaluative feedback. *Cognition*, 167:91–106, 2017.

[48] T. Inamura, M. Inaba, and H. Inoue. Acquisition of probabilistic behavior decision model based on the interactive teaching method. In *Proceedings of the Ninth International Conference on Advanced Robotics, ICAR99*, 1999.

[49] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone. A Social Reinforcement Learning Agent. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 377–384, New York, NY, USA, 2001. ACM.

[50] K. Judah, A. Fern, P. Tadepalli, and R. Goetschalckx. Imitation Learning with Demonstrations and Shaping Rewards. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1890–1896, Québec City, Québec, Canada, 2014. AAAI Press.

[51] K. Judah, S. Roy, A. Fern, and T. G. Dietterich. Reinforcement Learning via Practice and Critique Advice. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, pages 481–486, Atlanta, Georgia, 2010. AAAI Press.

[52] F. Kaplan, P.-Y. Oudeyer, E. Kubinyi, and A. Miklsi. Robotic clicker training. *Robotics and Autonomous Systems*, 38(34):197 – 206, 2002.

[53] R. J. Kate and R. J. Mooney. Using String-kernels for Learning Semantic Parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 913–920, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[54] E. S. Kim and B. Scassellati. Learning to refine behavior using prosodic feedback. In *2007 IEEE 6th International Conference on Development and Learning*, pages 205–210, July 2007.

[55] W. B. Knox and P. Stone. Interactively Shaping Agents via Human Reinforcement: The TAMER Framework. In *Proceedings of the Fifth International Conference on Knowledge Capture*, K-CAP '09, pages 9–16, New York, NY, USA, 2009. ACM.

[56] W. B. Knox and P. Stone. Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 5–12, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[57] W. B. Knox and P. Stone. Augmenting Reinforcement Learning with Human Feedback. In *ICML 2011 Workshop on New Developments in Imitation Learning*, July 2011.

[58] W. B. Knox and P. Stone. Understanding Human Teaching Modalities in Reinforcement Learning Environments: A Preliminary Report. In *IJCAI 2011 Workshop on Agents Learning Interactively from Human Teachers (ALIHT)*, July 2011.

[59] W. B. Knox and P. Stone. Reinforcement learning from human reward: Discounting in episodic tasks. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 878–885, Sept. 2012.

[60] W. B. Knox and P. Stone. Reinforcement Learning from Simultaneous Human and MDP Reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '12, pages 475–482, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[61] W. B. Knox, P. Stone, and C. Breazeal. Training a Robot via Human Feedback: A Case Study. In *Proceedings of the 5th International Conference on Social Robotics - Volume 8239*, ICSR 2013, pages 460–470, New York, NY, USA, 2013. Springer-Verlag New York, Inc.

[62] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement Learning in Robotics: A Survey. *Int. J. Rob. Res.*, 32(11):1238–1274, Sept. 2013.

[63] E. Kohler, C. Keysers, M. A. Umilt, L. Fogassi, V. Gallese, and G. Rizzolatti. Hearing Sounds, Understanding Actions: Action Representation in Mirror Neurons. *Science*, 297(5582):846–848, 2002.

[64] S. Krening, B. Harrison, K. M. Feigh, C. L. Isbell, M. Riedl, and A. Thomaz. Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55, March 2017.

[65] G. Kuhlmann, P. Stone, R. J. Mooney, and J. W. Shavlik. Guiding a Reinforcement Learner with Natural Language Advice: Initial Results in RoboCup Soccer. In *The AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*, July 2004.

[66] A. Len, E. F. Morales, L. Altamirano, and J. R. Ruiz. Teaching a Robot to Perform Task Through Imitation and On-line Feedback. In *Proceedings of the 16th Iberoamerican Congress Conference on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, CIARP'11, pages 549–556, Berlin, Heidelberg, 2011. Springer-Verlag.

[67] J. Lieberman and C. Breazeal. Improvements on action parsing and action interpolation for learning through demonstration. In *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, volume 1, pages 342–365 Vol. 1, Nov. 2004.

[68] L.-J. Lin. Programming Robots Using Reinforcement Learning and Teaching. In *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2*, AAAI'91, pages 781–786, Anaheim, California, 1991. AAAI Press.

[69] A. Lockerd and C. Breazeal. Tutelage and socially guided robot learning. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 4, pages 3475–3480, Sept. 2004.

[70] R. Loftin, J. MacGlashan, B. Peng, M. E. Taylor, M. L. Littman, J. Huang, and D. L. Roberts. A Strategy-aware Technique for Learning Behaviors from Discrete Human Feedback. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 937–943, Québec City, Québec, Canada, 2014. AAAI Press.

[71] R. Loftin, B. Peng, J. Macglashan, M. L. Littman, M. E. Taylor, J. Huang, and D. L. Roberts. Learning Behaviors via Human-delivered Discrete Feedback: Modeling Implicit Feedback Strategies to Speed Up Learning. *Autonomous Agents and Multi-Agent Systems*, 30(1):30–59, Jan. 2016.

[72] M. Lopes, T. Cederbourg, and P. Y. Oudeyer. Simultaneous acquisition of task and feedback models. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–7, Aug. 2011.

[73] T. Lozano-Perez. Robot programming. *Proceedings of the IEEE*, 71(7):821–841, July 1983.

[74] J. MacGlashan, M. Babes-Vroman, M. DesJardins, M. Littman, S. Muresan, and S. Squire. Translating english to reward functions. Technical report, Technical Report CS14-01, Computer Science Department, Brown University, 2014.

[75] J. MacGlashan, M. K. Ho, R. Loftin, B. Peng, G. Wang, D. L. Roberts, M. E. Taylor, and M. L. Littman. Interactive learning from policy-dependent human feedback. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2285–2294. JMLR. org, 2017.

[76] J. MacGlashan, M. Littman, R. Loftin, B. Peng, D. Roberts, and M. E. Taylor. Training an agent to ground commands with reward and punishment. In *Proceedings of the AAAI Machine Learning for Interactive Systems Workshop*, 2014.

[77] R. Maclin, J. Shavlik, L. Torrey, T. Walker, and E. Wild. Giving Advice About Preferred Actions to Reinforcement Learners via Knowledge-based Kernel Regression. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, pages 819–824, Pittsburgh, Pennsylvania, 2005. AAAI Press.

[78] R. Maclin, J. Shavlik, T. Walker, and L. Torrey. Knowledge-based support-vector regression for reinforcement learning. In *IJCAI 2005 Workshop on Reasoning, Representation, and Learning in Computer Games*, page 61, 2005.

[79] R. Maclin and J. W. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 22(1):251–281, 1996.

[80] O. L. Mangasarian, J. W. Shavlik, and E. W. Wild. Knowledge-Based Kernel Approximation. *J. Mach. Learn. Res.*, 5:1127–1141, Dec. 2004.

[81] K. W. Mathewson and P. M. Pilarski. Simultaneous Control and Human Feedback in the Training of a Robotic Agent with Actor-Critic Reinforcement Learning. *arXiv preprint arXiv:1606.06979*, 2016.

[82] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to Parse Natural Language Commands to a Robot Control System. In J. P. Desai, G. Dudek, O. Khatib, and V. Kumar, editors, *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, pages 403–415. Springer International Publishing, Heidelberg, 2013. DOI: 10.1007/978-3-319-00065-7_28.

[83] J. McCarthy. Programs with Common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes (December 1958)*, pages 75–91, London, 1959. Her Majesty's Stationary Office.

[84] R. J. Mooney. Learning to Connect Language and Perception. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1598–1601, Chicago, Illinois, 2008. AAAI Press.

[85] C. Mueller, J. Venicx, and B. Hayes. Robust robot learning from demonstration and skill repair using conceptual constraints. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6029–6036, Oct 2018.

[86] A. Najar. *Shaping robot behaviour with unlabeled human instructions*. PhD thesis, Paris 6, 2017.

[87] A. Najar, E. Bonnet, B. Bahrami, and S. Palminteri. Imitation as a model-free process in human reinforcement learning. *bioRxiv*, 2019.

[88] A. Najar, O. Sigaud, and M. Chetouani. Social-Task Learning for HRI. In A. Tapus, E. Andr, J.-C. Martin, F. Ferland, and M. Ammi, editors, *Social Robotics: 7th International Conference, ICSR 2015, Paris, France, October 26-30, 2015, Proceedings*, pages 472–481. Springer International Publishing, Cham, 2015. DOI: 10.1007/978-3-319-25554-5_47.

[89] A. Najar, O. Sigaud, and M. Chetouani. Training a robot with evaluative feedback and unlabeled guidance signals. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 261–266, Aug. 2016.

[90] A. Najar, O. Sigaud, and M. Chetouani. Interactively shaping robot behaviour with unlabeled human instructions. *Auton. Agents Multi Agent Syst.*, 34(2):35, 2020.

[91] C. L. Nehaniv and K. Dautenhahn. The Correspondence Problem. In K. Dautenhahn and C. L. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 41–61. MIT Press, Cambridge, MA, USA, 2002.

[92] A. Y. Ng, D. Harada, and S. J. Russell. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[93] A. Y. Ng and S. J. Russell. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[94] M. N. Nicolescu and M. J. Mataric. Learning and Interacting in Human-robot Domains. *Trans. Sys. Man Cyber. Part A*, 31(5):419–430, Sept. 2001.

[95] M. N. Nicolescu and M. J. Mataric. 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*, AAMAS '03, pages 241–248, New York, NY, USA, 2003. ACM.

[96] A. Olsson, E. Knapska, and B. Lindström. The neural and computational systems of social learning. *Nature Reviews Neuroscience*, pages 1–16, 2020.

[97] V. Paléologue, J. Martin, A. K. Pandey, and M. Chetouani. Semantic-based interaction for teaching robot behavior compositions using spoken language. In *Social Robotics - 10th International Conference, ICSR 2018, Qingdao, China, November 28-30, 2018, Proceedings*, pages 421–430, 2018.

[98] K. V. N. Pradyot, S. S. Manimaran, and B. Ravindran. Instructing a Reinforcement Learner. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, pages 23–25, Marco Island, Florida., May 2012.

[99] K. V. N. Pradyot, S. S. Manimaran, B. Ravindran, and S. Natarajan. Integrating Human Instructions and Reinforcement Learners: An SRL Approach. *Proceedings of the UAI workshop on Statistical Relational AI*, 2012.

[100] K. V. N. Pradyot and B. Ravindran. Beyond Rewards: Learning from Richer Supervision. In *Proceedings of the 9th European Workshop on Reinforcement Learning*, Athens Greece, 2011.

[101] G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature reviews neuroscience*, 2(9):661, 2001.

[102] M. T. Rosenstein, A. G. Barto, J. Si, A. Barto, W. Powell, and D. Wunsch. Supervised Actor-Critic Reinforcement Learning. In *Handbook of Learning and Approximate Dynamic Programming*, pages 359–380. John Wiley & Sons, Inc., 2004. DOI: 10.1002/9780470544785.ch14.

[103] P. E. Rybski, K. Yoon, J. Stolarz, and M. M. Veloso. Interactive robot task training through dialog and demonstration. In *2007 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 49–56, Mar. 2007.

[104] J. Saunders, C. L. Nehaniv, and K. Dautenhahn. Teaching Robots by Moulding Behavior and Scaffolding the Environment. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, HRI '06, pages 118–125, New York, NY, USA, 2006. ACM.

[105] S. P. Singh. Transfer of Learning by Composing Solutions of Elemental Sequential Tasks. *Machine Learning*, 8(3-4):323–339, May 1992.

[106] M. Sridharan. Augmented Reinforcement Learning for Interaction with Non-expert Humans in Agent Domains. In *Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops - Volume 01*, ICMLA '11, pages 424–429, Washington, DC, USA, 2011. IEEE Computer Society.

[107] H. B. Suay and S. Chernova. Effect of human guidance and state space size on Interactive Reinforcement Learning. In *2011 RO-MAN*, pages 1–6, July 2011.

[108] H. B. Suay, R. Toris, and S. Chernova. A Practical Comparison of Three Robot Learning from Demonstration Algorithm. *International Journal of Social Robotics*, 4(4):319–330, 2012.

[109] K. Subramanian, C. L. Isbell, Jr., and A. L. Thomaz. Exploration from Demonstration for Interactive Reinforcement Learning. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 447–456, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[110] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[111] U. Syed and R. E. Schapire. Imitation Learning with a Value-based Prior. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, UAI'07, pages 384–391, Arlington, Virginia, United States, 2007. AUAI Press.

[112] M. E. Taylor, H. B. Suay, and S. Chernova. Integrating Reinforcement Learning with Human Demonstrations of Varying Ability. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 617–624, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[113] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, Aug. 2011.

[114] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villaseor-Pineda. Dynamic Reward Shaping: Training a Robot by Voice. In A. Kuri-Morales and G. R. Simari, editors, *Advances in Artificial Intelligence IBERAMIA 2010: 12th Ibero-American Conference on AI, Baha Blanca, Argentina, November 1-5, 2010. Proceedings*, pages 483–492. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. DOI: 10.1007/978-3-642-16952-6_49.

[115] A. L. Thomaz. *Socially Guided Machine Learning*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.

[116] A. L. Thomaz and C. Breazeal. Reinforcement Learning with Human Teachers: Evidence of Feedback and Guidance with Implications for Learning Performance. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 1000–1005, Boston, Massachusetts, 2006. AAAI Press.

[117] A. L. Thomaz and C. Breazeal. Transparency and Socially Guided Machine Learning. In *the 5th International Conference on Developmental Learning*, 2006.

[118] A. L. Thomaz and C. Breazeal. Asymmetric Interpretations of Positive and Negative Human Feedback for a Social Learning Agent. In *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, pages 720–725, Aug. 2007.

[119] A. L. Thomaz and C. Breazeal. Robot learning via socially guided exploration. In *2007 IEEE 6th International Conference on Development and Learning*, pages 82–87, July 2007.

[120] A. L. Thomaz and M. Cakmak. Learning About Objects with Human Teachers. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, HRI '09, pages 15–22, New York, NY, USA, 2009. ACM.

[121] A. L. Thomaz, G. Hoffman, and C. Breazeal. Reinforcement Learning with Human Teachers: Understanding How People Want to Teach Robots. In *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 352–357, Sept. 2006.

[122] L. Torrey, T. Walker, R. Maclin, and J. W. Shavlik. Advice Taking and Transfer Learning: Naturally Inspired Extensions to Reinforcement Learning. In *AAAI Fall Symposium: Naturally-Inspired Artificial Intelligence*, volume FS-08-06 of *AAAI Technical Report*, pages 103–110. AAAI, 2008.

[123] G. G. Towell and J. W. Shavlik. Knowledge-based Artificial Neural Networks. *Artif. Intell.*, 70(1-2):119–165, Oct. 1994.

[124] A. M. Turing. Computing Machinery and Intelligence. *Mind*, 59(236):433–460, 1950.

[125] P. E. Utgoff and J. A. Clouse. Two Kinds of Training Information for Evaluation Function Learning. In *In Proceedings of the Ninth Annual Conference on Artificial Intelligence*, pages 596–600. Morgan Kaufmann, 1991.

[126] A. Vogel and D. Jurafsky. Learning to Follow Navigational Directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 806–814, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[127] A.-L. Vollmer, B. Wrede, K. J. Rohlfing, and P.-Y. Oudeyer. Pragmatic Frames for Teaching and Learning in HumanRobot Interaction: Review and Challenges. *Frontiers in Neurorobotics*, 10:10, 2016.

[128] S. D. Whitehead. A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning. In *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2*, AAAI'91, pages 607–613, Anaheim, California, 1991. AAAI Press.

[129] S. D. Whitehead and D. H. Ballard. Learning to Perceive and Act by Trial and Error. *Mach. Learn.*, 7(1):45–83, July 1991.

[130] M. A. Wiering and H. van Hasselt. Ensemble Algorithms in Reinforcement Learning. *Trans. Sys. Man Cyber. Part B*, 38(4):930–936, Aug. 2008.

[131] E. Wiewiora. Potential-Based Shaping and Q-Value Initialization are Equivalent. *J. Artif. Intell. Res. (JAIR)*, 19:205–208, 2003.

[132] L. S. Zettlemoyer and M. Collins. Learning Context-dependent Mappings from Sentences to Logical Form. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 976–984, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.