frontiers in Robotics and AI

REVIEW
published: 01 June 2021
doi: 10.3389/frobt.2021.584075

Reinforcement Learning With Human Advice: A Survey

Anis Najar[1]* and Mohamed Chetouani[2]

[1] Laboratoire de Neurosciences Cognitives Computationnelles, INSERM U960, Paris, France, [2] Institute for Intelligent Systems and Robotics, Sorbonne Université, CNRS UMR 7222, Paris, France

In this paper, we provide an overview of the existing methods for integrating human advice into a reinforcement learning process. We first propose a taxonomy of the different forms of advice that can be provided to a learning agent. We then describe the methods that can be used for interpreting advice when its meaning is not determined beforehand. Finally, we review different approaches for integrating advice into the learning process.

Keywords: advice-taking systems, reinforcement learning, interactive machine learning, human-robot interaction, unlabeled teaching signals

OPEN ACCESS

Edited by:
Iolanda Leite,
Royal Institute of Technology, Sweden

Reviewed by:
Garrett Warnell,
United States Army Research Laboratory, United States
Tesca Fitzgerald,
Carnegie Mellon University, United States

***Correspondence:**
Anis Najar
anis.najar@ens.fr

Specialty section:
This article was submitted to Human-Robot Interaction, a section of the journal Frontiers in Robotics and AI

Received: 16 July 2020
Accepted: 03 March 2021
Published: 01 June 2021

Citation:
Najar A and Chetouani M (2021) Reinforcement Learning With Human Advice: A Survey. Front. Robot. AI 8:584075. doi: 10.3389/frobt.2021.584075

1. INTRODUCTION

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."* (Turing, 1950). 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 (Chernova and Thomaz, 2014). 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 (Vollmer et al., 2016).

The interactive machine learning literature references a plethora of teaching signals such as instructions (Pradyot et al., 2012b; Najar et al., 2020b), demonstrations (Argall et al., 2009), and feedback (Knox and Stone, 2009; Najar et al., 2016). 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, learning from evaluative feedback (or critique), and learning from demonstration (LfD) (Knox and Stone, 2009, 2011b; Judah et al., 2010). While this taxonomy is commonly used in the literature, it is not infallible as these categories can overlap. For example, in some papers, evaluative feedback is considered as a particular type of advice (Judah et al., 2010; Griffith et al., 2013). In more rare cases, demonstrations (Whitehead, 1991; Lin, 1992) were also referred to as advice (Maclin and Shavlik, 1996; Maclin et al., 2005a). The definition of advice in the literature is relatively vague with no specific constraints on what type of input can be provided to the learning agent. For example, it has been defined as *"concept definitions, behavioral constraints, and performance heuristics"* (Hayes-Roth et al., 1981), or as *"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"* (Pradyot and Ravindran, 2011). Although more specific definitions can be found, such as *"suggesting an action when a certain condition is true"* (Knox and Stone, 2009), in other works advice also represents state preferences (Utgoff and Clouse, 1991), action preferences (Maclin et al., 2005a), constraints on action values (Maclin et al., 2005b; Torrey et al., 2008), explanations (Krening et al., 2017), instructions (Clouse and Utgoff, 1992; Maclin and Shavlik, 1996; Kuhlmann et al., 2004; Rosenstein et al., 2004), feedback (Judah et al., 2010; Griffith et al., 2013; Celemin and Ruiz-Del-Solar, 2019), or demonstrations (Whitehead, 1991; Lin, 1992; Maclin and Shavlik, 1996). In some papers, the term feedback is used as a shortcut for evaluative feedback (Thomaz and Breazeal, 2006; Leon et al., 2011; Griffith et al., 2013; Knox et al., 2013; Loftin et al., 2016). However, the same term is sometimes used to refer to corrective feedback (Argall et al., 2011). 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 toward 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 to clarify some of the terminology used in the interactive machine learning literature by providing a taxonomy of the different forms of advice, and to review how these teaching signals can be integrated into a reinforcement learning (RL) process (Sutton and Barto, 1998). In this survey, we define advice as *teaching signals that can be communicated by the teacher to the learning system without executing the task*. Thus, we do not cover LfD, since demonstration is different from advice given this definition, and comprehensive surveys on this topic already exist (Argall et al., 2009; Chernova and Thomaz, 2014).

Although the methods we cover belong to various mathematical frameworks, we mainly focus on the RL perspective. We equivalently use the terms of "agent," "robot," and "system," by making abstraction of the support over which the RL algorithm is implemented. Throughout this paper, we use the term "shaping" to refer to the mechanism by which advice is integrated into the learning process. Although 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 (Gullapalli and Barto, 1992; Singh, 1992; Dorigo and Colombetti, 1994; Knox and Stone, 2009; Judah et al., 2014; Cederborg et al., 2015), the general meaning of shaping is equivalent to training, which is to make an agent's *"behavior converge to a predefined target behavior"* (Dorigo and Colombetti, 1994).

The paper is organized as follows. We first introduce some background about RL in section 2. We then provide an overview of the existing methods for integrating human advice into an RL process in section 3. The different methods are discussed in section 4, before concluding the paper in section 5.

2. REINFORCEMENT LEARNING

RL refers to family of problems where an autonomous agent has to learn a sequential decision-making task (Sutton and Barto, 1998). These problems are generally represented as Markov decision process (MDP), defined as a tuple $< S, A, T, R, \gamma >$. S represents the state-space over which the problem is defined and A is the set of actions the agent is able to perform on every time-step. $T : S \times A \rightarrow Pr(s'|s, a)$ defines a state-transition probability function, where $Pr(s'|s, a)$ represents the probability that the agent transitions from state s to state s' after executing action a. $R : S \times A \rightarrow \mathbb{R}$ is a reward function that defines the reward $r(s, a)$ that the agent gets for performing action a in state s. When at time t, the agent performs an action a_t from state s_t, it receives a reward r_t and transitions to state s_{t+1}. The discount factor, γ, represents how much future rewards are taken into account for the current decision.

The behavior of the agent is represented as a policy π that defines the probability to select each action in every state: $\forall s \in S$, $\pi(s) = \{\pi(s, a); a \in A\} = \{Pr(a|s); a \in A\}$. The quality of a policy is measured by the amount of rewards it enables the agent to collect over the long run. The expected amount of cumulative rewards, when starting from a state s and following a policy π, is given by the state-value function and is written as:

$$V^{\pi}(s) = \sum_{a} \pi(s, a)[R(s, a) + \gamma \sum_{s'} Pr(s'|s, a)V^{\pi}(s')]. \quad (1)$$

Another form of value function, called action-value function and noted Q^{π}, provides more directly exploitable information than V^{π} for decision-making, as the agent has direct access to the value of each possible decision:

$$Q^{\pi}(s, a) = R(s, a) + \gamma \sum_{s'} Pr(s'|s, a)V^{\pi}(s') \quad ; \forall s \in S, a \in A. \quad (2)$$

To optimize its behavior, the agent must find the optimal policy π^* that maximizes V^{π} and Q^{π}. When both the reward and transition functions are unknown, the optimal policy must be learnt from the rewards the agent obtains by interacting with its environment using an RL algorithm. RL algorithms can be decomposed into three categories: value-based, policy-gradient, and Actor-Critic (Sutton and Barto, 1998).

2.1. Value-Based RL

In value-based RL, the optimal policy is obtained by iteratively optimizing the value function. Examples of value-based algorithms include Q-learning (Watkins and Dayan, 1992) and SARSA (Sutton, 1996).

In Q-learning, the action-value function of the optimal policy π^* is computed iteratively. On every time-step t, when the agent transitions from state s_t to state s_{t+1} by performing an action a_t, and receives a reward r_t, the Q-value of the last state-action pair is updated using:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a_t)], \quad (3)$$

where $\alpha \in [0, 1]$ is a learning rate.

At decision time, the policy π can be derived from the Q-function using different action-selection strategies. The ϵ-*greedy* action-selection strategy consists of selecting most of the time the optimal action with respect to the Q-function, $a_t = \max_{a \in A} Q(s_t, a)$, and selecting with a small probability ϵ a random action. With the *softmax* action-selection strategy, the policy π is derived at decision-time by computing a softmax distribution over the Q-values:

$$\pi(s,a) = Pr(a_t = a|s_t = s) = \frac{e^{Q(s,a)}}{\sum_{b\in A} e^{Q(s,b)}}. \tag{4}$$

The SARSA algorithm is similar to Q-learning, with one difference at the update function of the Q-values:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)], \tag{5}$$

where a_{t+1} is the action the agent selects at time-step $t + 1$. At decision time, the same action-selection strategies can be implemented as for Q-learning.

2.2. Policy-Gradient RL

In contrast to value-based RL, policy-gradient methods do not compute a value function (Williams, 1992). Instead, the policy is directly optimized from the perceived rewards. In this approach, the policy π is controlled with a set of parameters $w \in \mathbb{R}^n$, such that $\pi_w(s, a)$ is differentiable in w; $\forall s \in S, a \in A$. For example, w can be defined so that $w(s, a)$ reflects the preference for taking an action in a given state by expressing the policy as a softmax distribution over the parameters:

$$\pi_w(s,a) = Pr(a_t = a|s_t = s) = \frac{e^{w(s,a)}}{\sum_{b\in A} e^{w(s,b)}}. \tag{6}$$

A learning iteration is composed of two stages. First, the agent estimates the expected returns, G, by sampling a set of trajectories. Then, the policy π_w is updated using the gradient of the expected returns with respect to w. For example, in the REINFORCE algorithm (Williams, 1992), a trajectory of T time-steps is first sampled from one single episode. Then, for every time-step t of the trajectory, the return G is computed as $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} r_t$, and the policy parameters are updated with:

$$w \leftarrow w + \gamma^t G \nabla_w \ln \pi_w(a_t|s_t). \tag{7}$$

2.3. Actor-Critic RL

Actor-Critic architectures constitute a hybrid approach between value-based and policy-gradient methods by computing both the policy (the actor) and a value function (the critic) (Barto et al., 1983). The actor can be represented as a parameterized softmax distribution as in Equation (6). The critic computes a value function that is used for evaluating the actor. The reward r_t received at time t is used for computing a temporal difference (TD) error:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t). \tag{8}$$

The TD error is then used for updating both the critic and the actor, using respectively, Equations (9) and (10):

$$V(s_t) \leftarrow V(s_t) + \alpha\delta_t, \tag{9}$$

$$w(s_t, a_t) \leftarrow w(s_t, a_t) + \beta\delta_t, \tag{10}$$

where $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are two learning rates. A positive TD error increases the probability of selecting a_t in s_t, while a negative TD error decreases it.

The main advantage of RL algorithms is the autonomy of the learning process. Given a predefined reward function, they allow an agent to optimize its behavior without the intervention of a human supervisor. However, they present several limitations. For instance, they involve a time-consuming iterative process that limits their applicability to complex real-world problems (Kober et al., 2013). Some existing techniques, such as reward shaping, aim at overcoming this limitation by defining intermediate rewards (Gullapalli and Barto, 1992; Mataric, 1994). However, they generally require expert knowledge for designing an appropriate reward shaping function (Ng et al., 1999; Wiewiora et al., 2003). Also, the exploration aspect of autonomous learning methods raises several safety issues (Garcia and Fernandez, 2015).

Interactive learning constitutes a complementary approach that aims at overcoming these limitations by involving a human teacher in the learning process. In the next section, we show how a human teacher can provide an RL agent with various forms of advice to convey different information about the task. We then show how advice can be interpreted by the agent, for instance by grounding its meaning in the learning process using either the reward function, the value function or the policy. Finally, we show how advice can be used, in turn, to intervene at different levels of the learning process, by influencing either the reward function, the value function, the policy, or the action-selection strategy.

3. REINFORCEMENT LEARNING WITH HUMAN ADVICE

In one of the first papers of artificial intelligence, John McCarthy described an *"Advice Taker"* system that could learn by being told (McCarthy, 1959). This idea was then elaborated in Hayes-Roth et al. (1980) and Hayes-Roth et al. (1981), where a general framework for learning from advice was proposed. This framework can be summarized in the following five steps (Cohen and Feigenbaum, 1982; Maclin and Shavlik, 1996):

1. Requesting or receiving the advice.
2. Converting the advice into an internal representation.
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 human advice can be provided to the system. Different forms of advice can be distinguished based on this criterion. Step 2 refers to the encoding the perceived advice into an internal representation. Most of existing advice-taking systems assume that the internal representation of advice is predetermined by the system designer. However, some recent works tackle the problem of letting the system learn how to interpret raw advice in order to make the interaction protocol less constraining for the human teacher (Vollmer et al., 2016).

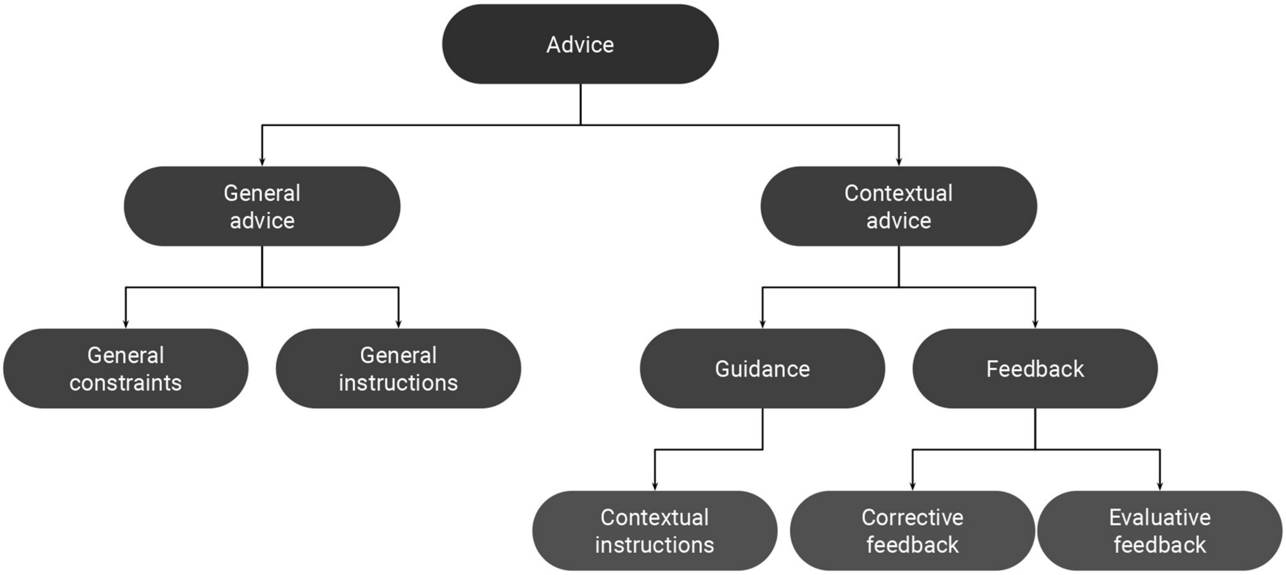

FIGURE 1 | Taxonomy of advice.

Steps 3–5 describe how human advice can be used by the agent for learning. These three steps are often confounded into one single process, that we call shaping, which consists of integrating advice into the agent's learning process.

In the remainder of this section, we first propose a taxonomy of different categories of advice based on how they can be provided to the system (step 1). Then we detail how advice can be interpreted (step 2). Finally, we present how advice can be integrated into an RL process (steps 3–5).

3.1. Providing Advice

The means by which teaching signals can be communicated to a learning agent vary. They can be provided via natural language (Kuhlmann et al., 2004; Cruz et al., 2015; Paléologue et al., 2018), computer vision (Atkeson and Schaal, 1997; Najar et al., 2020b), hand-written programs (Maclin and Shavlik, 1996; Maclin et al., 2005a,b; Torrey et al., 2008), artificial interfaces (Abbeel et al., 2010; Suay and Chernova, 2011; Knox et al., 2013), or physical interaction (Lozano-Perez, 1983; Akgun et al., 2012). Despite the variety of communication channels, we can distinguish two main categories of teaching signals based on how they are produced: advice and demonstration. Even though advice and demonstration can share the same communication channels, like computer vision (Atkeson and Schaal, 1997; Najar et al., 2020b) and artificial interfaces (Abbeel et al., 2010; Suay and Chernova, 2011; Knox et al., 2013), they are fundamentally different from each other in that demonstration requires the task to be executed by the teacher (demonstrated), while advice does not. In rare cases, demonstration (Whitehead, 1991; Lin, 1992) has been referred to as advice (Maclin and Shavlik, 1996; Maclin et al., 2005a). However, it is more common to consider demonstration and advice as two distinct and complementary approaches for interactive learning (Dillmann et al., 2000; Argall et al., 2008; Knox and Stone, 2009, 2011b; Judah et al., 2010). Based on this distinction, we define advice as *teaching signals that can be communicated by the teacher to the learning system without executing the task*.

TABLE 1 | Types of advice.

Category	References
General constraints	Hayes-Roth et al., 1981; Kuhlmann et al., 2004; Mangasarian et al., 2004; Maclin et al., 2005a,b; Torrey et al., 2008
General instructions	Maclin and Shavlik, 1996; Kuhlmann et al., 2004; Branavan et al., 2009, 2010; Vogel and Jurafsky, 2010
Guidance	Thomaz, 2006; Thomaz and Cakmak, 2009; Suay and Chernova, 2011; Chu et al., 2016; Subramanian et al., 2016
Contextual instructions	Utgoff and Clouse, 1991; Clouse and Utgoff, 1992; Nicolescu and Mataric, 2003; Rosenstein et al., 2004; Rybski et al., 2007; Thomaz and Breazeal, 2007b; Branavan et al., 2010; Tenorio-Gonzalez et al., 2010; Pradyot et al., 2012b; Grizou et al., 2013; MacGlashan et al., 2014a; Cruz et al., 2015; Mathewson and Pilarski, 2016; Najar et al., 2020b
Corrective feedback	Nicolescu and Mataric, 2003; Chernova and Veloso, 2009; Argall et al., 2011; Celemin and Ruiz-Del-Solar, 2019
Evaluative feedback	Dorigo and Colombetti, 1994; Colombetti et al., 1996; Isbell et al., 2001; Kaplan et al., 2002; Thomaz et al., 2006; Kim and Scassellati, 2007; Knox and Stone, 2009, 2010, 2011a, 2012a,b; Judah et al., 2010; Tenorio-Gonzalez et al., 2010; Lopes et al., 2011; Grizou et al., 2013 Griffith et al., 2013; Grizou et al., 2014b; Loftin et al., 2014, 2016; Ho et al., 2015; Mathewson and Pilarski, 2016; Najar et al., 2016, 2020b; MacGlashan et al., 2017

We mainly distinguish two forms of advice depending on how it is provided to the system: *general advice* and *contextual advice*

(**Figure 1**, **Table 1**). *General advice* can be communicated to the system, non-interactively, prior to the learning process (offline). This type of advice represents information about the task 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). Examples include specifying general constraints about the task and providing general instructions about the desired behavior. *Contextual advice*, on the other hand, is context-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 (Knox and Stone, 2009; Celemin and Ruiz-Del-Solar, 2019; Najar et al., 2020b). *Contextual advice* can also be provided in an offline fashion, with the teacher interacting with previously recorded task executions by the learning agent (Judah et al., 2010; Argall et al., 2011). Even in this case, each piece of advice has to be provided at a specific moment of the task execution. Examples of *contextual advice* include evaluative feedback (Knox and Stone, 2009; Najar et al., 2016), corrective feedback (Argall et al., 2011; Celemin and Ruiz-Del-Solar, 2019), guidance (Thomaz and Breazeal, 2006; Suay and Chernova, 2011), and contextual instructions (Clouse and Utgoff, 1992; Rosenstein et al., 2004; Pradyot et al., 2012a; Najar et al., 2020b).

3.1.1. General Advice

Advice can be used by the human teacher to provide the agent with general information about the task prior to the learning process. These information can be provided to the system in a written form (Hayes-Roth et al., 1980; Maclin and Shavlik, 1996; Kuhlmann et al., 2004; Branavan et al., 2009; Vogel and Jurafsky, 2010).

General advice can specify *general constraints* about the task such as domain concepts, behavioral constraints, and performance heuristics. For example, the first ever implemented advice-taking system relied on general constraints that were written as LISP expressions, to specify concepts, rules and heuristics for a card-playing agent (Hayes-Roth et al., 1981).

A second form of general advice, *general instructions*, explicitly specifies to the agent what actions to perform in different situations. It can be provided either in the form of *if-then* rules (Maclin and Shavlik, 1996; Kuhlmann et al., 2004), or as detailed action plans describing the step-by-step sequence of actions that should be performed in order to solve the task (Branavan et al., 2009; Vogel and Jurafsky, 2010). Action plans can be seen as a sequence of low-level or high-level *contextual instructions* (cf. definition below). For example, a sequence like (e.g., *"Click start, point to search, and then click for files or folders."*), can be decomposed into a sequence of three low-level *contextual instructions* (Branavan et al., 2009).

3.1.2. Contextual Advice

In contrast to *general advice*, a *contextual advice* 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, and must be inferred by the learner from the current context. Consequently, *contextual advice* must be progressively provided to the learning agent along the task. Contextual advice can be divided into two main categories: guidance and feedback. Guidance informs about future actions, whereas feedback informs about past ones.

3.1.3. Guidance

Guidance is a term that is encountered in many papers and has been made popular by the work of Thomaz (2006) 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 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 (Thomaz and Cakmak, 2009). For instance, in Subramanian et al. (2016), demonstrations were provided in order to teach the agent how to explore interesting regions of the state space. In Chu et al. (2016), 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 of suggesting a limited set of actions from all the possible ones (Thomaz and Breazeal, 2006; Suay and Chernova, 2011).

3.1.4. Contextual Instructions

One particular type of guidance is to suggest only one action to perform. We refer to this type of advice as *contextual instructions*. For example, in Cruz et al. (2015), the authors used both terms of advice and guidance for referring to contextual instructions. Contextual instructions can be either low-level or high-level (Branavan et al., 2010). Low-level instructions indicate the next action to perform (Grizou et al., 2013), whereas high-level instructions indicate a more extended goal without explicitly specifying the sequence of actions that should be executed (MacGlashan et al., 2014a). High-level instructions were also referred to as commands (MacGlashan et al., 2014a; Tellex et al., 2014). In RL terminology, high-level instructions would correspond to performing *options* (Sutton et al., 1999). Contextual instructions can be provided through speech (Grizou et al., 2013), gestures (Najar et al., 2020b), or myoelectric (EMG) interfaces (Mathewson and Pilarski, 2016).

3.1.5. 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 (Knox and Stone, 2009; Judah et al., 2010). Corrective feedback, also called instructive feedback, implicitly implies that the performed action is wrong (Argall et al., 2011; Celemin and Ruiz-Del-Solar, 2019). However, it goes beyond simply criticizing the performed action, by informing the agent about the correct one.

3.1.6. Corrective Feedback

Corrective feedback can be either a corrective instruction (Chernova and Veloso, 2009) or a corrective demonstration (Nicolescu and Mataric, 2003). The main difference with instructions (respectively, demonstrations) is that they are provided after an action (respectively, a sequence of actions) is executed by the agent, not before. So, operationalization is made with respect to the previous state instead of the current one.

So far, corrective feedback has been mainly used for augmenting LfD systems (Nicolescu and Mataric, 2003; Chernova and Veloso, 2009; Argall et al., 2011). For example, in Chernova and Veloso (2009), while the robot is reproducing the provided demonstrations, the teacher could interactively rectify any incorrect action. In Nicolescu and Mataric (2003), corrective demonstrations were delimited by two predefined verbal commands that were pronounced by the teacher. In Argall et al. (2011), 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 Celemin and Ruiz-Del-Solar (2019), 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 (Knox and Stone, 2009).

3.1.7. Evaluative Feedback

Teaching an agent by evaluating its actions is an alternative solution to the standard RL approach. Evaluative feedback can be provided in different forms: a scalar value $f \in [-1, 1]$ (Knox and Stone, 2009), a binary value $f \in \{-1, 1\}$ (Thomaz et al., 2006; Najar et al., 2020b), a positive reinforcer $f \in \{\text{“}Good!''\text{, “}Bravo!''\}$ (Kaplan et al., 2002), or a categorical information $f \in \{Correct, Wrong\}$ (Loftin et al., 2016). These values can be provided through buttons (Kaplan et al., 2002; Suay and Chernova, 2011; Knox et al., 2013), speech (Kim and Scassellati, 2007; Grizou et al., 2013), gestures (Najar et al., 2020b), or electroencephalogram (EEG) signals (Grizou et al., 2014a).

Another form of evaluative feedback is to provide preferences between demonstrated trajectories (Christiano et al., 2017; Sadigh et al., 2017; Cui and Niekum, 2018). Instead of critiquing one single action or a sequence of actions, the teacher provides a ranking for demonstrated trajectories. The provided human preferences are then aggregated in order to infer the reward function. This form of evaluative feedback has been mainly investigated within the LfD community as an alternative to the standard Inverse Reinforcement Learning approach (IRL) (Ng and Russell, 2000), by relaxing the constraint for the teacher to provide demonstrations.

3.2. Interpreting Advice

The second step of the advice-taking process stipulates that advice needs to be converted into an internal representation. Predefining the meaning of advice by hand-coding the mapping between raw signals and their internal representation has been widely used in the literature (Clouse and Utgoff, 1992; Nicolescu and Mataric, 2003; Lockerd and Breazeal, 2004; Rosenstein et al., 2004; Rybski et al., 2007; Thomaz and Breazeal, 2007b; Chernova and Veloso, 2009; Tenorio-Gonzalez et al., 2010; Pradyot et al., 2012a; Cruz et al., 2015; Celemin and Ruiz-Del-Solar, 2019). However, this solution has many limitations. First, programming the meaning of raw advice signals for new tasks requires expert programming skills, which is not accessible to all human users. Second, it limits the possibility for different teachers to use their own preferred signals.

One way to address these limitations is to teach the system how to interpret the teacher's raw advice signals. This way, the system would be able to understand advice that can be expressed through natural language or non-verbal cues, without predetermining the meaning of each signal. In this case, we talk about learning with unlabeled teaching signals (Grizou et al., 2014b; Najar et al., 2020b). To achieve this goal, different approaches have been taken in the literature. **Table 2** summarizes the literature addressing the question of interpreting advice. We categorize them according to the type of advice, the communication channel, the interpretation method, and the inputs given to the system for interpretation.

3.2.1. Supervised Interpretation

Some methods relied on interpreters trained with supervised learning methods (Kate and Mooney, 2006; Zettlemoyer and Collins, 2009; Matuszek et al., 2013). For example, in Kuhlmann et al. (2004), the system was able to convert general instructions expressed in a constrained natural language into a formal representation using *if-then* rules, by using a parser that was previously trained with annotated data. In Pradyot et al. (2012b), two different models of contextual instructions were learned in the first place using Markov logic networks (MLN) (Domingos et al., 2016), and then used for guiding a learning agent in a later phase. The most likely interpretation was taken from the instruction model with the highest confidence. In Kim and Scassellati (2007), a binary classification of prosodic features was performed offline, before using it to convert evaluative feedback into a numerical reward signal for task learning.

3.2.2. Grounded Interpretation

More recent approaches take inspiration from the *grounded language acquisition* literature (Mooney, 2008) to learn a model that grounds the meaning of advice into concepts from the task. For example, general instructions expressed in natural language can be paired with demonstrations of the corresponding tasks to learn the mapping between low-level contextual instructions and their intended actions (Chen and Mooney, 2011; Tellex et al., 2011; Duvallet et al., 2013). In MacGlashan et al. (2014a), the authors proposed a model for grounding general 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 behaviors were then inferred using expectation maximization (EM) (Dempster et al., 1977). 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. The authors extended their model in MacGlashan et al. (2014b) to ground command meanings in reward functions using evaluative feedback instead of demonstrations.

In a similar work (Grizou et al., 2013), a robot learned to interpret both low-level contextual instructions and evaluative feedback, while inferring the task using an EM algorithm. Contextual advice was interactively provided through speech. As

TABLE 2 | Interpreting advice.

References	Advice	Channel	Method	Inputs
Kate and Mooney, 2006	GI	Text	SVM	Demonstration*
Kim and Scassellati, 2007	EFB	Speech	kNN	Binary EFB classes
Chen and Mooney, 2011	GLI	Text	SVM	Demonstration
Tellex et al., 2011	GHI	Text	Graphical model	Demonstration
Artzi and Zettlemoyer, 2013	GHI	Text	Perceptron	Rewards or demonstration + language model
Duvallet et al., 2013	GLI	Text	MCC	Demonstration + language model
Tellex et al., 2014	GHI	Text	Gradient descent	Demonstration
Pradyot et al., 2012b	CLI	Gestures	MLN	Demonstration*
Lopes et al., 2011	EFB and CFB	Simulation	IRL	EFB and CFB
Grizou et al., 2013	EFB or CLI	Speech	EM	Task models
Grizou et al., 2014b	EFB	EEG	EM	Task models
MacGlashan et al., 2014a	GHI	Text	EM	Task and language models
MacGlashan et al., 2014b	GHI	Text	EM	EFB + language model
Loftin et al., 2016	EFB	Buttons	EM	Task models
Branavan et al., 2009	GLI	Text	PGRL	Rewards
Branavan et al., 2010	GHI	Text	MB-PGRL	Rewards
Vogel and Jurafsky, 2010	GLI	Text	SARSA	Demonstration
Najar et al., 2015b	CLI	Simulation	XCS	Rewards
Najar et al., 2015a	CLI	Gestures	XCS	EFB
Najar et al., 2016	CLI	Gestures	Q-learning	EFB
Mathewson and Pilarski, 2016	CLI	EMG	ACRL	Rewards and/or EFB
Najar et al., 2020b	CLI	Gestures	ACRL	Rewards and/or EFB

*GI, General instruction; GLI, general low-level instruction; GHI, general high-level instruction; CLI, contextual low-level instruction; EFB, evaluative feedback; CFB, corrective feedback; SVM, Support Vector Machines; kNN, k-nearest neighbors; MCC, multi-class classification; MLN, Markov Logic Networks; IRL, Inverse Reinforcement Learning; PGRL, policy-gradient RL; MB-PGRL, model-based policy-gradient RL; ACRL, Actor-Critic RL. *The term demonstration here is taken in the general sense as a trajectory, not necessarily the optimal one.*

in MacGlashan et al. (2014b), 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 also used for interpreting evaluative feedback provided through EEG signals (Grizou et al., 2014b). In Lopes et al. (2011), a predefined set of known feedback signals, both evaluative and corrective, were used for interpreting additional signals with IRL.

3.2.3. RL-Based Interpretation

A different approach relies on RL for interpreting advice (Branavan et al., 2009, 2010; Vogel and Jurafsky, 2010; Mathewson and Pilarski, 2016; Najar et al., 2020b). In Branavan et al. (2009), the authors used a policy-gradient RL algorithm with a predefined reward function to interpret general low-level instructions for a software application. This model was extended in Branavan et al. (2010) to allow for the interpretation of high-level instructions by learning a model of the environment. In Vogel and Jurafsky (2010), a similar approach was used for interpreting general low-level instructions, in a path-following task, using the SARSA algorithm. The rewards were computed according to the deviation from a provided demonstration.

In Mathewson and Pilarski (2016), contextual low-level instructions were provided to a prosthetic robotic arm in the form of myoelectric control signals and interpreted using evaluative feedback with an Actor-Critic architecture. In Najar et al. (2015b), a model of contextual low-level instructions was built using the XCS algorithm (Butz and Wilson, 2001) in order to predict task rewards, and used simultaneously for speeding-up the learning process. This model was extended in Najar et al. (2015a) to predict action values instead of task rewards. In Najar et al. (2016), interpretation was based on evaluative feedback using the Q-learning algorithm. In Najar (2017), several methods for interpreting contextual low-level instructions were compared. Each contextual low-level instruction was defined as a *signal policy* representing a probability distribution over the action-space in the same way as an RL policy:

$$\pi(i) = \{\pi(i, a);\ a \in A\} = \{Pr(a|i);\ a \in A\}, \tag{11}$$

where i is an observed instruction signal, such as a pointing gesture or a vocal command. Two types of interpretation methods were proposed: batch and incremental. The main idea of batch interpretation methods is to derive a state policy for an instruction signal by combining the policies of every task state in which it has been observed. Different combination methods were investigated. The Bayes optimal solution derives the signal policy by marginalizing the state policies over all the states where the signal has been observed:

$$\pi(i, a) = Pr(a|i) = \sum_{s \in S} Pr(a|s) \times Pr(s|i) \tag{12}$$

$$= \sum_{s \in S} \pi(s, a) \times Pr(i|s) \times Pr(s)/Pr(i), \tag{13}$$

where $Pr(i|s), Pr(s)$, and $Pr(i)$ represent, respectively, the probability of observing the signal i in state s, the probability of being in state s and the probability of observing the signal i.

Other batch interpretation methods were inspired from ensemble methods (Wiering and van Hasselt, 2008), which have been classically used for combining the policies of different learning algorithms. These methods compute preferences $p(i, a)$ for each action, which are then transformed into a policy using the softmax distribution as in Equation (6). Boltzmann Multiplication consists in multiplying the policies:

$$p(i, a) = \prod_{s \in S; i^*(s)=i} \pi(s, a), \tag{14}$$

where $i^*(s)$ represents the instruction signal associated to the state s.

Boltzmann Addition consists in adding the policies:

$$p_t(i, a) = \sum_{s \in S; i^*(s)=i} \pi_t(s, a). \tag{15}$$

In Majority Voting, the most preferred interpretation for a signal i is the action that is optimal the most often over all its contingent states:

$$p(i, a) = \sum_{s \in S; i^*(s)=i} I(\pi^*(s), a), \tag{16}$$

where $I(x, y)$ is the indicator function that outputs 1 when $x = y$ and 0 otherwise.

In Rank Voting, the most preferred action for i is the one that has the highest cumulative ranking over all its contingent states:

$$p(i, a) = \sum_{s \in S; i^*(s)=i} R(s, a), \tag{17}$$

where $R(s, a)$ is the rank of action a in state s, such that if a_j and a_k denote two different actions and $\pi(s, a_j) \geq \pi(s, a_k)$ then $R(s, a_j) \geq R(s, a_k)$.

Incremental interpretation methods, on the other hand, incrementally update the meaning of each instruction signal using information from the task learning process such as the rewards, the TD error, or the policy gradient. With Reward-based Updating, instruction signals constitute the state space for an alternative MDP which is solved using a standard RL algorithm. This approach is similar to the one used in Branavan et al. (2010), Branavan et al. (2009), and Vogel and Jurafsky (2010). In Value-based Updating, the meaning of an instruction is updated with the same amount as the Q-values of its corresponding state:

$$\delta p_t(i, a_t) = \delta Q(s_t, a_t), \tag{18}$$

whereas in Policy-based Updating, it is updated using the policy update:

$$\delta \pi(i, a_t) = \delta \pi(s_t, a_t). \tag{19}$$

These methods were compared using both a reward function and evaluative feedback. Policy-based Updating presented the best compromise in terms of performance and computation cost.

3.3. Shaping With Advice

We can distinguish several strategies for integrating advice into an RL system, depending on which stage of the learning process is influenced by the advice. The overall RL process can be summarized as follows. First, the main source of information to an RL agent is the reward function. In value-based RL, the reward function is used for computing a value function, which is then used for deriving a policy. In policy-based RL, the policy is directly derived from the reward function without computing any value function. Finally, the policy is used for decision-making. Advice can be integrated into the learning process at any of these four different stages: the reward function, the value function, the policy, or the decision.

We qualify the methods used for integrating advice as shaping methods. In the literature, this term has been used exclusively for evaluative feedback, especially as a technique for providing extra-rewards. For example, we find different terminologies such as reward shaping (Tenorio-Gonzalez et al., 2010), interactive shaping (Knox and Stone, 2009), and policy shaping (Griffith et al., 2013; Cederborg et al., 2015). In some works, the term shaping is not even adopted (Loftin et al., 2016). In this survey, we generalize this term to all types of advice by considering the term shaping in its general meaning as influencing an RL agent toward a desired behavior. In this sense, all methods for integrating advice into an RL process are considered as shaping methods, especially that similar shaping patterns can be found across different categories of advice.

We distinguish four main strategies for integrating advice into an RL system: reward shaping, value shaping, policy shaping, and decision biasing, depending on the stage in which advice is integrated into the learning process (cf. **Table 3**). Orthogonal to this categorization, we distinguish model-free from model-based shaping strategies. In model-free shaping, the perceived advice is directly integrated into the learning process, whereas model-based shaping methods build a model of the teacher that is kept in parallel with the agent's own model of the task. Both models can be combined using several combination techniques that we review in this section.

3.3.1. Reward Shaping

Traditionally, reward shaping has been used as a technique for providing an RL agent with intermediate rewards to speed-up the learning process (Gullapalli and Barto, 1992; Mataric, 1994; Ng et al., 1999; Wiewiora, 2003). One way for providing intermediate rewards is to use evaluative feedback (Isbell et al., 2001; Thomaz et al., 2006; Tenorio-Gonzalez et al., 2010; Mathewson and Pilarski, 2016). 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. 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 using standard RL algorithms (cf. **Figure 2**) (Isbell et al., 2001; Thomaz et al., 2006; Tenorio-Gonzalez et al., 2010; Mathewson and Pilarski, 2016). This means that the effect of the provided feedback extends beyond the last performed action. When the RL agent has also access to a predefined reward

TABLE 3 | Shaping methods.

Shaping method	Model	Advice	References
Reward shaping	Model-free	Contextual instructions	Clouse and Utgoff, 1992
		Evaluative feedback	Isbell et al., 2001; Thomaz et al., 2006; Tenorio-Gonzalez et al., 2010; Mathewson and Pilarski, 2016
	Model-based	Contextual instructions	Najar et al., 2015b
		Evaluative feedback	Knox and Stone, 2010, 2011a, 2012b
Value shaping	Model-free	General instructions	Utgoff and Clouse, 1991; Maclin and Shavlik, 1996; Kuhlmann et al., 2004; Maclin et al., 2005a,b; Torrey et al., 2008
		Evaluative feedback	Dorigo and Colombetti, 1994; Colombetti et al., 1996; Najar et al., 2016
	Model-based	Contextual instructions	Najar et al., 2015a, 2016
		Evaluative feedback	Knox and Stone, 2010, 2011a, 2012b
Policy shaping	Model-free	Contextual instructions	Rosenstein et al., 2004
		Evaluative feedback	Ho et al., 2015; MacGlashan et al., 2017; Najar et al., 2020b
	Model-based	Contextual instructions	Pradyot et al., 2012b; Grizou et al., 2013; Najar et al., 2020b
		Evaluative feedback	Knox and Stone, 2010, 2011a, 2012b; Lopes et al., 2011; Griffith et al., 2013; Loftin et al., 2016
		Corrective feedback	Lopes et al., 2011
Decision biasing		Guidance	Thomaz and Breazeal, 2006; Suay and Chernova, 2011
		Contextual instructions	Nicolescu and Mataric, 2003; Rosenstein et al., 2004; Rybski et al., 2007; Thomaz and Breazeal, 2007b; Tenorio-Gonzalez et al., 2010; Cruz et al., 2015

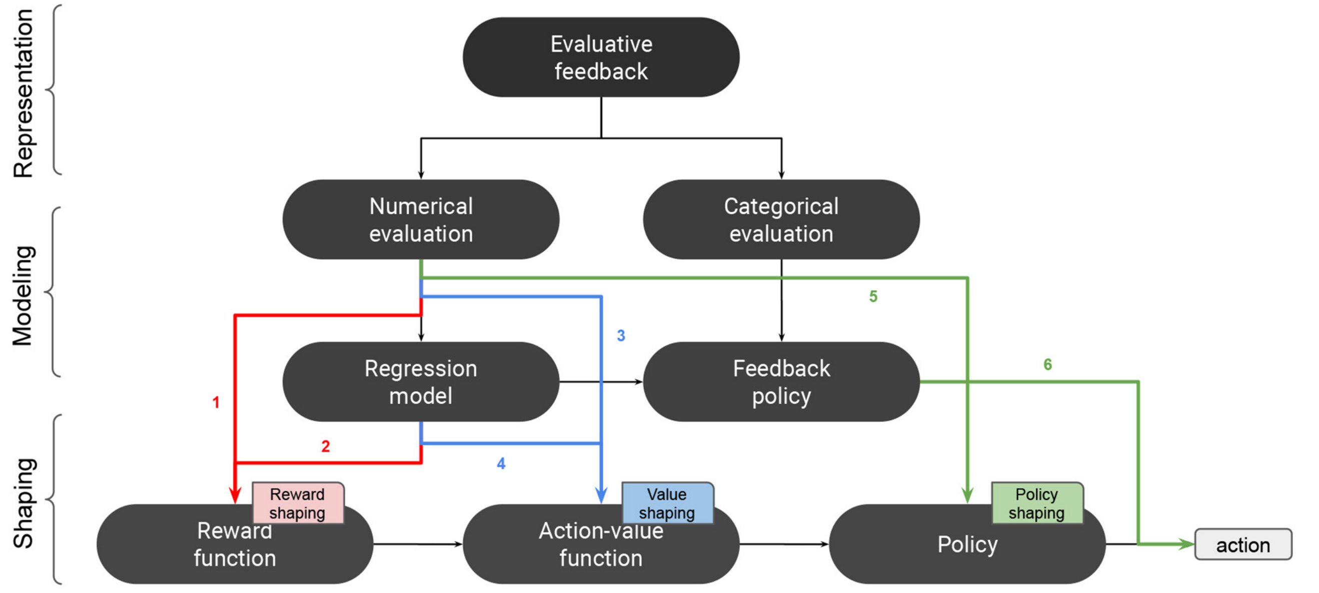

FIGURE 2 | Shaping with evaluative feedback. 1: model-free reward shaping. 2: model-based reward shaping. 3: model-free value shaping. 4: model-based value shaping. 5: model-free policy shaping. 6: model-based policy shaping.

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. This way of shaping with is model-free in that the numerical values provided by the human teacher are directly used for augmenting the reward function.

Reward shaping can also be performed with instructions (cf. **Figure 3**). For example, in Clouse and Utgoff (1992), *contextual instructions* were integrated into an RL algorithm by positively reinforcing the proposed actions in a model-free fashion.

Other works considered building an intermediate model of human rewards to perform model-based reward shaping. In the TAMER framework (Knox and Stone, 2009), evaluative feedback was converted into rewards and used for computing a regression model $\hat{H}$, called the "*Human Reinforcement Function*." This model predicted the amount of rewards $\hat{H}(s, a)$ that the human provided for each state-action pair (s, a). Knox and Stone (2010, 2011a, 2012b) 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:

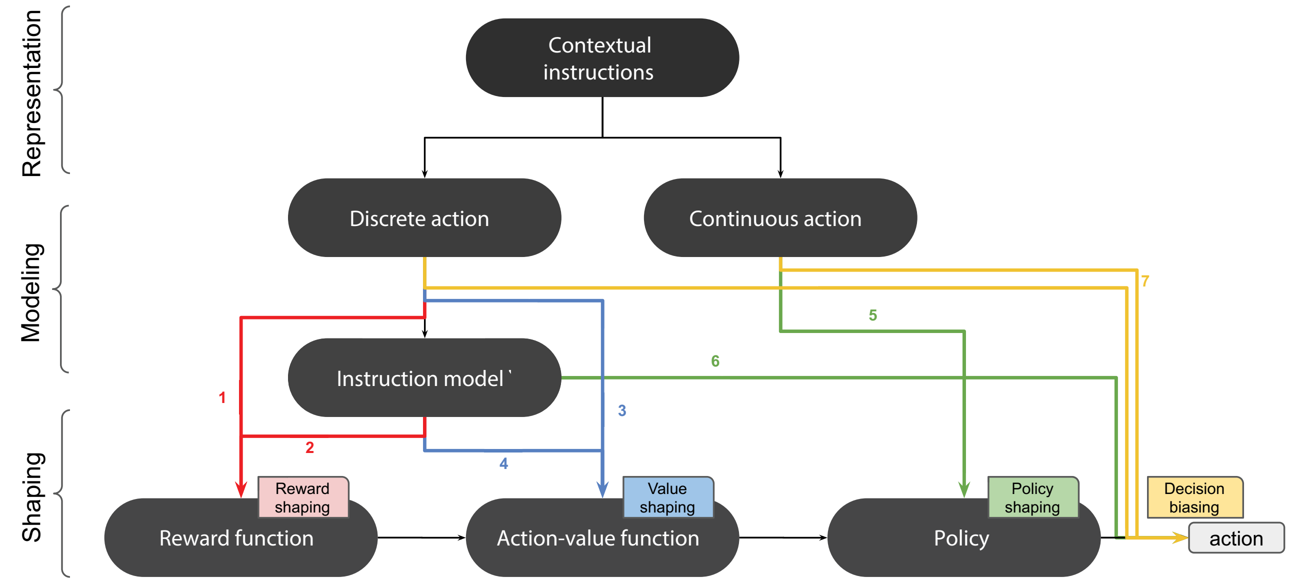

FIGURE 3 | Shaping with contextual instructions. 1: model-free reward shaping. 2: model-based reward shaping. 3: model-free value shaping. 4: model-based value shaping. 5: model-free policy shaping. 6: model-based policy shaping. 7: decision biasing.

$$R'(s,a) = R(s,a) + \beta * \hat{H}(s,a). \tag{20}$$

Model-based reward shaping can also be performed with *contextual instructions*. In Najar et al. (2015b), a human teacher provided social cues to humanoid robot about the next action to perform. A model of these cues was built in order to predict task rewards and used simultaneously for reward shaping.

3.3.2. Value Shaping

While investigating reward shaping, some authors pointed out the fundamental difference that exists between immediate and delayed rewards (Dorigo and Colombetti, 1994; Colombetti et al., 1996; Knox and Stone, 2012a). Particularly, they considered evaluative feedback as an immediate information about the value of an action, as opposed to standard MDP rewards (Ho et al., 2017). For example, in Dorigo and Colombetti (1994), the authors used a *myopic discounting* scheme by setting the discount factor γ to zero. In this way, evaluative feedback constituted *immediate reinforcements in response to the actions of the learning agent*, which comes to consider rewards as equivalent to action values. So, value shaping constitutes an alternative to reward shaping by considering evaluative feedback as an action-preference function. The work of Dorigo and Colombetti (1994) was one of the earliest examples of model-free value-shaping. Another example can be found in Najar et al. (2016), where evaluative feedback was directly used for updating a robot's action values with *myopic discounting*.

Model-free value shaping can also be done with *general advice*. For example, *if-then* rules can be incorporated into a kernel-based regression model by using the Knowledge-Based Kernel Regression (KBKR) method (Mangasarian et al., 2004). This method was used for integrating *general constraints* into the value function of a SARSA agent using Support Vector Regression for value function approximation (Maclin et al., 2005b). 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 Maclin et al. (2005a) 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 Torrey et al. (2008). Another possibility is given by the Knowledge-Based Neural Network (KBANN) method, which allows incorporating knowledge expressed in the form of *if-then* rules into a neural network (Towell and Shavlik, 1994). This method was used in RATLE, an advice-taking system based on Q-learning that used a neural network to approximate its Q-function (Maclin and Shavlik, 1996). *General instructions* written in the form of *if-then* rules and *while-repeat* loops were incorporated into the Q-function using an extension of KBANN method. In Kuhlmann et al. (2004), a SARSA agent was augmented with an *Advice Unit* that computed additional action values. *General instructions* were expressed in a specific formal language in the form of *if-then* rules. Each time a rule was activated in a given state, the value of the corresponding action was increased or decreased by a constant in the Advice Unit, depending on whether the rule advised for or against the action. These values were then used for augmenting the values generated by the agent's value function approximator.

Model-based value shaping with evaluative feedback has been investigated by Knox and Stone (2012a) by comparing different discount factors for the *human reinforcement function* $\hat{H}$. The authors demonstrated that setting the discount factor to zero was better suited, which came to consider $\hat{H}$ as an action-value function more than a reward function.[1] The numerical representation of evaluative feedback is used for modifying the Q-function rather than the reward function. One of the shaping methods that they proposed, Q-Augmentation (Knox and Stone, 2010, 2011a, 2012b), uses the human reinforcement function $\hat{H}$ for augmenting the MDP Q-function using:

$$Q'(s,a) = Q(s,a) + \beta * \hat{H}(s,a), \tag{21}$$

where β is the same decaying weight factor as in Equation (20).

Model-based value shaping can also be done with *contextual instructions*. In Najar et al. (2015a) and Najar et al. (2016), a robot built a model of contextual instructions in order to predict action values, which were used in turn for updating the value function.

3.3.3. Policy Shaping

The third shaping strategy is to integrate the advice directly into the agent's policy. Examples of model-free policy shaping with evaluative feedback can be found in MacGlashan et al. (2017) and Najar et al. (2020b). In both methods, evaluative feedback was used for updating the actor of an Actor-Critic architecture. In MacGlashan et al. (2017), the update term was scaled by the gradient of the policy:

$$w \leftarrow w + \alpha \nabla_w \ln \pi_w(a_t|s_t) f_t, \tag{22}$$

where f_t is the feedback provided at time t. In Najar et al. (2020b), however, the authors did not consider a multiplying factor for evaluative feedback:

$$w \leftarrow w + \alpha f_t. \tag{23}$$

Model-free policy shaping with *contextual instructions* was considered in Rosenstein et al. (2004), in the context of an Actor-Critic architecture, where the error between the instruction and the *actor*'s decision was used as an additional term to the TD error for updating the *actor*'s parameters:

$$w \leftarrow w + \alpha[k\delta_t(a^E - a^A) + (1-k)(a^S - a^A)]\nabla_w \pi^A(s), \tag{24}$$

where a^E is the actor's exploratory action, a^A is its deterministic action, a^S is the teacher's action, $\pi^A(s)$ is the actor's deterministic policy, and k is an interpolation parameter.

Knox and Stone proposed two model-based policy shaping methods for evaluative feedback (Knox and Stone, 2010, 2011a, 2012b). Action Biasing uses the same equation as Q-Augmentation (Equation 21) but only in decision-making, so that the agent's Q-function is not modified:

$$a^* = argmax_a[Q(s,a) + \beta * \hat{H}(s,a)]. \tag{25}$$

The second method, Control Sharing, arbitrates between the decisions of both value functions based on a probability criterion. A parameter β is used as a threshold for determining the probability of selecting the decision according to $\hat{H}$:

$$Pr(a = argmax_a[\hat{H}(s,a)]) = min(\beta, 1). \tag{26}$$

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

Other model-based policy shaping methods do not convert evaluative feedback into a scalar but into a categorical information (Lopes et al., 2011; Griffith et al., 2013; Loftin et al., 2016). The distribution of provided feedback is used within a Bayesian framework in order to derive a policy. The method proposed in Griffith et al. (2013) 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: $\pi \propto \pi_R \times \pi_F$, where π_R is the policy derived from the reward function and π_F the policy derived from evaluative feedback. In Lopes et al. (2011), both evaluative and corrective feedback were considered under a Bayesian IRL perspective.

Model-based policy shaping can also be performed with *contextual instructions*. For example, in Pradyot et al. (2012b), 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:

$$\kappa_\pi(s) = \max_{a \in A} \pi(s,a) - \max_{b \in A; b \neq a} \pi(s,b). \tag{27}$$

The same arbitration criterion was used in Najar et al. (2020b) to decide between the outputs of an Instruction Model and a Task Model.

3.3.4. Decision Biasing

In the previous paragraphs, we said that policy shaping methods can be either model-free, by directly modifying the agent's policy, or model-based, by building a model that is used at decision-time to bias the output of the policy. A different approach consists of using advice to directly bias the output of the policy at decision-time without corrupting the policy nor modeling the advice. This strategy, that we call decision biasing, is the simplest way of using advice as it only biases the exploration strategy of the agent, without modifying any of its internal variables. In this case, learning is done indirectly by experiencing the effects of following the advice.

This strategy has been mainly used in the literature with guidance and contextual instructions. For example, in Suay and Chernova (2011) and Thomaz and Breazeal (2006) guidance reduces the set of actions that the agent can perform at a given time-step.

Contextual instructions can also be used for guiding a robot along the learning process (Thomaz and Breazeal, 2007b; Tenorio-Gonzalez et al., 2010; Cruz et al., 2015). For example, in Nicolescu and Mataric (2003) and Rybski et al. (2007), an LfD system was augmented with verbal instructions in order to make the robot perform some actions during the demonstrations. In Rosenstein et al. (2004), in addition to model-free policy shaping, the provided instruction was also used for decision

[1] The authors proposed another mechanism for handling temporal credit assignment in order to alleviate the effect of highly dynamical tasks (Knox and Stone, 2009). In their system, human-generated rewards were distributed backward to previously performed actions within a fixed time window.

biasing. 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:

$$a \leftarrow ka^E + (1-k)a^S, \tag{28}$$

where a^E is the actor's exploratory action, a^S the supervisor's action, and k an interpolation parameter.

4. DISCUSSION

In this section, we first discuss the difference between the various forms of advice introduced in section 3.1. We then discuss the approaches presented in sections 3.2 and 3.3. Finally, we open some perspectives toward a unified view of interactive learning methods.

4.1. Comparing Different Forms of Advice

When designing an advice-taking system, one may ask which type of advice is best suited (Suay et al., 2012). In this survey, we categorized different forms of advice according to how they are provided to the system. Even though the same interpretation and shaping methods can be applied to different categories of advice, each form of advice 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 about the policy, so the learning process mainly depends on the human teacher.

This aspect has been described in the literature as the guidance-exploration spectrum (Breazeal and Thomaz, 2008). In section 3.1, 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 (**Figure 4**). In the following paragraphs, we compare different forms of advice along this spectrum, by putting them into perspective with respect to other learning schemes such as autonomous learning and LfD.

4.1.1. Autonomous Learning

At one end of the exploration-control spectrum, autonomous learning methods assume that the robot is able to autonomously evaluate its performance on the task, through a predefined evaluation function, such as a reward function. The main advantage of this approach is the autonomy of the learning process. The evaluation function being integrated on board, the robot is able to optimize its behavior without requiring help from a supervisor.

However, this approach has some limitations when deployed in real-world settings. First, it is often hard to design, especially in complex environments, an appropriate evaluation function that could anticipate all aspects of a task (Kober et al., 2013). Second, this approach relies on autonomous exploration, which raises some practical challenges. For example, exploring the space of behaviors makes the convergence of the learning process very slow, which limits the feasibility of such approach in complex problems. Also, autonomous exploration may lead to dangerous situations. So, safety is an important issue that has to be considered when designing autonomous learning systems (Garcia and Fernandez, 2015).

4.1.2. Evaluative Feedback

Evaluative feedback constitutes another way to evaluate the agent's performance that has many advantages over predefined reward functions. First, like all other types of teaching signals, it can alleviate the limitations of autonomous learning, by allowing faster convergence rates and safer exploration. Whether it is represented as categorical information (Griffith et al., 2013) or as immediate rewards (Dorigo and Colombetti, 1994), it provides a more straightforward evaluation of the policy, as it directly informs about the optimality of the performed action (Ho et al., 2015). Second, from an engineering point of view, evaluative feedback is generally easier to implement than a reward function. If designing a proper reward function can be challenging in practice, evaluative feedback generally takes the form of binary values that can be easily implemented (Knox et al., 2013).

Nevertheless, the informativeness of evaluative feedback is still limited, as it is only given as a reaction to the agent's actions, without communicating 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 agent 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 (Thomaz et al., 2006). 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 (Thomaz and Breazeal, 2006). This reflects the limitations of evaluative feedback and demonstrates that human teachers also need to provide guidance.

4.1.3. Corrective Feedback

One possibility for improving the feedback channel is to allow for corrections and refinements (Thomaz and Breazeal, 2007a). Corrective instructions improve the informativeness of evaluative feedback by allowing the teacher to inform the agent about the optimal action (Celemin and Ruiz-Del-Solar, 2019). Being also reactive to the agent's actions, they still require exploration. However, they prevent the agent from waiting until it tries the correct action by its own, so they require less exploration compared to evaluative feedback.

On the other hand, corrective instructions require more engineering efforts than evaluative feedback, as they are generally more than a binary information. Since they operate over the action space, they require from the system designer to encode the mapping between contextual instruction signals and their corresponding actions.

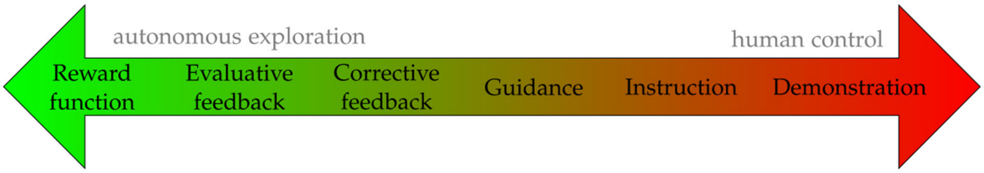

FIGURE 4 | Exploration-control spectrum. As we move to the right, teaching signals inform more directly about the optimal policy and provide more control to the human over the learning process.

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 (Chernova and Veloso, 2009). 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 (cf. the paragraph about demonstrations below).

4.1.4. Guidance

The experiments of Thomaz and Breazeal have shown that human teachers want to provide guidance (Thomaz and Breazeal, 2006). In contrast to feedback, guidance allows the agent to be informed about future aspects of the task, such as the next action to perform (contextual instruction) (Cruz et al., 2015), an interesting region to explore (demonstration) (Subramanian et al., 2016) or a set of interesting actions to try (guidance) (Thomaz and Breazeal, 2006).

Even though guidance requires less exploration compared to feedback by informing about future aspects of the task, the control over the learning process is exerted indirectly through decision biasing (cf. section 3.3). By performing the communicated guidance, the agent does not directly integrate this information as being the optimal behavior. Instead, it will be able to learn only through the experienced effects, for example by receiving a reward. So guidance is only about limiting exploration, without providing full control over the learning process, as it still depends on the evaluation of the performed actions.

4.1.5. 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 more directly into the learning process via reward shaping, value shaping, or policy shaping.

In section 3.1, we presented two main strategies for providing instructions: providing general instructions in the form of *if-then* rules, or interactively providing contextual 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. Even though in the literature they are generally provided offline prior to the learning process, there is no reason they cannot be integrated at any moment of the task. For example, in works like (Kuhlmann et al., 2004), we can imagine that different rules being activated and deactivated at different moments of the task. Their integration into the learning process will only depend on the validity of their conditions, not on the moment of their activation by the teacher. This puts less interactive load on the teacher as he/she does not need to stay concentrated in order to provide the correct information at the right moment.

General instructions also 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 (Kuhlmann et al., 2004). Furthermore, they require from the teacher to know about the robot's sensors and effectors in order to correctly express the desired behaviors. So, formulating rules requires expertise about the task, the environment, and the robot. Second, general instructions can be difficult to communicate. They require either expert programming skills from the teacher or sophisticated natural language understanding capabilities from the agent.

Contextual instructions, on the other hand, communicate a less sophisticated message at a time, which makes them easier to formulate and to provide. Compared to general instructions, they only inform about the next 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. So they provide more control over the learning process with respect to general instructions. However, this can be challenging in highly dynamical tasks, as the teacher needs a lapse of time to communicate each contextual instruction.

4.1.6. Demonstration

Formally, a demonstration is defined as a sequence of state-action pairs representing a trajectory in the task space (Argall et al., 2009). So, from a strictly formal view, a demonstration is not very different from a general instruction providing a sequence of actions to perform (Branavan et al., 2009; Vogel and Jurafsky, 2010). The only difference is the sequence of states that the robot is supposed to experience. In many LfD settings, such as teleoperation (Abbeel et al., 2010) and kinesthetic teaching (Akgun et al., 2012), the states visited by the robot are controlled by the human. So, controlling a robot through these devices can be seen as providing a continuous stream of contextual instructions: the commands sent via the joystick or the forces exerted on the robot's kinesthetic device. So the difference between action plans and demonstrations provided under these settings goes beyond their formal definitions as sequences of actions or state-action pairs.

The main difference between demonstrations and general instructions (actually, all forms of advice) is that demonstrations provide 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 is systematically executed. With instructions, however, the robot is in control of its own actions. Even though the instruction can be integrated into the learning process, via any shaping methods, the robot is still free to execute or not the communicated action.

One downside of this control is that 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, like with contextual instructions.

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. Comparing Different Interpretation Methods

In section 3.2, we presented three main approaches for interpreting advice. The classical approach, supervised interpretation, relies on annotated data for training linguistic parsers. Even though this approach can be effective for building systems that are able to take into account natural language advice, they come at the cost of constituting large corpora of language-to-command alignments.

The second approach, grounded interpretation, relaxes this constraint by relying on examples of task executions instead of perfectly aligned commands. This approach is easier to implement by taking advantage of crowd-sourcing platforms like Amazon Mechanical Turk. Also, the annotation process is facilitated as it can be performed in the reverse order compared to the standard approach. First, various demonstrations of the task are collected, for example in the form of videos (Tellex et al., 2011, 2014). Then, each demonstration is associated to a general instruction. Even though this approach is more affordable than standard language-to-command annotation, it still comes at the cost of providing demonstrations, which can be challenging to provide in some contexts, as discussed in the previous section.

The third approach, RL-based interpretation, relaxes these constraints even more by relying only on a predefined performance criterion to guide the interpretation process (Branavan et al., 2009, 2010). Some intermediate methods also exists, for example by deriving a reward function from demonstrations and then using an RL algorithm to interpret advice (Vogel and Jurafsky, 2010; Tellex et al., 2014). Given that reward functions can also be challenging to design, some methods rely on predefined advice for interpreting other advice (Lopes et al., 2011; Mathewson and Pilarski, 2016; Najar et al., 2016), or a combination of both advice and reward functions (Mathewson and Pilarski, 2016; Najar et al., 2020b).

Orthogonal to the difference between supervised, grounded, and RL-based interpretation methods, we can distinguish two different strategies for teaching the system how to interpret unlabeled advice. The first strategy is to teach the system how to interpret advice without using it in parallel for task learning. For example, a human can teach an agent how to interpret continuous streams of contextual instructions by using evaluative feedback (Mathewson and Pilarski, 2016). Here, the main task for the agent is to learn how interpret unlabeled instructions, not to use them for learning another task. Another example is when the agent is first provided with general instructions, either in the form of *if-then* rules or action plans, and then teaching it how to interpret these instructions using either demonstrations (Tellex et al., 2011; MacGlashan et al., 2014a), evaluative feedback (MacGlashan et al., 2014b) or a predefined reward function (Branavan et al., 2009, 2010; Vogel and Jurafsky, 2010). In this case, even though the agent is allowed to interact with its environment, the main task is still to learn how to interpret advice, not to use it for task learning.

The second strategy consists of guiding a task-learning process by interactively providing the agent with unlabeled contextual advice. In this case, the agent learns how to interpret advice at the same time as it learns to perform the task (Grizou et al., 2013; Najar et al., 2020b). For example, in Grizou et al. (2013), the robot is provided with a set of hypotheses about possible tasks and advice meanings. The robot then infers the task and advice meanings that are the most coherent with each other and with the history of observed advice signals. In Najar et al. (2020b), task rewards are used for grounding the meaning of contextual instructions, which are used in turn for speeding-up the task-learning process.

It is important to understand the difference between these two strategies. First, when the agent learns how to interpret advice while using it for task learning, we must think about which shaping method to use for integrating the interpreted advice into the task-learning process (cf. section 3.3). Second, when the goal is only to interpret advice, there is no challenge about the optimality nor the sparsity of the unlabeled advice.

With the first strategy, advice cannot be erroneous as it constitutes the reference for the interpretation process. Even though the methods implementing this strategy do not explicitly assume perfect advice, the robustness of the interpretation methods against inconsistent advice is not systematically investigated. When advice is also used for task learning, however, we need to take into account whether or not advice is correct with respect to the target task. For example, in Grizou et al. (2013), the authors report the performance of their system under erroneous evaluative feedback. In Najar et al. (2020b), the system is evaluated in simulation against various levels of error for both evaluative feedback and contextual instructions. Also with the first strategy, advice signals cannot be sparse since they constitute the state-space of the interpretation process. For instance, the standard RL methods that have been used for interpreting general instructions (Branavan et al., 2009, 2010; Vogel and Jurafsky, 2010) cannot be used for interpreting sparse contextual instructions. In these methods, instructions constitute the state-space of an MDP over which the RL algorithm is deployed, so they need to be instantiated on every time-step. This problem has been addressed in Najar et al. (2020b), where the system was able to interpret sporadic contextual instructions by using the TD error of the task-learning process.

4.3. Comparing Different Shaping Methods

In section 3.3, we presented different methods for integrating advice into an RL process: reward shaping, value shaping, policy shaping, and decision biasing. The standard approach, reward shaping, has been effective in many domains (Clouse and Utgoff, 1992; Isbell et al., 2001; Thomaz et al., 2006; Tenorio-Gonzalez et al., 2010; Mathewson and Pilarski, 2016). However, this way of providing intermediate rewards has been shown to cause sub-optimal behaviors such as positive circuits (Knox and Stone, 2012a; Ho et al., 2015). Even though these effects have been mainly studied under the scope of evaluative feedback, they can also be extended to other forms of advice such as instructions, since the positive circuits problem is inherent to the reward shaping scheme regardless of the source of the rewards (Mahadevan and Connell, 1992; Randlov and Alstrom, 1998; Ng et al., 1999; Wiewiora, 2003).

Consequently, many authors considered value shaping as an alternative solution to reward shaping (Knox and Stone, 2012b; Ho et al., 2017). However, when comparing different shaping methods for evaluative feedback, 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"* (Knox and Stone, 2012b). In fact, this can be explained by the specificity of the Q-function with respect to other preference functions. Unlike other preference functions (e.g., Advantage function, Harmon et al., 1994), a Q-function also informs about the proximity to the goal via temporal discounting. Contextual advice such as evaluative feedback and contextual instructions, however, only inform about local preferences like the last or the next action, without including such information (Ho et al., 2015). So, like reward shaping, value shaping with contextual advice may also lead to convergence problems.

Overall, policy shaping methods show better performance compared to other shaping methods (Knox and Stone, 2012b; Griffith et al., 2013; Ho et al., 2015). 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.

4.4. Toward a Unified View

Overall, all forms of advice overcome the limitations of autonomous learning by providing more control over the learning process. Since more control comes at the cost of more interaction load, the autonomy of the learning process is important for minimizing the burden on the human teacher (Najar et al., 2020b). Consequently, many advice-taking systems combine different learning modalities in order to balance between autonomy and control. For example, RL can be augmented with evaluative feedback (Judah et al., 2010; Sridharan, 2011; Knox and Stone, 2012b), corrective feedback (Celemin et al., 2019), instructions (Maclin and Shavlik, 1996; Kuhlmann et al., 2004; Rosenstein et al., 2004; Pradyot et al., 2012b), instructions and evaluative feedback (Najar et al., 2020b), demonstrations (Taylor et al., 2011; Subramanian et al., 2016), demonstrations and evaluative feedback (Leon et al., 2011), or demonstrations, evaluative feedback, and instructions (Tenorio-Gonzalez et al., 2010). Demonstrations can be augmented with corrective feedback (Chernova and Veloso, 2009; Argall et al., 2011), instructions (Rybski et al., 2007), instructions and feedback, both evaluative and corrective (Nicolescu and Mataric, 2003), or with prior RL (Syed and Schapire, 2007). In Waytowich et al. (2018), the authors proposed a framework for combining different learning modalities in a principled way. The system could balance autonomy and human control by switching from demonstration to guidance to evaluative feedback using a set of predefined metrics such as performance.

Integrating different forms of advice into one single and unified formalism remains an active research question. So far, different forms of advice have been mainly investigated separately by different communities. For example, some shaping methods have been designed exclusively for evaluative feedback and were not tested with other forms of advice such as contextual instructions, and the converse is also true. In this survey, we extracted several aspects that were shared across different forms of advice. Regardless of the type of advice, we must ask the same computational questions as we go through the same overall process (**Figure 5**): First, we must think about how advice will be represented and whether its meaning will be predetermined or interpreted by the learning agent. Second, we must decide whether to aggregate advice into a model, or directly use it for influencing the learning process (model-based vs. model-free shaping). Finally, we must choose a shaping method for integrating advice (or its model) into the learning process. From this perspective, all shaping methods that were specifically designed for evaluative feedback could also be used for instructions and *vice versa*. For example, all the methods proposed by Knox and Stone for learning from evaluative

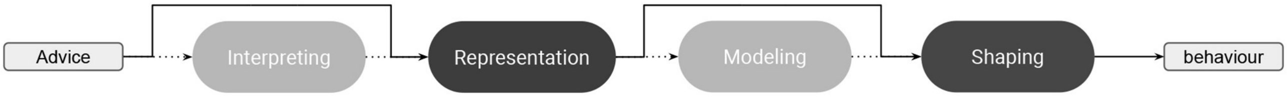

FIGURE 5 | Shaping with advice, a unified view. When advice is provided to the learning agent, it has first to be encoded into an appropriate representation. If the mapping between teaching signals and their corresponding internal representation is not predetermined, then advice has to be interpreted 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 gray.

feedback (Knox and Stone, 2010, 2011a, 2012b), can be recycled for learning from instructions. Similarly, the confidence criterion used in Pradyot et al. (2012b) for learning from contextual instructions constitutes another Control Sharing mechanism, similar to the one proposed in Knox and Stone (2010), Knox and Stone (2011a), and Knox and Stone (2012b) for learning from evaluative feedback.

It is also interesting to think about the relationship between interpretation and shaping. For example, we can notice the similarity between interpretation and shaping methods. In Section 3.2, we mentioned that some interpretation methods relying on the task-learning process can be either reward-based, value-based, or policy-based. This scheme is reminiscent of the different shaping methods: reward shaping, value shaping, and policy shaping. For instance, the policy shaping method proposed in Griffith et al. (2013) for combining evaluative feedback with a reward function is mathematically equivalent to the Boltzmann Multiplication method used in Najar (2017) for interpreting contextual instructions. So by extension, the other ensemble methods that have been used for interpreting contextual instructions could also be used for shaping. We also note that the confidence criterion in Pradyot et al. (2012b) was used for both interpreting instructions and policy shaping. So, we can think of the relationship between shaping and interpretation as a reciprocal influence scheme, where advice can be interpreted from the task-learning process in a reward-based, value-based, or a policy-based way, and in turn can influence the learning process in a reward-based, value-based, or policy-based shaping way (Najar, 2017). This view contrasts with the standard flow of the advice-taking process, where advice is interpreted before being integrated into the learning process (Hayes-Roth et al., 1981). In fact in many works, interpretation and shaping happen simultaneously, sometimes by using the same mechanisms (Pradyot and Ravindran, 2011; Najar et al., 2020a).

Under this perspective, we can extend the similarity between all forms of advice to include also other sources of information such as demonstration and reward functions. At the end, even though these signals can sometimes contradict each other, they globally inform about one same thing, i.e., the task (Cederborg and Oudeyer, 2014). Until recently, advice and demonstration have been mainly considered as two complementary but distinct approaches, i.e., communication vs. action (Dillmann et al., 2000; Argall et al., 2008; Knox and Stone, 2009, 2011b; Judah et al., 2010). However, these two approaches share many common aspects. For example, the counterpart of interpreting advice in the LfD literature is the correspondence problem, which is the question of how to map the teacher's states an actions into the agent's own states and actions. With advice, we also have a correspondence problem that consists of interpreting the raw advice signals. So, we can consider a more general correspondence problem that consists of interpreting raw teaching signals, independently from their nature. 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 agent 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 (Adornetti and Ferretti, 2015). In this view, language is grounded in action, just like imitation (Corballis, 2010). 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 (Kohler et al., 2002), 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 (Aziz-Zadeh et al., 2006).

So, one challenging question is 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. In Cederborg and Oudeyer (2014), the authors proposed a mathematical framework for learning from different sources of information. The main idea is to relax the assumptions about the meaning of teaching signals by taking advantage of the coherence between the different sources of information. When comparing demonstrations with instructions, we mentioned that some demonstration settings could be considered as a way of providing continuous streams of contextual instructions, with the subtle difference that demonstrations are systematically executed by the robot. Considering this analogy, the growing literature about interpreting instructions (Branavan et al., 2010; Vogel and Jurafsky, 2010; Grizou et al., 2013; Najar et al., 2020b) could provide insights for designing new ways of solving the correspondence problem in imitation.

Unifying all types of teaching signals under the same view is a relatively recent research question (Cederborg and Oudeyer, 2014; Waytowich et al., 2018), and this survey aims at pushing toward this direction by clarifying some of the concepts used in the interactive learning literature and highlighting the similarities that exist between different approaches. 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 addressed by the Cognitive Neuroscience community (Biele et al., 2011; Najar et al., 2020a; Olsson et al., 2020). We hope this survey will contribute in bridging the gap between both communities.

5. CONCLUSION

In this paper, we provided an overview of the existing methods for integrating human advice into an RL process. We first proposed a taxonomy of the different forms of advice that can be provided to a learning agent. We then described different methods that can be used for interpreting advice, and for integrating it into the learning process. Finally, we discussed the different approaches and opened some perspectives toward a unified view of interactive learning methods.

AUTHOR CONTRIBUTIONS

AN wrote the manuscript. MC supervised the project. All authors contributed to the article and approved the submitted version.

ACKNOWLEDGMENTS

This work was supported by the Romeo2 project. This manuscript has been released as a pre-print at arXiv (Najar and Chetouani, 2020).

REFERENCES

Abbeel, P., Coates, A., and Ng, A. Y. (2010). Autonomous helicopter aerobatics through apprenticeship learning. *Int. J. Robot. Res.* 29, 1608–1639. doi: 10.1177/0278364910371999

Adornetti, I., and Ferretti, F. (2015). The pragmatic foundations of communication: an action-oriented model of the origin of language. *Theor. Histor. Sci.* 11, 63–80. doi: 10.12775/ths-2014-004

Akgun, B., Cakmak, M., Yoo, J. W., and Thomaz, A. L. (2012). "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* (New York, NY: ACM), 391–398. doi: 10.1145/2157689.2157815

Argall, B. D., Browning, B., and Veloso, M. (2008). "Learning robot motion control with demonstration and advice-operators," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Nice: IEEE), 399–404. doi: 10.1109/IROS.2008.4651020

Argall, B. D., Browning, B., and Veloso, M. M. (2011). Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. *Robot. Auton. Syst.* 59, 243–255. doi: 10.1016/j.robot.2010.11.004

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robot. Auton. Syst.* 57, 469–483. doi: 10.1016/j.robot.2008.10.024

Artzi, Y., and Zettlemoyer, L. (2013). Weakly supervised learning of semantic parsers for mapping instructions to actions. *Trans. Assoc. Comput. Linguist.* 1, 49–62. doi: 10.1162/tacl_a_00209

Atkeson, C. G., and Schaal, S. (1997). "Learning tasks from a single demonstration," in *Proceedings of International Conference on Robotics and Automation* (Albuquerque, NM), 1706–1712. doi: 10.1109/ROBOT.1997.614389

Aziz-Zadeh, L., Wilson, S. M., Rizzolatti, G., and Iacoboni, M. (2006). Congruent embodied representations for visually presented actions and linguistic phrases describing actions. *Curr. Biol.* 16, 1818–1823. doi: 10.1016/j.cub.2006.07.060

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybernet.* 13, 834–846. doi: 10.1109/TSMC.1983.6313077

Biele, G., Rieskamp, J., Krugel, L. K., and Heekeren, H. R. (2011). The neural basis of following advice. *PLoS Biol.* 9:e1001089. doi: 10.1371/journal.pbio.1001089

Branavan, S. R. K., Chen, H., Zettlemoyer, L. S., and Barzilay, R. (2009). "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* (Stroudsburg, PA: Association for Computational Linguistics), 82–90. doi: 10.3115/1687878.1687892

Branavan, S. R. K., Zettlemoyer, L. S., and Barzilay, R. (2010). "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* (Stroudsburg, PA: Association for Computational Linguistics), 1268–1277.

Breazeal, C., and Thomaz, A. L. (2008). "Learning from human teachers with socially guided exploration," in *2008 IEEE International Conference on Robotics and Automation* (Pasadena, CA), 3539–3544. doi: 10.1109/ROBOT.2008.4543752

Butz, M. V., and Wilson, S. W. (2001). "An algorithmic description of XCS," in *Advances in Learning Classifier Systems: Third International Workshop, IWLCS 2000*, eds P. Luca Lanzi, W. Stolzmann, and S. W. Wilson (Paris), 253–272. doi: 10.1007/3-540-44640-0

Cederborg, T., Grover, I., Isbell, C. L., and Thomaz, A. L. (2015). "Policy shaping with human teachers," in *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15* (Buenos Aires: AAAI Press), 3366–3372.

Cederborg, T., and Oudeyer, P.-Y. (2014). A social learning formalism for learners trying to figure out what a teacher wants them to do. *Paladyn J. Behav. Robot.* 5, 64–99. doi: 10.2478/pjbr-2014-0005

Celemin, C., Maeda, G., del Solar, J. R., Peters, J., and Kober, J. (2019). Reinforcement learning of motor skills using policy search and human corrective advice. *Int. J. Robot. Res.* 38, 1560–1580. doi: 10.1177/0278364919871998

Celemin, C., and Ruiz-Del-Solar, J. (2019). An interactive framework for learning continuous actions policies based on corrective feedback. *J. Intell. Robot. Syst.* 95, 77–97. doi: 10.1007/s10846-018-0839-z

Chen, D. L., and Mooney, R. J. (2011). "Learning to interpret natural language navigation instructions from observations," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI'11* (San Francisco, CA: AAAI Press), 859–865.

Chernova, S., and Thomaz, A. L. (2014). Robot learning from human teachers. *Synthesis Lect. Artif. Intell. Mach. Learn.* 8, 1–121. doi: 10.2200/S00568ED1V01Y201402AIM028

Chernova, S., and Veloso, M. (2009). Interactive policy learning through confidence-based autonomy. *J. Artif. Int. Res.* 34, 1–25. doi: 10.1613/jair.2584

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems*, eds U. Von Luxburg et al. (Long Beach, CA; Neural Information Processing Systems Foundation, Inc. (NIPS)), 4299–4307.

Chu, V., Fitzgerald, T., and Thomaz, A. L. (2016). "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* (Piscataway, NJ: IEEE Press), 221–228. doi: 10.1109/HRI.2016.7451755

Clouse, J. A., and Utgoff, P. E. (1992). "A teaching method for reinforcement learning," in *Proceedings of the Ninth International Workshop on Machine Learning, ML '92* (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 92–110. doi: 10.1016/B978-1-55860-247-2.50017-6

Cohen, P., and Feigenbaum, E. A. (1982). *The Handbook of Artificial Intelligence*, Vol. 3. Los Altos, CA: William Kaufmann & HeurisTech Press.

Colombetti, M., Dorigo, M., and Borghi, G. (1996). Behavior analysis and training-a methodology for behavior engineering. *IEEE Trans. Syst. Man Cybernet. B* 26, 365–380. doi: 10.1109/3477.499789

Corballis, M. C. (2010). Mirror neurons and the evolution of language. *Brain Lang.* 112, 25–35. doi: 10.1016/j.bandl.2009.02.002

Cruz, F., Twiefel, J., Magg, S., Weber, C., and Wermter, S. (2015). "Interactive reinforcement learning through speech guidance in a domestic scenario," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Killarney), 1–8. doi: 10.1109/IJCNN.2015.7280477

Cui, Y., and Niekum, S. (2018). "Active reward learning from critiques," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD: IEEE), 6907–6914. doi: 10.1109/ICRA.2018.8460854

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* 39, 1–38. doi: 10.1111/j.2517-6161.1977.tb01600.x

Dillmann, R., Rogalla, O., Ehrenmann, M., Zöliner, R., and Bordegoni, M. (2000). "Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm," in *Robotics Research*, eds J. M. Hollerbach and D. E. Koditscheck (Snowbird, UT: Springer), 229–238. doi: 10.1007/978-1-4471-0765-1_28

Domingos, P., Lowd, D., Kok, S., Nath, A., Poon, H., Richardson, M., et al. (2016). "Unifying logical and statistical AI," in *2016 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (New York, NY: IEEE), 1–11. doi: 10.1145/2933575.2935321

Dorigo, M., and Colombetti, M. (1994). Robot shaping: developing autonomous agents through learning. *Artif. Intell.* 71, 321–370. doi: 10.1016/0004-3702(94)90047-7

Duvallet, F., Kollar, T., and Stentz, A. (2013). "Imitation learning for natural language direction following through unknown environments," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 1047–1053. doi: 10.1109/ICRA.2013.6630702

Garcia, J., and Fernandez, F. (2015). A Comprehensive Survey on Safe Reinforcement Learning. *J. Mach. Learn. Res.* 16, 1437–1480. doi: 10.5555/2789272.2886795

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., and Thomaz, A. (2013). "Policy shaping: integrating human feedback with reinforcement learning," in *Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS'13* (Lake Tahoe, CA: Curran Associates Inc.), 2625–2633.

Grizou, J., Iturrate, I., Montesano, L., Oudeyer, P.-Y., and Lopes, M. (2014a). "Calibration-free BCI based control," in *Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, QC), 1–8.

Grizou, J., Iturrate, I., Montesano, L., Oudeyer, P.-Y., and Lopes, M. (2014b). "Interactive learning from unlabeled instructions," in *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI'14* (Arlington, VA: AUAI Press), 290–299.

Grizou, J., Lopes, M., and Oudeyer, P. Y. (2013). "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)* (Osaka), 1–8. doi: 10.1109/DevLrn.2013.6652523

Gullapalli, V., and Barto, A. G. (1992). "Shaping as a method for accelerating reinforcement learning," in *Proceedings of the 1992 IEEE International Symposium on Intelligent Control* (Glasgow), 554–559. doi: 10.1109/ISIC.1992.225046

Harmon, M. E., Baird, L. C., and Klopf, A. H. (1994). "Advantage updating applied to a differential game," in *Proceedings of the 7th International Conference on Neural Information Processing Systems, NIPS'94* (Cambridge, MA: MIT Press), 353–360.

Hayes-Roth, F., Klahr, P., and Mostow, D. J. (1980). *Knowledge Acquisition, Knowledge Programming, and Knowledge Refinement*. Santa Monica, CA: Rand Corporation.

Hayes-Roth, F., Klahr, P., and Mostow, D. J. (1981). Advice-taking and knowledge refinement: an iterative view of skill acquisition. *Cognit Skills Acquisit*. 231–253.

Ho, M. K., Littman, M. L., Cushman, F., and Austerweil, J. L. (2015). "Teaching with rewards and punishments: reinforcement or communication?" in *Proceedings of the 37th Annual Meeting of the Cognitive Science Society* (Pasadena, CA).

Ho, M. K., MacGlashan, J., Littman, M. L., and Cushman, F. (2017). Social is special: a normative framework for teaching with and learning from evaluative feedback. *Cognition* 167, 91–106. doi: 10.1016/j.cognition.2017.03.006

Isbell, C., Shelton, C. R., Kearns, M., Singh, S., and Stone, P. (2001). "A social reinforcement learning agent," in *Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS '01* (New York, NY: ACM), 377–384. doi: 10.1145/375735.376334

Judah, K., Fern, A., Tadepalli, P., and Goetschalckx, R. (2014). "Imitation learning with demonstrations and shaping rewards," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14* (Qubec City, QC: AAAI Press), 1890–1896.

Judah, K., Roy, S., Fern, A., and Dietterich, T. G. (2010). "Reinforcement learning via practice and critique advice," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI'10* (Atlanta, GA:AAAI Press), 481–486.

Kaplan, F., Oudeyer, P.-Y., Kubinyi, E., and Miklosi, A. (2002). Robotic clicker training. *Robot. Auton. Syst*. 38, 197–206. doi: 10.1016/S0921-8890(02)00168-9

Kate, R. J., and Mooney, R. J. (2006). "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* (Stroudsburg, PA: Association for Computational Linguistics), 913–920. doi: 10.3115/1220175.1220290

Kim, E. S., and Scassellati, B. (2007). "Learning to refine behavior using prosodic feedback," in *2007 IEEE 6th International Conference on Development and Learning* (London), 205–210. doi: 10.1109/DEVLRN.2007.4354072

Knox, W. B., and Stone, P. (2009). "Interactively shaping agents via human reinforcement: the TAMER framework," in *Proceedings of the Fifth International Conference on Knowledge Capture, K-CAP '09* (New York, NY: ACM), 9–16. doi: 10.1145/1597735.1597738

Knox, W. B., and Stone, P. (2010). "Combining manual feedback with subsequent MDP reward signals for reinforcement learning," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '10* (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 5–12.

Knox, W. B., and Stone, P. (2011a). "Augmenting reinforcement learning with human feedback," in *ICML 2011 Workshop on New Developments in Imitation Learning* (Bellevue, WA).

Knox, W. B., and Stone, P. (2011b). "Understanding human teaching modalities in reinforcement learning environments: a preliminary report," in *IJCAI 2011 Workshop on Agents Learning Interactively from Human Teachers (ALIHT)* (Barcelona).

Knox, W. B., and Stone, P. (2012a). "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* (Paris), 878–885. doi: 10.1109/ROMAN.2012.6343862

Knox, W. B., and Stone, P. (2012b). "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* (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 475–482.

Knox, W. B., Stone, P., and Breazeal, C. (2013). "Training a robot via human feedback: a case study," in *Proceedings of the 5th International Conference on Social Robotics - Volume 8239, ICSR 2013* (New York, NY: Springer-Verlag), 460–470. doi: 10.1007/978-3-319-02675-6_46

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: a survey. *Int. J. Robot. Res*. 32, 1238–1274. doi: 10.1177/0278364913495721

Kohler, E., Keysers, C., Umilte, M. A., Fogassi, L., Gallese, V., and Rizzolatti, G. (2002). Hearing sounds, understanding actions: action representation in mirror neurons. *Science* 297, 846–848. doi: 10.1126/science.1070311

Krening, S., Harrison, B., Feigh, K. M., Isbell, C. L., Riedl, M., and Thomaz, A. (2017). Learning from explanations using sentiment and advice in RL. *IEEE Trans. Cogn. Dev. Syst*. 9, 44–55. doi: 10.1109/TCDS.2016.2628365

Kuhlmann, G., Stone, P., Mooney, R. J., and Shavlik, J. W. (2004). "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* (San Jose, CA).

Leon, A., Morales, E. F., Altamirano, L., and Ruiz, J. R. (2011). "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* (Berlin; Heidelberg: Springer-Verlag), 549–556. doi: 10.1007/978-3-642-25085-9_65

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.* 8, 293–321. doi: 10.1007/BF00992699

Lockerd, A., and Breazeal, C. (2004). "Tutelage and socially guided robot learning," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Sendai), 3475–3480. doi: 10.1109/IROS.2004.1389954

Loftin, R., MacGlashan, J., Peng, B., Taylor, M. E., Littman, M. L., Huang, J., et al. (2014). "A strategy-aware technique for learning behaviors from discrete human feedback," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14* (Qubec City, QC: AAAI Press), 937–943.

Loftin, R., Peng, B., Macglashan, J., Littman, M. L., Taylor, M. E., Huang, J., et al. (2016). Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Auton. Agents Multiagent Syst.* 30, 30–59. doi: 10.1007/s10458-015-9283-7

Lopes, M., Cederbourg, T., and Oudeyer, P. Y. (2011). "Simultaneous acquisition of task and feedback models," in *2011 IEEE International Conference on Development and Learning (ICDL)* (Frankfurt am Main), 1–7. doi: 10.1109/DEVLRN.2011.6037359

Lozano-Perez, T. (1983). Robot programming. *Proc. IEEE* 71, 821–841. doi: 10.1109/PROC.1983.12681

MacGlashan, J., Babes-Vroman, M., DesJardins, M., Littman, M., Muresan, S., and Squire, S. (2014a). *Translating English to Reward Functions*. Technical Report CS14-01, Computer Science Department, Brown University.

MacGlashan, J., Ho, M. K., Loftin, R., Peng, B., Wang, G., Roberts, D. L., et al. (2017). "Interactive learning from policy-dependent human feedback," in *Proceedings of the 34th International Conference on Machine Learning* (Sydney, NSW), 2285–2294.

MacGlashan, J., Littman, M., Loftin, R., Peng, B., Roberts, D., and Taylor, M. E. (2014b). "Training an agent to ground commands with reward and punishment," in *Proceedings of the AAAI Machine Learning for Interactive Systems Workshop* (Québec City, QC).

Maclin, R., Shavlik, J., Torrey, L., Walker, T., and Wild, E. (2005a). "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* (Pittsburgh, PA: AAAI Press), 819–824.

Maclin, R., Shavlik, J., Walker, T., and Torrey, L. (2005b). "Knowledge-based support-vector regression for reinforcement learning," in *IJCAI 2005 Workshop on Reasoning, Representation, and Learning in Computer Games* (Edinburgh), 61.

Maclin, R., and Shavlik, J. W. (1996). Creating advice-taking reinforcement learners. *Mach. Learn.* 22, 251–281. doi: 10.1007/BF00114730

Mahadevan, S., and Connell, J. (1992). Automatic programming of behavior-based robots using reinforcement learning. *Artif. Intell.* 55, 311–365. doi: 10.1016/0004-3702(92)90058-6

Mangasarian, O. L., Shavlik, J. W., and Wild, E. W. (2004). Knowledge-based kernel approximation. *J. Mach. Learn. Res.* 5, 1127–1141. doi: 10.5555/1005332.1044697

Mataric, M. J. (1994). "Reward functions for accelerated learning," in *Proceedings of the Eleventh International Conference on Machine Learning* (New Brunswick, NJ), 181–189. doi: 10.1016/B978-1-55860-335-6.50030-1

Mathewson, K. W., and Pilarski, P. M. (2016). Simultaneous control and human feedback in the training of a robotic agent with actor-critic reinforcement learning. *arXiv [Preprint]. arXiv:1606.06979*.

Matuszek, C., Herbst, E., Zettlemoyer, L., and Fox, D. (2013). "Learning to parse natural language commands to a robot control system," in *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, eds J. P. Desai, G. Dudek, O. Khatib, and V. Kumar (Heidelberg: Springer International Publishing), 403–415 doi: 10.1007/978-3-319-00065-7

McCarthy, J. (1959). "Programs with common sense," in *Proceedings of the Teddington Conference on the Mechanization of Thought Processes* (London: Her Majesty's Stationary Office), 75–91.

Mooney, R. J. (2008). "Learning to connect language and perception," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3, AAAI'08* (Chicago, IL: AAAI Press), 1598–1601.

Najar, A. (2017). *Shaping robot behaviour with unlabeled human instructions* (Ph.D. thesis). Paris, France: University Paris VI.

Najar, A., Bonnet, E., Bahrami, B., and Palminteri, S. (2020a). The actions of others act as a pseudo-reward to drive imitation in the context of social reinforcement learning. *PLoS Biol.* 18:e3001028. doi: 10.1371/journal.pbio.3001028

Najar, A., and Chetouani, M. (2020). Reinforcement learning with human advice. A survey. *arXiv [Preprint]. arXiv:2005.11016*.

Najar, A., Sigaud, O., and Chetouani, M. (2015a). "Social-task learning for HRI," in *Social Robotics: 7th International Conference, ICSR 2015*, eds A. Tapus, E. Andre, J. C. Martin, F. Ferland, and M. Ammi (Cham: Springer International Publishing), 472–481 doi: 10.1007/978-3-319-25554-5

Najar, A., Sigaud, O., and Chetouani, M. (2015b). "Socially guided XCS: using teaching signals to boost learning," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO Companion '15* (Madrid: ACM), 1021–1028. doi: 10.1145/2739482.2768452

Najar, A., Sigaud, O., and Chetouani, M. (2016). "Training a robot with evaluative feedback and unlabeled guidance signals," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (New York, NY), 261–266. doi: 10.1109/ROMAN.2016.7745140

Najar, A., Sigaud, O., and Chetouani, M. (2020b). Interactively shaping robot behaviour with unlabeled human instructions. *Auton. Agents Multiagent Syst.* 34:35. doi: 10.1007/s10458-020-09459-6

Ng, A. Y., Harada, D., and Russell, S. J. (1999). "Policy invariance under reward transformations: theory and application to reward shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning, ICML '99* (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 278–287.

Ng, A. Y., and Russell, S. J. (2000). "Algorithms for inverse reinforcement learning," in *Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00* (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 663–670.

Nicolescu, M. N., and Mataric, M. J. (2003). "Natural methods for robot task learning: instructive demonstrations, generalization and practice," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '03* (New York, NY: ACM), 241–248. doi: 10.1145/860575.860614

Olsson, A., Knapska, E., and Lindström, B. (2020). The neural and computational systems of social learning. *Nat Rev Neurosci* 21, 197–212. doi: 10.1038/s41583-020-0276-4

Paléologue, V., Martin, J., Pandey, A. K., and Chetouani, M. (2018). "Semantic-based interaction for teaching robot behavior compositions using spoken language," in *Social Robotics - 10th International Conference, ICSR 2018* (Qingdao), 421–430. doi: 10.1007/978-3-030-05204-1_41

Pradyot, K. V. N., Manimaran, S. S., and Ravindran, B. (2012a). "Instructing a reinforcement learner," in *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference* (Marco Island, FL), 23–25.

Pradyot, K. V. N., Manimaran, S. S., Ravindran, B., and Natarajan, S. (2012b). "Integrating human instructions and reinforcement learners: an SRL approach," in *Proceedings of the UAI workshop on Statistical Relational AI* (Catalina Island, CA).

Pradyot, K. V. N., and Ravindran, B. (2011). "Beyond rewards: learning from richer supervision," in *Proceedings of the 9th European Workshop on Reinforcement Learning* (Athens).

Randlov, J., and Alstrom, P. (1998). "Learning to drive a bicycle using reinforcement learning and shaping," in *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98* (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 463–471.

Rosenstein,M. T., Barto, A. G., Si, J., Barto, A., Powell, W., and Wunsch, D. (2004). "Supervised actor-critic reinforcement learning," in *Handbook of Learning and Approximate Dynamic Programming*, eds J. Si, A. Barto, W. Powell, and D. Wunsch (JohnWiley & Sons, Inc.), 359–380.

Rybski, P. E., Yoon, K., Stolarz, J., and Veloso, M. M. (2007). "Interactive robot task training through dialog and demonstration," in *2007 2nd ACM/IEEE*

International Conference on Human-Robot Interaction (HRI) (Arlington, VA), 49–56. doi: 10.1145/1228716.1228724

Sadigh, D., Dragan, A. D., Sastry, S., and Seshia, S. A. (2017). "Active preference based learning of reward functions," in *Robotics: Science and Systems*, eds N. Amato, S. Srinivasa, N. Ayanian, S. Kuindersma (Robotics: Science and Systems Foundation). doi: 10.15607/RSS.2017.XIII.053

Singh, S. P. (1992). Transfer of learning by composing solutions of elemental sequential tasks. *Mach. Learn.* 8, 323–339. doi: 10.1007/BF00992700

Sridharan, M. (2011). "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* (Washington, DC: IEEE Computer Society), 424–429. doi: 10.1109/ICMLA.2011.37

Suay, H. B., and Chernova, S. (2011). "Effect of human guidance and state space size on interactive reinforcement learning," in *2011 RO-MAN* (Atlanta, GA), 1–6. doi: 10.1109/ROMAN.2011.6005223

Suay, H. B., Toris, R., and Chernova, S. (2012). A practical comparison of three robot learning from demonstration algorithm. *Int. J. Soc. Robot.* 4, 319–330. doi: 10.1007/s12369-012-0158-7

Subramanian, K., Isbell, J.r., C. L., and Thomaz, A. L. (2016). "Exploration from demonstration for interactive reinforcement learning," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, AAMAS '16* (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 447–456.

Sutton, R. S. (1996). "Generalization in reinforcement learning: Successful examples using sparse coarse coding," in *Advances in Neural Information Processing Systems*, eds D. Touretzky, M. C. Mozer, and M. Hasselmo (Denver, CO: MIT Press), 1038–1044.

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press. doi: 10.1109/TNN.1998.712192

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.* 112, 181–211. doi: 10.1016/S0004-3702(99)00052-1

Syed, U., and Schapire, R. E. (2007). "Imitation learning with a value-based prior," in *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, UAI'07* (Arlington, VA: AUAI Press), 384–391.

Taylor, M. E., Suay, H. B., and Chernova, S. (2011). "Integrating reinforcement learning with human demonstrations of varying ability," in *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS '11* (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 617–624.

Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S. J., et al. (2011). "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (San Francisco, CA).

Tellex, S., Thaker, P., Joseph, J., and Roy, N. (2014). Learning perceptually grounded word meanings from unaligned parallel data. *Mach. Learn.* 94, 151–167. doi: 10.1007/s10994-013-5383-2

Tenorio-Gonzalez, A. C., Morales, E. F., and Villasenor-Pineda, L. (2010). "Dynamic reward shaping: training a robot by voice," in *Advances in Artificial Intelligence - IBERAMIA 2010: 12th Ibero-American Conference on AI*, eds A. Kuri-Morales and G. R. Simari (Berlin; Heidelberg: Springer), 483–492. doi: 10.1007/978-3-642-16952-6

Thomaz, A. L. (2006). *Socially guided machine learning* (Ph.D. thesis). Massachusetts Institute of Technology, Cambridge, MA, United States.

Thomaz, A. L., and Breazeal, C. (2006). "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* (Boston, MA: AAAI Press), 1000–1005.

Thomaz, A. L., and Breazeal, C. (2007a). "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* (Jeju-si), 720–725. doi: 10.1109/ROMAN.2007.4415180

Thomaz, A. L., and Breazeal, C. (2007b). "Robot learning via socially guided exploration," in *2007 IEEE 6th International Conference on Development and Learning* (London, UK), 82–87. doi: 10.1109/DEVLRN.2007.4354078

Thomaz, A. L., and Cakmak, M. (2009). "Learning about objects with human teachers," in *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, HRI '09* (New York, NY: ACM), 15–22. doi: 10.1145/1514095.1514101

Thomaz, A. L., Hoffman, G., and Breazeal, C. (2006). "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* (Hatfield), 352–357. doi: 10.1109/ROMAN.2006.314459

Torrey, L., Walker, T., Maclin, R., and Shavlik, J. W. (2008). "Advice taking and transfer learning: naturally inspired extensions to reinforcement learning," in *AAAI Fall Symposium: Naturally-Inspired Artificial Intelligence (AAAI)* (Arlington, VI), 103–110.

Towell, G. G., and Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artif. Intell.* 70, 119–165. doi: 10.1016/0004-3702(94)90105-8

Turing, A. M. (1950). Computing machinery and intelligence. *Mind* 59, 433–460. doi: 10.1093/mind/LIX.236.433

Utgoff, P. E., and Clouse, J. A. (1991). "Two kinds of training information for evaluation function learning," in *Proceedings of the Ninth Annual Conference on Artificial Intelligence* (Anaheim, CA: Morgan Kaufmann), 596–600.

Vogel, A., and Jurafsky, D. (2010). "Learning to follow navigational directions," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10* (Stroudsburg, PA: Association for Computational Linguistics), 806–814.

Vollmer, A.-L., Wrede, B., Rohlfing, K. J., and Oudeyer, P.-Y. (2016). Pragmatic frames for teaching and learning in human-robot interaction: review and challenges. *Front. Neurorobot.* 10:10. doi: 10.3389/fnbot.2016.00010

Watkins, C. J., and Dayan, P. (1992). Q-learning. *Mach. Learn.* 8, 279–292. doi: 10.1023/A:1022676722315

Waytowich, N. R., Goecks, V. G., and Lawhern, V. J. (2018). Cycle-of-learning for autonomous systems from human interaction. *arXiv [Preprint]. arXiv:1808.09572*.

Whitehead, S. D. (1991). "A complexity analysis of cooperative mechanisms in reinforcement learning," in *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2, AAAI'91* (Anaheim, CA: AAAI Press), 607–613.

Wiering, M. A., and van Hasselt, H. (2008). Ensemble algorithms in reinforcement learning. *Trans. Syst. Man Cyber. B* 38, 930–936. doi: 10.1109/TSMCB.2008.920231

Wiewiora, E. (2003). Potential-based shaping and Q-value initialization are equivalent. *J. Artif. Intell. Res.* 19, 205–208. doi: 10.1613/jair.1190

Wiewiora, E., Cottrell, G., and Elkan, C. (2003). "Principled methods for advising reinforcement learning agents," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML'03* (Washington, DC: AAAI Press), 792–799.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* 8, 229–256. doi: 10.1007/BF00992696

Zettlemoyer, L. S., and Collins, M. (2009). "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, ACL '09* (Stroudsburg, PA: Association for Computational Linguistics), 976–984. doi: 10.3115/1690219.1690283

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.