A social learning formalism for learners trying to figure out what a teacher wants them to do

Thomas Cederborg, Pierre-Yves Oudeyer

Research Article **Open Access**

Thomas Cederborg* and Pierre-Yves Oudeyer

A social learning formalism for learners trying to figure out what a teacher wants them to do

Abstract: This article presents a theoretical foundation for approaching the problem of how a learner can infer what a teacher wants it to do through strongly ambiguous interaction or observation. The article groups the interpretation of a broad range of information sources under the same theoretical framework. A teacher's motion demonstration, eye gaze during a reproduction attempt, pushes of "good"/"bad" buttons and speech comment are all treated as specific instances of the same general class of information sources. These sources all provide (partially and ambiguously) information about what the teacher wants the learner to do, and all need to be interpreted concurrently. We introduce a formalism to address this challenge, which allows us to consider various strands of previous research as different related facets of a single generalized problem. In turn, this allows us to identify important new avenues for research. To sketch these new directions, several learning setups are introduced, and algorithmic structures are introduced to illustrate some of the practical problems that must be overcome.

Keywords: Social learning; human robot interaction; formalizations

DOI 10.2478/pjbr-2014-0005
Received February 7, 2014; accepted August 11, 2014.

1 Introduction

Various strands of research about social learning in animals and machines have dealt with similar problems, but have made different assumptions and used different formalisms. Many of the problems that these learners face are similar when described in the abstract. A human teacher might be wrong, creating similar problems for the learner regardless of the type of interaction that is taking place. A teacher being wrong, uninformed or confused could for example lead to imperfect human evaluations, or bad demonstrations. The theoretical problem of how to define success when the human is flawed is similar across any domain where an artificial learner is trying to do what a flawed human teacher would like it to do. When a non-expert human and an artificial learner are interacting in an unstructured environment, a number of problems arise. In this setting it is not clear how to obtain a good success measure from the learner's sensors, meaning that some attention needs to be given to how success is best defined, and to how a learner can estimate this success based on its sensors. To bring theoretical foundations to several different fields, a formalism is proposed for any situation in which a learner is trying to figure out what a teacher wants it to do, based on observations or interactions. This is meant to cover any type of information that the learner might be analyzing, and tries to not make assumptions regarding the types of strategies the learner uses to obtain information. The learner has to make probabilistic updates regarding what to do, without ever knowing for certain how successful it is.

The problems with hard coding a reward function for a robot learner in an unstructured environment with a human is that for the learner to use the reward signal, it must be specified in terms of variables that are available to the robot. See [1] for a detailed discussion of the problem with specifying this reward function. In the formalism proposed in the current paper, a reward function defined in robot sensors is therefore instead treated as an approximation by a learner, not something to be maximized. In [1] the idea of Inverse Reinforcement Learning (IRL) is proposed, using a teacher's demonstrations to estimate the reward function. The interpretation of those demonstrations is, however, difficult. See, for example, [2] for a discussion of the correspondence problem. Other problems include a teacher failing to achieve some state, or trying to achieve the wrong thing due to a misunderstanding.

Since defining a reward function in terms of a robots sensors is difficult, and interpreting a demonstration is also problematic, a human teacher could observe the learner and simply tell it how well it performed. Studies have, however, shown that there are serious problems with

***Corresponding Author: Thomas Cederborg:** Inria Bordeaux Sud-Ouest, E-mail: thomascederborgsemail@gmail.com
Pierre-Yves Oudeyer: Inria Bordeaux Sud-Ouest, E-mail: pierre-yves.oudeyer@inria.fr

directly maximizing human numerical input. In [3] it is for example shown that there is sometimes a motivational component to the rewards given by human subjects. The behavior of giving positive rewards to bad states is not always caused by failure to observe the world, or lack of knowledge about what the correct action is, but is sometimes instead a way of "encouraging the robot". The fundamental problem is a mismatch between the actual human teacher, and the assumptions built into the algorithms. In [3] two paths to mitigating the problem are discussed, understanding what humans actually do, and modify learning algorithms to better match this (see also [4] for an algorithm based on a different interpretation of evaluations), or the modification of the setup to make the teacher's behavior better match the assumptions of the algorithm (giving a teacher a dedicated button for motivational feedback reduces the tendency to use the "reward button" for such communication). See [5],[6], [7], [8] and [9] for more on these issues. Further problems include a teacher giving an evaluation based on not seeing some object, or due to a misunderstanding.

Since a programmer defined reward function, a demonstration and a teacher generated evaluation are all problematic and in need of interpretation, the presented formalism treats all of them as members of the same abstract class of objects, referred to as information sources. Besides evaluations, demonstrations and reward signals this class of information sources include anything that can be used to infer what a teacher wants a learner to do, for example pointing gestures, facial expressions, EEG readings, hand gestures, tone of voice, speech comments, eye gaze and anything else that can be used to guess what a human teacher wants the learner to do. The formalism presents a coherent way of describing learners that autonomously learns how to interpret information sources, regardless of what those sources are. For example by learning a simple task with the help of a reasonably good understanding of demonstrations and speech comments, and then using this to re-interpret the meaning of a "reward button" pushed by a human.

The learner's goal is not to maximize positive evaluations, but instead to find better interpretations of evaluations and other information sources in order to figure out what the teacher actually wants the learner to do. This can lead to different types of behavior compared to a "positive evaluation maximizer", for example when hiding a mistake will lead to higher reward[1], or when consistently performing well will result in the teacher deciding that the learner knows the task and therefore stop using the buttons[2].

The formalism defines a problem. What is presented is a formalism, not a model, and what this article attempts to answer is: "what, exactly, do we mean by: figuring out what a human wants done?" and "how can we describe different types of solutions to this problem in a unified way?". The practical goal is to be able to build better artificial learners that can learn from humans in environments that are as unstructured as possible. Such learners need to operate effectively in situations where the human teacher is not an expert, might be wrong, or is not certain of how the learner will interpret various types of feedback. This article, therefore, tries to give a solid theoretical foundation for a variety of research projects, gathering several different types of social learning under the same formalism. Each information source is treated in a similar way, and a learner is seen as something that learns to interpret these information sources.

This type of investigation can help us better understand social learning in humans by building computational models. To the extent that a biological organism is trying to solve the formalized problem, it might be a good idea to use the notation as a descriptive/predictive model of that biological system. And to the degree that biological systems are successful at solving the formalized problem, their strategies can inspire solutions in artificial agents (just as a good solution to the problem is a candidate explanation for some specific behavior of a biological system (if something works, a biological system might have found it)). The actual details of biological systems are however not the only determining factor regarding how to formalize the problem or how to best solve it. The problem is interesting in it's own right, regardless of what biological systems do. And the solution strategies of biological systems are not necessarily better than designed strategies.

1 Revealing a possible mistake to a teacher who provides a numerical value when evaluating the learner might sometimes result in both lower expected numerical value provided, and at the same time higher expected accuracy in the evaluation. If the goal is to maximize the numerical value given by the teacher, the optimal action might be different from the optimal action of a learner as defined by the presented formalism.

2 If the teacher will stop evaluating the learner when the learner performs the task correctly, and the teacher gives more positive than negative rewards, then learning the task as fast as possible might not be the optimal way of maximizing the value received by the teacher. Thus the optimal action of a learner as defined here could be different (the numerical value is a teacher signal to be interpreted, not a value to be maximized).

If the teacher often fails (perhaps missing a target on some occasions), then the learner could learn much more if it could interpret the demonstrations and make a guess about which attempts were failures (for example, by analyzing facial expressions, body language or what the teacher says after the demonstration). This boost in learning is of the same type as the boost in learning attained by better estimating which part of a demonstration was relevant, or which action was evaluated. These problems all share the very important property; a perfect solution is not needed to improve on the status quo. Any solution that is better than what is produced when ignoring the problem will boost learning. Being aware of the problem (that the evaluated action is not necessarily the one directly preceding the evaluation, that not all parts of the context are necessarily equally important), and mitigating it, can be better than ignoring the problem, even if a complete solution is very far away (a “failed demonstration detector" does not need to be perfect to improve on the status quo).

This new way of looking at social learning allows us to see new types of learning situations that can be explored. An example could be building a learner that needs to learn how to tell a failed demonstration from a successful demonstration. Such a determination might be based on facial expressions and body language, and can be used to learn a set of difficult tasks. This can boost learning in cases with large failure rates or with to expensive/few demonstrations. Even without the ability to separate failed and successful demonstrations, the learner might be able to confidently learn one simple task by observing many demonstrations. When the task is known, the learner knows which demonstrations were failures, and can find correlations between failures and facial expressions or body language. The updated interpretation hypothesis can now be used to learn the more difficult tasks.

We have thus far talked about the learner’s ability to interpret various sources of information. This is, however, ambiguous and should be made concrete and formalized. A human’s demonstrations, facial expressions, eye gaze or EEG readings do not contain a crisply specified meaning that the learner can decode. The teacher might not even be consciously aware of producing all the information that the learner is interpreting. This makes it problematic to define success in terms of how well meaning in the mind of the teacher is transferred to the mind of the learner. Instead, a success criterion is defined in terms of what an informed version of the teacher would think of a learner’s action choice. The interpretation of an information source is thus successful to the extent that it causes the learner to take actions that are successful according to this criterion.

A creative new learner strategy could, for example, result in states that the teacher does not see, have long term consequences the teacher is unable to predict, or have side effects the teacher has never considered as a possibility. To choose between two strategies, it is now necessary to estimate what the teacher would have thought about a strategy if that teacher was fully informed. This is subtly different from predicting how much it will like the strategy based on what the teacher will observe the learner doing.

The formalism for the unsimplified setup starts by redescribing existing learning algorithms. Each algorithm is seen as a certain way of interpreting an information source. The idea is introduced where the way of interpreting an information source can be modified based on observations, which means that any specific interpretation of a source is a hypothesis (for example that an observable scalar value indicates absolute success, or that it indicates incremental progress). A learning algorithm is thus referred to as an interpretation hypothesis. The concept of the teacher’s informed preferences is defined and a learning algorithm is now viewed as a hypothesis of how information sources relate to the informed preferences of the teacher. Informed preferences can be roughly described as what the teacher would want the learner to do if the teacher knew and understood everything it considers relevant to the learner’s action choice. Informed preferences is designed to deal with cases such as “the teacher would like the learner to perform an action, but if the teacher knew the consequences of that action, would prefer another action instead" or “the teacher is very happy with the end result after the learner has cleaned the apartment, but if the teacher knew that the cleaning produced a lot of noise that disturbed the neighbors, the teacher would not like the cleaning strategy". The success of a learner’s action choice is determined by what an informed teacher’s opinion would have been regarding how good these actions are for the teacher.

The problem is unfortunately no longer an inference problem with a numerical success value visible to the experimenter. This means that it is not possible to describe an algorithm as searching for an approximation to a known optimal solution. The learner has some rule for selecting actions, which we will refer to as a policy. In addition to suggesting updates to this policy, the interpretation hypotheses make predictions regarding what will be observed. Let’s take the example where a human teacher is given one button with a plus sign on it, and one button with a minus sign on it (both including a “volume" control), and the teacher is told that it can use these to give feedback to the robot learner. A learner can now have several competing hypotheses regarding how to interpret this

teacher signal. For example, the buttons might be pushed comparing the learner's most recently performed action to: (i) the best action the teacher has seen, (ii) the optimal action, (iii) the previous seven actions performed by the learner, (iv) the previous two actions performed, or various other possibilities. For a limited data set, the suggested policy updates can be quite different. They also make different predictions regarding what interaction histories will be observed. If the learner performs the same action in the same context, for example, then the different hypotheses will predict different changes in evaluation. In general, this results in different probabilities for a given interaction history. Thus, given an observed interaction history, it is possible to update the set of interpretation hypotheses by discarding, changing probabilities or modifying parameters. If the learner has access to a few reliable task policies that are learnt by interpreting some other information source, and the interaction history of the time these tasks were learnt includes button push data, then these task models can be very useful when the learner chooses between the different ways of interpreting the reward button (the correct action, the action actually taken, and the button pushing behavior are all known). It is in general possible to use the interpretation of one information source to learn how to interpret another source; if the well understood source can be used to learn a task policy, then this task policy can be used when learning to interpret another information source.

Some of the information sources will be described as "teaching signals", but this should not be taken to imply that the teacher is following an interaction protocol shared with the learner.

After covering related work in the following section, the scope of the formalism is described. Then a success criterion is introduced in a series of simplified worlds, each with a mathematically defined success criterion. These are presented in stages where some simplifications are removed at each stage. In the following sections, the simplifications are removed step by step in order to gradually approach the desired setup containing an actual human teacher in an unstructured environment. At one stage of removing assumptions/simplifications, the problem stops being a well defined inference problem with a mathematically defined success criterion (there will be no way of obtaining a number representing the level of success of a learner in a specific situation), and we must fall back to the informed preferences of the teacher. In section 8 existing learning algorithms are described as interpretation hypotheses.

2 Related work

The formalism attempts to cover all agents, referred to as learners, trying to figure out what a human teacher wants them to do. The formalism tries to avoid making assumptions about the semantics of the information sources that the learner uses to figure this out. This means that research into all forms of social interaction where an agent can be described as trying to do what a teacher wants it to do is related, including research on animal, human and artificial learners. Formalisms of related setups are also relevant, including those concerned with learning from demonstration, inverse reinforcement learning and reinforcement learning. By framing social learning in this way, the formalism opens up many new possible avenues of research, with perhaps the most prominent example being a learner that is learning how various information sources should be interpreted concurrently with learning to perform multiple tasks. These new research avenues are a significant contribution of this article, and in this section they will be discussed while covering the previous work that is most related to the new setups.

This is a very large research area and all aspects cannot be covered here. Focus will instead be on a few representative research projects, and on various ways in which they can be extended, as well as on other formalisms (since these have similar goals as the presented formalism), and studies of human teachers (since human teachers is what the learner is trying to interpret).

2.1 Robot Learning from Demonstration

The teacher can provide examples of good policy which can be used to infer what it wants the learner to do. One of the difficulties in interpreting these demonstrations arises if they are not provided to the learner in the form of its own input and output spaces, giving rise to the correspondence problem [2]. If demonstrations can be obtained, or transformed into, the input and output spaces of the learner then there are a lot of different methods for turning this into a policy.

If demonstrations are interpreted as flawless policy executions perturbed by Gaussian noise, then a model of the optimal policy in the form of a Gaussian Mixture Model (GMM) can be found using an Expectation Maximization (EM) [10] algorithm. Modeling a generating policy using an GMM with parameters set by EM was introduced in [11], and later Calinon et al. showed, through a series of advanced robotic experiments [12–15], that the method was

easy to use and could successfully generalize and reproduce smooth motor trajectories. The assumption made is that each data point have been generated by one out of a set of Gaussian distributions. The parameters of these Gaussians are not known, and which one generated what data point is also not known. If the parameters of the Gaussians were known, it would be easy to estimate the probabilities of which Gaussian generated which data point, and vice versa. If it were known what data points were generated, it would be easy to estimate the parameters of the Gaussians (in both cases one can relatively easily find the best possible estimate given the available data). The solution presented in [10] is to make an initial estimate of both parameters and probabilities, and then concurrently re-estimate both. If one estimate gets better, then this can be used to improve the estimate of the other, which in turn can be used to improve the estimate of the first.

Besides being a technical solution to using demonstrations to find the parameters of a GMM, the basic ideas behind the EM algorithm will be used later in example solutions during the simplified setups. It will also be used as an analogy in the examples where multiple interpretation hypotheses and multiple tasks are learnt concurrently in the unsimplified setup. Knowing how to interpret one source of information can enable the learning of tasks and the interpretation of other information sources, as well as re-interpreting specific interactions. This is similar to a learner that has an imperfect and uncertain model of how to interpret speech comments, facial expressions and demonstrations, as well as imperfect and uncertain models of a set of tasks. Observations of a new demonstration can be used to update a task model, that in turn can be used to update the model of the speech comments; if the task being commented upon is better understood, it is possible to learn more about what the speech comments meant. The updated model of the speech comments can then be used to update another task model. This new task model can now be used to update the model of how to interpret demonstrations and facial expressions. This is the exact same principle: update a set of uncertain models with interlinked probabilities.

The usual setup is either a single task or labelled demonstrations. One exception is [16], where two different tasks are learnt from unlabeled demonstrations (the starting position determines which of two different ping pong swings to perform). The imitator is not told the number of tasks or which demonstrations is of which task. The problem of multiple tasks is also dealt with in[17] and [18], where an imitator learns how to respond to the instructions of a human. They also deal with multiple unsegmented demonstrations, where the number of tasks and the number of demonstrations are both unknown.

To allow new unlabeled demonstrations to be incrementally and immediately incorporated online during the teaching process, Local and Online formulation of Gaussian Mixture Regression (ILO-GMR), was introduced in [19] and [20]. The central idea of this technique is to build online and on-demand local Gaussian Mixture Models of the task(s).

When an imitator is observing a demonstrator performing an action, the imitator must decide how to map the actions of the demonstrator into actions in its own action space, referred to as the correspondence problem [2]. This problem becomes more difficult the more the embodiments of the demonstrator and imitator differ. Imagine a child observing a much bigger adult demonstrator clap its hands. Should the hands of the imitator be at the same height above the floor, or should the angles of the arms be the same? Should the maximum distance between hands be the same, or should the angles be the same? If the maximum distance should be smaller (so that arm angles can be mimicked), it is not physically possible to simultaneously mimic the speed of clapping, the speed the hands hit each other and the relative speed curve of the hands (unless the angles are much larger, then the hands of the imitator can only impact each other at the same speed as the adult, if the child either increases the pace of clapping or takes a pause at some point during the movement). This simple, “clap your hands" example shows that there are difficult questions to answer about what the essence of a movement is, even if embodiment only differ in size, and the movement is only relative to the agent itself. One method that has been used to learn this mapping is presented in [21] where a robot observes a human that is mimicking its motions. If a human performs a movement that it considers to be best corresponding to the movement of the robot, the robot can learn the intended mapping of the human. One common method of demonstration that avoids dealing with the correspondence problem is to tele-operate the robot, for example using a joystick to remote control a robot helicopter [22]. The same effect can be achieved by physically directing a robot’s body, as in for example [23].

This research is usually referred to as *imitation learning*, *programming by demonstration* or *learning from demonstration*. In [24], an agent-based view of imitation learning is presented and five central questions are put forward. The imitator must decide who, when, what and how to imitate, and someone must address the question of what constitutes success (for example formalized by an experimenter or the creator of an artificial imitator). What constitutes success is defined by the formalism relative to what

an informed version of a teacher would think. If the learner knows what it is trying to do, it can use a human teacher to achieve this known goal. The answer to the question of "who to imitate" is then determined by whomever provides the demonstrations that are most useful for achieving this goal. The answer to the question of "when to imitate" is then determined by when a given demonstrator is providing demonstrations that are useful for this goal. The learner, as defined in the presented formalism, is in a different situation since it does not know what it is supposed to do, and would use a teacher's demonstrations to figure this out.

In [24], the question of what the imitator is trying to achieve is left open (and depends on factors such as what type of animal the imitator is or what type of robot it is). If the imitator has a specific goal whose progress it can measure and that is unrelated to the teacher/demonstrator, then the "who" and "when" questions can be answered with respect to that goal (the formalism presented is different since the goal is dependent on a teacher whose mind is not necessarily easy to read). We can take the example of a robot that is trying to maximize the amount of states it can reach in a measurable outcome space, and an animal that is trying to maximize the amount of food obtained when food is given as rewards for imitation by a human. "Who" to imitate is then a question of who gives the most informative demonstrations relative to current skill levels (producing reproducible action with outcomes it could not previously achieve) or a question of who gives it the most food. "When" to imitate depends on how effective this is for learning relative to other learning strategies or a question of whether imitation is the most effective way of getting food. The "how" and "what" questions also becomes well formalized (imitation of which parts of the demonstration results in food), meaning that the simultaneous exploration of all four questions is a well formed problem. In [25] all four questions are simultaneously addressed by an artificial imitator that is trying to expand the amount of outcomes it can reach. But in general, the setup where an imitator faces these four questions simultaneously is not well explored.

The field has mostly focused on the "what" and the "how" questions where two large technical issues that algorithms must take into consideration have been to maximize at the same time *genericity* and *accurate generalization*. The goal has been to develop techniques that may allow a robot to learn various context-dependant skills without the need for tuning of parameters for each new skill: this implies that the dimensions/variables that the robot can measure in demonstrations should be higher than the number of dimensions that are relevant for determining the action to be done during a (sub-part of) a skill. Indeed, different skills might be determined by different variables/dimensions/constraints. If a robot can to learn the "essence" of a skill, this will allow it to reproduce demonstrations successfully in contexts which are not exactly the same as those of the demonstration. If all details of demonstrations are considered then the learner will never find itself in a situation that is similar to a demonstration, since there is always something that is different. If the wrong details or the wrong abstractions are used, demonstrations might seem to have been made in similar situations even if they are not. If the learner frames the demonstrations correctly (that is, attends to the relevant details or abstracts the demonstration in the right way) it can find those demonstrations that really are relevant to the current context. In [26] several learning from demonstration projects are covered, classified according to how demonstrations are gathered, and according to the type of algorithm used, as well as discussing some ways of dealing with imperfect demonstrators. It also classifies inverse reinforcement learning as an instance of learning from demonstration. In [27] research into robot programming by demonstration is summarized and different types of algorithms and ways of encoding tasks are presented.

Demonstrations can be encoded at the trajectory level. Instead of building a model that operates on discrete primitive actions, models are built that operate in continuous spaces. These spaces could, for example, be in the joint space of the robot or in the operational/task space, such as the position, speed or torque space of its hands, mapping sensory inputs to motor outputs or desired hand velocities (which can be seen as different levels of granularity). In the early work of [28] the set of acceptable trajectories is spanned by the trajectories seen during the demonstrations, and [29] introduces a nonparametric regression technique based on natural vector splines to build a representation of a trajectory either in Cartesian (sometimes referred to as task space) or joint space, from several demonstrations. In [30] a method inspired by dynamical systems and attractors is presented using recurrent neural networks to learn different motions and switch between them. The mimesis model proposes to encode a trajectory as a HMM. Reproduction is achieved using a stochastic algorithm on the transition probabilities of the HMM.

A number of authors have proposed to encode sensorimotor policies as sub-symbolic dynamical systems, whose dynamical output is determined by both internal parameters and an input context [27]. These dynamical systems can be implemented/modelled as complex recurrent neural networks [31][32][33] or as more traditional statistical regression techniques [34][35]. These are interesting as

they in general do not make assumptions regarding the type of outputs and inputs they are dealing with, making them suitable for use when the context has been extended to include the communicative acts of an interactant. In practice, output has typically been motor commands and input context has been a compact encoding of the past sensorimotor flow [15][27][22] including potentially internal variables that may encode simple things such as motivational states but also predictions of the future or hypotheses about properties of the environment that cannot be observed directly [36].

It has also been proposed that learning context-dependent skills could be achieved through inverse reinforcement learning (see [1] for some early work and [37] for a recent overview): instead of directly modeling the skill at the trajectory level with a dynamical system, a first inference step is performed that consists in trying to infer the reward/cost function that the observed demonstrations are supposed to optimize. Such a function can for example be a numerical assessment of how much food gets into the mouth of the doll in the case of a set of doll-feeding demonstrations, or how close to a rabbit the stone of a teacher landed if the inferred intention of the teacher was to hit the rabbit with the stone. Thus, this is a technical approach to directly learn the goal/intention of the demonstrator. When a hypothesis of reward function has been generated, then the learning agent can search for the adequately encoded dynamical system (including encoding of the context) that allows it to maximize the corresponding rewards. The drawback of this approach is that it is difficult to design a hypothesis space of reward functions that is at the same time flexible enough to learn a variety of tasks and allows for efficient statistical search and inference. There are two main advantages of this approach. Firstly, it allows potentially better generalization by letting the robot self-explore alternative strategies to achieve the goal that may be more efficient or more robust than those used by the demonstrator (e.g. see [22]). The second advantage is that the approach naturally makes it possible to take advantage of reinforcement feedback from the demonstrator during reproduction attempts made by the robot.

In [38], a learner is presented who updates interpretation hypotheses of a teacher's comments, described as an extension to inverse reinforcement learning. The teacher say things such as: "go right", "bad robot", "no", "good robot", "go left", and the learner starts with an incomplete model of how to interpret these utterances. The learner then concurrently updates this model of what the words mean, as well as a reward function used to describe the task. In [38], discrete symbols are used, and the learner starts with a partial lexicon. In [39], [40] and [41], these limitations are relaxed, and the information sources being interpreted is extended to include EEG. EEG is an information source for which the need of interpretation is intuitively obvious (raw EEG readings look incomprehensible to a human, while the interpretation of a reward button or a speech comment might be mistakenly seen as obvious).

In [42], different types of social learning are classified based on the degree to which a learner is trying to do what a teacher wants it to do. A distinction is made between higher level teacher goals and low level teacher actions, forming a triangle. Learners that do not care at all about what the teacher wants the learner to do is in one corner (for example learning how an object functions by observing a teacher, but without any consideration of that teacher's goals). Learners that care only about high level teacher goals are in another corner, and learners caring about only low level actions are in the third corner. Learners caring about multiple things are somewhere in the middle. Within the framework of [42], a learner as formalized here would be somewhere on the side of the triangle opposite the "don't care about the teacher corner". Where the learner is on that line depends on what the teacher wants it to do (if the teacher is showing the learner how to dance it would be at one end of this side, and if the teacher wants to achieve some specific world state by any means, then the learner would be at the opposite end).

2.2 Reinforcement learning

A numerical evaluation of an action is another information source that can be used by an agent to try to determine what a human wants it to do. These types of signals are often maximized, which is what a learner, as defined here, would do under certain interpretations. If the numerical value is interpreted as coming from an informed teacher, as evaluating absolute performance, and as coming from a teacher who will always send an evaluative signal[3], then it makes sense to maximize this value.

3 Even if a teacher sends numerical values corresponding exactly to the performance of the learner, the teacher might under some circumstances decide that the learner now knows the task and consequently stop sending signals. This might mean that the learner's optimal action does not maximize the value sent by the teacher (there is no incentive to drag out learning just to get the teacher to keep pressing the button). The learner would therefore only be a pure value maximizer if it interpreted the teacher as a tireless evaluator, in which case, the absence of the signal cannot be a result of consistently successful actions.

Other interpretations will lead to somewhat different strategies. In an example based on the experiment with the Leo robot in [43], we can imagine a learner and a teacher sitting at opposite sides of a table, each with only partial ability to observe a set of colored buttons that the learner can press. If the learner can estimate what the teacher sees, there are now many ways to treat the numerical value besides as a pure indicator of success. The value could be ignored if the teacher does not see everything relevant, or it could be used to make an update with a smaller step size. The value could also be considered as an evaluation of what the teacher saw. If two identical learner actions in identical setups are judged differently, this could be interpreted as the teacher evaluating performance based on buttons that the learner does not see. Another way to interpret the inconsistent evaluation of two identical actions in seemingly identical situations is that the value represents incremental progress. Attempting to model what a teacher sees, and how to interpret the teacher's actions is a central idea behind this formalism. Except for [43], this situation is severely under explored.

As well as taking into account the level of informedness of a teacher when interpreting a numerical value, a learner might also need to determine whether the reinforcement signal indicates incremental progress or absolute performance. The two models will, in general, make different predictions about which interaction histories are more probable, and imply different policies, making it possible and useful to estimate which model is correct. The learner can also act proactively to increase its information content. If the learner is unsure that the action it is about to take is good, it can, for example, wait until the teacher is paying attention, and/or call on the attention of the teacher before performing the action. If the learner believes it might have made a mistake, such as damaging something, it can draw the teacher's attention to this in order to check if it will receive a negative value in response.

If the best way of getting reward is to do what the teacher wants it to do; the problem faced by the reward maximizer is identical to the problem faced by the learner as specified in this formalism. The learner wants to figure out what the teacher wants it to do since doing what the teacher wants it to do is the learner's primary goal. In this case the reward maximizer wants to figure out the same thing, since it is the best way to get a reward. This means that algorithms maximizing a reward can be used as solutions to the problem faced by a learner, possibly in combination with a system that estimates when this approximation is appropriate and useful. The optimal behavior of the two formalisms diverge in the event that hiding the result of an action from the teacher will result in higher expected reward. A learner could, for example, be built as a combination of a reward maximizer and a system that checks for this type of situation and then switches to another type of behavior.

2.3 Learning to interpret speech, gestures and facial expressions

If trying to get someone to understand a task, one of the most intuitive ways to help them understand what they should do is to just tell them directly. This raises the question of how a learner might figure out how verbal utterances relate to what a teacher wants the learner to do. For this purpose, the learner can use observations of speech in three different ways. One category is of the type "good robot", an evaluative comment in the space of teacher signals (an information source that should lead to policy updates). Another category is illustrated when a teacher throws a basketball to another player shouting "here", treated as part of the context and input to a policy (an aspect of the scene that should lead to a specific action). A third type is when the teacher shouts "here" when the teacher is in a good position and a player on the same team has the ball, a demonstration that is to be imitated in a specific circumstance (something that should be mapped to a state in the learner's action space, and reproduced if the learner is in the same context as the teacher was when performing that action). It should be clear by now that from the point of view of the learner, it is possible to treat "good robot" type utterances in the same way as facial expressions. In the same way it is possible to treat a team player's hand waving, its "here" speech act, and the fact that there is no opponent between the learner and a team player in the same way (they are all part of the context).

In this sense there is no longer a need to define success in language learning as the successful transfer of a meaning from one mind to another (a definition that suffers from the problem that it is hard to locate these meanings inside the head of human speakers). This point has been elaborated further in [44], which also investigates how a single learner can learn to respond accurately to speech acts, gestures and object positions using the same strategy and treating them in the same way. In this situation the learner is not told the number of tasks, and not told whether or not speech or gestures are at all relevant to the tasks of the individual demonstrations.

2.4 Combined approaches

The idea to let the task model be complemented by feedback from the demonstrator upon reproductions, as well as by self-exploration actions done by the robot, is presented in [45][34]. These studies provide an example of how to combine multiple sources of information from social interaction. In [22] an imitator learns to perform helicopter acrobatics and its skill surpasses that of the teacher; this research exemplifies how important it is for a formalization of social learning to allow a learner to become better than the teacher.

It has to be noted that these various approaches to learning by imitation can also naturally be augmented by more complex interactions involving things such as the demonstrator explicitly drawing the attention of the learner towards the relevant aspects of the context using social cues (e.g. see [36]). Thus, this family of approaches adopt a non-restrictive broad view of learning by imitation, in contrast to more restrictive definitions sometimes used. The more restrictive definitions have lead some researchers to argue that "learning by imitation is limited because the observed action does not always reveals its meaning [...] In order to understand an action, a learner will typically need to be provided with additional observation given by a teacher who demonstrates what is crucial: the goal, the means and - most importantly - the constraints of a task" [46] (for the same line of thinking, see also [47] and [48]).

Combinations of evaluative feedback and demonstrations have been explored in several different settings. In [49] the learner is provided with demonstrations, and the teacher is able to provide evaluative feedback by indicating parts of a reproduction where the learner performed well or badly. While [50] does not explicitly say that it combines different kinds of social learning feedback, if we view the motor primitives as a set of demonstrations (or a set of demonstrated skills), then the reinforcement signal acts as a second source of information.

2.5 Studies of human teachers, and algorithms adapted to their observed behavior

The difficulty of the problem faced by a learner is strongly influenced by how well the priors over interpretation hypotheses fits with the actual behavior of the human teachers that the learner will encounter. This aspect can be improved by a systematic study of how humans actually teach artificial learners. Therefore the problem of autonomously re-interpreting information sources are tightly linked to three complimentary research avenues related to the study of human teachers. First is the experimental paradigm trying to figure out how humans actually behave, leading to better algorithms if assumptions are static or better initial interpretation hypotheses if assumptions can be updated. Second is trying to figure out how various learner actions influence a teacher in a social situation, something that can lead to better feedback (displaying confusion or understanding seems to help people give good feedback for example). The third is how the behavior of teachers can be modified by researchers so that they give useful feedback (for example explaining to teachers what types of demonstrations will be useful, give evaluators a dedicated button for motivation in order to make them stop using a reward button for this purpose, etc).

In [36], [6], [7], [8] and [9], human teacher intentions were investigated, showing, for example, that teachers attempted to include multiple communicative intents in a single channel, that negative feedback had a different interpretation than positive feedback, and that teachers sometimes gives positive feedback attempting to guide future actions. Humans also tend to give more positive than negative rewards, even in the very beginning of a learning episode, before performance is good (the surplus of positive reward is not caused by high performance). It shows that interpreting the total number of positive rewards minus negative rewards as a measure of success is a very inaccurate model of many human teachers. Various ways of modifying the algorithms to better fit the actual intent of human teachers were shown to improve learning.

On a more abstract level [36], [6], [7], [8] and [9] shows that pre specifying an interpretation of a human teacher is in general very difficult to do at programming time, and also that better interpretations can lead to better learning. That failure to pre specify an interpretation (valid for all teacher types and situations a learner might encounter) is both likely and costly shows that this mismatch is a problem that should be investigated. These studies thus provide a motivation for the formalism presented in this paper. The formalism provides a systematic way of describing strategies that the learner can employ to autonomously improve its interpretations. Giving the learner this ability is useful since flawed interpretations are likely, and improving them is both possible and useful.

In [51] different interaction protocols are tested on non-experts to see how they perform in actual situations. In [52], the cobot software agent interacts in an online social situation where several of the usual assumptions regarding the reward signal is violated. The average reward over time is for example not an appropriate measure of

success (due to the human tendency to stop giving positive rewards when some part of a social environment consistently does what it's supposed to do).

In [53], several ways in which human teachers break implicit assumptions of learning algorithms are discussed, further underscoring the usefulness of improving the interpretations of human behavior. In [54] a modification to the TAMER framework [55], is designed to allow a learner to take advantage of feedback on future actions. This might allow the learner to take advantage of "no don't do that" type feedback, if future intent can be displayed by the learner and understood by the teacher. Even when it is impossible to find out what specific imagined future learner action was evaluated by the teacher, determining that a particular evaluation was not an evaluation of any of the performed actions can still reduce noise. Building on these findings, [56] presents a study observing teaching behaviors in five different navigation tasks. In [57] studies of human teachers are used to build a parameterized model of how a teacher gives feedback, with a given teacher being describable as a point in a three dimensional space. There is the error rate in determining whether an action was correct or not, the probability of providing positive feedback given that an action was determined to be correct, and a probability of providing negative feedback given that an action was determined to be incorrect. These three values, an hypothesized correct policy, and an action taken, implies a probability distribution over the observable evaluation space. The presented formalism aims to provide a structured way of denoting these types of models, valid for any information source. This will hopefully facilitate the description of learners that is concurrently updating a policy, as well as several models of this type (each operating on a different information source).

In terms of modifying the behavior of teachers, [58] shows that researchers explaining what types of demonstrations will be useful to the learner helps produce useful demonstrations. A modification to the setup where a dedicated motivation button is added can reduce the tendency to use a "reward button" for communicating this message [3]. For learners modifying teacher behavior, see [59] which investigates active learning with the specific viewpoint of how a learner can act to maximize informative observations during interaction with a human teacher, or [60], where a robot learner influences the way a human teacher gives movement demonstrations.

In [61] a survey of different methods for extracting information sources from human behavior is conducted. This concerns the construction of the information sources that the presented formalism hopes to interpret. Unless the learner is able to handle raw sensor inputs of a human giving feedback, there will need to be an intermediate step, where manageable input spaces are constructed, making these methods highly relevant to implemented solution strategies. If two different methods for extracting body language are available to the learner, it can estimate how these correlate with what should be done. If some task is known, the learner can evaluate which of these two spaces are most useful for learning a new task. The learner cannot, however, directly evaluate correctness as this might not be perfectly correlated with usefulness; one method might fail to pick up anger and boredom, and the other might fail to pick up disappointment and fear. The usefulness of the resulting input spaces can be different even if their accuracy is the same. If the extraction methods come with parameters, then this usefulness estimate can result in a parameter search, finding the input spaces that are most conductive for learning. See also [62] for a survey focused on sensor fusion in the domain of social signals. The presented formalism implies a way of fusing sensors, based on its correlation with tasks, but the resulting input spaces are selected on a different measure (specifically, the learner would ignore accurate models of emotions that the learner is unable to use for the purpose of finding out what to do).

In [63] we can see another approach to studying imitation learning, in this case using a parrot called Alex. Alex was shown to be able to learn a large number of quite complex tasks when trained in a very specific type of setup. Alex was motivated by food, but the computational problems he had to solve were similar to those of our artificial agents. Specifically Alex could not use the simulation theory of mind to figure out what a human wanted him to do (he could not think "what would I have meant if I was doing that"), since his cognitive architecture was so different from the humans he was learning from. This example is interesting for its similarity with artificial learners that do not share the human cognitive architecture (for example due to the technical difficulties in implementing such an architecture, or because it might be desirable to build a learner with a different type of mind). The experiments with Alex thus showed that it is possible to learn quite a lot of interesting skills without using the simulation theory of mind.

2.6 Other formalisms

Formalisms of imitation learning have taken two main forms: classification of tasks and mathematical formalisms.

In the classical work [64] an algebraic framework is specified and a success criterion is defined for imitation learning. The set of imitator and environment states is referred to as X, consisting of the states at each instance of a time series (where each time instance contain for example: states internal to the imitator, the fact that the learner is holding an apple, states in the environment, etc). The set of demonstrator and environment states is referred to as Y. Both X and Y are contained in the state set Z. Finally, success is defined as minimizing a distance metric $d : Z \times Z \rightarrow \Re$ (where 0 is optimal imitation). There are three ways in which this differs from the presented formalism; (i) it is not clear how such a distance metric should be obtained, as demonstrator evaluation is problematic, for example when a human demonstrator is not aware of everything that happened, (ii) the formalism in [64] does not include the possibility of other types of information sources besides demonstrations (which is an important part of the presented formalism), and (iii) the formalism in [64] cannot formalize the situation of non-optimal demonstrations (even given a correct framing and a perfect distance metric between imitator behavior and a demonstration, the situation where a demonstrator simply failed at achieving the task perfectly cannot be handled properly). Imagine, for example, a demonstrator trying to shoot a basketball at a hoop and failing most of the time. This is a situation an imitator can infer a goal from, especially given complementary information sources. Even if the imitator has identical embodiment, and is in an identical situation as what the demonstrator was in during a demonstration, it should not miss on purpose if it knows what the goal was and is able to achieve it, even if the demonstrator did miss in the same situation. But according to the formalism in [64], missing the shot in this situation is always an optimal action, no matter what the demonstrator thinks about this (as long as the shot is missed in the exact same way, it is optimal per definition).

The summary provided in [26] also offers a formalism for learning from demonstration. The demonstrations are seen as generated from a function mapping inputs to outputs, and the goal of the learner is defined as approximating that function. This leaves the question on how to do better than the demonstrator (see for example [22] for an imitator that outperforms the demonstrator). The question is discussed in [26] and one solution offered is to either filter out bad demonstrations or smooth them over with regression techniques. This diverges from the stated definition of success where the learner is to approximate the function that generates the demonstrations, and it does not deal with the case where the teacher is never able to achieve optimal performance (for example attempting to teach a robot how to throw a ball as far as possible, where the robot could in principle throw the ball much further than the teacher). The other approach presented is to seek feedback. This is strongly in line with the approach taken in the presented formalism where multiple sources of information are used by a learner to figure out what a teacher wants it to do, but seeking feedback falls outside of the [26] formalism. In [65] for instance, where reinforcement learning is used to improve on sub-optimal demonstrations, the success criterion of the reinforcement learning framework is used and the demonstrations are useful to the extent that they speed up learning (a learner that is only interested in maximizing a reward function could tackle the questions of “who" and “when" to imitate in a way similar to [25]).

In [66] an attempt to categorize imitation learning is made. The focus of this work is to classify various imitation learning tasks in terms of what type of goal the demonstrator would like the imitator to perform, for example replicating the exact movement, or replicating the end state of an object manipulated. This focus is not the same as what is attempted in the current paper, where defining success is an important aspect.

In [67], a formalism is presented for learning from demonstration. It deals with tele-operated robots in cases where the demonstrator has a clear understanding of its goal. Besides the fact that it is not restricted to tele-operated robots, the presented formalism is also more general in that it covers other types of information sources (eye gaze, a reward button, facial expressions, speech comments, EEG readings, etc) as well as demonstrations. The information spaces and related ideas are however quite similar, and the presented formalism can be seen as building on the ideas of [67].

The switch from a static interpretation of a teacher’s behavior to a parameterized hypothesis space, updated based on observations, is similar to how many have suggested moving from planning in a static world model, to re-estimating world dynamics based on observations. This has been detailed for example in [68] and [69]. A family of formalisms are proposed for dealing with unknown world dynamics. They all present different ways of describing how to model worlds with partially known dynamics, while the presented formalism describes how to model a very specific situation involving a human teacher’s behaviors, and is focused on finding out what should be done, instead of finding out how the world works. Hypotheses regarding how a human teacher’s behaviors relate to what should be done are updated instead changing hypotheses regarding world dynamics.

One of the formalisms covered in [68] and [69] is the Partially Observable Markov Decision Process (POMDP) formalism. POMDP is a way of trying to deal with uncertainty and lack of observability. The difference from the proposed formalism is that POMDP deals with uncertainty in the world, not uncertainty over observations of success. To demonstrate the difference an intermediate step is described, where the reward is a hidden state, affecting observations in a way that is not completely known. An extra hidden state H_r (hidden reward) would need to be added, and the reward signal removed from the observable space. The state H_r is now affected by states and actions, and it affects observable states. The learner has a prior over how states impact H_r, and a prior over how H_r impacts observable states. The goal of the learner is now to maximize H_r. Even though the learner might never be able to observe H_r perfectly, it can still make probabilistic updates regarding how H_r is affected (both by other hidden states, and by observable states), and how H_r affects observable states. It is also possible to take information gathering actions that are specifically chosen because, when those actions are taken, different hypotheses regarding how H_r interacts with other states implies different predictions for observable states.

3 What types of learners are being formalized

The existing research that the formalism attempts to cover includes any situation where a learner is trying to figure out what a teacher wants it to do. In order to cover all those types of situations, the learner **cannot** just perform the actions that the teacher would have performed, or the actions that would make the teacher say “good robot", or the actions that would make the teacher push a reward button, or maximize any other directly observable value. Finally, the learner does **not** have access to a sensor that tells it how successful it was. And it does **not** have access to a function over its inputs that specify how successful it was.

The typical learner that the formalism is aimed at is analyzing multiple sources of information at the same time and is refining its understanding of some of them, based on what it learnt from others. Let’s take the example of a learner that is able to see, but not fully understand, demonstrations, speech comments, facial expressions, a numerical value provided by the teacher, tone of voice, EEG readings of the teacher and eye gaze. If at least one of these is reasonably well understood, in at least some situations, it is possible to learn some types of tasks. If the learner is able to interpret some types of demonstrations, it can learn a task by looking at only this modality. Then it might be able to see from its history that some type of speech inputs are related to performance, and that if the teacher is observing the learner, then some facial expressions are often made right after the learner made a mistake during reproduction. If there is not enough data to decide how to interpret some speech comments, the learner might form multiple hypotheses. The ability to interpret facial expressions and speech can now be tested and refined in a new task and, if validated, can be used to learn new tasks. These new tasks might allow the learner to figure out that eye gaze at an object is correlated with it being important. The learner could also learn that when the teacher has observed all relevant aspects of the reproduction, the numerical value provided by the teacher is correlated with the performance of the action that the learner has just performed (relative to the average performance of a few of its most recent actions). New tasks allows new hypotheses for interpretation of information sources to be formulated, validated and refined. This, in turn, allows the learning of new tasks and perhaps the reinterpretation of old data. The learner could for example re-examine a large amount of old data in light of everything it has learned, and discover that it is very important to not bump into some object, and that when the learner does so, or is close to doing so, there will be a certain type of EEG reading. The learner could also figure out that, for this particular teacher, the numerical value is actually more related to policy similarity with good actions than with absolute performance. The learner could now use this understanding when learning new tasks, and it could also actively test these new ideas by performing actions that it expects to generate observations that will allow it to validate or invalidate them. The class of agents described above is the archetype that the formalism is designed to deal with, and it provides a general framework that is able to describe any agent that is trying to figure out what a human wants it to do. The formalism also tries to make as few assumptions as possible regarding the types of behaviors that are interpreted.

The formalism will be presented in a series of seven progressively more complex situations or setups. Some issues are easier to explain when other complexities are removed, and hopefully the first steps will convey some fundamental insights that will make it easier to describe the fully unsimplified setup. First, a set of simplifications are introduced which reduces this setup into an inference problem with a mathematically well defined success criterion, and these are then gradually relaxed. In the first setup, the learner is assumed to know how to interpret the teacher’s behavior, and the learner only needs to learn tasks while in the second setup this assumption is relaxed.

This is followed by five other setups where simplifications are removed step by step. At each step, the formalism is modified to deal with the new complexity, and new solutions are discussed. In the final, seventh step, the learner is dealing with an unstructured real-world situation.

4 A formalism for step one: finding $u^\star$

In the most simple setup the learner is required to output a policy based on a given data set of interactions. The teacher's utility function $u^\star$ is not known, but it can be estimated, and this estimate can then be used to find a good policy. A learner is located in a perfectly visible world with exactly one correct representation, known to the learner; we can say that the world has a single correct ontology, known to the learner). The learner is also perfectly observing the teaching signal. The teacher is also known to perfectly observe the world, and it has direct access to a utility function over world-action pairs, that takes everything into account (including all future consequences). This utility function maps states in the learner's action space and the world states to a real valued number (the situation can be roughly described as "the teacher knows what actions it wants the learner to take").

The learner also knows the mapping from what the teacher wants and what the teacher observes to states in a teaching signal space. A simple teacher giving demonstrations could, for example, be of the form "the teacher gives demonstrations which are perfect with probability 0.8 and otherwise the teacher performs random actions" or "the teacher rewards incremental progress and gives a scalar feedback value feedback equal to the utility of the current action minus the average utility of the learner's 6 previous actions". Knowing this mapping allows inference even if it is stochastic as each possible utility function results in a probability or a probability density for the actually observed feedback. To be more specific, it is necessary to first introduce some notation:

- **World state** $s = (x_1, x_2, ..., x_{N_S}) \in C^s$. An N_S dimensional vector in (in $\Re^{N_S}$ if it is continuous and unbounded), describing the state of the world. C^s is the space of possible world states. The learner has direct access to the world state in this step.
- **Action** $\alpha = (y_1, y_2, ..., y_{N_\alpha}) \in C^\alpha$. An N_α dimensional vector describing a learner action.
- **Policy** $\pi \in C^\pi : C^s \rightarrow C^\alpha$. Since the world is fully visible, the learner's policy is definable as a transform from world states to actions.
- **Situation** $\Xi = \{s, \alpha\} \in C^\Xi$. A world state and a learner action. This is what the teacher will respond to by giving a teaching signal.
- **Teaching signal** $f = (z_1, z_2, ..., z_{N_F}) \in C^f$. An N_f dimensional vector describing the teaching signal response to a setup Ξ, for example a demonstration of what action should have been performed, a speech comment on the learner's action, a scalar value evaluation of the action, the eye gaze towards an important object, etc.
- **Interaction** $I = \{\Xi, f\} \in C^I$. A setup, and the feedback that was given in that setup (a world state s, a learner action α, and the feedback f that was produced by the teacher as a response).
- **Interaction history** $h = \{I_1, I_2, ...\} \in C^h$ is a set of interactions. To refer to an element E in h of interaction number t, we use E_t, for example: Ξ_t and f_t (the setup and response at interaction t). Note that the interaction history consists of states in spaces that are observable to the learner.
- **Learning algorithm** $Y \in C^Y : C^h \rightarrow C^\pi$. An interaction-history-to-policy-transform. Since the learner's job in this step is only to output a policy as a response to data, a learner is defined by an Y and a history h. A learning algorithm/learner can be defined using an iterative update rule, modifying a policy based on one interaction at a time (since this recursively implies a unique Y). If Y is stochastic, a current policy π is also needed to define the learner [4].
- **Utility function** $u \in C^u : C^s \times C^\alpha \rightarrow \Re$. Mapping world state-action pairs to a real number (expressing preferences over the action space, conditioned on the current world state).
- **Teacher's utility function** $u^\star \in C^u : C^s \times C^\alpha \rightarrow \Re$. The teacher is assumed to have access to a utility function $u^\star$ that represents exactly what the teacher would have wanted if it were fully informed (for example regarding future consequences of the action). The sole evaluation criterion of the success of the learner is: $E[u^\star(\hat{\pi}(s_R))]$, where s_R is a randomly generated world state (the expected utility of its policy $\hat{\pi}$ when the state of the world is not known). Finally, the learner is assumed to know this fact (although it does not know $u^\star$).
- $u^\star$ **generating distribution:** $\mathcal{D}^{u^\star} : \Theta^{u^\star} \rightarrow \Re$. The utility function $u^\star$ is drawn from a distribution known to the learner. A known function class has a parameter

4 There is no possibility to choose actions in order to get informative feedback.

space Θ^{u^*}, and each possible state is assigned a probability, or probability density, by the known distribution $\mathcal{D}^{u^*}$. This distribution could for example be over discrete outcomes, or a density function over the continuous parameter space of a function class, or a probability distribution consisting of a density function over a continuous space as well as a set of Dirac deltas for certain values in that space, etc. We denote the utility function that parameter θ^{u^*} generate as $u(\theta^{u^*})$.

- **Teacher signal generating transform** $\Omega \in C^{\Omega} : C^{\Xi} \times C^h \times C^u \rightarrow C^f$. A stochastic transform[5] from the current situation Ξ, the interaction history h, and a utility function u into feedback f (the current world state, the current learner action, a utility function and the interaction history determines what distribution a state in the teaching signal space is drawn from). In this step the learner is assumed to have access to this transform (the next section formalizes a setup where this assumption is relaxed).

As the generating distribution $\mathcal{D}^{u^*}$ over possible u^*s is given, what is needed to get the posterior probabilities of possible u^*s is the probability or the probability density of the observed feedback conditioned on all the different u^* hypotheses. Since $\Omega : C^{\Xi} \times C^h \times C^u \rightarrow C^f$ is known and the states in all the other input spaces are known, the probability of the observed feedback is only dependent on u^*. If the probability density of observing the feedback f_t at interaction t is denoted p^{f_t}, and $\mathcal{D}^{u^*}$ is a density function over a continuous Θ^{u^*} space, we get the following equation:

$$p^{f_t} = \int_{\theta^{u^*} \in \Theta^{u^*}} D^{u^*}(\theta^{u^*}) D(f_t | h(t), \Xi_t, u(\theta^{u^*})) d\theta^{u^*} \qquad (1)$$

If p^{f_t} is the probability of observing the feedback f_t at interaction t, and $\mathcal{D}^{u^*}$ is a probability distribution over a discrete space Θ^{u^*} with N_{u^*} number of hypotheses u_i, and the prior probability that $u_i = u^*$ is denoted $p_i^{u^*}$, then we have:

$$p^{f_t} = \sum_{i=1}^{N_{u^*}} p_i^{u^*} p(f_t | h(t), \Xi_t, u_i) \qquad (2)$$

Now, if the probability or probability density for observing f_t is denoted p, the posterior probability p_{px} of u^* hypothesis number x being correct is simply $p_{px} = p_{pa} p / p^{f_t}$, where p_{pa} is the a priori probability of u^* hypothesis number x being correct. The update factor p/p^{f_t} basically measures how good the hypothesis was at predicting the observed feedback compared to other plausible hypotheses.

Since Ω is a known mapping and the probability distribution over possible u^*s is given, finding the posterior distribution over possible u^*s given a history h has thus been cast as an inference problem. This is not a solved problem, but the fact that there exists a research community dealing with it means that we do not have to re-invent the wheel. Finding an optimal policy π given a finite history h is now a matter of maximizing the expected utility function (the weighted sum of all u^* hypotheses, or the integral over u^* space). See, for example, particle swarm optimization [72] or various methods for approximate Bayesian inference [71]. u^* is defined in the action space given the observable world state, so the exact expected utility (given the known prior distribution over u^*, and the fully observable history) of each action is known to the learner. It could be that even if Ω is known, finding the optimal solution is intractable, necessitating the need for approximate solutions. As simplifications are dropped in later sections, intractability will become an increasing problem, and much effort will be put into discussing approximate solutions.

Since the problem has been formalized in this way, standard ideas and principles of approximate optimization can be used to find approximate solutions. If Θ^{u^*} is continuous and high dimensional, and there is a large history, then one possibility is creating a number of discrete u^* hypotheses, each with its own set of parameters. Then the probability of these hypotheses, and their parameters are updated iteratively, one interaction at a time. The probability that a hypothesis is correct is modified in each iteration, based on how well it predicted the actual teaching signal, and the parameters are modified so that it better predicts the observed history. Hypotheses that are very unsuccessful at predicting teaching signals from unobserved interactions can be rejected, and new ones constructed by doing alternate parameter modifications on good hypotheses. The parameter updating of each hypothesis takes the parameters of other hypotheses into account, to avoid crowding in small areas and/or move towards regions that are good/highly populated.

The point is not that all problems within the framework can be solved. It is instead that they can be cast as an instance of a well-studied type of problems. Standard ideas can then be used to solve these problems (in the ex-

5 This could denoted as $\Omega \in C^{\Omega} : C^{\Xi} \times C^h \times C^u \times C^f \rightarrow \Re$, but to emphasize that the thing generated is a teaching signal, the notation of a stochastic transform is used where $a : b \rightarrow c$ means that a stochastically generates states in c according to a distribution that is dependent on states in b.

ample above, using the basic ideas of particle swarm optimization).

5 A formalism for step two: finding Ω

In this step the learner must learn to interpret the feedback of the teacher. Specifically, the Ω transform is no longer known, but is drawn from a known distribution, and must be learnt in a way that is similar to how $u^\star$ was learnt in the previous step. Two different Ω hypotheses will in general give different probabilities, or densities, for an observed interaction history This again reduces the problem to an inference problem of a well-studied form (so that ideas from proposed approximate solutions can be used). New practical difficulties that arise, and new approximate strategies to deal with them will be discussed below.

The parameters of a known stochastic function class are drawn from a known distribution. Any parameter set results in a static (but not necessarily deterministic) mapping from an interaction history, the utility function and a current world-action pair to an output in a teaching signal space. Examples of this include: “if I demonstrate something, and then the learner reproduces it incorrectly, I demonstrate again", “if the learner is doing better than usual, I will press a plus button" or “if the learner fails a lot and look like it needs encouragement, I will push a plus button[6]".

As in the previous step, the learner’s job is to output a policy based on a given data set (an interaction history of known length). This means that the learner still does not have to deal with the problem of choosing actions in a way that trades off the maximization of information with actually performing the task. First some additional notation is needed:

- Ω **generating distribution:** $\hat{D}_{\hat{\theta}^\Omega} : \hat{\Theta}^\Omega \rightarrow \Re$. Ω is drawn from a distribution known to the learner. A known function class has a parameter space $\hat{\Theta}^\Omega$, and each possible state $\hat{\theta}^\Omega \in \hat{\Theta}^\Omega$ is assigned a probability, or probability density by the known distribution $\hat{D}_{\hat{\theta}^\Omega}$. This distribution could, for example, be: over discrete outcomes; a density function over the continuous parameter space of a function class, or a probability distribution consisting of a density function over a continuous space, as well as a set of Dirac deltas for certain values in that space, etc.
- Ω **estimate distribution** D^Ω. If the learner builds a model of Ω with parameters, then distribution over this space is denoted D^Ω (due to tractability issues, this does not have to be the same as $\hat{D}_{\hat{\theta}^\Omega}$). We denote the resulting Ω of parameter θ_k^Ω as Ω_k.

Just as in the previous step, the problem is to define an Y, and success is measured in how well the resulting policy optimizes $u^\star$. Since the prior probability of each possible feedback generating transform is known, and the prior probability of each possible utility function is known, for each interaction history $h \in C^h$ there is at least one policy π such that the expected utility is maximized. That is: there is at least one optimal policy that, given the known information, will give maximum expected utility, and finding it is an inference problem of a well explored type. Below, two examples are presented, and then generalized approximate solution strategies are discussed since intractability is a likely practical problem. The problem of finding a $u^\star$ from a partially known Ω is similar to finding $u^\star$ from a known stochastic Ω, as a set of stochastic Ω hypotheses (weighted by probability) reduces to a single stochastic function. The practical difference is that in an approximate solution, it is possible to update Ω hypotheses concurrently with $u^\star$ hypotheses in an EM inspired way.

To illustrate how the various algorithms that will be defined for re estimating interpretation hypotheses interact with other elements of the learner, two very simple “algorithms" are described. They show the step between the later algorithms and static interpretations of teacher behavior. The first “algorithm" is designed to deal with the case where human teachers are known to belong to one out of a small set of possible types. A learner starts with one interpretation hypothesis for each type. Each hypothesis has a set of “typical interaction histories" attached to it (provided by the programmers), and after a certain pre-

6 “Studies have shown that there is sometimes a motivational component to the rewards given by human subjects [3]. It does not seem like the behavior of giving positive rewards to bad states is always caused by failure to observe the world or lack of knowledge about what the correct action is, but is sometimes a way of “encouraging the robot". Therefore this possibility is still relevant to the setup presented. A human learner is capable of noticing this type of teaching signal (for example tone of voice in combination with a partial understanding of the task) and is able to take this into account when making policy updates. Thus, an artificial learner should in principle be able to do the same by, for example: (i) first failing at a task where the goal is known, then (ii) noticing that there is a statistical pattern in tone of voice space correlated with “failures getting positive feedback", (iii) confirming the theory in a unrelated setting, (iv) building a detailed model of when this happens, and with what probability, and finally (v) using this during learning in novel settings by keeping track of the probability that a particular teaching signal instance was generated like this (and take that into account during policy updates).

defined set of interactions, the observations are compared to these histories, and the one that is closest is selected and is permanently assumed accurate. This type of interpretation hypothesis selector is clearly not optimal for all possible situations, but it does demonstrate how it is possible to improve over the static assumption situation. Given the current best estimate of $u^\star$, the second "algorithm" assigns a gold star to the hypothesis whose observation prediction is closest to the actual prediction. The hypothesis with the largest number of gold stars at any given point is used to update $u^\star$. This represents an even larger improvement over the case with a static interpretation. Since we have described the situation in a standard form, it is however possible to directly tap into a large amount of existing research. In the following sections, example algorithms will be designed with a particular focus on using well established methods, and using very well-known ideas. The two central sources are famous forms of EM, and the idea of tracking several parameterized solutions. One family of algorithms well suited for dealing with this problem is Estimation of Distribution Algorithms (EDA, see [70] for a survey). But other types of solutions can of course also be used, including ad hoc solutions designed for specific implementations, well established standard methods with solid theoretical foundations, or new and innovative algorithms.

5.1 An example with a discrete set of possible teachers

$\hat{\Theta}^\Omega$ consists of N discrete possibilities, denoted $\theta_1^\Omega, \theta_2^\Omega, \ldots, \theta_N^\Omega$, where each θ_n^Ω results in a unique transform Ω_n. We denote the probability that $\Omega_n = \Omega$ as p_n^Ω. Before observing the interaction history the learner's estimate of Ω is in this case identical to the generating distribution $\hat{\Theta}^\Omega$ (if the generating distribution had been to difficult to handle computationally, the learner's initial estimate could have been something simpler). The learner observes a single interaction $I^1 = \{\Xi^1, F^1\}$.

The learner also has M hypotheses of what $u^\star$ looks like (also initialized with the generating distribution), $\theta_1^{u^\star}, \theta_2^{u^\star}, \ldots, \theta_M^{u^\star}$, where each hypothesis is denoted $u_m^\star$. Since there is a discrete set of $u^\star$ and Ω hypotheses as well as a single single interaction $I^1 = \{\Xi^1, F^1\}$, the probabilities of the $u^\star$ hypotheses can be updated with a simple equation. The probability of observing the feedback F^1 given the history, setup, hypothesized $u^\star$ and hypothesized θ_n^Ω is denoted $p(F^1|\Xi^1, h, u_m^\star, \Omega_n)$. We have $M \times N$ possibilities, each corresponding to a utility function-transform-hypothesis pair. Each pair has a prior and each assigns a probability to observing the actually observed feedback. This means that each pair can be assigned a posterior probability. For transform hypothesis n and utility function hypothesis m the posterior pair probability is $p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n)$

Thus the probability (after updating on the new observation F^1) of each utility function hypothesis and transform hypothesis is given by simply summing the posterior pair probabilities. We denote the probability at time step t as ${}^t p_m^{u^\star}$ so that the probability ${}^2 p_m^{u^\star}$ is the probability that utility function hypothesis number m is correct, after updating on observing F^1. ${}^2 p_m^{u^\star}$ is thus simply the sum:

$$ {}^2 p_m^{u^\star} = p_m^{u^\star} \frac{\sum_{n=1}^N p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n)}{\sum_{m=1}^M (p_m^{u^\star} \sum_{n=1}^N p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n))} \quad (3) $$

And in just the same way we have the new probability ${}^2 p_n^\Omega$ (the new probability that transform hypothesis number n is correct, after updating on observing F^1) in the sum:

$$ {}^2 p_n^\Omega = p_n^\Omega \frac{\sum_{m=1}^M p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n)}{\sum_{n=1}^N (p_n^\Omega = \sum_{m=1}^M p_n^\Omega p_m^{u^\star} p(F^1|\Xi^1, h, u_m^\star, \Omega_n))} \quad (4) $$

5.2 An example with a continuous space of Ω parameters

Now we take the exact same setup, but we have a continuous parameter space of possible Ω transforms. The exact same reasoning applies when it comes to updating the discrete set of ${}^2 p_m^{u^\star}$, with the only difference being that integrals replaces sums, so that we get:

$$ {}^2 p_m^{u^\star} = p_m^{u^\star} \frac{\int_{\Theta^\Omega} D_{\theta^\Omega} p_m^{u^\star} D(F^1|\Xi^1, h, u_m^\star, \theta^\Omega) d\theta^\Omega}{\sum_{m=1}^M p_m^{u^\star} \int_{\Theta^\Omega} D_{\theta^\Omega} p_m^{u^\star} D(F^1|\Xi^1, h, u_m^\star, \theta^\Omega) d\theta^\Omega} \quad (5) $$

If the parameter space of possible transforms is high-dimensional (that is, there are many ways in which the teacher's feedback behavior could vary), this integral might be completely intractable. But the problem has been cast in a more standard form. And the question of "what is being approximated" has been clarified.

One approximate solution is to create a set of hypotheses for how feedback is generated, test them against data, and continuously modify them, discard them or create new ones. We need a hypothesis generating algorithm, an algorithm that tests and modifies or discards hypotheses and an iterative procedure for concurrently updating the $u^\star$ hypotheses and the Ω hypotheses. Let's call such an Ω

hypothesis an interpretation hypothesis (as it is a hypothesis regarding how feedback should be interpreted) and denote interpretation hypothesis number i as Π_i.

To test the quality of a hypothesis in this example (with a single data point), we cannot do better than check how well the transform model predicts the observed behavior (and of course look at the known density function of transform generators). Any test will be strongly dependent on the current estimate of u^*, since the feedback is dependent on both u^* and Ω.

Given a test that rates a Π_i conditioned on the current best guess of the u^*, we can update our set of p^{u^*} probabilities based on the current set of Πs, concurrently with updating our set of Πs based on the current set of p^{u^*}. This shows how the old ideas behind various EM algorithms can be used when the imitation learning problem has been reduced to this form. The basic idea that can be taken from these algorithms is that when there are two unknowns, and knowing one helps finding the other, updating both concurrently can lead to a functioning and tractable algorithm. An example is when a set of points is known to be generated by a known number of Gaussian distributions of with unknown parameters. Knowing which generator produced which points help when estimating the parameters of the generators. And knowing the parameters of the generators helps when estimating which points where generated by what generator. See for example Dempster and Laird's 1977 paper [10] presenting an EM algorithm.

5.3 An example with continuous parameter spaces and a large number of interactions

To illustrate the problem faced by a learner in this step, a more specific setup and solution strategy is introduced. Standard solutions are used to solve the formalized problem in order to be specific and to illustrate the basic concepts.

Consider a problem where the teacher is known to have been drawn from the high-dimensional, independent distributions D_Ω and D_{u^*}, and where there is a large history h to learn from. It is in principle possible to find the posterior D_{u^*} conditioned on the a priori D_Ω and the history, but let's look at a class of tractable approximate solutions. We need a bit of notation:

- $\hat{U} = \{\hat{u}_1, \hat{u}_2, ...\}$: The set of discrete hypotheses regarding u^*.
- $\hat{\Omega} = \{\Pi_1, \Pi_2, ...\}$: The set of interpretation hypotheses.
- $h(t) = \{I_1, I_2, ..., I_t\}$: The history up until time t.
- D^{u^*}: The a priori density function over the possible utility functions that the teacher might have.
- $GenerateNew - \Omega - Hypotheses(h(t), \hat{\Omega}, \hat{U}, D^\Omega)$. An algorithm that generates Ω hypotheses. If the set of hypotheses $\hat{\Omega}$ has any empty slots (either due to not being initialized or due to some Πs having been discarded), this function needs to create hypotheses Π that makes a tradeoff between being probable according to the prior probability D^Ω, being consistent with the data $h(t)$ (where the accuracy of the consistency estimate is dependent on the current estimate $\hat{U}$ of the teacher's utility function u^*) and being well distributed in the space (these Πs will be modified as a response to data, so several in the same small region could be wasteful as they might converge to the same point).
- $GenerateNew - u^* - Hypotheses(h(t), \hat{U}, \hat{\Omega}, D^{u^*})$. The tradeoffs are similar with the situation detailed above, and in this case the accuracy of the consistency estimate is dependent on the current estimate $\hat{\Omega}$ instead of $\hat{U}$.
- $Discard - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$. The Πs have been modified without any access to the interaction I_t, so it is suitable to test them. If a hypothesis predicts new observations badly enough compared to the others, it can be eliminated. Another reason to eliminate a hypothesis is that the modification process has made it too similar to another hypothesis. As before, the accuracy of the consistency estimate is dependent on the quality of the current $\hat{U}$ estimate.
- $Discard - u^* - Hypotheses(I_t, \hat{U}, \hat{\Omega})$. The same concerns as above apply in this step.
- $Modify - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$. This algorithm updates the set $\hat{\Omega}$ of interpretation hypotheses based on the new interaction I_t and the current estimate $\hat{U}$ of the utility function. As the quality of the update is dependent on the accuracy of the current $\hat{U}$ estimate, it makes sense to update both estimates concurrently a few times in an EM inspired way.
- $Modify - u^* - Hypotheses(I_t, \hat{U}, \hat{\Omega})$. This is the other half of the above mentioned EM pair.

With these functions we can build an iterative algorithm, (see Algorithm 1) that could hopefully approximate the integrals, while remaining tractable.

Input: D^{u^*}, D^{Ω}, T, $h(T)$
• D^{u^*}: *The probability distribution that the teacher's utility function* u^* *is known to be drawn from.*
• D^{Ω}: *The distribution that the teacher's feedback generating transform is known to be drawn from.*
• T: *The number of interactions in the history.*
• $h(T)$: *The history of interactions.*
$\hat{U} \leftarrow GenerateNew - u^* - Hypotheses$
$(h(0), \hat{U}, \hat{\Omega}, D^{u^*})$
$\hat{\Omega} \leftarrow GenerateNew - \Omega - Hypotheses$
$(h(0), \hat{\Omega}, \hat{U}, D^{\Omega})$ **for** $t = 1$ ***to*** T **do**
  $\hat{\Omega} \leftarrow Discard - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$
  $\hat{U} \leftarrow Discard - u^* - Hypotheses(I_t, \hat{U}, \hat{\Omega})$
  $\hat{\Omega} \leftarrow$
  $GenerateNew{-}\Omega{-}Hypotheses(h(t), \hat{\Omega}, \hat{U}, D^{\Omega})$
  $\hat{U} \leftarrow GenerateNew - u^* -$
  $Hypotheses(h(t), \hat{U}, \hat{\Omega}, D^{u^*})$
  while *Stopping criterion not met* **do**
    $\hat{U} \leftarrow Modify - \Omega - Hypotheses(I_t, \hat{\Omega}, \hat{U})$
    $\hat{\Omega} \leftarrow Modify - u^* - Hypotheses(I_t, \hat{U}, \hat{\Omega})$
    end
  end

Algorithm 1: Approximate solution to example 3

5.4 An example solved by a multiple-generator-based algorithm

The teacher is approximated as having a number of different teaching signal generators or interaction protocols denoted Γ. In each situation the teacher selects one based on the interaction history and the current setup.

Ω is approximated as a combination of generators $\Gamma : C^{\Xi} \times C^h \times C^u \rightarrow C^f$. Roughly speaking, the teacher is approximated as having several ways in which it can interact, and as choosing which way of interacting (choosing which Γ will generate the teaching signal). More precisely, each Γ has a probability to be activated that is state space dependent, and is otherwise a stochastic transform of the same class as Ω, mapping the same input spaces to the feedback space C^f. Γs are not hypotheses in the sense of the Πs mentioned above since Ω is not hypothesized to be equal to any one Γ, Ω is instead modeled as built up by a set of Γ transforms. The proposed algorithm concurrently estimates how each generator will produce teaching signals, in what type of situations a generator is used (encoded as a triggering region in situation space for each generator), and what data was generated by what Γ. This is done in a way that is very similar to old and well-known EM methods (just as in the examples discussed above). First some additional notation:

- **Feedback generator number** n**.** $\Gamma^n :\in C^{\Gamma} : C^{\Xi} \times C^h \times C^u \rightarrow C^f$. Generator number n that Π is built from.
- Θ^n_{Γ} is the parameter space of Γ number n.
- $D^t_{\Theta^n_{\Gamma}}$ is the probability density function over Θ^n_{Γ} at time t (the current estimate of the parameters of Γ^n).
- $\mathcal{D}^t_{\Gamma} = (D^t_{\Theta^1_{\Gamma}}, D^t_{\Theta^2_{\Gamma}}, \ldots, D^t_{\Theta^N_{\Gamma}})$ is a set of Γ parameter estimates.
- **Generating tendency** $G^n : C^{\Xi} \times C^h \times C^u \rightarrow \Re$. The tendency of basis function number n to generate the feedback. The probability that Γ^n will generate feedback is $\frac{NG^n(\Xi,h)}{\sum_{m=1}^{N} G^m(\Xi,h)}$ (the probability that a specific observed feedback was generated by Γ^n depends on the type of feedback that Γ^n tends to generate and what the generating tendency in the current state is).
- ${}^G\Theta^n$ is the parameter space of G^n.
- $D^t_{{}^G\Theta^n}$ is the probability density function over ${}^G\Theta^n$ at time t (the current estimate of the parameters of the generating tendency G^n).
- $\mathcal{D}^t_G = \{D^t_{{}^G\Theta^1}, D^t_{{}^G\Theta^2}, \ldots, D^t_{{}^G\Theta^N}\}$ is the estimate at time t of the generating tendencies.
- Θ^{u^*} is the parameter space of u^*.
- $D^t_{\Theta^{u^*}}$ is the probability density function over Θ^{u^*} at time t (the current estimate of the parameters of u^*).
- **Generating probability** p^t_n**:** The estimated probability that the feedback f^t, observed at time t, was generated by Γ^n.
- **Generated feedback** y^n**:** The current estimate $y^n = \{p^1_n, p^2_n, p^3_n, \ldots\}$ of what feedback was generated by Γ^n.
- $\mathcal{P}^t_y = \{y^1, y^2, \ldots, y^N\}$**:** the estimate at time t of what feedback was generated by what Γ.

It is now possible to concurrently re-estimate: (i) The feedback behavior of the basis functions, (ii) their generating tendency, (iii) the set of feedback instances that was generated by each Γ, and (iv) the utility function u^*. We can see this in Algorithm 2, which is in turn based on the following sub-algorithms:

- ${}^{s+1}\mathcal{P} \leftarrow estimateGen({}^s\mathcal{P}, {}^s D^{u^*}, h, {}^s\mathcal{D}^{\Gamma}, {}^s\mathcal{D}^G)$**:** Given the model at step s of u^* and the Γs, the estimate of wich Γs generated which feedback is updated. The previous step updated the feedback behavior ${}^s\mathcal{D}^{\Gamma}$, the generating tendencies ${}^s\mathcal{D}^G$ and the estimate $D^t_{\Theta^{u^*}}$ of u^*. Given these new estimates, *estimateGen* must approximate the probability that a given basis function was the one that generated the feedback, under the new u^* estimate. Which Γ generated the feedback is straightforwardly dependent on where that Γ is expected to generate feedback, and on how probable a

Γ is to generate the observed type of feedback (conditioned on the new $u^\star$ estimate).

- ${}^{s+1}D^{u^\star} \leftarrow update - u^\star - estimates({}^{s}D^{u^\star}, {}^{s+1}\mathcal{P}, {}^{s}\mathcal{D}^{\Gamma})$**:** The estimate of which Γ generated the data and the feedback behavior of those Γs has been updated, which means that the data can be re-interpreted and used to update $u^\star$. Given a fixed current model of Ω (consisting of a set of Γs), this reduces to a standard form of supervised learning problem with data that has a known noise structure (the uncertainty of the Ω estimate and the stochastic nature of the various Γs). Again, part of the problem is reduced to a well-studied type of sub-problem.
- ${}^{s+1}\mathcal{D}^{\Gamma} \leftarrow update - \Gamma - estimates({}^{s}\mathcal{D}_{\Gamma}, {}^{s+1}\mathcal{P}, {}^{s+1}D^{u^\star})$**:** The estimates of what data a given Γ has generated has been updated, but also in what situation it was generated, since $u^\star$ has been updated. For example: a learner has observed a “good robot" comment when shooting a basketball close to a hoop. If the learner manages to figure out that the teacher only cares about whether or not a basketball lands inside or outside a hoop, then the learner can re-interpret the feedback-generating function. Specifically it can figure out that a failed attempt gets a “good robot" comment if the outcome is closer to good outputs than previous attempts, instead of for example rewarding incremental increase in performance (which would have been more likely if the teacher had instead wanted something like “shoot as close as possible to the hoop"). Given the known world state, and taking the current estimate of $u^\star$ for granted and, the current estimate of which points were generated by which Γ for granted, this reduces to a function-approximation problem of a known form. Each Γ can have its own type of parameter space but the basic idea is still that of “freezing" all the other estimates and using them to update the generators (it is not necessary to do the same type of update for each generator).
- ${}^{s+1}\mathcal{D}^{G} \leftarrow estGenTend({}^{s}\mathcal{D}^{G}, {}^{s+1}\mathcal{P}, h, {}^{s+1}D^{u^\star})$**:** Given the current best estimate of which generator actually generated which point, this is a supervised learning problem with a labeled data set.

The algorithm rests on the same principle as building a Gaussian Mixture Model (GMM) with an EM algorithm that concurrently estimates which data points was generated by which Gaussian (based on the current estimated properties of the Gaussians), and estimating the properties of that Gaussian (based on the current estimate of which points they generated). The algorithm illustrates how a vague problem to “do what the teacher intended the learner to do" has been formalized to the point where the exact solution integrals can be set up, so that tractable approximation can be found using standard techniques, see for example Dempster and Lairds paper [10] from 1977, explaining an EM method based on a very similar idea[7], and solving a very similar problem. It is doing essentially the same thing as all the classical EM algorithms (even though the Γs can have different parameter spaces). The purpose of presenting this algorithm is not to present a general solution strategy to any imitation learning problem, but simply to demonstrate that the problem has now been formalized to the point where old standard ideas can be used. The question of solvability is now dependent on factors similar to those that determine whether or not the classical EM algorithms would find a solution (dimensionality, size and quality of the data set, etc).

Input: $\mathcal{D}_0^G$, $\mathcal{D}_0^\Gamma$, $D_0^{u^\star}$, h, S
• $\mathcal{D}_0^G = \{D_0^{G1}, D_0^{G2}, \ldots, D_0^{GN}\}$ *is the initial estimate of the generating tendencies (at time 0).*
• $\mathcal{D}_0^\Gamma = \{D_0^{\Gamma 1}, D_0^{\Gamma 2}, \ldots, D_0^{\Gamma N}\}$ *is the initial estimate of the feedback behaviors.*
• $D_0^{u^\star}$ *is the initial estimate of* $u^\star$ *(a probability distribution)*
• h *is the interaction history*
• S *is the number of update steps*
for $s = 1$ ***to*** S **do**
  ${}^{s+1}\mathcal{P} \leftarrow estimateGen({}^{s}\mathcal{P}, {}^{s}D^{u^\star}, h, {}^{s}\mathcal{D}^{\Gamma}, {}^{s}\mathcal{D}^{G})$
  ${}^{s+1}D^{u^\star} \leftarrow update - u^\star - estimates({}^{s}D^{u^\star}, {}^{s+1}\mathcal{P}, {}^{s}\mathcal{D}^{\Gamma})$
  ${}^{s+1}\mathcal{D}^{\Gamma} \leftarrow update - \Gamma - estimates({}^{s}\mathcal{D}_{\Gamma}, {}^{s+1}\mathcal{P}, {}^{s+1}D^{u^\star})$
  ${}^{s+1}\mathcal{D}^{G} \leftarrow estGenTend({}^{s}\mathcal{D}^{G}, {}^{s+1}\mathcal{P}, h, {}^{s+1}D^{u^\star})$
end

Algorithm 2: A multiple generator algorithm to solve the problem in example 4

Let’s give a few examples of generators that might be used to model actual humans:

- **Demonstrating as a response to failed reproduction attempt Γ^1:**. The triggering region G^1 would be where a demonstration was followed by a failed reproduction attempt, and could have parameters relating

7 In both cases there is a data set produced by a set of generators. The properties of the generators are not known, but if they were known, it would be possible to estimate which generator generated which data point. This is not known, but if it were, it would allow us to determine the properties of the generators. Both things are given initial estimates, and then the estimates are updated concurrently (conditioned on the best current other estimates).

to how badly the demonstration has to fail, or if the relevant distance is in policy space or outcome space: a close basketball trow can be very close to optimal policy, but still have an outcome that is no better than any other failed attempt. If a reproduction needs to be far from optimal in policy space to elicit another demonstration, this would not be within the triggering region. G^1 could also contain an arbitrary amount of other parameters, such as the behavior being more likely when the task is easy for the teacher to perform, or when the teacher looks irritated directly after a learner reproduction attempt, etc. The irritated facial expression could be either a single value attached to the binary output of a fixed "irritated facial expression detector" (multiplying the triggering tendency with the parameter value for instance) or it could include parameters regulating what counts as "irritated facial expression" (or more technically "facial expression that is correlated with triggering Γ^1"). Θ^1_Γ describes the feedback generating behavior, and could include parameters of how many mistakes the teacher makes. It could, for instance, be that the teacher does this only when irritated, which is correlated with tasks that are simple for the teacher, which correlates with good performance (lower noise than other demonstrations).

- **Verbally evaluating progress Γ^2:** Saying things like "Good robot","No!", or "Great!" based on the learner's performance relative to its recent interaction history. Parameters could include the length and weigthing of the recent history, the strength of the different words (does "Great!" indicate better performance than "Good robot", and if so, how much better?), the parameters of how to map speech input to a set of pre-defined categories (with pre-defined interpretation), the parameters of a transform from speech space to evaluation space, etc, etc. Γ^2 could include parameters regarding how much more likely this feedback behavior is in the case of eye contact, or in the case of a long interaction history consisting of the same types of actions, etc, etc (speech could for example be more likely to be relevant in the case of eye contact).
- **Pushing a reward/punish button based on absolute and relative performance Γ^3:** The reward button is pushed with a value based on: (i) how good the outcome is in an absolute sense, (ii) how good the outcome is compared to recent history, (iii) how close the action was in action space to good actions compared to recent history. The triggering of this behavior could be dependent on anything from the number of demonstrations made to the attitude (angry, happy, etc) of the teacher, leading to a large number of possible parameters of G^3. Θ^3_Γ could include a value defining what constitutes "recent history", and the relative weighting of the different considerations.
- **Pushing a reward/punish button to punish the robot for breaking something Γ^4:** Maximal punishment and a surprised and angry facial expression indicate that something was broken, which can help with credit assignment (the problem was not that the basketball was far from the hoop, it was that the basketball went through the window).
- **Pushing a reward button to encourage a robot that has failed a lot and who looks sad Γ^5:** The generating tendency G^5 can have parameters related to teacher facial expressions and eye contact (for example a distribution encoding something like: "Γ^5 was not the generator if the immediate teacher response after looking at the outcome of the learner action was a triumphant smile and a "Great!" speech utterance"). This type of feedback is actively harmful to the learner's ability to figure out what the teacher wants it to do, but it is still important for the learner to understand this behavior so that it can classify feedback as having been generated by Γ^5. If the feedback was likely to have been generated by Γ^5, it can for example be ignored, which is already a big improvement compared to updating policy as if it indicated success.
- **Looking at an object that the learner interacted with badly Γ^6:** When the learner fails, and one particular object is important for that failure, the teacher will tend to look at that object.

The idea behind the algorithm is also similar to Simultaneous Localization And Mapping (SLAM) in that knowing what position a robot had at each time step will allow the building of a good map, and knowing the map makes finding the positions much easier. The analogy with a robot moving around and trying to build a map and at the same time figuring out where it is within that map is useful as it makes the idea of active information gathering obvious (the robot can move to different places, or just direct its sensors as a way of testing competing hypotheses regarding both what the area looks like, and where it is within that area). This solution strategy will be discussed in simplification step three, where the data set is not fixed, and the learner must take actions for the purpose of obtaining maximally useful data. This will lead us to another old field known as optimal experiment design. As the resulting "expected information gain integrals" will normally be intractable, we will be making contact with various approximate methods for finding actions that result in good information, for example using biological systems for inspi-

ration, and described in terms such as artificial curiosity. Intuitively, finding good strategies for gathering informative data seems to be a central question, at least as important as analyzing that data. Even though this research area is very active and making progress, it is not close to finding neat solutions applicable to any problem, meaning that one might have to dig into all the messy details and integrate a specially designed solution to the active information gathering problem into the full learner architecture from the beginning. A neat, of-the-shelf and fully general solution that can be plugged in as a separate module will probably not be available.

6 Step three: allowing the learner to actively gather valuable data.

Let's allow the learner to choose information-gathering actions, allowing it to actively collect the data that will allow the learner to distinguish between competing interpretation hypotheses. For example, if one hypothesis is that the teacher is giving rewards corresponding to performance, and another hypothesis is that the teacher is giving rewards in response to incremental improvements (similar to how one does when training a dog for example), then repeating an action can help the learner distinguish between these hypotheses (as they make different predictions in the observable reward space). Actions can be chosen in order to understand the way feedback is generated, and/or to understand what the teacher wants the learner to do, just as a SLAM robot can take actions designed to build a map and/or find out where the SLAM robot is within that map. This can hopefully make an intractable inference problem tractable by actively gathering the information that will allow it to understand the world well enough to make reasonably accurate simplifications. The teacher's utility function $u^\star$ is still defined in the same way, and the success criterion is still judged only based on $u^\star$. Choosing actions so that the learner can best estimate the teacher signal generating transform Ω is, however, probably a good strategy since $u^\star$ is easier to find with a better Ω estimate.

We denote an interaction protocol as the stochastic transform $\wp : C^h \times C^s \rightarrow C^\alpha$. A $\wp$ is a strategy for generating an action based on the interaction history and the current world state. A protocol can for example be defined by a rule for how to modify some data structure (such as a policy) at each interaction, and then select the next type of interaction based only on the current state of this data structure. The data structure update rule, and the rule for selecting interactions based on current state together imply a unique $\wp$.

To keep things simple, we keep the same success criterion, meaning that exploratory actions only serve to gather information that can be used to build a policy. We are thus still sidestepping the issue of making tradeoffs between *learning* what should be done, and actually *doing* what should be done (the learner simply tries to act in the way that will lead to the best possible policy, not needing to worry about how well it is performing tasks during the learning phase). We do not make any strong assumptions regarding the interaction behavior of the teacher, meaning that the teacher could stop giving feedback at unknown times, possibly dependent on how the learner act. The teacher could stop interacting or start interacting in a less engaged way because the learner is "not learning", or "done learning", or because the learner is now "boringly repeating the same actions", etc).

Since the problem of building a policy based on a given data set was treated in simplification step two, the only thing left to deal with from a theoretical point of view is "how to select actions with the highest expected usefulness of information". This is a relatively easy step from the point of view of a formalism, but leads us to an active, but basically open, research field when we look for tractable solutions. From a theoretical point of view, we need to find the action that will result in the highest expected amount of useful information. Ω includes all feedback behavior, so there is no need to introduce any additional transform for the "stop interacting in certain situations" or "start giving less informed feedback if bored" situations. The usefulness of an action is thus dependent on the usefulness of the feedback it will generate immediately, as well as how the action will impact on the future interaction behavior of the teacher. The problem of determining the expected amount of immediate information from an action is related to the field of optimal experiment design. Tractability issues are different, but the theoretical framework is the same.

The expected usefulness of a single action is simply the weighted sum of the expected usefulness of the action according to all $u^\star$-Ω pairs (weighted by probability), or the corresponding integrals in the case of continuous parameter spaces. For the hypothesized stochastic Ω^j, and a hypothesized $u^{\star i}$, the usefulness of a single action is the weighted sum of the usefulness of the change in policy from the possible feedback responses. In the continuous case, the sums turns into integrals. Determining the expected usefulness of a single discrete action, even given absolute knowledge of the world, is thus a completely intractable triple integral. We therefore need to start look-

ing at approximate solutions. One could, for example, try to optimize the expected information gain regarding what $u^\star$ looks like. This is an approximation, as discriminating between some possible $u^\star$s can be completely useless, even when they are very different (they could result in identical policies, or policies that have identical utility according to $u^\star$). It would also be possible to optimize the information gained about Ω, under the very reasonably sounding approximation that learning to understand the teacher's feedback will allow the learner to do what the teacher wants it to do. It is also possible to make some even more radical approximations and just maximize surprise, but then a TV showing static noise (or any other situation providing completely unpredictable sensory information) would become an attention trap. Luckily there is an entire field of research that is actively exploring what types of approximations can be made, and when some of them leads to traps of the type mentioned above. See [73] for early work and for example [74] and [75] for more recent experiments. In the previously mentioned [25], these methods are used for determining when and who to imitate. See also [76] for the optimal experiment design framework from which some of the basic principles come from. One common setting is building a forwards model for robot control by selecting exploratory actions to be as informative as possible, but the findings can be used without much modification. Below we can see Algorithm 3, which builds on the multiple generator algorithm discussed previously. It is built on the extremely old and very basic idea that if a hypothesis is formed based on a history, and then found to be good at predicting newly observed data, it is probably good. Actions are selected so as to discriminate between competing hypotheses. Hypotheses are discarded or modified based on the new observations, and new hypotheses are created based on history. Just like in the previous multiple generator algorithm, it concurrently estimates $u^\star$, Γs, which points where generated by which Γ, and what the generating regions are.

Input: $\mathcal{D}_0^G$, $\mathcal{D}_0^\Gamma$, $D_0^{u^*}$, T,$h(T)$,S
• $\mathcal{D}_0^G = \{D_0^{G1}, D_0^{G2}, \ldots, D_0^{GN}\}$ *is the initial estimate of the generating tendencies (at time 0).*
• $\mathcal{D}_0^\Gamma = \{D_0^{\Gamma 1}, D_0^{\Gamma 2}, \ldots, D_0^{\Gamma N}\}$ *is the initial estimate of the feedback behaviors.*
• D_0^Ω *is the initial estimate of* $u^\star$
• S *is the number of update steps done as a response to each interaction*
while *teacher still giving feedback* **do**
  $a_t \leftarrow determineAction(\mathcal{P}_t, D_t^{u^*}, \mathcal{D}_t^\Gamma, \mathcal{D}_t^G)$
  $I_t \leftarrow \{\Xi, observeDemAction(a_t)\}$
  $\mathcal{D}_t^\Gamma discardΓHyptheses(I_t, h(t-1), \mathcal{P}_t, D_t^{u^*}, \mathcal{D}_t^\Gamma, \mathcal{D}_t^G)$
  $h(t) \leftarrow \{I_1, I_2, \ldots, I_t\}$
  for $s = 1$ ***to*** S **do**
    ${}^{s+1}\mathcal{P}_t \leftarrow estimateGen({}^s\mathcal{P}_t, {}^s D_t^{u^*}, h(t), {}^s\mathcal{D}_t^\Gamma, {}^s\mathcal{D}_t^G)$
    ${}^{s+1}D_t^{u^*} \leftarrow update-u^\star-estimates({}^s D_t^{u^*}, {}^{s+1}\mathcal{P}^t, {}^s\mathcal{D}_\Gamma^t)$
    ${}^{s+1}\mathcal{D}_t^\Gamma \leftarrow update-\Gamma-estimates({}^s\mathcal{D}_\Gamma^t, {}^{s+1}\mathcal{P}_t, {}^{s+1}D_t^{u^*})$
    ${}^{s+1}\mathcal{D}_t^G \leftarrow estGenTend({}^s\mathcal{D}_t^G, {}^{s+1}\mathcal{P}_t, h(t), {}^{s+1}D_t^{u^*})$
  end
  $\mathcal{P}_{t+1} \leftarrow^S \mathcal{P}_t$
  $D_{t+1}^{u^*} \leftarrow^S D_t^{u^*}$
  $\mathcal{D}_{t+1}^\Gamma \leftarrow^S \mathcal{D}_t^\Gamma$
  $\mathcal{D}_{t+1}^G \leftarrow^S \mathcal{D}_t^G$
  $t \leftarrow t+1$
end

Algorithm 3: Active multiple generator algorithm

7 Relaxing assumptions on visibility and known distributions

In this section, three steps are briefly detailed. In these cases, removal of assumptions does not lead to the same level of additional algorithmic complexity as in the previous setups.

7.1 Step four: dealing with a world that is not perfectly visible to the teacher

Let's remove the perfect visibility of the teacher, and allow some world dynamics. $u^\star$ is now a mapping with the inputs expanded to include the teacher's world model. The teacher can now care about both the real world, and its own world model. The teacher could for example want dust to be removed from an apartment, and/or want to believe that the dust has been removed (a teacher could dislike a dusty apartment, and/or dislike that the apartment looks dusty, leaving the learner to deal with the old "to sweep dust under the rug, or to not sweep dust under the rug" question discussed earlier). The output of $u^\star$ is no longer directly visible to the teacher since the actual world state is not directly visible (and must instead be modeled based on inputs). Success is still defined in exactly the same way however. This changes little for the learner from

a theoretical point of view as it never had access to the outputs of $u^\star$ anyway. From a practical point of view it changes everything. Solution strategies such as "wait to sweep the dust under the rug until the teacher is looking in order to improve usefulness of the feedback" becomes central.

What the teacher sees, and what types of worlds/actions are easy for it to see/understand will now have to be monitored, and actions will also have to be chosen so that future world states are informative. The learner also has access to sensor readings of the teacher that might be informative regarding what is visible to the teacher (for example a camera image of the teacher, from which the learner can estimate what the teacher is looking at, and which objects are in the teacher's line of sight). Ω now maps the teacher's world model to feedback. Given a data set, the learner can in principle build a composite transform consisting of one transform from the actual world to the world model of the teacher, and then simply use that world model instead of the world state as input to Ω. The teacher still shares a given ontology with the learner in this step, meaning that the teacher's world model is a point in the same space as the actual world state (but not necessarily the same point). Again, it is easy to extend the formalism to include the analysis of a given data set by simply giving the full transform including a learner created transform from world state to teacher world model and Ω, and give this full transform the same place in the equations as Ω had earlier. Let's introduce some notation:

- **teacher sensor readings:** $z \in Z$. These could include camera images of the teacher, is observable to the learner, which can be used to estimate which objects are visible to the teacher.
- **teacher world model:** $w \in \mathcal{W}$: The teacher's best guess concerning the world state and the learner action.
- **Estimated teacher world building apparatus:** $\mathcal{V} : Z \times S \rightarrow \mathcal{W}$. Since the teacher's feedback behavior is now based on the teacher's world model (not the actual world state), it would be useful for the learner to estimate this transform.
- **Changed inputs for Ω:** $\Omega : \mathcal{W} \times C^h \times C^u \rightarrow C^f$. The learner must now estimate the teacher's world model in order to interpret the feedback generated from Ω.
- **Changed inputs for the utility function:** $u^\star : \mathcal{W} \times C^s \times C^a \rightarrow \Re$: Preferences can now be defined in both actual world states and estimated states (the teacher can want the apartment to be clean and/or want to avoid the sight of dirt).

Practical difficulty is increased, but this is still a well-defined inference problem with an analytical optimal solution (in terms of expected utility) for any finite data set and any generating distributions (if the teacher's world model building apparatus is also drawn from a known distribution). The problem was intractable even before, so not much has actually changed in terms of needing approximate solutions to evaluating data. But new types of information gathering strategies might be needed.

An obvious solution strategy would be to create the types of situation that are visible to the teacher in order to get more informative feedback. For example: "make sure to sweep the dust under the rug while the teacher is looking, so that feedback will be more informative". The analogy for a map-building robot would be to find that estimates based on camera images are less reliable in dark rooms so light conditions can be taken into account when updating based on observations. And the learner could turn on the light whenever possible.

7.2 Step five: dealing with a world that is not perfectly visible to the learner

Let's remove the perfect visibility of the learner, so that sensor reading and internal states take the place of world states. The interpretation hypotheses become transforms from teacher preferences to sensor readings, or states that are obtained by transforming sensor readings. The inputs to interpretation hypotheses and policies are now either input spaces or the results of transforms from input spaces. Very little actually changes from a theoretical point of view. The world states were not visible to the teacher in the previous step, so this space already functioned like sensor readings that are used to estimate what the teacher was perceiving.

7.3 Step six: finding Ω without a known generating distribution

Let's remove the aspect of the learner knowing how likely each possible way to generate the teacher signal is. In other words, the teacher signal generating transform Ω is no longer drawn from a known distribution. If the learner has a set of hypotheses regarding the distribution from which the teacher's Ω transform is drawn, this collapses (from the learner's point of view) into an equivalent problem. If the learner is able to investigate interaction histories from multiple teachers, the learner could, in principle, revise its estimate of each individual teacher's Ω transform concurrently with estimating the distribution from which Ω is drawn.

As the interpretation hypotheses are already defined over sensor readings, and they already output policy changes, nothing changes from the point of view of the learner. The initial set of hypotheses will imply a prior distribution, but finding it would be completely intractable (and of no practical value to the learner). The success criterion has not changed at all. We are now almost at an unstructured, real world environment, with the only remaining difference being the existence of the teacher's utility function $u^\star$. $u^\star$ is however not, however, visible to anyone at this point, and does not actually influence anything, so removing it would not change anything dramatically from the point of view of the learner. The task of interacting with a teacher in step six is indistinguishable from interacting with an actual human in an actual unstructured environment from the point of view of the learner.

8 Step seven: viewing existing learning algorithms, operating in the unsimplified setup, as testable interpretation hypotheses

We finally remove the assumption of a $u^\star$ in the head of the teacher. As pointed out above, this will change nothing from the point of view of the learner, as $u^\star$ was not actually affecting anything. An interpretation hypothesis Π now maps sensor readings, or states in more abstract spaces that are ultimately obtained from sensor readings, to policy updates. This means that a Π is now identical to a learning algorithm, and existing learning algorithms can be seen as interpretation hypotheses. Existing learning algorithms will often only be a hypothesis of how some limited set of the input space should be interpreted, making a set of learning algorithms with different types of inputs very suitable for concurrent modifications. By viewing a learning algorithm in this way, we see when and how one learning algorithm can be used to modify another. This section will focus on how to denote this type of concurrent modification of existing learning algorithms as concurrent updates of a set of hypotheses. The focus will be on a graphical representation, and on sketching numerous examples.

A learning algorithm is now re-interpreted as encoding assumptions about some information source, such as "demonstrated actions are more likely to be good than random actions", or "a reward button is pushed by a teacher, who very accurately compares the end result of an action with the end results of 7 previous actions". If these assumptions are made explicit as well as modifiable and/or falsifiable, then "interpretation hypothesis" is a more suitable name than "learning algorithm". It is possible to turn the number of previous actions that the current action is compared to into a variable, that can then be updated based on observations (evaluating the current action compared to the previous 3 actions will lead to different expected interaction histories than if it is compared to the 7 previous actions, allowing us to update the respective probabilities/update the parameter). Perhaps the most basic application would be to take two learning algorithms using the same set of inputs, then make their assumptions explicit so that it is possible to calculate what types of interaction histories they predict. Finally, observations can be used to determine which interpretation hypothesis/learning algorithm to use. Let's look more closely at a two-step algorithm that first estimates the teacher's goal by looking at how highly the teacher evaluates the learner's actions, and then learns how to achieve this goal. The hidden assumption is that the teacher evaluates the performance of the learner. We can then take another learning algorithm that changes the policy to be closer to policies that get a higher evaluation. The hidden assumption here is that the teacher evaluates how close the policy is to good policies. The learner uses one of these algorithms (or some other method, like interpreting demonstrations) to learn a simple task. The learner can now go through its history and see how the various types of actions were evaluated, and choose one interpretation hypothesis over the other (for example checking how actions that are close to good action in policy space, but very bad in outcome space, are evaluated).

Another example would be when there is a probability estimate of how often demonstrated actions are better than random, and how far they are from optimal. With a limited data set, improving these estimates will likely improve the policy updates. One idea arising from this way of viewing learning algorithms is to use multiple interpretation hypotheses. Each hypothesis interprets a different type of input, and each has a set of parameters (such as history length of a reward button). These hypotheses can then be used to learn a task concurrently with changing the parameters of the hypotheses, and estimating their usefulness. If the teacher pushing the reward button does not compare the current action to history, but instead to the optimal action, then it should be possible to discard the interpretation hypothesis as no parameter value will result in an accurate model. The ability to assess the usefulness of an interpretation hypotheses becomes impor-

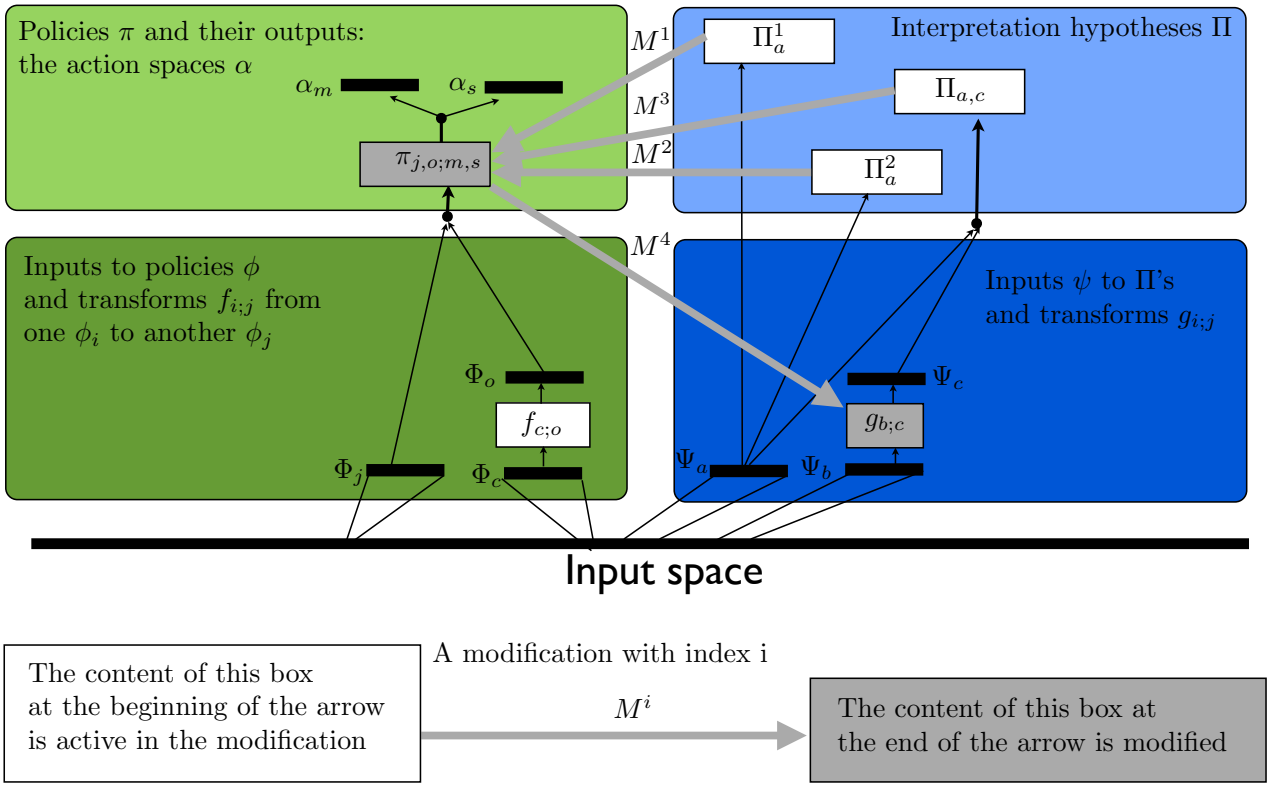

Fig. 1. In this setup, a policy $\pi_{j,o;m,s}$ is modified by three interpretation hypotheses; Π^1_a, Π^2_a and $\Pi_{a,e}$. Inputs to policies are denoted Φ and inputs to interpretation hypotheses are denoted Ψ. The black arrows mark inputs and the grey arrows mark modifications. The policy $\pi_{j,o;m,s}$ is used to modify the transform $g_{b;c}$. If the policy $\pi_{j,o;m,s}$ can be reasonably well learnt using only information in Ψ_a (for example a demonstration of a task), then it can later be used to check if the state in another space Ψ_c contains any useful information (for example, if one state is more frequent in the case where the demonstration was a failure, then this can be detected using $\pi_{j,o;m,s}$). The informedness of states in a space Ψ_c can be judged using $\pi_{j,o;m,s}$ (for example how useful states in Ψ_c are for separating failed demonstrations from successful ones) . It is now possible to use $\pi_{j,o;m,s}$ to choose from two different parameter settings of $g_{b;c}$ (two different parameter sets result in two different Ψ_c spaces, which $\pi_{j,o;m,s}$ can be used to choose between).

tant if there are competing hypotheses on how to interpret the same type of information, or if there are other information sources that can be used instead. The ability to discard some sources of information means that a learner can in theory learn how to deal with a real human in an unstructured environment that provides a diverse and redundant amount of information, and where some of the information is much harder to interpret than others, and where it is not known at programing time what information sources will be most useful (which could be heavily dependent on the type of teacher and the type of task). If a teacher often uses a reward button to encourage a learner that looks sad after failing, and the learner is unable to differentiate the two different types of reward button uses, then it might be best to simply learn from other information sources when dealing with this teacher. Similarly, if demonstrations often fail, and the learner is unable to separate a failure from a success (for example by using facial expressions), then it might be best to avoid requesting demonstrations, or to avoid wasting computational resources trying to learn something useful from them. The next step is to try to better describe what exactly these hypotheses are models of. They are supposed to model "what the information means", but this has to be stated more exactly (see section 9.1 where the concept of informed preferences is introduced).

Let's look more closely at one algorithm that translates demonstrations into policy changes, and a second algorithm that translates a scalar value following a learner action into policy changes. To make use of these information sources it is necessary to make some sort of assumptions about them, at the very least it is necessary to assume something along the lines of: "demonstrations are more likely than random actions to be good", or that: "good learner actions are more likely to be followed by high scalar values than bad actions". If a learner knows

one of these facts, it is possible to infer the other from observations. If the learner correctly assumes that demonstrations are good actions, it can learn what to do in some restricted circumstance and then notice that good actions are more likely to be followed by high scalar values. And if it correctly assumes that high scalar values indicate good actions, it can learn what to do in some situations and then notice that demonstrated actions are more likely to be examples of good actions. The correlation, once noticed, can be used when learning how to act in new situations (learning how to learn by learning how to interpret the various types of feedback given by the teacher). Active learning in this setting means seeking situations and performing interactions that will result in the type of information necessary to disambiguate between different hypotheses of how to interpret teacher behavior.

Knowing either the policy or the parameters of one interpretation hypothesis allows us to find the other two (the policy allows us to find the parameters of both interpretation hypotheses, and either interpretation hypothesis allows learning of the policy). The idea is that learning the interpretation hypotheses will be good for learning other tasks, but updating all three things concurrently can be useful even when only learning a single task.

The setup now consists of a learner that can be represented by transforms and input/output spaces, and an unstructured environment containing a human teacher. We have removed the simplification that there exists a utility function somewhere in the head of the teacher. The mathematical notation describing the teacher now only exists as a model that exists fully inside the learner architecture. The setup of step seven thus contain a real-world human and an unstructured world, but all the notation is now describing a computational system (the actual physical embodiment of the learner is outside this computational system, even if it's model of its own embodiment is part of the system). From the formalism point of view we will focus on this learner architecture and we choose a graphical representation in order to get a better overview. Using this representation, we will describe several different concrete architectures, learning from various information sources.

8.1 Graphical representation

We introduce a way of depicting a system/learner architecture graphically in Figure 1. Hopefully this representation will make it easier to see new extensions to existing research as well as enable us to describe proposed setups more clearly and more quickly. The top left rectangle contains the policies and the lower left contains the steps that lead to the inputs of the policy (which can be described as feature selection, finding the task space, finding the framing, etc). The top right rectangle contains the interpretation hypotheses, and the lower right rectangle contains the transforms that generate those inputs. The black arrows depicts inputs or outputs and the grey arrows depict modifications.

Examples of inputs include: current sensor readings (internal sensors such as battery life or external sensors such as cameras), past sensor readings, predicted future sensor readings, internal states (for example an estimated urgency of the current task), the estimated position of an object at some previous time (where object position is calculated, not present in sensor readings), the output of some opaque pre-processing step that the learner has no access to, or the estimated current common ground in a conversation[8], etc, etc.

Modifications of the M^4 kind uses a task in order to find a new input space for an interpretation hypothesis, for example learning which teacher facial expressions correspond to failed demonstrations (by using a known policy and the recorded history of demonstrations and facial expressions). The reasonably well-learnt policy $\pi_{j,o;m,s}$ can be used to determine how good individual demonstrations were. When we have a set of demonstrations with estimated quality, we can search for a way to predict this quality. This enables us to evaluate a space Ψ_c in terms of how well states in Ψ_c enables us to predict the quality of a new demonstration. The ability to evaluate a possible space Ψ_c enables us to modify the transform $g_{b;c}$ that results in Ψ_c (we can choose between two different parameter values of $g_{b;c}$ since we can choose between the two different resulting spaces Ψ_c). In short: $\pi_{j,o;m,s}$ modifies $g_{b;c}$ (which is denoted by a grey arrow from $\pi_{j,o;m,s}$ to $g_{b;c}$, and given the M^4 identifier for easy reference). From a technical point of view, this type of modification is not different in principle from other types of modification, but the result is that the learner can be said to "learn how to learn".

The states in Ψ_c could for example correspond to facial expressions of the teacher where some facial expressions indicates failure and other facial expressions indicate success. $\pi_{j,o;m,s}$ can help determine if states in Ψ_c are informative and so the "facial expression classifier" $g_{b;c}$ can be

8 What is appropriate to say and do is often dependent on the common ground, since an interlocutor will interpret actions based on this. A learner that can change how the current common ground is updated will be denoted by a modifiable policy with an appropriate action space (consisting of manipulations to its model of the current common ground).

modified. If the states in Ψ_c are informative in other situations, this could speed up learning in many other tasks (the teacher might make similar types of facial expressions no matter what task the learner is failing/succeeding at).

8.1.1 A learning from demonstration setup

In this setup the learner learns from demonstrations, as well as an estimate of how happy the teacher was with the demonstration the teacher just performed (the teacher is not always successful at performing the task and the learner is trying to predict if a new demonstration was a failure, and use that during learning). A graphical overview can be seen in Figure 2. The input to $\Pi_{e,d}$ is teacher actions represented in Ψ_d (a set of low dimensional context-action pairs) and an evaluation represented in Ψ_e, obtained by a transform $g_{f,t;e}$ with inputs in facial-expression space Ψ_f and tone of voice space Ψ_t. Ψ_f is obtained from a camera input Ψ_c using $g_{c;f}$ and Ψ_t is obtained from an audio input Ψ_a using $g_{a;t}$. Ψ_d is given to the learner directly and the learner cannot modify how it is obtained (it is not a sensor reading but, since it cannot be modified, it is an input to the learner). $\Pi_{e,d}$ updates a policy $\pi_{j,o;m,s}$ with inputs in joint- and estimated-object-position space, and performing actions in speech and motor α spaces.

$\Pi_{e,d}$ is an exact implementation of an interpretation hypothesis that could be verbally approximated as; "actions in Ψ_d are probably good for certain states in Ψ_e and probably bad for other states in Ψ_e", or even more crudely; "imitate the actions that the teacher seems pleased with". The update is denoted M^1 and is dependent on the details of the hypothesis, for example the assumed noise level of a favorable evaluation of some specific type of demonstration (for learning from demonstration algorithms that assume normally distributed noise, see the GMR based algorithms of [12–15]). If the update mechanism is static and ad hoc, only implicitly encoding assumptions about noise levels, it is still referred to as a hypothesis (it is just a hypothesis whose details are not easy to see and that is not updated based on observations). If the details of this hypothesis are made explicit there are several ways in which it could be updated: (i) demonstrations that involve heavy objects can be given a higher expected noise rate[9]. (ii) Adding a word recognizer that detects only the word "Nooo!" (giving a binary input to $\Pi_{e,d}$) and using this instead of the state in Ψ_e when it is present, but ignoring it when not present (the noise is of course dependent on the word recognizer and the usefulness is dependent on how often the word is used after failed demonstrations). (iii) A "triadic joint attention[10] detector" could be added based on the finding that the noise level is much lower when this is happening. The learner does not have to understand why the noise is lower in some states of the "triadic joint attention detector". The correlation could be detected, for example, if: the teacher is putting some real effort into trying to do a good demonstration, or if the facial expression estimator $g_{c;f}$ works better when it has this type of input. Other situation when correlation might be detected include if the type of verbalizations made in this type of interaction is easier to interpret by $g_{a;t}$, if the types of tasks that are demonstrated with triadic joint attention are easier to learn, or if the types of behavior the teacher performs in this type of interaction are the types of behavior that the teacher would like the learner to adopt. The learner can benefit from this "triadic joint attention detector" without fully understanding why it works.

8.1.2 A feedback learning setup

We can see this setup in Figure 3, where the policy $\pi_{j,o;m,s}$ has the same inputs and outputs as before. $\pi_{j,o;m,s}$ contains a list of previously performed actions (context and output) where each action has two scores, one estimated evaluation and one estimate of how informed the teacher was at the time of evaluation. It also consists of a rule for how to select a subset from the list based on the current context, and finally a regression algorithm that gives an output based on the subset of actions selected. If there are no previous actions with contexts close to the current context, a fixed default $\pi^2_{j,o;m,s}$ is executed (not shown in the figure).

$\pi_{j,o;m,s}$ is modified by Π_{v,e^2,a^2} based on the estimated visibility of the teacher Ψ_v (how much of the scene the teacher sees when giving the feedback), the estimated evaluation Ψ_{e^2} and the action performed Ψ_{a^2} (the action

9 If the teacher is not very proficient at manipulating heavy objects, and furthermore states in Ψ_e mainly captures how pleased it is with its own performance relative to the difficulty level of a task, then it is perfectly possible that the same state in Ψ_e indicate a higher expected noise level in the case of a type of Ψ_d that involves heavy objects. This can be utilized by a learner that simply notices that certain types of states in Ψ_d correspond to higher noise levels (even given the state in Ψ_e).

10 This is when the teacher is looking at the learner, as well as an object, and is following the learner's gaze to make sure they are both attending to the same object.

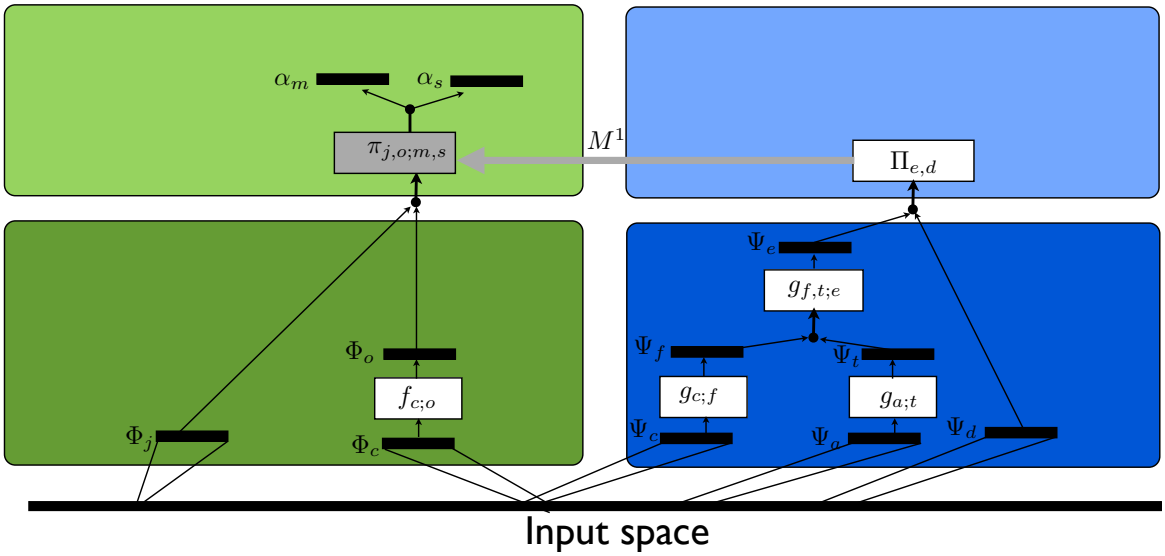

Fig. 2. A learning from demonstration setup where $\Pi_{e,d}$ modifies the policy $\pi_{j,o;m,s}$. The inputs to $\pi_{j,o;m,s}$ are in joint space Φ_j and estimated object position space Φ_o, and the policy is able to set the states in the action spaces α_m (motor outputs) and α_s (speech outputs). The policy is being modified by $\Pi_{e,d}$ based on an estimated teacher evaluation of the teacher's own demonstration Ψ_e and a representation of the demonstration in Ψ_d. The evaluation estimate Ψ_e is obtained by $g_{f,t;e}$ based on facial expression Ψ_f (obtained by $g_{c;f}$ from camera input Ψ_c) and tone of voice (obtained by $g_{a;t}$ from audio input Ψ_a).

that is assumed to be evaluated, and the parts of the context that are assumed to be relevant for the action to be performed). There are three input spaces to Π_{v,e^2,a^2} that could be improved. Ψ_{a^2} is the representation of actions and contexts, so any improvement of this would center around re-estimating which part of the context is relevant, or what part of the action is relevant. In this example the focus is on re-estimating what evaluation the teacher wanted to give, represented in Ψ_{e^2}, by modifying the transform $g_{a;e^2}$. And also on modifying the transform $g_{c;v}$ calculating how visible the scene was to the teacher at the time of the evaluation (represented in Ψ_v). For modification M^1 to work, the initial way of calculating Ψ_{e^2} must be at least approximately accurate. Even if the data is noisy, an accurate estimate of $\pi_{j,o;m,s}$ is still possible (using a larger data set than would have been needed if noise-free data were available). A somewhat accurate $\pi_{j,o;m,s}$ can then be used to create a data set that can be used to modify $g_{a;e^2}$ (denoted M^2) consisting of how correct the action was according to $\pi_{j,o;m,s}$ and the input in Ψ_a (the noise level will be dependent on the accuracy of $\pi_{j,o;m,s}$). One way of modifying $g_{a;e^2}$ would be to find a transform that, besides mapping audio input to the known word categories, also maps some audio input to a category that correlate strongly with very large performance improvement (for example corresponding to the teacher loudly saying "Great!").

The modification M^3 is done in a similar way. What is sought after is a space whose states can be used to predict if the evaluations in Ψ_{e^2} are accurate. $\pi_{j,o;m,s}$ tells us the accuracy of individual evaluations, which gives us a data set consisting of pairs of inputs in Ψ_c and this accuracy. What we want is a transform $g_{c;v}$ such that states in Ψ_v predict this accuracy. This problem is of a very well explored format (obtaining a function based on input-output pairs) and its solvability is strongly influenced by the accuracy of $\pi_{j,o;m,s}$ due to its influence on the noise level in the data set. One example where this can succeed is if 90% of evaluations are correct, and in the other 10%, the teacher is unable to see an object. An accurate $\pi_{j,o;m,s}$ can be learnt (M^1) with data of this noise level, leading to a data set where the accuracy of the individual evaluations are accurately estimated. Now this data set (not necessarily noisy, since $\pi_{j,o;m,s}$ is accurate) can be used to find a $g_{c;v}$ that optimally separates the 90% of accurate evaluations from the 10% of inaccurate ones.

The interesting part of this setup is to improve $g_{c;v}$ and $g_{a;e^2}$, which can hopefully be used to learn similar tasks.

9 Discussion

The task of the learner is to do what an informed version of the human would consider best for the uninformed version of the human. This setup is interesting as real robots placed in unstructured environments, with non-expert hu-

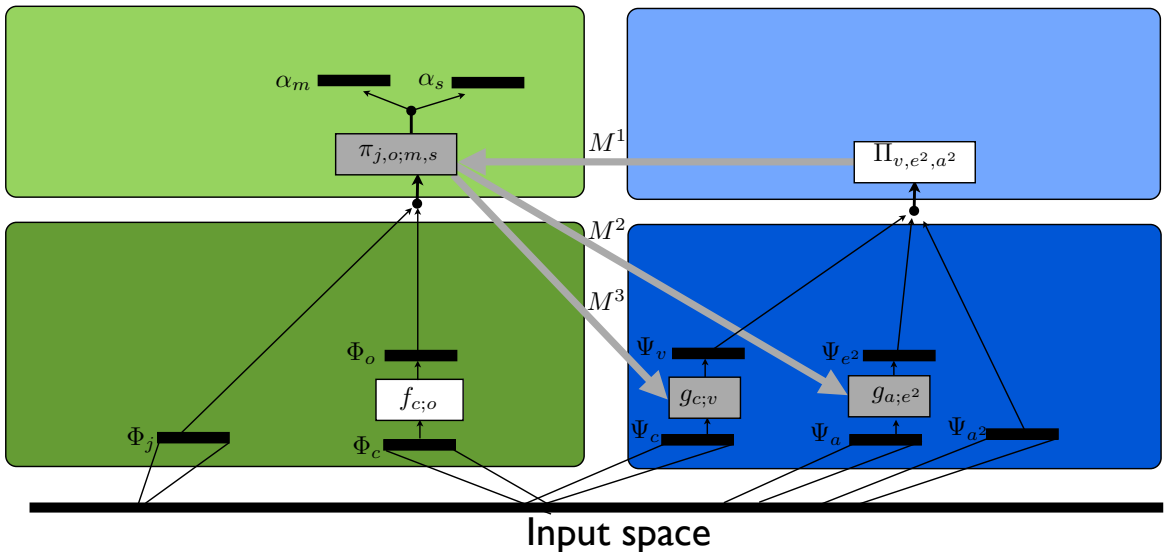

Fig. 3. This figure shows a learning from feedback setup where Π_{v,e^2,a^2} estimates how highly the learner's actions (represented in Ψ_a) were valued (estimated in Ψ_{e^2}) and how informed the teacher was Ψ_v, and uses this to choose whether or not to add the action (along with the context it was performed in) to a list of such actions contained in the policy $\pi_{j,o;m,s}$.

mans, will have to operate under these conditions. Non-expert humans in unstructured environments are not always well approximated as flawless feedback givers whose feedback has an easily encoded meaning. How to interpret the eye gaze or facial expression of a specific teacher will have to be learned, in a way similar to how it will have to learn to interpret a failed demonstration or a reward button pushed because the teacher failed to notice something or to encourage the learner.

9.1 Elaborating on the concept of informed preferences

The concept of informed preferences is designed to deal with cases such as "the teacher would like the learner to perform an action, but if the teacher knew the consequences of that action, would prefer another action" or "the teacher is very happy with the end result after the learner has cleaned the apartment, but if the teacher knew that the cleaning disturbed the neighbors by producing a lot of noise, it would not like the cleaning strategy". Being "informed" includes understanding of concepts (such as what types of actions are possible) and knowing specific facts, such as long term consequences of actions, or the contents of a box. These preferences are specified over the learner's action choices, and the goal of the learner is to execute preferred actions. Several thought experiments, including the example of a box whose contents the teacher is misinformed about, as well as several of the basic concepts of the formalism, are inspired by work done by Cynthia Breazeal, Andrea Thomaz and others, especially with the Leo robot. See [5] for an early publication that proposes to investigate genuine collaborative behavior as opposed to setups where the robot is conceptualized as a tool that a human is using. Conceptualizing the robot as an ally that shares intentionality with a human, instead of as a fancy hammer, makes it natural to start thinking about how the learner might solve some of the subtasks that socially collaborative humans routinely deals with. Especially relevant to the presented formalism would be: i) autonomously learning how the social signals of some particular human should be interpreted, ii) evaluating the competence of a collaborator (how likely is it that demonstrated actions are good actions?) iii) learning how to get more useful information from collaborators (postponing an action until the teacher is looking, displaying confusion with a facial expression or confirming understanding by nodding, asking good questions, etc).

A teacher might lack knowledge about the world, fail to understand certain concepts, not have imagined all possible strategies, be unaware of all consequences of an action, want things due to a misunderstanding, etc. If the teacher would consider knowledge relevant to the learner's action choice if it were made aware of it, then that piece of knowledge is considered relevant. For the purposes of evaluating the relative desirability of the actions that are available to a learner in a specific context, a subset of all knowledge that the teacher is unaware of will be relevant. This subset will be denoted Σ (a set of pieces

of knowledge that the teacher would consider relevant to a specific learner action choice, if the teacher was made aware of it). If a version of the teacher that knew about a fact would consider the fact relevant, it would be included in Σ, if a version of the teacher that understood a concept would consider it relevant, the concept would be included in Σ (and similarly with knowing about a possible strategy, or resolving a misunderstanding that made it want something, etc). There are many ways to segment actions. The very same learner's action can be evaluated by a preference ordering over states in its motor output space, or they can be evaluated according to a preference ordering over sequences of motor primitives. The correct segmentation is one that would be preferred by the teacher if informed. Anything considered relevant to its opinion about the segmentation is also considered relevant to the relative desirability of actions, and thus in Σ. If all things in Σ were acquired by the teacher (facts known, concepts understood, etc), then the resulting person is referred to as the informed version of the teacher. If the informed version of the teacher has an opinion about what would be best for the actually existing uninformed version of the teacher, then this is defined as the informed preferences of the teacher (a preference ordering over the learner actions that are available in the current situation). If the learner faces the decision of whether or not to show the teacher what is in a box, and the informed version of the teacher already knows what is in there, then the informed version of the teacher might want different things for itself than for the uninformed version of itself (since the decision can be different, it matters that the decision is about what is best for the uninformed version). The learner is now defined as a set of interpretation hypotheses and success is judged according to the informed preferences of the teacher.

Let's explore the case of a robotic learner providing security for a building using a camera, a microphone and an alarm. It is also able to move around the building, send video and microphone recordings over the internet to its teacher Steve, and receive commands and feedback from Steve. Steve is expecting that Bill will break into the building and has bought the robot as a way to get revenge on Bill (by showing video recordings of Bill committing a crime to the police), and has provided the learner with pictures of Bill, examples of what type of video would hold up in court, etc. The learner is very expensive (and knows this fact) and knows that if detected, it might get stolen. The learner sees a truck drive through a door, driven by a single masked person. The person gets out of the car and start taking things and putting them in the truck, and this person is very clearly much taller than Bill. The learner hides, triggers the alarm and starts sending a video feed to Steve. Steve is at home when he is alerted by the alarm and immediately sends the command "get a picture of Bills face", a huge negative scalar feedback when he sees that the learner is hiding, and then the command "move forward" (which would result in the learner and the robber seeing each other). In this case it is of course impossible to know for certain what will be in Σ or what Steve would want if acquiring everything in Σ, but it is, in principle, an empirical question. It is, however, possible to make a better-than-random educated guess, even if the number of things that (from the learner's perspective) might potentially be in Σ is huge. If someone other than Bill is breaking in, then they would take the expensive robot if they saw it, and further video would be useless. If Steve would consider this relevant in his decision of how the learner should respond to his commands, then these facts are part of Σ. If there is nothing else that Steve would consider relevant to his decision, and an informed version of Steve would think that the best thing for Steve would be that the robot stay hidden despite his commands, then this is Steve's informed preference. That the best possible action cannot be found with absolute certainty is abundantly clear in this case since the set of facts about the world that might be true and might change Steve's mind if he knew them is very big and some of them are very complex.

This is, however, not different in principle from a robot that maximizes the plus button pushes and that operates in an unstructured environment where an action can result in very bad rewards due to some impossible-to-predict effect (for example, some actions might make a reward button pusher think that the robot actually knows what to do but refuses to do it, and that it will start cooperating if it is punished enough with the minus button). And the problem can be dealt with in the same way, by making the best guess possible given the available information. It is easy to think of scenarios where impossible-to-know things impact Steve's decision in impossible-to-predict ways, but the problem is not fundamentally different from trying to fulfill any other success criterion in an intractable and unstructured world. The basic strategy of building the best probabilistic models possible given current ability, information and resources, continuously expanding them, continuously re-estimating what situation can be understood and always attempting to stay in situations the learner can handle, is still viable. It is possible that (i) the robber is Bills accomplice; (ii) Bill just walked in unmasked (and so, moving forward would result in Bill being convicted of breaking into the building); (iii) that if Steve understood some complex concepts of cognitive science regarding how his brain works and why he wants revenge on Bill, then he would conclude that he should

not seek revenge after all; (iv) if Steve understood some complicated concepts regarding long-term societal consequences of overcrowding in prisons, he would not want to send Bill to prison. Another possibility is (v) that if the robot moved forward, it would crush a butterfly under its wheel; the butterfly would then otherwise distract Steve while he was driving his car the next day, causing an accident that would kill Steve (and all versions of Steve agree that this outcome would be bad). In case (v) the learner should move forward, but fully understanding the situation is completely hopeless. It is however interesting to note that the effect of the butterfly poses the exact same problem to any robot, regardless of formalism (assuming the formalism is good enough that it classifies a dead Steve as a bad thing). The enormous set of things that might influence a decision is expanded to include a new category (consisting of things like how a teacher would modify what it wants as a response to understanding complex concepts), some of which can be hypothesized and be useful in a probabilistic model, but most of which will be just as unusable as hypotheses regarding the effects of crushing the butterfly (one can form as many such hypotheses as one likes, in favor of any decision one likes, but they cannot be tested, and doing this is not a useful strategy when searching for good decisions).

In a slightly different scenario where video of the learner is not routinely recorded for some reason, then Steve might never discover that it was not Bill who broke into the building unless the learner moves. Steve can record images sent to him after an alarm has been triggered, so the basics of the scenario remain the same. In this case the most important thing for Steve might be that he learns that it was not Bill who broke into the building. What is best for Steve is now different from what would have been best for an informed version of Steve in the same situation since he already knows that it is not Bill that is breaking in (and then the price of the robot would dominate the decision). That is why the formal success criterion cannot be to do what would have been best for an informed version of Steve in the same situation (even correctly answering the question “what action would Steve have preferred me to do if he were informed" will sometimes result in incorrect actions since uninformed versions sometimes have different needs, for example a need to know certain things that the informed version already knows). Doing what would be best for the informed version of the teacher if it existed does not seem to make any sense (the informed version is not present, and the informed version has for example different informational needs than the uninformed version).

If a robot sweeps dust under a rug and a teacher who is unaware of this considers the robot’s performance good, then the knowledge about the dust might be part of Σ. If the teacher considers the task to be “make the apartment clean", and would consider the learner’s actions bad if it knew about the dust, then it is part of Σ. But if the teacher considers the task to be “make the apartment look clean before the guests arrive", the information could be completely irrelevant, and thus not part of Σ. If the learner spent a large amount of energy cleaning the apartment, and there exist other cleaning strategies that would consume less energy, then this fact might be part of Σ. If the more energy efficient strategies had unacceptable side effects, it might not be part of Σ. If there are both unfamiliar concepts and unknown facts relating to societal effects of limited resources, then the teacher might prioritize energy efficiency differently. Again, these possibilities are not different in principle from the possibility that a meteor will strike, causing a blackout so that the learner cannot re charge, and that the learner’s removable batteries will actually be extremely important for some complicated reason. A robot operating in unstructured environments will face these types of hypotheses regardless of formalism, and they can be handled in a similar way. Formulating a success criterion in this way means that there might be a few hypotheses that can be tested and that does advocate different actions. In this case the learner can wait until the teacher is watching before sweeping the dust under the rug. It acts differently from a robot maximizing the additive output of a reward button (which would wait until the teacher is not watching to sweep the dust, so as to avoid risking negative reward) because in this situation it can actually test the two competing hypotheses that (i) the teacher wants a clean apartment and (ii) the teacher wants a presentable apartment (both of which seem like something a human might want and they could both be viable given available demonstrations, feedback, etc).

In the example where the cleaning robot is sweeping dust under the rug when Steve is not looking, success is not very visible, even if the learner receives positive feedback, since it does not know if Steve has an informed preference for this type of behavior. This is basically always the case to some extent since for most possible teachers it is not possible to know for certain what their fully informed preferences would be. Observability of success is thus a matter of degree, and potential experimental setups can be evaluated based on how observable the success is expected to be. Learners can choose their actions partly based on how observable success will be (for example waiting until Steve is looking before sweeping the dust under the rug).

A messy success criterion is needed because the real world is messy and intractable, which means that all non-messy success criteria are inaccurate, and thus only move the messy part to deciding when the success criterion is useful[11]. The initial set of interpretation hypotheses, along with update algorithms operating on them, can be interpreted as forming a prior distribution over possible informed preferences of the teacher, and over ways in which those are connected to the learner's inputs. If this is taken as an axiom, then the problem becomes an inference problem again. Success is now, however, possibly separated from the optimal solution to this problem, since the initial assumptions built into system might be inaccurate. This leaves us with an inference problem that can be approximated, but we still do not have access to a number that is guaranteed to represent actual success (the closest we can in principle get to guaranteed success is optimality given assumptions and information).

One strategy for dealing with an intractable problem in an uncontrolled environment is to autonomously extend the situations the learner can handle reasonably well and the types of teacher behavior it can interpret reasonably well, and constantly re-estimate the boundaries of what can be handled and what can be interpreted. This combines the nice feature of a success criterion where a strategy that is successful in the formalism is actually successful, with the possibility of a robot that can actually do things.

Extending the situations in which a learner knows how to act can be done concurrently with extending the types of teacher behaviors it can understand. For example, if the learner starts with an interpretation hypothesis Π_d that is able to learn from demonstrations reasonably well (at least in some situations), then it can extend the types of teacher behaviors it can understand by building a feedback interpretation hypothesis Π_f (after learning a task, it goes through the history of demonstrations and reproductions, and notice that what the teacher said was actually related to how good it was performing). When learning a new task, the learner can check if Π_f is accurate in this task as well, and later use Π_f to extend the types of tasks it can learn.

An analogy with this concurrent learning of tasks and interpretation hypotheses can be made with trying to build a model of some objects at the same time as trying to understand a set of languages that describe the object. The tasks are analogous to a set of unobservable objects, the interaction history is analogous to a set of descriptions of objects, and the interpretation hypotheses are analogous to the models of the languages that the descriptions are written in. A flawed understanding of a language can be used to build a good model of an object if there is enough redundant information about it. An example would be using a large number of separate, detailed descriptions of the object from many people, using different vocabulary, and describing the object at different levels of abstraction, different level of detail and from different complementary perspectives, such as descriptions considering the object's function, shape, component materials, durability, methods of manufacture, etc. If enough redundant information is available to build a model that is known to be accurate with respect to some aspects of one object, it is then possible to update the model of any language describing the object. In practice, it might be convenient to concurrently update the model of the object and the model of each language. It is not necessary to directly observe the object being described, or have access to any description in a perfectly understood language. The objects can be modeled and the languages can be learned by concurrently updating interconnected hypotheses. According to the same principle, it is possible to refine an interpretation hypothesis without being able to directly observe the informed preferences of the teacher, or having any flawless interpretation hypothesis. In some sense, interpretation hypotheses are very similar to the different possible world models of an agent with a specified utility function in the ontology of those world models; if they suggest different actions it is useful to distinguish between them, and if they predict different observations, it is possible to distinguish between them (the "actions" being analogous to policy updates, and "what the world actually looks like" to "what the teacher behavior actually means").

9.2 What is the purpose of the simplified setups

The simplified setups are introduced so that some problems can be examined without distraction and in order to make it possible to use more beautiful and crisp math.

11 In the case of a non-messy success criterion, without any complicated or unobservable parts, it is instead the suitability of the success criterion that is difficult to observe. It is sometimes obvious that the success criterion was bad, such as when a dust minimizing robot burns down the building and thereby clearly fails and simultaneously performs perfectly according to its non messy success criterion. But at other times the suitability might be difficult to observe. The difference is that there is no formal way to determine the suitability of a success criterion, and an agent that is optimizing an inappropriate criterion does not care that it is unsuitable, and will therefore not even try to fix the situation.

These setups can also serve as a pedagogical tool since the formalism in the simplified setups are easier to explain, and if the reader understands them it will be easier to explain the formalism of the unsimplified setup.

Let's take the example where the learner only has access to noisy sensor readings of the world, and does not perfectly hear the speech comments that the teacher uses to evaluate its performance; in this case the learner needs to interpret two different, inconsistent evaluations (two different evaluations of the same action in the same world state). It is now natural to investigate possibilities such as: (i) the evaluation was misheard, or (ii) the world model was wrong (so that it was the same action in two different world states that was evaluated), or (iii) the action was not the same in the dimensions that actually matters (which can happen in the case of incorrect assumptions regarding what aspects of an action is relevant), (iv) the world is viewed in the wrong framing (i.e. the world model is correct both times, but there is some relevant aspect of the world that is not captured by the model). In a noisy world, one of these could very well be the problem, and it makes a lot of sense to investigate all of these possibilities. But the danger is that one overlooks other types of potential problems. Let's say that the world state is observable and given in a known ontology (the world is neatly divided into world states that are shared by the learner and teacher). The teacher has access to a flawless policy, and only cares about things represented in the world state, and finally that the teacher is giving a fully observable scalar value as feedback (instead of a noisy speech comment). What can the learner do if it observes inconsistent behavior in such a setup? It is now forced to investigate an entirely new class of possibilities, for example: (i) the teacher cannot see all relevant objects, or (ii) the teacher is giving rewards for incremental progress, or (iii) the teacher is giving high rewards as encouragement since the robot has failed a lot and looks sad (real humans do this), or (iv) the teacher did not observe the entire action that it was evaluating[12], or any number of similar possibilities.

A simplified setup makes it possible to investigate these types of problems rigorously and without distractions. Inference problems can be intractable, and some are impossible to solve perfectly, even in principle. For the intractable inference problems, this formalism aims to provide a clear description of what it is that solutions are an approximation of. In some setups the best course of action can be impossible to find even in principle, and these are cast as an inference problem that contain a set of hypotheses such that each one: (i) has non negligible probability, (ii) imply a different optimal policy, (iii) make the same identical prediction in all observable spaces[13].

Since these problems exists in a simple setup, it seems obvious that they are much worse in more complex setups (at the very least they must be equally bad). As in most problems, the types of solutions that are appropriate in a simple world are not guaranteed to be appropriate in complex worlds. Thus the simplifications are removed gradually so that more realistic setups can be investigated, leading to modifications of both descriptions and solutions. A learner always contain a stochastic transform from an interaction history space $h \in C^h$ to a policy space $\pi \in C^\pi$ at each simplification step. In the first steps, the interaction history is over observable world states and a well separated feedback space, and later this is replaced by inputs. C^π also at first takes inputs in observable world states, but is later changed to have sensor-reading-type inputs. Thus a learner always contains the same type of stochastic transform : $C^h \rightarrow C^\pi$, even though the relevant spaces are given a different interpretation in later steps, as simplifications are removed.

In the first steps, the learner simply analyzes a fixed data set and outputs a policy, so that this transform is a complete specification of a learner. Any update rule that modifies a policy based on a single interaction and then forgets the information, is recursively defining a stochastic transform, so this is also a way of fully specifying a learner (even if this transform is unknown to the programmers and very difficult to find or interpret, any iterative learning rule is still identical to a unique stochastic transform $C^h \rightarrow C^\pi$). In later steps, the learner needs to perform information gathering actions (meaning that an aditional element is needed to fully specify a learner).

12 For example, observing the full action sequence of a cleaning behavior in one instance, but only the end result in the other instance. The evaluations could be different if the teacher missed that the learner swept the dust under the rug, or made a lot of noise while moving the furniture (which annoys the neighbors), or damaged the floor under the sofa, etc.

13 There are a set of hypothesis pairs (a teacher informed preference hypothesis and an interpretation hypothesis) that makes identical predictions regarding what feedback will be observed. If the informed preferences are modeled by a utility function, the best that can be done is to collapse them into a weighted sum, according to prior probabilities. This reduces the problem to the same type of inference problem as before.

10 Conclusion and future research

We have presented a formalism that provides a common theoretical foundation for a large number of research projects that have so far been considered as separate fields of research. By considering for example a teacher demonstration, an evaluative speech comment and a teacher-provided numerical value as the same type of information source, a structured way of using several of these information sources simultaneously has been established.

A mathematical success criterion was presented in a set of simplified setups and several possible solution methods were sketched, opening up new avenues of future research. New research projects can start from the simplified setups and explore ways of making approximate inference (along the lines of the example algorithms that were sketched along with the formalism). Then simplifications can be incrementally removed when solutions to the simpler setups are validated in experiments.

Another avenue for future research was opened up by re describing existing learning algorithms as interpretation hypotheses. This provided a structured way in which an information source can be reinterpreted based on observations, and introduced the idea of updating several learning algorithms concurrently. This means that a new research project can start with a set of existing algorithms, introduce parameters and then design a rule for when to change those parameters as a response to observations, concurrently with learning tasks. It is also possible to start with minor additions to individual existing learning algorithms. Any learning-from-demonstration algorithm can be augmented by a system that checks facial expressions of the teacher and learns that some facial expressions means that the teacher failed. One very simple way of using this information, that will work with any learning-from-demonstration algorithm, is to avoid using data that is above some threshold of probability of being a failure. It is also possible to make a small addition to any reinforcement learning algorithm by making the size of a learning rate dependent on an estimation of the level of informedness of the teacher. It is sometimes possible to confidently learn a task even in the face of sometimes incorrect reinforcement signals. When the real performance is known, the interaction history can be re-examined and it can be determined in what situations values given by the teacher are accurate, and when they are inaccurate. Now it is just a matter of estimating the probability of the signal being accurate given the context (for example the presence of an obstacle between the teacher's eyes and an important object), and reducing the learning rate based on the probability estimates.

The formalism presented can thus be used to see existing research in a new light, giving old experiments a new interpretation and making alternative avenues of future research visible.

Acknowledgement: The authors thank Manuel Lopes, Jonathan Grizou and the anonymous reviewers for their insightful comments. This research was partially funded by Region Aquitaine and ERC Grant EXPLORERS 240007.

References

[1] Ng, A.Y. and Russell, S. Algorithms for inverse reinforcement learning, *Proceedings of the Seventeenth International Conference on Machine Learning*. pp 663-670. 2000.

[2] Nehaniv, C., and Dautenhahn, K., The correspondence problem, *Imitation in animals and artifacts*., MIT Press, pp 41-61. 2002.

[3] A. L. Thomaz and C. Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners *Artificial Intelligence Journal*, 172:716-737, 2008.

[4] S. Griffith, K. Subramanian, J. Scholz C.L. Isbell, and A. L. Thomaz. Policy Shaping: Integrating Human Feedback with Reinforcement Learning. *proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2013.

[5] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. L. Thomaz, and D. Chilongo.,Tutelage and collaboration for humanoid robots, *International Journal of Humanoid Robotics*, vol. 1, no. 02, pp. 315?348, 2004.

[6] A. L. Thomaz, and C. Breazeal. "Reinforcement Learning with Human Teachers: Evidence of feedback and guidance with implications for learning performance." In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), 2006.

[7] A. L. Thomaz, G. Hoffman, and C. Breazeal. Reinforcement learning with human teachers: Understanding how people want to teach robots, *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2006.

[8] A. L. Thomaz and C. Breazeal. Asymmetric Interpretations of Positive and Negative Human Feedback for a Social Learning Agent, *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2007.

[9] A. L. Thomaz and C. Breazeal. Experiments in Socially Guided Exploration: Lessons learned in building robots that learn with and without human teachers. *Connection Science, Special Issue on Social Learning in Embodied Agents*, pages 91-110, 2008.

[10] A. Dempster and N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society.* B, 39(1):, pp 1-38. 1977.

[11] Zoubin Ghahramani and Michael I. Jordan, Supervised learning from incomplete data via an EM approach, *Advances in*

Neural Information Processing Systems. vol. 6, pp 120-127. 1994.

[12] S. Calinon and F. Guenter and A. Billard, On Learning, Representing and Generalizing a Task in a Humanoid Robot, *IEEE Transactions on Systems, Man and Cybernetics, Part B* , vol. 37, pp 286-298. 2007.

[13] Calinon, S. and Billard, A, Incremental Learning of Gestures by Imitation in a Humanoid Robot, *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI),.*, pp255-262. 2007.

[14] Aude Billard, Sylvain Calinon, Ruediger Dillmann, Stefan Schaal, Robot Programming by Demonstration, *Springer Handbook of Robotics.*, pp 1371-1394. 2008.

[15] Calinon, S *Robot Programming by Demonstration: A Probabilistic Approach*.,EPFL/CRC Press , 2009.

[16] Calinon, S. and D'halluin, F. and Sauser, E. L. and Caldwell, D. G. and Billard, A. G., Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression, Robotics and Automation Magazine, vol. 17, pp 44-54. 2010.

[17] Yasser Mohammad and Toyoaki Nishida, Learning Interaction Protocols using Augmented Baysian Networks Applied to Guided Navigation, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010.

[18] Yasser Mohammad, Toyoaki Nishida, and Shogo Okada, Unsupervised Simultaneous Learning of Gestures, Actions and their Associations for Human-Robot Interaction *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*., pp. 2537-2544. 2009.

[19] Baranes, A., Oudeyer, P-Y. R-IAC: Robust Intrinsically Motivated Exploration and Active Learning, *IEEE Transactions on Autonomous Mental Development*, 1(3), pp. 155-169. 2009.

[20] Cederborg, T., Ming, L., Baranes, A., Oudeyer, P-Y: Incremental Local Online Gaussian Mixture Regression for Imitation Learning of Multiple Tasks, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* 2010.

[21] M, Ogino., H, Toichi., Y, Yoshikawa., M, Asada., Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping. In *Robotics and Autonomous Systems*, 54, 5, pp 414-418. 2006.

[22] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous Helicopter Aerobatics through Apprenticeship Learning *International Journal of Robotics Research*, Nov. vol. 29, no. 13, 1608-1639. 2010.

[23] A. Billard., S. Calinon., F. Guenter. Discriminative and Adaptive Imitation in Uni-Manual and Bi-Manual Tasks. *In Robotics and Autonomous Systems*. volume 54", number 5, pages 370-384. 2006.

[24] K. Dautenhahn and C. L. Nehaniv. The agent-based perspective on imitation, *Imitation in animals and artifacts*, pages 1-40. MIT Press, 2002.

[25] S, M, Nguyen. P-Y Oudeyer. Socially Guided Intrinsic Motivation for Robot Learning of Motor Skills. *Autonomous Robots, Springer*, 36 (3), pp. 273-294. 2014.

[26] B.D. Argall, S. Chernova, M. Veloso and B. Brett. A survey of robot learning from demonstration in *Robot. Auton. Syst.*, 57, 5, pp 469-483. 2009.

[27] Billard, A., Calinon, S., Dillmann R. and Schaal, S. Robot Programming by Demonstration. *In Siciliano, B. and Khatib, O. (eds.). Handbook of Robotics*, pp. 1371-1394. Springer. 2008.

[28] N. Delson and H. West., Robot programming by human demonstration: Adaptation and inconsistency in constrained motion., *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).*, pp 30-36. 1996.

[29] Ude, A, Trajectory generation from noisy positions of object features for teaching robot paths, *Robotics and Autonomous Systems.*, pp 113-127. 1993.

[30] Ito, M., Noda, K., Hoshino, Y., and Tani, J, Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model, *Neural Networks.* , 19 (3), pp 323-337. 2006.

[31] Tikhanoff V., Cangelosi A and Metta G. Language understanding in humanoid robots: iCub simulation experiments. *IEEE Transactions on Autonomous Mental Development*. 3(1), 17-29. 2011.

[32] Massera, G., Tuci, E., Ferrauto, T., and Nolfi, S. The facilitatory role of linguistic instructions on developing manipulation skills, *Comp. Intell. Mag.*, 5(3): 33-42. 2010.

[33] Sugita, Y. and Tani, J. Learning Semantic Combinatoriality from the Interaction between Linguistic and Behavioral Processes. *Adaptive Behavior*, 13(1):33–52. 2005.

[34] Schaal, S. and Peters, J. and Nakanishi, J. and Ijspeert, A. learning movement primitives in, *international symposium on robotics research*. springer. 2004.

[35] Calinon, S. and Billard, A. Teaching a humanoid robot to recognize and reproduce social cues. *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 346-351. 2006.

[36] Thomaz, A., Berlin, M., and Breazeal, C.. Robot science meets social science: An embodied computational model of social referencing. *Workshop toward social mechanisms of android science (CogSci)*. pp. 7-17. 2005.

[37] Neu, G. and Szepesvári, C. Training parsers by inverse reinforcement learning. *Machine learning*, vol 77, number 2, pp 303-337. 2009.

[38] Lopes M, Cederborg T, Oudeyer PY Simultaneous acquisition of task and feedback models. *IEEE International Conference on Development and Learning*. 2011.

[39] J, Grizou and M, Lopes and P-Y, Oudeyer. Robot Learning Simultaneously a Task and How to Interpret Human Instructions *Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*. 2013.

[40] Grizou, Jonathan and Iturrate, Iñaki and Montesano, Luis and Oudeyer, Pierre-Yves and Lopes, Manuel. Calibration-Free BCI Based Control *AAAI Conference on Artificial Intelligence*. 2014.

[41] Grizou, Jonathan and Iturrate, Iñaki and Montesano, Luis and Oudeyer, Pierre-Yves and Lopes, Manuel. Interactive Learning from Unlabeled Instructions, *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, 2014.

[42] Manuel Lopes, Francisco S. Melo, Ben Kenward and Jose Santos-Victor. *A Computational Model of Social-Learning Mechanisms. Adaptive Behaviour*, 467(17), 2009.

[43] Cynthia Breazeal and Jesse Gray and Matt Berlin, An Embodied Cognition Approach to Mindreading Skills for Socially Intelligent Robots, *I. J. Robotic Res.* vol. 28, 5, pp 656-680. 2009.

[44] Thomas Cederborg and Piere-Yves Oudeyer. From Language to Motor Gavagai: Unified Imitation Learning of Multiple Linguistic and Non-linguistic Sensorimotor Skills, *IEEE Transactions on Autonomous Mental Development*, 2013, accepted.

[45] Duy Nguyen-Tuong and Matthias W. Seeger and Jan Peters, Real-Time Local GP Model Learning, *in From Motor Learning to Interaction Learning in Robots*, pp 193-207. 2010.

[46] Angelo Cangelosi, Giorgio Metta, Gerhard Sagerer, Stefano Nolfi, Chrystopher Nehaniv, Kerstin Fischer, Jun Tani, Tony Belpaeme, Giulio Sandini, Luciano Fadiga, Britta Wrede, Katharina Rohlfing, Elio Tuci, Kerstin Dautenhahn, Joe Saunders, Arne Zeschel: Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics. *IEEE Transactions on Autonomous Mental Development*, 2010.

[47] Csibra, G., and G. Gergely. Social learning and social cognition: The case of pedagogy. *In Progress of Change in Brain and Cognitive Development. Attention and Performance, vol. XXI, edited by Y. Munakata and M. H. Johnson. Oxford: Ofxord University Press*. pp 249-274. 2006.

[48] Zukow-Goldring P., Assisted imitation: Affordances, effectivities, and the mirror system in early language development *in From Action to Language, Arbib, M.A. Ed. Cambridge: CUP*, pp. 469-500. 2006.

[49] B.D. Argall, B. Browning, M. Veloso. Teacher feedback to scaffold and refine demonstrated motion primitives on a mobile robot. *Robotics and Autonomous Systems*. 59(3-4). pp 243-255. 2011.

[50] Kober J, Wilhelm A, Oztop E, Peters J. Reinforcement learning to adjust parametrized motor primitives to new situations. *Autonomous Robots*. pp 1-19. 2012.

[51] Akgun B, Cakmak M, Yoo J, Thomaz A. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In: *International Conference on Human-Robot Interaction*. 2012.

[52] Isbell, C., Kearns, M., Singh, S., Shelton, C., Stone, P., Kormann, D. Cobot in LambdaMOO: An Adaptive Social Statistics Agent. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2006.

[53] W. Bradley Knox, Brian D. Glass, Bradley C. Love, W. Todd Maddox, and Peter Stone. How Humans Teach Agents: A New Experimental Perspective., *International Journal of Social Robotics*. 4(4) , 409-421. July 2012.

[54] Knox, W.B., Breazeal, C., Stone, P.: Learning from feedback on actions past and intended. *Proceedings of 7th ACM/IEEE International Conference on Human-Robot Interaction HRI*. 2012.

[55] Knox, W., Stone, P.: Interactively shaping agents via human reinforcement: The TAMER framework. *The 5th International Conference on Knowledge Capture*. 2009.

[56] W. Bradley Knox, Cynthia Breazeal, and Peter Stone. Training a Robot via Human Feedback: A Case Study. *Proceedings of the International Conference on Social Robotics (ICSR)*., 2013.

[57] Robert Loftin, Bei Peng, James MacGlashan, Michael L. Littman, Matthew E. Taylor, Jeff Huang, and David L. Roberts. Learning Something from Nothing: Leveraging Implicit Human Feedback Strategies. *Proceedings of the Twenty-Third IEEE International Symposium on Robot and Human Communication (ROMAN)*. 2014.

[58] Algorithmic and Human Teaching of Sequential Decision Tasks, Maya Cakmak and Manuel Lopes. AAAI Conference on Artificial Intelligence (AAAI), Toronto, Canada, 2012.

[59] M. Cakmak and A.L. Thomaz. Designing Robot Learners that Ask Good Questions, *International Conference on Human-Robot Interaction (HRI)*, 2012.

[60] A.-L. Vollmer, M. Muhlig, J. J. Steil, K. Pitsch, J. Fritsch, K. J. Rohlfing, and B. Wrede, Robots show us how to teach them: Feedback from robots shapes tutoring behavior during action learning, *PloS one, vol. 9, no. 3,*, 2014.

[61] A.Vinciarelli, M.Pantic, D.Heylen, C.Pelachaud, I.Poggi, F.D'Errico and M.Schroeder. Bridging the Gap Between Social Animal and Unsocial Machine: A Survey of Social Signal Processing, *IEEE Transactions on Affective Computing*, Vol. 3, No. 1, pp. 69-87. 2012.

[62] Z. Zeng, M. Pantic, G. Roisman, and T. Huang. A Survey of Affect Recognition Methods: Audio, Visual and Spontaneous Expressions, IEEE Trans. *Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39-58, Jan. 2009.

[63] Irene M. Pepperberg and Diane V. Sherman. Training behavior by imitation: from parrots to people ... to robots? *In, Imitation and Social Learning in Robots, Humans and Animals Behavioural, Social and Communicative Dimensions. Edited by C. L. Nehaniv and K. Dautenhahn*. pp. 383-406. 2007.

[64] Nehaniv, C., and Dautenhahn, K., Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications, *Interdisciplinary approaches to robot learning.*,World Scientific Press, vol. 24, pp 136-161. 2000.

[65] M. Stolle, C.G. Atkeson, Knowledge transfer using local features, in: Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, ADPRL'07, 2007.

[66] Chrystopher L. Nehaniv, Nine Billion Correspondence Problems, *In C. L. Nehaniv and K. Dautenhahn (Eds.), Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, Cambridge University Press, 2007.

[67] E. A. Billing, T. Hellstrom., A formalism for learning from demonstration. Paladyn , vol. 1, no. 1, pp. 1-13, 2010.

[68] Bessère, Pierre and Laugier, Christian and Siegwart, Roland. Probabilistic reasoning and decision making in sensory motor systems. *Springer tracts in advanced robotics*. 2008.

[69] João Filipe Ferreira, and Jorge Dias. Probabilistic Approaches to Robotic Perception. *Springer Tracts in Advanced Robotics Springer*. 2014.

[70] Mark Hauschild and Martin Pelikan. An Introduction and Survey of Estimation of Distribution Algorithms. *Swarm and Evolutionary Computation* Volume 1, Issue 3, September, Pages 111-128. 2011.

[71] Thomas P Minka. A family of algorithms for approximate bayesian inference. PhD thesis, 2001.

[72] Kennedy, J.; Eberhart, R. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942-1948. 1995.

[73] J. Schmidhuber. Curious model-building control systems. In: *Proc. Int. Joint Conf. Neural Netw.*, vol 2, pp: 1458-1463. 1991.

[74] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation. *IEEE Transactions on Autonomous Mental Development* 2(3): pp: 230-247. 2010.

[75] Baranes A, Oudeyer PY Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems* 61(1):49-73. 2013.

[76] V. Fedorov. Theory of Optimal Experiment. *Academic Press, Inc., New York, NY*. 1972.