Observational Learning by Reinforcement Learning

Diana Borsa
DeepMind
borsa@google.com

Nicolas Heess
DeepMind
heess@google.com

Bilal Piot
DeepMind
piot@google.com

Siqi Liu
DeepMind
liusiqi@google.com

Leonard Hasenclever
DeepMind
leonardh@google.com

Remi Munos
DeepMind
munos@google.com

Olivier Pietquin
Google Brain
pietquin@google.com

ABSTRACT

Observational learning is a type of learning that occurs as a function of observing, retaining and possibly imitating the behaviour of another agent. It is a core mechanism appearing in various instances of social learning and has been found to be employed in several intelligent species, including humans. In this paper, we investigate to what extent the explicit modelling of other agents is necessary to achieve observational learning through machine learning. Especially, we argue that observational learning can emerge from pure Reinforcement Learning (RL), potentially coupled with memory. Through simple scenarios, we demonstrate that an RL agent can leverage the information provided by the observations of an other agent performing a task in a shared environment. The other agent is only observed through the effect of its actions on the environment and never explicitly modeled. Two key aspects are borrowed from observational learning: i) the observer behaviour needs to change as a result of viewing a 'teacher' (another agent) and ii) the observer needs to be motivated somehow to engage in making use of the other agent's behaviour. The later is naturally modeled by RL, by correlating the learning agent's reward with the teacher agent's behaviour.

KEYWORDS

Reinforcement Learning; Observational Learning; Learning from other agents; Information seeking; Imitation.

ACM Reference Format:
Diana Borsa, Nicolas Heess, Bilal Piot, Siqi Liu, Leonard Hasenclever, Remi Munos, and Olivier Pietquin. 2019. Observational Learning by Reinforcement Learning. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 8 pages.

1 INTRODUCTION

Humans have evolved to live in societies and a major benefit of that is the ability to leverage the knowledge of parents, ancestries or peers to aid their understanding of the world and more rapidly develop skills deemed crucial for survival. Most of this learning is done by observing the behaviour of the other agents with core learning mechanisms emerging such as role modeling, imitation or observational learning. In this paper, we are particularly interested in the latter. We define observational learning as an agent's ability to modify its behavior or to acquire information from purely from observing another agent, that happens to share its environment, without explicitly modeling it as an agent.

In the machine learning literature, one of the most popular and successful ways of modeling goal-motivated learning agents is via Reinforcement Learning (RL) [20, 32]. In the recent years, combining RL with the increased representational power of deep learning [16] and the memory capabilities of recurrent models (LSTMs/GRUs) [7, 12] has lead to a string of impressive successes ranging from video-game playing [20] to 3D navigation tasks [18, 19] and robotics [17]. Motivated in part by these, here we want to study if observational learning can naturally emerge in DeepRL agents empowered with memory. Thus the main questions we would want to answer are: *is (deep) RL coupled with memory enough to successfully tackle observational learning?* Will the RL agent learn to ignore or leverage the teacher? Is the RL signal enough for the emergence of more complex behaviour like *imitation*, *goal emulation* or *information seeking*? In other words, we want to understand to what extent other agents have to explicitly be modeled as such by a learning agent. Is the combination of perception (deep nets), memory (recurrent nets) and motivation (RL) enough to learn from the sole observation of other agents' effects on a shared environment?

It is worth noting that similar questions have been investigated in the cognitive and behaviour science community. In their work, Bandura and Walters [4, 5] proposed and coined the term 'observational learning' or social learning. According to them, observational learning differs from imitative learning in that it does not strictly require a duplication of the behavior exhibited by the teacher. Heyes [9] distinguished imitation and non-imitative social learning in the following way: imitation occurs when animals learn about behavior from observing conspecifics, whereas non-imitative social learning occurs when animals learn about the environment from observing others. Meaning that one can learn about the dynamics of its environment only by observing others evolving in this environment.

Learning with the help of a teacher or demonstrator is by no means a new idea in machine learning neither. Imitation learning

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

has a long standing in the machine learning literature [2, 30]. In this body of work, one can distinguish two major ideas: i) *behaviour cloning*, where we are regressing directly onto the policy of another agent/teacher [26, 27], or ii) *inverse RL*, where we are trying to infer a reward function from the behaviour of other agents [22, 29] and then use this, in conjunction with RL techniques, to optimize this inferred function – this is closer to goal emulation. Imitation learning has also been combined to RL in recent works [6, 8, 13, 24]. While these methods have been successfully applied to a variety of tasks [3, 10, 11, 15, 21, 25, 26, 28, 34], one problematic aspect of all these scenarios is that they almost always need to provide the learning agent with the teacher trajectories in the same state-action space as the learner. Otherwise, some explicit mapping between the learner and the teacher state space has to be discovered [31]. As previously argued/recognized in [31], these are somewhat restrictive and unrealistic requirements. Furthermore, in inverse RL one has to explicitly model the behaviour/trajectories coming from another agent and infer the reward signal. For this problem to become tractable, most of the time we need to make some structural assumptions about this reward signal – like linearity in a given feature space, or smoothness [1, 14, 24]. These assumptions might not hold for the true reward signal and minor approximation errors can be easily amplified when planning onto this faulty signal [23].

Given these increased complexities, we propose to study the simpler, yet more natural alternative of observational learning, moving away from the traditional setup of learning from teachers. We do not claim we can address all of the problems tackled by the imitation learning literature. We are merely arguing that there might be scenarios where this level of modelling is not required and the RL agent can learn directly through pure observations. In this context, our main contribution is to exhibit scenarios where observational learning is emerging from a standard DeepRL algorithm (A3C [19]) when combined or not with memory. In all scenarios we will look at, the A3C agent (learner) shares its environment with another agent (teacher) that has a better knowledge of the task to solve. The learner observes the teacher through its sole perception of the environment. It is only rewarded for performing the task and does not receive any incentive to follow, imitate or interact with the teacher. The teacher is not aware that it is watched by the learner and is not meant to teach or provide extra information to the learner neither. It is only performing its own task independently from the presence of the learner. By building tasks of increasing difficulty, we show that complex behaviours such as imitative and non-imitative learning emerge without explicit modeling of the teacher. In addition, we provide some theoretical insights to explain why these behaviours are possible.

In the next section, we describe our experimental design. Section 3 provides the general background of RL and the theoretical foundations of this work. In Section 4 we provide our experimental results before concluding in Section 5.

2 EXPERIMENTAL DESIGN

As explained in the introduction, we are primarily interested to see if an RL agent can learn to leverage the behaviour of a teacher, based solely on 1) external reward (from the environment), ii) its ability to observe the consequences of the teacher's actions in the environment. The learner does not have a notion of the agency of the teacher. This additional agent is simply part of the learner's environment and it can choose to ignore the presence of the teacher if it deems this signal unimportant. Note that we call the teacher "agent" as it is an entity that acts in the environment. It has its own task and its own algorithm to solve the task. This information is hidden to the learner and not modeled by it. So it cannot only be considered as some additional hint introduced in the environment to help the learner. It rather simulates the presence of other goal-directed entities in the environment. It is also worth noting that in all our case studies, the presence of the teacher does not impact the dynamics nor rewards of the learner. This is a necessary assumption that will be formalized in Sec. 3.

Thefi rst question we want to answer is whether the teacher's presence has any impact on the learner. For this we look at two scenarios: 1) the learner has perfect information and can learn an optimal behaviour on its own, 2) the learner has only partial information about the environment/task, but the teacher's behaviour can provide additional information by showing a demonstration of the desired behaviour. In thefi rst case, we do not expect a difference between the learnt policies with or without the teacher. Nevertheless, when adding the teacher in the same environment, we are effectively expanding the observation space of the learner. Thus, on top of the RL policy, now the learner also needs to learn to ignore this extra signal in its observation space. In the second scenario however, the teacher's behaviour contains crucial information for improving the learner's policy. In this case, by ignoring the teacher, the learner can still complete the task at hand, but can only do so, sub-optimally. Now, there is a real incentive for the learning agent to pay attention to the teacher. Nevertheless, the learner's own reward signal is still the one coming directly from the environment (which it would experience even without the other agent), but now our agent needs to somehow correlate this (potentially sparse) reward signal with the behaviour exhibited by the teacher. This is a highly non-trivial association the learner needs to make and then learn to exploit it, in order to improve its policy.

If both the teacher and the learner have the same reward structure, a good strategy for the learner would be to imitate, if possible, the teacher's behaviour. This, in principle, is a much easier and safer policy than attempting to randomly explore the environment on its own. The learner would only need to solve the local problem of following the teacher, but would not need to worry about the global task - the global planning that is now done by the teacher. Although this might not be optimal, this kind of behaviour is transferable between tasks and/or environments and could potentially provide the learner with a better initial policy for exploring an unfamiliar environment. This could lead to a more principled/guided way to explore and could have a major impact on the speed at which the agent discovers areas of interest, especially in a sparse reward setting.

We are also interested in showing that the learner can become autonomous and still perform the task optimally in the absence of the teacher after learning from observations. Indeed, thefi nal goal of a learning agent is to solve tasks on its own and it should still be able to reach that goal after having learned optimal policies from a teacher.

To support our claims, we perform additional experiments in environments where the teacher and the learner don't share the same state or action space. In addition, we use environments of different complexities going from a grid world to a continuous control problem within a simulated physical environment [33] so as to show the genericity and scalability of the framework.

3 LEARNING BY OBSERVING A TEACHER

In this section, we formalize the experimental design described in Sec. 2. More precisely, the primary goal is to show that an agent (the learner) can still learn via an RL algorithm in an environment where a teacher is added. The environment is modelled by a Markov Decision Process (MDP) where the dynamics is one step Markovian and stationary. The Markovian and stationary properties are essential to allow the agent to learn properly as they guarantee that the environment is stable and predictable. Therefore, after introducing some notations in Sec. 3.1, we show, in Sec. 3.2, how an MDP where a teacher agent is introduced still remains an MDP. Then, we provide a formal setup where the experimental design is formalized and we show how this setup can still be seen as an MDP even when a teacher is added. Finally, we show how adding a teacher in an environment can improve the global policy of the learner.

3.1 Notation

An MDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is the set of actions available to the agent, $\mathcal{P}$ is the transitional kernel modelling the one-step Markovian dynamics and gives, for each state and action, a probability distribution over next states, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function that represents the local benefit of doing action a in state s and $\gamma \in [0, 1]$ is a discount factor.

A stochastic policy π maps each state to a probability distribution over actions $\pi(\cdot|s)$ and gives the probability $\pi(a|s)$ of choosing action a in state s. Given such a policy π, the value function $V^\pi(s)$ is the expected cumulative discounted reward associated with following this policy:

$$V^\pi(s) = \mathbb{E}^\pi \left[\sum_{t=0}^{+\infty} \gamma^t R(s_t, a_t) \right],$$

where $\mathbb{E}^\pi$ is the expectation over the distribution of admissible trajectories $(s_0, a_0, s_1, a_1, \dots)$ obtained by executing the policy π starting from $s_0 = s$ and $a_0 \sim \pi(\cdot|s_0)$. In RL, we are interested in finding an optimal policy π^* that results in the maximum value function $V^* = V^{\pi^*} = \max_\pi V^\pi$.

3.2 MDP with a teacher agent

By introducing the teacher (following policy π_e) in the learner's environment and making it visible in the observational state of the learner, we change the learner's MDP. The resulting MDP can be parameterised as follows: $\tilde{\mathcal{M}} = \langle \tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}, \gamma \rangle$, where now the state space consists of: i) a part of the state space that can be directly influenced by the learner, we will refer to this part of the state space as the *controllable* part of the state space $\mathcal{S}_c$ and ii) a part of state space that the learner does not have any direct control over, but this is still part of its observational state and includes useful information, $\mathcal{S}_{\neg c}$. In our case, $\mathcal{S}_{\neg c}$ will include observations corresponding to the presence of the teacher. Given this factorization, $\tilde{s} = (s_c, s_{\neg c}) \in \mathcal{S}_c \times \mathcal{S}_{\neg c}$, we assume that the transition dynamics $\tilde{\mathcal{P}}$ factorizes as follows:

$$\tilde{\mathcal{P}}((s'_c, s'_{\neg c})|(s_c, s_{\neg c}), a) = \mathcal{P}(s'_c|s_c, a)\mathcal{P}^{\pi_e}(s'_{\neg c}|s_{\neg c}),$$

where $\mathcal{P}^{\pi_e}(s'_{\neg c}|s_{\neg c})$ is the probability to reach $s'_{\neg c}$ starting from $s_{\neg c}$ and following the teacher. This assumption simply means that the controllable part of the next state depends only on the controllable part of the state and the action of the agent, and that the non-controllable part of the next state depends only on the non-controllable part of the state and the teacher policy. In addition, if the teacher has a stationary behaviour (w.r.t. the non-controllable part of the state) then this implies that $\tilde{\mathcal{P}}$ is a well defined transitional kernel. Therefore $\tilde{\mathcal{M}}$ is a well-defined MDP.

3.3 Formal setup

In this section, we will show formally that at least two of the desired behaviours are made possible in the context of observational reinforcement learning: imitation and active task identification (a.k.a. information seeking). We consider a set of MDPs that share states, actions, transition dynamics and discount factor, but differ in the reward function $G = \{\mathcal{M}_t | \mathcal{M}_t = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}_t, \gamma \rangle\}$. Let us consider uniformly sampling one of these MDPs $\mathcal{M}_t \sim U(G)$ and unrolling one episode given this choice. Once the episode has terminated, we re-sample from G and repeat this process. Please note that this procedure, defines another MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where $\mathcal{R} = \mathbb{E}_{\mathcal{M}_t}[\mathcal{R}_t]$[1]. This holds only when the transitional dynamics $\mathcal{P}$ is shared across the candidate MDPs. We are interested in the policy π^* that performs well, in expectation across this family of MDPs:

$$\pi^* \in \arg\max_\pi \left(\mathbb{E}_{\mathcal{M}_t \sim U(G)}[V^\pi_{\mathcal{M}_t}] \right)$$

Introducing a teacher into the formal setup. Once a teacher is introduced, the set of MDPs becomes $\tilde{G} = \{\tilde{\mathcal{M}}_t | \tilde{\mathcal{M}}_t = \langle \tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{P}}, \mathcal{R}_t, \gamma \rangle\}$ where $\tilde{\mathcal{M}}_t$ is the augmented MDP relative to $\mathcal{M}_t$ (as seen in Sec.3.2). We are interested in the policy $\tilde{\pi}^*$ that performs well, in expectation across $\tilde{G}$:

$$\tilde{\pi}^* \in \arg\max_\pi \left(\mathbb{E}_{\tilde{\mathcal{M}}_t \sim U(\tilde{G})}[V^\pi_{\tilde{\mathcal{M}}_t}] \right).$$

Now, we would like to know if it is possible to do better than the stationary policy π^*, when placing the learner in the augmented setup $\tilde{G}$. More precisely we want to know if $\mathbb{E}_{\tilde{\mathcal{M}}_t \sim U(\tilde{G})}[V^{\tilde{\pi}^*}_{\tilde{\mathcal{M}}_t}] \geq \mathbb{E}_{\mathcal{M}_t \sim U(G)}[V^{\pi^*}_{\mathcal{M}_t}]$. The answer is yes. Indeed, a direct consequence of moving from $\mathcal{M}_t$ to $\tilde{\mathcal{M}}_t$ is an augmentation in the state space, which results in an expansion of the policy space. Since the set of possible policies in $\tilde{\mathcal{M}}_t$ is a superset of the policies in $\mathcal{M}_t$, it is easy to see that the optimal policy in the augmented space is at least as good as the optimal policy in the original MDP.

This means that the learner can leverage the behaviour of the teacher in order to learn a policy $\pi(a|s_c, s_{\neg c})$ that is better than $\pi(a|s_c)$. Given this setup, let us take a closer look at two particular case studies, where this might occur:

[1]Note:

$$\mathbb{E}_{\mathcal{M}_t}[V^\pi_{\mathcal{M}_t}] = \mathbb{E}_{\mathcal{M}_t}\left[\mathcal{R}_t(s, a) + \gamma \mathbb{E}_{s' \sim P(.|s,a)} V^\pi_{\mathcal{M}_t}(s') \right],$$
$$= \mathbb{E}_{\mathcal{M}_t}\left[\mathcal{R}_t(s, a)\right] + \gamma \mathbb{E}_{s' \sim P(.|s,a)} \left[\mathbb{E}_{\mathcal{M}_t}[V^\pi_{\mathcal{M}_t}(s')] \right].$$

i) **Imitation**. Let us consider the case where $\mathcal{A} = \mathcal{A}_e$, where $\mathcal{A}_e$ is the action-space of the teacher. Let us assume the teacher's policy is better than the stationary policy defined in eq. (3.3), π^* – otherwise there is no incentive to deviate from π^*. If $s_{\neg c}$ includes the actions taken by the teacher, then imitation can trivially take place, by just reading off the desired action from $s_{\neg c} : \pi(a|s_c, s_{\neg c}) := \pi(a_e|s_c, s_{\neg c})$[2]. It is actually a known result that observing teacher's actions is mandatory to achieve actual imitation [10]. In a purely observational setting though, the learner would not have access directly to the teacher's actions. Nevertheless, we also know that in a deterministic environment, observing the effects of one's actions in the environment is enough to infer the action that was performed. To learn this mapping we need to 'remember' or keep track of at least two consecutive observations of the teacher's behaviour. Thus, if imitation is to emerge, it can only do so in a system able to distill this information from the observational state. For this our agents will need memory or augmentation of the state space to include several time steps.

ii) **Information seeking behaviour**. If the learner knows the task (i.e. the MDP ($\mathcal{M}_t$) is identified), then its policy can become independent of the teacher as it can optimally perform the task without any input from the teacher. We thus study here the case where the teacher's behavior can help in identifying the task. We denote $\pi^*_{\mathcal{M}_t}$ the optimal policy in $\mathcal{M}_t$. We will make the additional assumption that, given t, the optimal policy in the MDP $\mathcal{M}_t$ with or without the teacher is the same. Formally, $\pi^*_{\mathcal{M}_t} = \pi^*_{\tilde{\mathcal{M}}_t}$. Thus, if the identity t of the task is known, the optimal behaviour of the agent would be to just switch between these optimal policies given the context t: $\tilde{\pi}(a|\tilde{s}, t) = \pi^*_{\mathcal{M}_t}(a|s_c)$. This policy is optimal for each of the sampled MDPs and results in an optimal behaviour in $\tilde{\mathcal{M}}$, provided the context t is known. If this information can be distilled from the observation of the other teacher, the learner can improve over the stationary policy defined in eq. (3.3). Thus if $\exists g : \mathcal{S}_l \rightarrow \mathbb{N}_T$ s.t. $t = g(s_{\neg c})$, then $\exists$ a stationary policy in the augmented state space $\tilde{\mathcal{S}}$ that (can) outperfom $\pi^*_{\mathcal{M}}$:

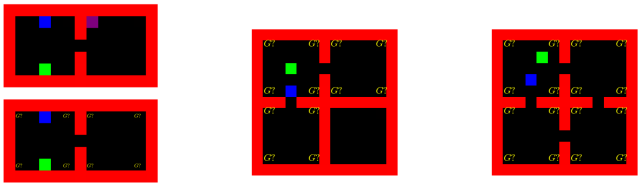
$$\tilde{\pi}(a|\tilde{s}) = \tilde{\pi}(a|\tilde{s}, g(s_{\neg c})) = \pi^*_{\mathcal{M}_t}(a|s_c)$$

Note that g can take into account in its computation, a series of observations of the teacher (several steps of its trajectory). This can be practically implemented by stacking several recent observations in our current state, or relying on a memory/recurrent model to distill temporal information as part of its hidden state and infer the current context.

4 EXPERIMENTS

For our experiments, we choose a widely used DeepRL algorithm: the Asynchronous Advantage Actor-Critic (A3C) algorithm [19]. This learns both a policy (the actor), $\pi_{\theta_\pi}(a_t|s_t)$ and value function (the critic) $V_{\theta_V}(s_t)$ given a state observation s_t. The two approximations share the intermediate representation and only diverge in thefi nal fully-connected layer. The policy is given by a softmax over actions. The setup closely follows [19] including the entropy regularization and the addition of an LSTM layer to incorporate memory. We use both a simple feed-forward and a recurrent version in our experiments. For further details, we refer the reader to the original paper.

[2]We assume the two agents have the same starting position, but this is can relaxed, by first employing a policy that gets the learning agent close enough to the teacher, after which the mimicking behaviour can occur. We assume the learner to always be one step behind the teacher, but longer time difference can be negotiated via a memory component.



(a) Level 1 (b) Level 2 (c) Level 3

Figure 1: Environment snapshots

As scenarios, we consider a series of simple navigation tasks. Wefi rst start with a two-room layout (Fig. 1a) and place the two agents in this environment. Possible goal locations considered are the corners of each room (8 in total). At the start of each episode, we sample uniformly the location of the goal and make this information available to the teacher at all times. For the learning agent, we consider both including and occluding the goal in its observational space. In the second version of this task, the learner does not know which goal is activated at this point in time. It can only learn that the possible positions of the reward are the 8 corners. If it ignores the teacher's behaviour the optimal stationary policy would be to visit all of these locations in the minimum time possible. On the other hand, the teacher has access to the goal and can seek it directly. By observing its behaviour, the learner could potentially disentangle which corner is active, or at least narrow down the correct room, and use this information to improve its performance.

4.1 Global view: task identity occluded, but perfect information otherwise.

We provide the learner with a top-view representation of the environment. We use a one-hot encoding for each element present in the environment: a channel for the layout (L), one for the position of the agent (A), one for the position of the teacher (T), one for the position of the goal (G). We implement four variations of this observational space: LA, LAG, LAT, LAGT. Once the teacher reaches the goal, it will respawn in a random position in the environment and re-demonstrate the task. This gives the learner multiple chances of seeing the teacher and witness demonstrations from various parts of the environment. This is particularly important in the partially observable setting, as the learner might lose track of the teacher especially in the beginning.

We run this experiment with an A3C agent with two small convolutional layers, followed by a fully connected layer. The results are displayed in Fig. 2a (Feed-forward network), 2b (LSTM with 32 units). Thefi rst thing to notice is that, with perfect information, the learned policies with or without the teacher have the same asymptotic performance. Moreover the speed at which this is reached is not negatively impacted by the expansion of the state space. Thus, for this experiment we conclude that the presence or absence of

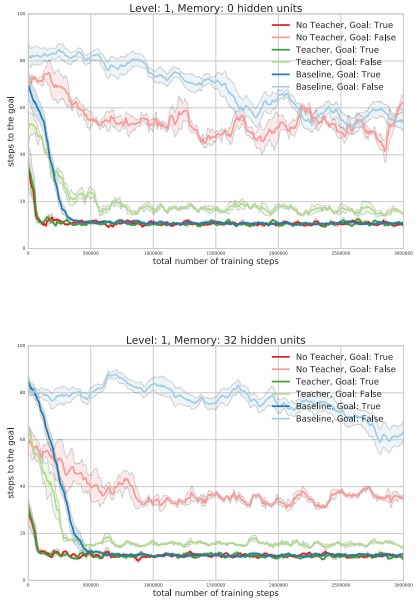


Figure 2: Level 1: Performance during training as measured by the number of steps to the goal. Red curves: learning agent alone in the environment. Green curves: learning agent shares the environment with a teacher. Blue curves: learning from scratch. Bold coloured curves: the goal is present (LAGT, LAG). Light coloured curves: the goal is occluded (LA, LAT).

the teacher does not have an impact on the learning process – this is observed across multiple experiments.

In the second case, when the learner has impoverished knowledge of the environment, we can clearly spot a difference between the performance of the learner when sharing in the same environment with the teacher and acting on its own. For thefi rst part of the training, the agent achieves similar performance, but at some point, the agent sharing the same environment with the teacher manages to leverage this extra signal to significantly improve its policy.

As we observed that the learner is able to exploit the teacher's behaviour for its own benefit, we now want to know if it is a transferable knowledge across different environments. To test this out, we constructed two other "levels", by adding an additional room each time and thus smoothly increasing the difficulty of the task. A depiction of these levels can be found in Fig. 1b, 1c. When extending the environment, we also naturally extend the number of possible goal locations. The average number of steps to the goal increases and when the goal location is occluded, the 'blind' stationary policy will be quite expensive, thus it becomes more and more crucial for the agent to leverage the teacher.

We train on these new levels in a curriculum fashion:fi rst we train on level 1, then continue training on level 2, andfi nally on level 3. This is mainly done to speed up training time, but also we expect the agent to learn the importance of the teacher in level 1 and continue to employ this knowledge in its learning process in

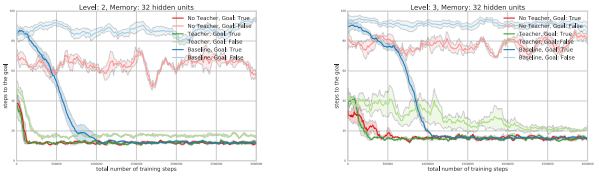


Figure 3: Level 2 and 3: Performance during training as measured by the number of steps to the goal. Red curves: learning agent alone in the environment. Green curves: learning agent shares the environment with a teacher. Blue curves: learning from scratch (w/o curriculum). Bold coloured curves: the goal is present (LAGT, LAG). Light coloured curves: the goal is occluded (LA, LAT). When the goal is occluded we can see that the agent can do substantially better by leveraging the teacher's behavior.

the next levels. The specification of the observational state of the learner is maintained throughout the curriculum: with/without the teacher, with/without the goal visible. The results are compiled in Fig. 3a, 3b. For reference, we included the baseline of learning in these levels starting from a random initialization, with the curriculum. One can see that the curriculum helps the learning process both when the goal is visible and when it is occluded. The performance is slightly better to begin with and then convergence is achieved considerably faster. Moreover, the presence of the teacher consistently improves the policy of the 'blind' learner across all levels. It is also worth noting, that for the last level when the goal is occluded, both the baseline and the lone agent are not always able to complete the task, whereas the agent leveraging the teacher can almost instantaneously complete the task at each episode (the transfer to level 3 is almost zero-shot).

4.2 Local view: agent needs to actively keep track of the teacher, if useful

We have seen in the previous section that by just using RL on the same reward signal, we can obtain different policies when environment is augmented with the presence of a teacher. Although occluding the goal might seem like a somewhat artificial task, please keep in mind that this is just a proxy for impoverished or imperfect information. A more natural example of this would be partial observability – when the agent has only a local view of the environment. We thus simulate this by taking a local window of observation centered around the agent. The rest of the setup remains the same, but note that we are changing the nature of the problem quite drastically. This is especially true if one thinks about the setting in which the teacher is in the picture. In this new scenario, the learner will have to actively pursue the teacher if it has any hope of benefiting from its presence. If it cannot see the teacher, the learner cannot learn from it. Because of the increased complexity, we now start with only one of the two rooms in level 1 and we basically treat this as pre-training of simple navigation skills with local view. In this level, the learner can learn to navigate to the goal and visit the potential locations where the goal is hidden, but we do not observe any benefit from having the teacher present. The size of the local window of observation is the size of the room – if the agent is in

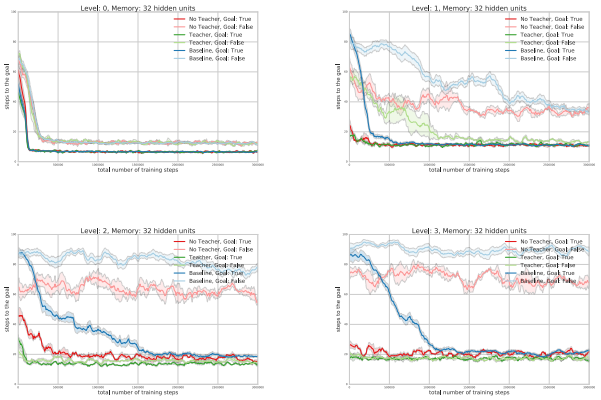


Figure 4: Local View: Performance during training (average steps to the goal), in all levels of the curriculum. Green curves: the learning agent shares the environment with the teacher. Blue curves: learning from scratch in this environment. Bold coloured curves: the goal is present (LAGT, LAG). Light coloured curves: the goal is occluded (LA, LAT). The teacher's presence has a significant impact on the performance in the training and quality of the end policy. It universally improving, or at least matching the performance of the lone agent (level 0).

the middle of the room it can see the whole room. This always means that the problem of keeping track of the other agent is not very difficult in thisfi rst setting. The learning curves can be found in Figure 4. Nevertheless, a different story emerges when we look at the training in level 1. First, we observe that when the goal is hidden, but the teacher is present, we get an improvement in policy over the scenario where the agent is alone in the environment. This is consistent with what we have seen before in the global view. When the potential benefit over the 'blind' stationary policy is big enough, the learner begins to 'pay' attention to the teacher and uses these observations to improve its own policy. This carries on to levels 2 and 3. Furthermore, at these last levels, we can see that in the cases where the teacher is present, the asymptotic performance of the agent matches or slightly outperforms that of the lone agent with the *goal visible*. This is remarkable as we are now seeing an improvement in policy even when the goal is (locally) visible. This is because in partial observability the trajectory of the teacher still contains valuable information about where the goal might be. At each step, the teacher narrows down the possible locations.

4.3 Breaking away from the teacher

The fact that thefi nal performance is independent of the goal's presence or absence in the state space suggests that the only information the agent learns to exploit is the behaviour the teacher. Visual inspection of thefi nal policies fully supports this intuition. The behaviour that emerges is the following: the agent seeks the teacher and tries to get as close as possible to it, after which it simply follows it to the goal.[3] This is potentially a very useful, transferable strategy to a different environment. To test this idea, we expand further our environment to 9 rooms and without any further learning, the previously trained agent can successfully negotiate this new environment with the help of the teacher and succeeds infinding new goals it has never seen before. Nevertheless, relying always on the teacher is somewhat unsatisfactory. In fact, if we do eliminate the teacher, the learning agent is quite lost. It will only reach for the goals currently visible if any, and otherwise will continue waiting for the other agent to appear, not even attempting to leave spawning room. Yet, this is an agent that has previously negotiates this environment with the help of the teacher, but has not retained the particularities of this environment, mainly because it did not need to. In order to break this dependence, we propose a simple (curriculum) strategy: we mask the presence of the teacher with some probability that increases over time. When this probability reaches 1 and the agent becomes completely independent of the teacher. We start with a masking probability of 0.25, then after 250k steps, we increase it to 0.5, then 0.75 andfinally 1.0. At the end of this process we successfully achieved an agent that can tackle the environment on its own, without the teacher.

4.4 Different action space

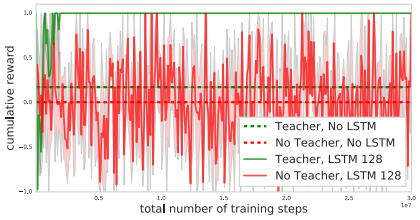
We ran a slight modification of the above, to show that the two agents (the learner and the expert) do not need to share body or action space. We were able to replicate the previous results with an slight modified learning agent, which now have four additional actions (NW, NE, SW and SE) in addition to the four primitive cardinal directions, shared with the expert. We observe that these new actions are used quite extensively by the learner – more than 50% of the time – thus the learner does not simply default to copying the actions of the teacher, but uses its whole actions space as this often offer an more effective way of navigating the environment.

4.5 Information Seeking

Although the previously setting is quite natural and we see a level of imitation emerging when the agents share the task, observation learning can be useful even if the agents do not have the same task – as long as the behaviour of another agent can inform a learning agent about the task at hand or its behaviour contains information that is useful in achieving the task optimally. We will explore two simple scenarios. Firstly, we consider a scenario where the two agents have inversely correlated rewards. Consider two goals (fruits) in the environment – their positions are randomized each episode, and one give positive reward (+1 to the learner, -1 to the teacher) and one give negative reward (-1 to the learner, +1 to the expert). But both of them look the same to the learner. Only the teacher knows, at each trial, which fruit it should pick up. Because the rewards of the two agents are inversely correlated, now the learner, instead of following the teacher, has to infer or observe which fruit the teacher is going for and head for the opposite fruit. An agent with memory successfully learns this behavior Fig. 5a.

The second scenario is very similar: two fruits in the environment, they look the same for both agents now, one give positive reward (+1), the other one gives negative reward(−0.1). The two agents share the reward, and now the teacher has the same sensory information as the learner (it can not distinguish between the

[3] Videos of these behaviours can be found at here.



(a) Negatively correlated reward

(b) Observing a visual effect on the other agent

Figure 5: Cumulative reward during training. Green curves: the teacher is present in the environment. Red curves: the learner is alone. The curves represent the performance of the a memory agent with and without the another agent. For reference (in dotted line) the performance of the same agent with a LSTM.

fruits, so it has to take a gamble if it want to receive any positive reward). We introduce a new element to the setup, which is quite natural in a true observational setting: in addition to observing the behaviour of an agent, one would (sometimes) observe the effect of an interaction – in a real situation, we would observe if the agent is happy, sad, if it got hurt etc. These are very informative cues. To simulate that, when the teacher eats one of the fruits, it will momentarily change color (red or white), indicating if it received positive or negative reward, if it ate the right or wrong fruit. Another important difference to the previous scenarios, is that now the teacher does not have additional or privileged information anymore. Information that could be somehow communicated via its behaviour. Instead, here additional information is provided by the reaction of the teacher when eating the right/wrong fruit. In this case, the teacher is suboptimal as it will try at random one of the fruits. If the teacher is not present, when endowed with memory, our agent will just try one fruit at random and remembers if it was good or bad. If not successful in thefi rst try, it will go for the good fruit after eating the bad fruit. If the learner has no memory, its overall reward is much worse as it can not remember which fruit was previously visit. This can be seen from the performance curves in Fig. 5b. Finally, if the teacher is present in the environment, a agent with memory, eventually learns not to take this gamble, but wait for the teacher to do so; observe its 'reaction', and directly seek the right fruit, without risking the negative penalty (Fig. 5b).

4.6 Scaling up: Mujoco experiments

Lastly, we wanted to validate the previous insights in a more challenging and realistic scenario. For this, we used the MuJoCo physics engine [33] to create two small maze-like tasks. We opted for two

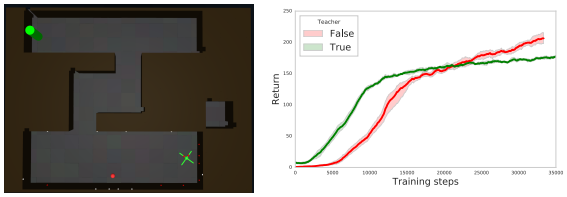


Figure 6: Mujoco: Learning curves with/without the teacher

different agents with different bodies and hence very different action spaces, one simple ball that can roll forward and backwards, steer and jump (3 action dimensions, 5 DoF) and a quadrupedal body (14 DoF; 8 action dimensions). Both agents were equipped with a simple LIDAR-like raycasting system that allowed to them to sense the distance to the nearest object and its type (wall, other walker body, target) in each direction in the horizontal plane. A scenario particularly challenging for imitation learning, as both observations and actions require some kind of explicit or implicit mapping from one agent state-action space to the other. This is especially non trivial as not all states can be achieved by both agents. For instance, the quadrupedal body cannot get as close to the walls and can easily get stuck at corners. Furthermore, the agents interfere with each other physically (which violated the assumption made in Sec. 3).

We start by placing both agents in a small room that also contains a randomly positioned goal. This goal is visible only to the teacher, although getting close to it rewards both agents. This is effectively a scaled up version of the scenario investigated above, but with notable additional difficulties due to the physical embodiment. Training the two agents in this scenario serves two purposes: first, it teaches both agents to locomote (note that especially the quadruped does not move coherently at the beginning of learning). Secondly, it serves to establish the bond between the two agents, as this has been something that we have seen leads to positive transfer when the environment increases in complexity. The second phase of this experiment takes place in a two room enclosure (Fig. 6a). Now the goal is visible to both agents and the new teacher has been previously trained to reliably achieve navigation to this goal.

What we observe again, is that the learner tries to keep close to the new teacher and sees the reward much sooner than our baseline, the quadruped trained in the initial scenario but without the teacher present[4]. This is despite the fact that the scenario is considerably more challenging and the quadruped needs tofirst learn to handle additional challenges not encountered in the initial training scenario (like navigating around corners), and that physical interference between the agents is possible when the teacher is present. Nevertheless, we can see that the learner manages to get to the goal and experience the positive reward much sooner in learning when the teacher is present. This extra signal it learnt during the previous phase, acts a more targeted exploration strategy which speeds up learning. Learning curves comparing the training of these two agents are provided in Fig. 6b. We observe that learning slows down at some point. We speculate that this is, at least partially, due to the physical interference between the agents (bumping into each

[4]Note that this is a strong baseline in that in this setting the quadruped benefits from having learned to walk in the initial scenario and will coherently explore the maze randomly.

other; see video). However, asymptotically both agents perform on par when the teacher is removed.

5 CONCLUSION AND FURTHER WORK

This paper demonstrates that observational learning can emerge from reinforcement learning, combined with memory. We argued that observational learning is an important learning mechanism that can be a 'cheaper' alternative than some of the current approaches to learning from teachers. In particular, this approach does not require any explicit modelling of the teacher, nor mapping their experience in the learner's state and action space. We have shown in our experiments, that relying only on the (sparse) reward signal given by the environment coupled with only the sheer presence of a teacher, can lead to a variety of behaviours, ranging smoothly between imitation and information seeking. We also demonstrated that, via a curriculum, we could end up with an autonomous agent that solves tasks without the presence of the teacher although it learned through observations of a teacher. We also showed that the teacher and the learner did not have to share the same action and state space which adds to our claim that only effects on the environment are important and not actual teacher's actions.

This is an initial work which should be extended further, in other settings. Especially, we want to test scenarios where the goal of the teacher and of the learner are not strictly aligned. We performed a preliminary study where there is a negative correlation (*e.g.* they are in opposite directions in the room). In the same way, we should study how optimal the teacher has to be, how much its motivation should be correlated to the one of the learner.

REFERENCES

[1] Peter Abbeel and Andrew Y. Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[2] Brenna. Argall, Sonnia Chernova, Manuella Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.

[3] Chris. Atkeson and Stefan Schaal. 1997. Robot learning from demonstration. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[4] Albert Bandura and Richard H Walters. 1963. *Social learning and personality development*. Vol. 14. JSTOR.

[5] Albert Bandura and Richard H Walters. 1977. Social learning theory. (1977).

[6] Jessica Chemali and Alessandro Lazaric. 2015. Direct Policy Iteration with Demonstrations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[7] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[8] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. 2018. Learning from Demonstrations for Real World Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

[9] Cecilia M Heyes. 1993. Imitation, culture and cognition. *Animal Behaviour* 46, 5 (1993), 999–1010.

[10] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NIPS)*.

[11] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. 2016. Showing versus doing: Teaching by demonstration. In *Advances in Neural Information Processing Systems (NIPS)*.

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] Beomjoon Kim, Amir massoud Farahmand, Joelle Pineau, and Doina Precup. 2013. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems (NIPS)*.

[14] Edouard Klein, Matthieu Geist, Bilal Piot, and Olivier Pietquin. 2012. Inverse reinforcement learning through structured classification. In *Advances in Neural Information Processing Systems (NIPS)*.

[15] J Zico Kolter, Pieter Abbeel, and Andrew Y Ng. 2008. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Advances in Neural Information Processing Systems (NIPS)*.

[16] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[17] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research* 17, 1 (2016), 1334–1373.

[18] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andy Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. 2017. Learning to navigate in complex environments. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[19] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[21] Gergely Neu and Csaba Szepesvári. 2009. Training parsers by inverse reinforcement learning. *Machine learning* 77, 2 (2009).

[22] Andrew Ng and Stuart Russell. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[23] Bilal Piot, Matthieu Geist, and Olivier Pietquin. 2013. Learning from demonstrations: is it worth estimating a reward function?. In *Proceedings of the Joint European Conference on Ma-chine Learning and Knowledge Discovery in Databases(ECML/PKDD)*.

[24] Bilal Piot, Matthieu Geist, and Olivier Pietquin. 2014. Boosted bellman residual minimization handling expert demonstrations. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*.

[25] Bilal Piot, Olivier Pietquin, and Matthieu Geist. 2014. Predicting when to laugh with structured classification.. In *Proceedings of INTERSPEECH*.

[26] Dean A. Pomerleau. 1989. *Alvinn: An autonomous land vehicle in a neural network*. Technical Report. DTIC Document.

[27] Nathan Ratliff, J.Andrew Bagnell, and Siddhartha S. Srinivasa. 2007. Imitation learning for locomotion and manipulation. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*.

[28] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on AI and Statistics (AISTATS)*.

[29] Stuart Russell. 1998. Learning agents for uncertain environments. In *Proceedings of the Conference on Learning Theory (COLT)*.

[30] Stefan Schaal. 1999. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences* 3, 6 (1999), 233–242.

[31] Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. 2017. Third-Person Imitation Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[32] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. Cambridge Univ Press.

[33] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*.

[34] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum Entropy Inverse Reinforcement Learning.. In *Proceedings of the Annual Meeting of the Association for Advances in Artificial Intelligence (AAAI)*.