Advances in Cognitive Systems 7 (2018) 117–136 Submitted 6/2018; published 12/2018

Bridging the Gap: Converting Human Advice into Imagined Examples

Eric Yeh ERIC.YEH@SRI.COM
Melinda Gervasio MELINDA.GERVASIO@SRI.COM
Artificial Intelligence Center, SRI International, 333 Menlo Park, CA 94025 USA

Daniel Sanchez DANIEL.SANCHEZ@SRI.COM
Matthew Crossley MATTHEW.CROSSLEY@SRI.COM
Computer Science Laboratory, SRI International, 333 Menlo Park, CA 94025 USA

Karen Myers KAREN.MYERS@SRI.COM
Artificial Intelligence Center, SRI International, 333 Menlo Park, CA 94025 USA

Abstract

Advice is a powerful tool for learning. But advice also presents the challenge of bridging the gap between the high-level representations that easily capture human advice and the low-level representations that systems must operate with using that advice. Drawing inspiration from studies on human motor skills and memory systems, we present an approach that converts human advice into synthetic or imagined training experiences, serving to scaffold the low-level representations of simple, reactive learning systems such as reinforcement learners. Research on using mental imagery and directed attention in motor and perceptual skills motivates our approach. We introduce the concept of a cognitive advice template for generating scripted, synthetic experiences and use saliency masking to further conceal irrelevant portions of training observations. We present experimental results for a deep reinforcement learning agent in a Minecraft-based game environment that show how such synthetic experiences improve performance, enabling the agent to achieve faster learning and higher rates of success.

1. Introduction

Advice is a powerful tool for enhancing learning, but delivering information in a way that can be appropriately used to improve performance can be a complex endeavor. For example, corrective advice should focus the learner on external components (e.g., effects of the motor control) rather than internal components, such as the motor control itself, as internally focused attention harms performance (Wulf et al., 2010). This is hypothesized to be due to conflicting representations between abstract advice and low-level motor programs (Flegal & Anderson, 2008). This suggests that the ability to apply top-down, abstract advice on a simple, reactive learning system requires that the advice essentially "scaffold" the low-level representation (Petersen et al., 1998) rather than interact with it directly. While this difference in knowledge representations is captured both in cognitive architectures (Anderson, 1982; Sun et al., 2001) and human neurophysiology (Henke, 2010), the ability to capture this scaffolding interaction model has yet to be explored.

Table 1. Differences between human advice and inputs suitable for most reinforcement learning agents.

	Humans Advice	RL Inputs
Quantity	Low (10s)	Large (1,000s–1,000,000s)
Conceptual Level	Higher level, more abstract	No abstractions, grounded in environment
Representation	Linguistic	Instance-based

Given this inspiration from studies of human motor skills and memory systems, we explored how abstract advice may be used to guide reinforcement learning for a simple, reactive agent. Human advice has been recognized as a powerful source of guidance for learning systems since the early days of AI (McCarthy, 1959), and much work has been done on integrating advice into symbolic reasoning systems (Mostow, 1983; Golding et al., 1987; Myers, 1996). In the 1990s, reinforcement learning (RL) came onto the scene as an attractive paradigm for continuous, integrated learning and acting. While mathematically elegant, systems for reinforcement learning are often limited to small, toy domains due to their simplicity and inability to scale to complex problems. However, with the explosive success of deep learning during the previous several years has also come impressive gains through the use of neural function approximators to reduce complexity (Mnih et al., 2013; Silver et al., 2017). As RL-trained autonomous systems become more widely used, a critical component for their acceptance is the ability for users to advise and influence autonomy.

A large body of work already fuses the flexibility and learning capability of reinforcement learning with extensions to allow for more complex thought. Examples include using reinforcement learning to learn an operator-selection policy in a cognitive system (Nason & Laird, 2005); supporting hierarchical deep Q-learning networks (DQN) (Kulkarni et al., 2016); developing goal-directed Monte Carlo rollouts to identify courses of action that best match human preferences and constraints (Kaushik et al., 2016); and other work further reviewed in Section 6. However, these approaches implicitly assume that a fundamentally reactive learning algorithm, such as RL, cannot learn to exhibit more complex, goal-directed behavior. Ostensibly, general belief and intuition dictate that such simple algorithms must require additional cognitive machinery to exhibit complex behavior. We contend that another path is possible, one that forgoes heavy modification of the reinforcement learner or reactive learning algorithms.

A fundamental problem is that human-provided advice, which is abstract and rich in its representation, is often not in a form readily usable by RL-trained autonomous agents. In developing *playbooks* (libraries of procedural knowledge) for teams of autonomous agents, we have found that domain subject matter experts often impart their knowledge in the form of high-level goals or constraints, which current learning systems cannot use directly. Although attempts to incorporate advice into RL systems have met with some success (Schaal, 1999; Ng & Russell, 2000; Abbeel & Ng, 2004), these approaches tend to require heavy user involvement in the training process.

A key advantage of RL-based systems is that they generally require little or no domain knowledge, learning strictly from examples garnered through experimentation. However, this focus on instance-based learning leads to a fundamental disconnect between human communication and standard approaches. Table 1 lists some of the differences between human-given advice and in-

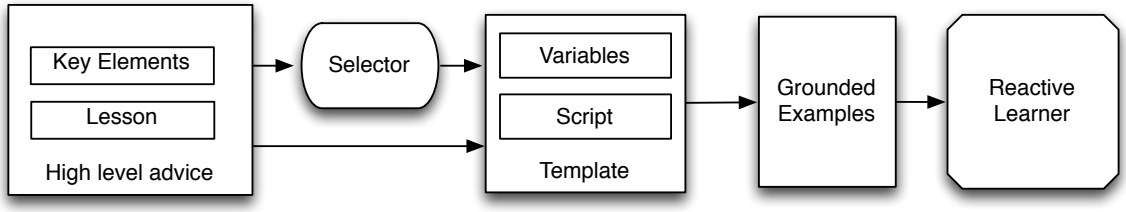

Figure 1. Scaffolding framework: Advice is matched by a selector to a template, which contains a model for generating multiple instanced permutations representing the lesson in the advice. These are then made readily available to a reactive learner.

puts accepted by such systems. Thus, the challenge is in developing techniques that allow abstract human advice to be used to guide reinforcement learning.

Our main research objective is to understand how to take natural human advice – typically a symbolic, language-based input that relies on robust models of the environment – and use it to guide low-level (even model-free) reactive learners such as reinforcement learning. To bridge this gap, we developed *scaffolding*, a framework for converting human advice into inputs that can influence reactive learning algorithms such as reinforcement learning. Scaffolding addresses the *conceptual level* of information and *quantity* of inputs (see Table 1). Our framework, further elaborated on in Section 2, is guided by multiple memory systems theory[1] (Sherry & Schacter, 1987; Squire, 2004) and is based on the following core theoretical ideas:

- *Tenet 1:* Cognitive systems should support complex interactions between abstract cognitive operations and simple, low-level reactive learning systems.
- *Tenet 2:* Because reactive learning processes are encapsulated, advice does not interact directly with the knowledge representation but instead acts as a scaffold to guide the learning.
- *Tenet 3:* Scaffolding can be realized through templates that generate synthetic training examples to shape the reactive learning.

Our primary claim is that simple reactive learning systems can be trained to deal with complex problems without requiring large changes to the underlying algorithms. By taking human advice and developing cognitive *templates* that model the specific components of the environment that are most relevant for learning, we should be able to guide a reactive agent to learn faster while also minimizing the amount of the environmental modeling required for high-level guidance. A theoretical outline for scaffolding is given in Figure 1. We purposefully avoid committing to any specific representation or reasoning mechanism at this level, as several different types of technologies can meet the needs of each component. Instead, we list a minimal set of criteria needed to embody the framework.

Advice is composed of key elements and a lesson. Key elements are the minimal set of world elements that are needed to convey the lesson in the advice. A selection mechanism aligns the

1. Of note, not to be confused with complementary learning systems (Kumaran et al., 2016; McClelland et al., 1995)

advice with the best matching template. Templates themselves have variables, which are matched with key elements from the advice, and a script. The script consists of a sequence of abstract interactions between its variables, along with labels indicating desirability of outcome. When a template is reified by binding advice elements to variables, the script uses these elements and domain knowledge to generate a larger amount of grounded instances.

In the remainder of the paper, we outline our research approach and hypotheses (Section 2), and briefly review reinforcement learning (Section 3). Following this, we describe our research platform (Section 4), our technical approach to applying cognitive-level advice to an RL system (Section 4), and the results of our novel architecture (Section 5) that show how advice-derived training memories improve the learning rate for a deep RL system. Finally, we discuss related work in cognitive psychology and in computer science (Section 6), and conclude with a discussion about future directions (Section 7).

2. An Approach to Learning from Advice

Our theoretical approach borrows principles from cognitive psychology and skill acquisition to develop methods for how an agent (human or system) can take high-level information and use it to guide low-level learning and representations. Expert skills rely on multiple, interacting memory systems, whereby a declarative system supports flexible knowledge representations that can be used to guide a procedural system that supports slow, inflexible learning through repetitive practice (*Tenet 1;* Milner et al., 1998; Fitts & Posner, 1967; Taylor & Ivry, 2012; Anderson, 1982; Henke, 2010). Akin to providing advice to a reactive learning agent, a coach provides verbalizable input (high-level advice) to a student to disrupt an incorrect motor program (low-level procedural representation), otherwise known as *deliberate practice* (Ericsson et al., 1993). The abstract representation of advice serves as a "scaffold" (Petersen et al., 1998) to guide the development and production of low-level motor programs (Chaffin et al., 2010), which are characterized by their inflexible, encapsulated representations (*Tenet 2;* Rozanov et al., 2010; Reber & Squire, 1998). Thus, our approach is to generate simple cognitive scaffolds, or *templates*, to guide an RL agent through the learning process – essentially constraining the search space for our novice learner. Because RL agents only learn through instance-based examples, much like a skill learner's procedural memory system, these templates are used to generate advice "episodes" to guide learning. Advice episodes were inspired by the technique of mental imagery, where a learner mentally rehearses a desired behavior to improve skill learning and performance (surveyed in Weinberg (2008)).

A potential complication is that the templates may lead to learning of incorrect information (i.e., spurious correlations). To combat this, we assessed "saliency masking," where only the most relevant information was retained in the episode (e.g., if an episode is focused on teaching an agent to avoid lava, it would only retain the lava in the environment). This was motivated by the visual attention literature in cognitive psychology, which hypothesizes that human attention focuses on only a portion of the visual field, following a "zoom lens" or an "attentional spotlight" model (Eriksen & St. James, 1986; Eriksen & Yeh, 1985; Posner et al., 1980). Moreover, studies in developing perceptual skills for sports have shown that expert-driven direction of visual attention can improve performance. For example, having a coach highlight important portions of a training video

improved a novice's ability to anticipate badminton plays (Hagemann et al., 2006). The important commonality that we leverage is the fact that only a portion of the visual field is considered useful for learning or making a decision. It is interesting to note that this issue with saliency is a hallmark of explicit, cognitive processing, whereby implicit learning (like a reactive learning agent) is traditionally able to learn complex rules from a high-dimensional space despite a lack of a clear, salient cue (Jiménez & Méndez, 2001).

To implement our advice scaffolding on a reactive learner we selected reinforcement learning, specifically DQN. Reinforcement learning was chosen because it is one of the most fundamental algorithms for learning stimulus-response behaviors, suggesting that any demonstrated improvements should be generalizable to other approaches (such as policy-gradient RL algorithms). We selected DQN because it is an effective state approximator that automatically learns a state representation without a large engineering overhead. We posit that the advice templates and subsequent advice episodes should be as simple as possible to appropriately constrain the RL agent and avoid common problems with machine learning systems. To this end, we utilize templates that reinforce a behavior (approach) or punish a behavior (avoidance) and isolate the minimal information required to focus the agent on the salient information that needs to be associated with the outcome. This approach is aimed at mitigating a major challenge for RL systems, which is learning spurious correlations between training observations and desired outcomes. This occurs because machine learning systems in general consider the entire input equally. For example, an autonomous driving system would favor braking when it saw heavy cloud cover simply because its training set was collected on a rainy and cloudy day. While statistics will eventually overcome this problem, getting enough data to reach this point may be difficult or infeasible.

By minimizing the information available in the training episodes, we are leveraging the concept of instructor-directed visual attention. Studies of human skill learning have found that novices learned much faster if their attention was directed to the portions of training videos that instructors deemed salient (Hagemann et al., 2006). Our artificial equivalent is using *saliency masking*, where we occlude portions of training observations to leave only the elements deemed essential for conveying the key lessons in a piece of advice.

Following our research motivation, we explored the following hypotheses:

- *Hypothesis 1:* Templates based on human advice can be used to generate training episodes that enhance learning of a model-free RL agent.
- *Hypothesis 2:* Masking the environment so that only the most salient information is available will enhance the RL agent's ability to learn from the templates.

Figure 2 depicts our scaffolding implementation. We author advice, such as "move toward target", in terms of existing templates and variables. While this also aligns advice to templates for us, automated solutions for aligning templates to less formal representations, such as natural language, are feasible as well. Templates are instantiated with world elements referenced in the advice, making it more concrete. We increase the quantity of data by having the system generate a variety of episodes illustrating the key element in the lesson, where episodes consist of actions, observations, and rewards. Unlike episodes drawn from interaction with the target environment, these "imaginary" episodes are constructed internally from a domain-specific action model and the selected template.

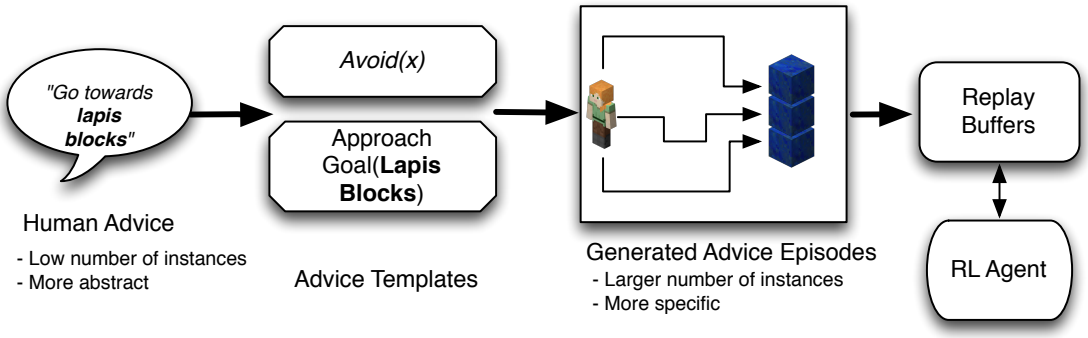

Figure 2. Overview of our scaffolding implementation. A single piece of human advice is matched against the domain-relevant templates, which are instantiated with world elements. Our system applies a simple action model to generate a larger number of training episodes that illustrate the key elements of the advice. These advice memories are fed into replay buffers, which the RL agent uses to learn from.

For example, "move toward target" generates several different paths for the agent to reach its target. These are stored into a bank of replay buffers, a memory store used by the RL agent to store its experienced memories and to draw from for learning. This approach was inspired by mental imagery studies, in which human subjects were directed to visualize themselves executing sports skills or playing out imaginary sports scenarios. When combined with practice, mental rehearsals improved the participants' performance on these tasks (Weinberg, 2008).

3. Review of Reinforcement Learning

A reinforcement learning agent learns how to operate in an environment to maximize cumulative reward (Sutton & Barto, 1998). It does so by taking exploratory action in the environment, then accumulating positive and negative rewards as a result. The environment is typically formulated as a Markov decision process (MDP), which consists of five elements:

- a finite set of states S;
- a finite set of actions A;
- a state transition function $T(s'|s,a) = \Pr(S_{t+1} = s'|S_t = s, A_t = a)$ that specifies the probability of transitioning from one state to another given a particular action;
- a reward function $R(s) \in \mathbb{R}$ over states; and
- a discount factor $\gamma \in [0,1]$ over future rewards.

The aim of an RL agent is to find an action-selection policy $\pi : S \times A \rightarrow [0,1]$ that will lead to the best reward outcome, without knowing either the state transition probability function or the reward function in advance.

There are numerous forms of RL; here we use Q-learning (Watkins & Dayan, 1992), a model-free algorithm that bypasses the state transition function and instead learns a function $Q^*(s, a)$ that captures the expected discounted reward from taking action a in state s and choosing actions. The optimal Q-value function, $Q^*(s, a)$, is computed by taking the action that leads to the greatest expected reward in subsequent time steps:

$$Q^*(s, a) = \max_{\pi} \mathbb{E}[r_t + \gamma r_{t+1} + \cdots | s_t = s, a_t = a, \pi] \quad (1)$$

In our work, we use deep Q-learning networks (DQN), a variant of Q-learning that uses neural networks to perform data-driven approximation of Q-values, yielding better generalizability over previous methods (Mnih et al., 2013). A key component of many deep RL systems, including DQN, is *experience replay* (Lin, 1992). Originally developed to improve sample efficiency and accelerate learning, experience replay has also been used to break temporal correlations that arise when estimating Q-values. In experience replay, the agent stores observed interactions as experience tuples in a *replay buffer*. An experience tuple $< s, a, s', r >$ consists of an initial state s, the action taken a, the resulting state s', and resulting reward r. RL agents update their Q-value estimates by sampling from tuples in the replay buffer rather than just the recent tuples from interaction with the environment.

We note that no requirement exists that replay buffers be given only actual environmental experience, and we harness this characteristic by inserting synthetically generated training memories into a replay buffer. By transforming user advice into these training memories and including them in the learning updates, we provide a mechanism for human guidance to influence the agent's learning. In the next section, we discuss our approach for operationalizing user advice by augmenting agent experience with synthetic training memories.

4. Learning from Advice in Minecraft

For our experiments, we used Project Malmo, an instrumented Minecraft[2] environment (Johnson et al., 2016). Minecraft is a 3D video-game environment, where the world is composed of different types of blocks, such as bedrock, cobblestone, logs, or lava. The game features basic mechanics that enable a variety of causal interactions between the agent and blocks in the environment, such as using pickaxes to mine ore or axes to remove trees. Game environments are excellent experimental platforms because they are controlled domains, generally inexpensive, and act as semi-realistic proxies for real-world scenarios. Minecraft, in particular, offers a highly flexible, controllable, and extensible environment. Its wide array of possible interactions supports modeling that ranges from simple tasks to complex multi-goal problems. This provides the necessary ability to create a training regime that consists of simple advice templates and complex environment exploration.

To focus on the key research problem of using advice to scaffold a reactive learner for improved learning, we simplified the percept and action-learning problem by using Malmo's discrete action mode: An agent can move forward and backward in one-block steps, it can turn in 90-degree increments, and it can use its pickaxe to remove any cobblestones blocks facing it.

2. `http://minecraft.net`

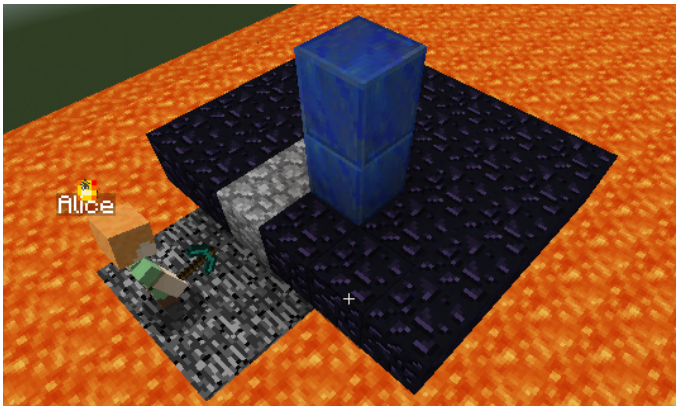

Figure 3. The test arena for the agent, from the point of view of a third-party observer. The agent (Alice) must step forward, remove the cobblestone (grey) with its pickaxe and then touch the blue lapis column. Falling into lava is instantly lethal and ends the episode. The walls (obsidian) and floor (bedrock) are indestructible and impassable.

As learning basic percepts from visual input has proven to be a challenge for complex game environments (de la Cruz et al., 2017; Lin et al., 2017), we used observations of the block identities within a rectangular volume centered on the agent. This environment encompassed a volume seven blocks wide, seven blocks long, and two blocks high, enabling the agent to observe the blocks constituting the floor and one block above. This let us focus on the core problem of advising reinforcement learning agents, although future work will instead start from pixels.

The test environment consisted of a small island surrounded by blocks of lava (Figure 3). If the agent falls into lava the episode terminates and it accrues a large negative reward. If the agent destroys the cobblestones and moves up to touch the blue lapis column, the episode ends and it earns a large positive reward. The floor and walls were made of bedrock and obsidian, and were impassable and indestructible. To incentivize exploration, a small negative reward was garnered with each step. The maximum duration for each run was set to ten seconds. We used the deep Q-learning network (DQN) algorithm (Mnih et al., 2013), implemented in the Keras-RL package (Plappert, 2016), modified to incorporate experience replay buffers for advice-derived training memories or experiences.

We manually frame advice in the form of simple templates (advice templates) that can be reified with elements from the agent's operating environment. The templates consist of a generic setup with corresponding scripts for generating sequences of actions and rewards. We then generate observations by running the scripted actions in a simplified recording environment to generate training memories. In the saliency-masked condition, we apply saliency masking to the observations. Lastly, they are inserted into the RL agent's replay buffer.

We examined several online Minecraft walkthroughs and playing guides to identify the types of basic advice used. From these, we selected and developed two advice templates, *Avoid Contact* and *Approach*, that we believed would be useful to the agent for playing in the Minecraft environment. We deliberately avoided coding advice for removing obstacles (e.g., cobblestone blocks) with the

Table 2. Generic advice templates.

Advice Template	Setting	Script
Avoid Contact(x)	Agent is near x.	Agent moves to contact x.
Approach(x, d)	x is within d blocks of Agent.	Agent moves to contact x.

agent's pickaxe. We wanted to assess how well the agent could learn to integrate the best-generated episodes that contain no explicit information about obstacle removal with experience in an environment that requires obstacle removal.

Each template consists of its arguments, the setting which describes how specific blocks and the agent are situated, and a script of actions to be performed, as shown in Table 2. For our scenario, we reified our advice templates as follows:

- **Avoid Lava**: Avoid contact with lava blocks, with contact earning a negative reward (-100).
- **Approach Lapis Column**: Approach and touch the lapis column. Contact earned a positive reward (+100), with incrementally increasing reward for moving toward the goal.

We collected observations for advice memories by having agents execute scripted actions in a recording environment. This was a simple flat plane with a floor composed of bedrock, and unlike typical RL, this environment was different and separate from the test environment.

Saliency masking can be considered a form of background subtraction, where portions of an observation deemed irrelevant to performing a task are removed. For example, an image-based car make and model classifier can simplify its learning problem by using background subtraction to identify which pixels are part of the background (non-vehicle). Setting these background elements to zero effectively removes them and lets the learner focus solely on vehicles. For our observation model, we represented non-salient blocks in the recording environment with a special "background" block that was filtered out when processing scripted observations. For our approach, we preserved only the objects used to reify the templates and "masked" all the other world elements in the observation (by converting these elements to zeros). Figure 4 illustrates our approach for the "Avoid Lava" advice. Here, saliency masking has effectively removed everything from the observation other than the lava block.

5. Experimental Studies of Learning from Advice

We now outline the training and testing protocol. For each training step, shown in Figure 5, the agent selects an action to take, randomly drawn from a Boltzmann distribution (Sutton & Barto, 1998). This distribution derives the probability of an action a given the current state s from the current Q-value estimates, $Q(s, a)$. In the case of DQN, a Q-value neural network is trained to approximate the Q-values.

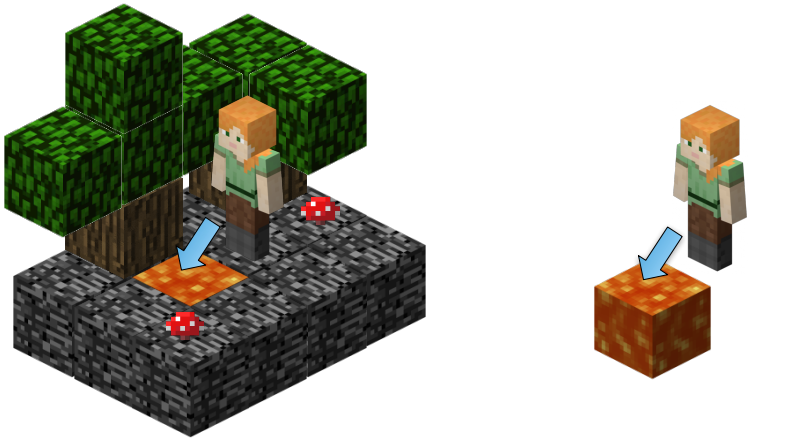

Figure 4. Saliency masking for generated observations: When collecting observations for "Avoid Lava" advice, we can either use full observations taken in the recording environment when executing the training script (left) or apply saliency masking (right). Standard observations incorporate not just elements such as the lava, the key point of this advice, but also extraneous blocks such as the bedrock floor and air. Saliency masking removes all nonessential blocks from the observation, leaving just what is needed to convey the lesson.

$$\Pr(a|s) = \frac{e^{Q(s,a)}}{\Sigma_{a'} e^{Q(s,a')}} \tag{2}$$

A transition tuple is collected and stored in the experience replay buffer. A tuple is then sampled from this buffer and used to update the Q-value network's parameters.

When advice-derived memories are used, we first convert human-provided advice into advice memories and insert the corresponding tuples into an advice replay buffer. At each step, two tuples are sampled: one from the experience replay buffer, the other from the advice buffer. Both are used to update the network parameters. The testing procedure is illustrated in Figure 6. Twenty trials were run for each experimental condition. Each trial consisted of 1000 training steps. At every 100 steps of training, performance was assessed with a test run, for a total of 10 test runs per trial. For each test run, the agent at that stage of training was evaluated by using a greedy action-selection policy, which selects the action with the maximal Q-value. The metric of test performance was whether the agent reached the goal (success) or not (failure). Test outcomes were combined to assess the probability of completion after a given number of training steps.

We ran 20 trials (one trial equaling 1000 training steps) for each experimental condition, evaluating the agent at every 100 steps. The three conditions were a standard DQN agent (DQN), a DQN agent augmented with advice episodes (DQN+Advice), and a DQN agent augmented with saliency-masked advice episodes (DQN+Masked Advice). Because of the highly stochastic nature of reinforcement learning, we use bootstrapping (Efron & Tibshirani, 1993), with a sample size of 1000, to derive the mean probability and standard errors of the agent reaching the goal. Even with

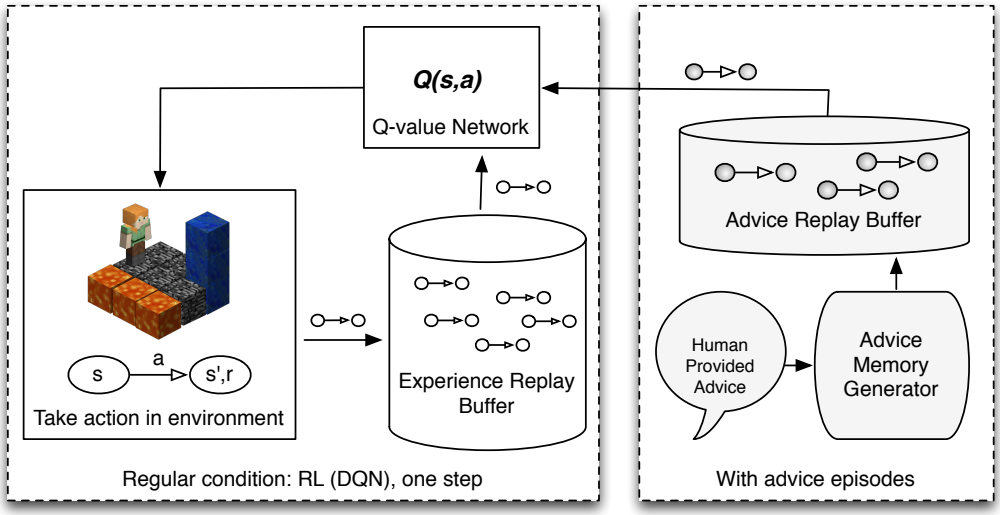

Figure 5. Order of events in a single training step: The agent randomly selects an action based on the current Q-value network estimates. An experience tuple $< s, a, s\prime, r >$ (start state s, action taken a, resulting state $s\prime$, and reward r) is stored into the experience replay buffer. A tuple is sampled from the buffer and used to update the Q-value network parameters. When advice memories are used, advice is converted into tuples, which are stored in an advice replay buffer. The Q-value network updates draw from tuples sampled from both the experience buffer and the advice buffer.

20 separate trials, there is a non-trivial amount of stochasticity in the learning, as evidenced by the fluctuations in goal probability.

Figure 7 shows the experimental results, comparing the mean probability of reaching the goal against total training steps on our three conditions. To assess our first hypothesis (*templates based on human advice can be used to generate training episodes that enhance learning of a model-free RL agent*) we compared a standard DQN agent (DQN) to agents augmented with the advice memories (DQN+Advice; Figure 7, left) and saliency-masked advice memories (DQN+Masked Advice; Figure 7, center). Compared to the standard DQN, we found that agents augmented with advice memories ($\chi^2(1) = 7.20$, p = .007) and agents augmented with saliency-masked advice memories ($\chi^2(1) = 9.04, p = .003$) achieved a higher overall probability of goal completion by the end of the 1000 steps.

To address our second hypothesis (*masking the environment so that only the most salient information is available will enhance the RL agent's ability to learn from the templates*), we examined the effect of saliency masking on the advice memories by comparing performance on the advice condition (DQN+Advice) and the saliency-masked advice condition (DQN+Masked Advice). We found that saliency-masked advice did not materially improve performance compared to an agent that just used advice memories, evidenced by a lack of performance difference at the final test,$\chi^2(1) = 0.10$, p = .752. However, as seen in the right panel of Figure 7, the saliency-masked advice condition did yield a better mean probability of reaching the goal at most of the evaluated training steps, particu-

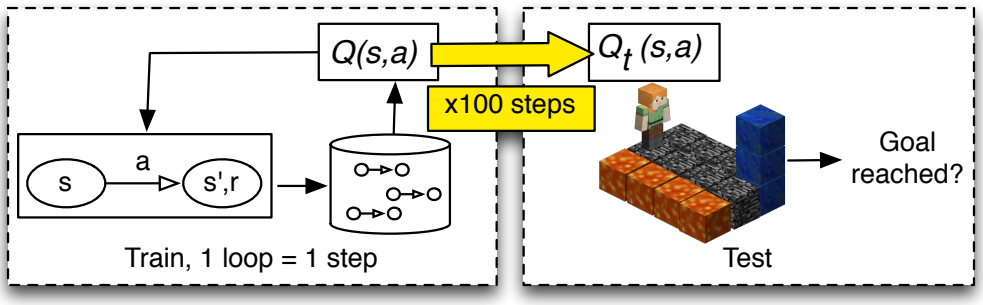

Figure 6. Test protocol for a single trial: After 100 steps of training, the agent uses its current Q-value network, $Q_t(s, a)$, to execute a test in the environment. Actions are selected by a greedy policy intended to maximize reward, unlike the exploration-focused stochastic policy used in training. When the test concludes, we record whether it was able to reach the goal.

larly in the early stages, such as at 300 steps of training, $\chi^2(1) = 5.00, p = 0.025$. It also exhibited less error overlap with the DQN condition than that of the unmasked advice condition.

These results show that advice memories, instantiated as synthetic training examples, can improve the performance of a baseline DQN agent. However, they also entail a notable alternate conclusion; use of advice memories did not harm overall performance. Because advice memories were generated in an "artificial" recording environment, their corresponding observations are unlikely to be distributionally similar to those obtained in the trial environment. Thus, this mismatch does not guarantee that the advice memories will positively impact the learning rate, as they even have the potential to harm the in-environment learning. The additional improvement in performance provided by saliency masking, assessed by comparing saliency-masked advice (DQN+Masked Advice) to unmasked advice (DQN+Advice), was not robust but did trend towards being beneficial, particularly during earlier trials. This indicates that removal of irrelevant observational elements may help with reducing the impact of the discrepancies between observations, but this benefit may be dependent on the stage of training.

We also noticed a conspicuous drop in performance for both the advice conditions (DQN+Advice, DQN+Masked Advice), at around 750 steps of training. One possible explanation is after a certain amount of in-environment experiences are accumulated, the advice memories began to hamper the learning. Our naive memory sampling regime sampled equally from advice and environmentally collected memories throughout training. This may strategy may not be optimal because as training progresses the model will become more tuned to the environment. In contrast, advice memories are static and do not change throughout training. At that point, updating the model with an equal proportion of unrealistic advice memories may confound the learning, which in turn may cause the agent to execute a different and less optimal behavior. This points to future investigations into how advice memories should be used, such as when and how much they should be incorporated during training. There is also the possibility that certain advice memories would be more valuable (or less harmful) at certain points of learning, and a more nuanced sampling strategy may be fruitful.

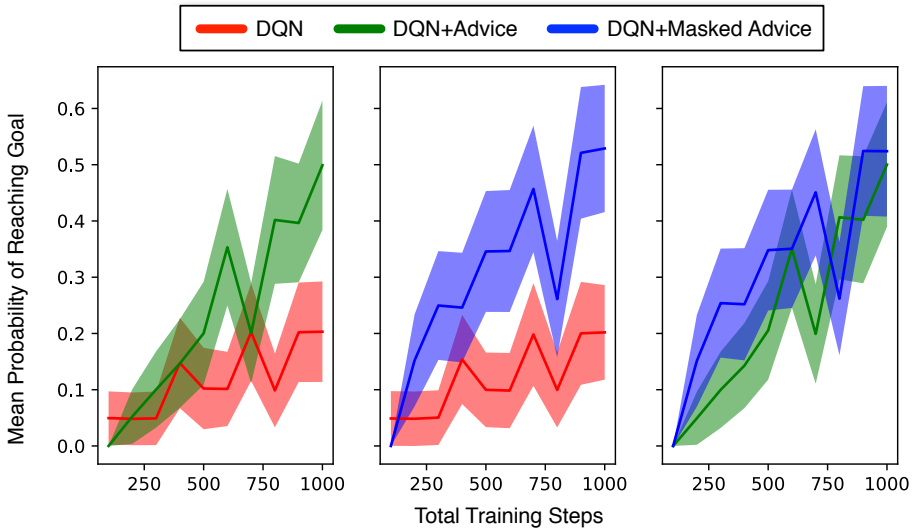

Figure 7. The mean probability of reaching the goal as a function of the number of training steps. Conditions compared are standard DQN vs. DQN with advice memories (left), DQN vs. DQNs with saliency masked advice memories (center), and DQNs with advice memories vs. those with saliency masked advice memories (right). Shaded regions correspond to one standard error of the mean (the estimated probability).

6. Related Work

Our approach demonstrates the ability for declarative advice to be transformed into a representation that can guide the learning of an autonomous reactive learning agent. In this section, we outline related research in understanding how humans and artificial intelligence (from cognitive systems to machine learning and connectionist networks) handle advice for enhanced learning and performance.

6.1 Advice in Artificial Intelligence

The idea of computer systems improving their performance through advice was first introduced by McCarthy (1959) in his seminal paper on Programs with Common Sense. He proposed a hypothetical program, the Advice Taker, that "automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows." The idea is for an advice giver to be able to improve an advice taker's performance by making statements about the environment and what is required, without the need for intimate knowledge of the advice-taker's internal representation and reasoning mechanisms. McCarthy laid out the representational requirements for the reasoning over situations and actions that would support such interaction.

Subsequent work further explored the idea of advice in symbolic reasoning systems. FOO (Mostow, 1983) formulated learning as the problem of *operationalizing* task advice by transforming it into executable procedures for accomplishing a task. FOO converted the advice by applying general transformation rules to perform inference over domain-specific knowledge. Golding et al.

1987 extended Soar to be able to use advice to set goals, relying on its natural chunking mechanism to learn how to use the advice. The Advisable Planner (Myers, 1996) introduced the concept of advice idioms to capture different classes of advice and developed translation strategies to convert (pseudo-)natural language advice into constraints to guide the plan-construction process.

RATLE (Maclin et al., 2005) was one of the first reinforcement learning systems to incorporate advice. Users provided advice in the form of simple if-then rules that were then converted into a knowledge-based neural network (Towell & Shavlik, 1994). Kuhlmann et al. (2004) took advantage of the restricted nature of the target task (RoboCup soccer) to successfully translate natural language advice into a formal semantic representation directly usable by the agent. In Toro Icarte et al. (2018), advice is represented as linear temporal logic formulas, which are converted into nondeterministic finite-state automata that are used to drive the exploration of RL agents. Krening et al. (2017) look at object-focused advice, which is essentially translated into policies whose recommendations are balanced against those based on learned object-oriented policies.

Like all this previous work, our use of templates is motivated by the need to operationalize user guidance into a form that can be used by the agent. However, rather than convert advice directly into rules or constraints that directly influence the system's behavior, our current approach converts it into training examples for the learning system. An advantage of this approach is that it is agnostic to the underlying learning system, enabling its use in any system that learns from training examples.

6.2 Human Guidance in Reinforcement Learning

Researchers in reinforcement learning have also explored other forms of human guidance to help alleviate the computational demands of reinforcement learning. One major form of human input for RL systems has been demonstration. Here, a human instructor performs the intended activity, which the RL agent then attempts to learn to perform itself. Notable examples of the work in this area include imitation learning (Schaal, 1999), inverse reinforcement learning (Ng & Russell, 2000), and apprenticeship learning for reinforcement learning (Abbeel & Ng, 2004). All take human demonstrations as input but differ in their learning objectives and in their learned models. Imitation learning attempts to learn a policy to replicate the human's policy through demonstrations and feedback; inverse reinforcement learning tries to learn a reward function that will cause the agent to behave similarly; and RL apprenticeship learning builds on inverse reinforcement learning to learn a policy from the induced reward function. In settings where demonstrations may be costly or otherwise impractical, our use of synthetic trajectories can replace or augment actual demonstrations.

Another large body of work, sometimes referred to as human-centered reinforcement learning or human-in-the-loop reinforcement learning, looks at various ways of incorporating human feedback into the training. *Reward shaping* involves using human feedback to modify the rewards an agent receives. For example, in TAMER (Knox & Stone, 2009), the human trainer provides simple scalar rewards for the agent's actions, and the agent learns to choose the action with the highest predicted reward. *Policy shaping* instead translates human feedback into direct policy feedback; the human-provided rewards over the agent's actions are used to infer the optimal policy rather than the reward function (Griffith et al., 2013). MacGlashan et al. (2017) interpret human feedback in a policy-dependent manner by passing human feedback through the *advantage function*, which estimates the value of an action compared to the current policy. In these approaches, the human's role is that of

a trainer, and thus frequent interaction is required. In contrast, we are interested in settings where human guidance is intended more as general advice.

Similar to TAMER's use of human feedback to label actions, the Action Advisor approach involves asking users to identify which actions are applicable in a state (Lin et al., 2017). These labels serves as input to an Action Advisor; an arbiter then decides between the recommendations from the Advisor and those from the learned policy. Another approach involves human labelers indicating their preferences between pairs of trajectory segments (Christiano et al., 2017). While user feedback over specific aspects of the training enable a reinforcement learner to achieve a higher level of performance with fewer training examples, it still requires a relatively large amount of human involvement, and for the user to examine individual instances instead of providing feedback at a higher level of abstraction.

Use of simulations for training RL agents was used in work such as Abel et al. (2016). As with advice memories, these simulations were approximations of the target environment. In contrast with our work, agents conducted trials in this environment, and did not execute actions corresponding to advice or desired behavior. However, it does demonstrate that agents can gain some benefit from learning over unrealistic data.

6.3 Visual Attention and Mental Imagery

The machine learning community has a long and continuing interest in developing mechanisms to guide learning *relevant* information, independent of whether it is a reinforcement learning agent or an image-classification network. It is critical that neither an RL agent develops a strong affinity for environmental rewards that prevent successful completion of a more complex goal, nor that an image-classification system fixates on spurious signals found only in a training set. Toward resolving these issues, related work explores selectively choosing which portions of an observation to use, in the form of attention mechanisms (Xu et al., 2015), where systems learn how to weight portions of their input. Along these same lines, others have used visual saliency mapping (Itti et al., 1998), where the observed area is given in a fixed form and not framed as a probability distribution. In contrast with other work, while our approach also employs selective input removal, we have the instructional advice framework perform this removal instead of having the agent attempt to learn it.

Perhaps the work most related to our approach are imagery (Wintermute, 2010) and imagined trajectories (Weber et al., 2017). Similar to our use of templates, both of these approaches use knowledge of the environment and its dynamics to synthesize projected outcomes given an initial state description. However, both of these methods use these projections to affect action selection. Imagery extrapolates possible consequences for taking an action from a given state by simulating projected outcomes with a domain-reasoning component. Imagined trajectories uses a similar idea, except a neural model is used to generate possible outcomes and integrate them in the policy function. Instead of governing action selection, we use domain models to synthesize training episodes. While this approach may not have as immediate an effect as directly governing agent policy, it requires fewer changes to existing RL algorithms. It also lets us model just the knowledge needed to convey the advice instead of the larger set necessary to generate projected outcomes.

7. Conclusions and Future Work

Motivated by research in skill acquisition and expertise from cognitive psychology, we demonstrated how to apply declarative, human-like advice to enhance the performance of a reinforcement learning agent. We presented a theoretical framework for how advice cannot directly interact with a reactive learning system's encapsulated knowledge representation, but can shape it through synthetic training examples. To implement this framework, we matched advice to templates that generated "imagined" training examples. These examples were scripted sequences of actions and observations, with saliency masking to focus attention on the most relevant aspects of the experience. This architectural allows for minimal *a priori* world modeling to guide a simple RL agent. Experimental results in a Minecraft-based test environment showed how these synthetic experiences can improve an RL agent's performance, achieving both faster learning and a higher success rate.

The initial experimental results reported here are promising; however, there remain several avenues for future work. Our approach relies on an experience replay buffer to store the advice-based memories, making it independent of the specific RL algorithm used. Thus, although we relied on basic Q-learning, our approach can be applied to systems with more advanced RL architectures that accommodate temporal abstractions and longer-range goals such as hierarchical RL (Kulkarni et al., 2016), and option critic architectures (Bacon et al., 2017). The more general concept of using advice templates to transform human advice into training examples applies to any learning system. For example, this approach may be useful for learning subsymbolic components of comprehensive architectures that span multiple levels of representation, such as training operator selection policies in SOAR-RL (Nason & Laird, 2005), or inference policies (Asgharbeygi et al., 2005).

As with other DQN approaches using experience replay, our approach randomly samples from the replay buffer. However, studies in episodic memory suggest that humans selectively retrieve memories, choosing the ones most pertinent to a given situation and using these for learning (Gershman & Daw, 2017). One line of future work is to implement this form of case-based retrieval as a form of a specialized situation-aware critic, and evaluate its effect on learning. Currently, we have templates to reinforce or to punish a behavior, but not both. Clearly humans will sometimes give advice that is more nuanced, e.g., *Do not brake suddenly unless you are about to hit something*. Thus, another avenue for future work is to develop mechanisms for handling such advice.

The work in this paper grew out of our work on a framework for explainable autonomy (Gervasio et al., 2018). Given system explanations that surface problems in the agent's knowledge, the natural next step is for humans to correct that knowledge and hence, this effort. We note that a duality exists between explanation and advice: good explanations often act as good advice. Thus, another important direction for future work is the use of explanation to elicit more effective advice and we are exploring the use of introspection mechanisms for this purpose.

Acknowledgements

This work was supported by SRI International's Internal Research and Development. The authors would also like to thank Amin Atrash, Boone Adkins, and Rodrigo de Salvo Braz for stimulating discussions and the anonymous reviewers for their insightful comments.

References

Abbeel, P., & Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. *Proceedings of the Twenty-First International Conference on Machine Learning*. Louisville, KY: ACM.

Abel, D., Salvatier, J., Stuhlmüller, A., & Evans, O. (2016). Agent-agnostic human-in-the-loop reinforcement learning. *Proceedings of the NIPS 2016 Workshop on the Future of Interactive Learning Machines*. Barcelona, Spain: Curran Associates.

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review*, *89*, 369.

Asgharbeygi, N., Nejati, N., Langley, P., & Arai, S. (2005). Guiding inference through relational reinforcement learning. *Proceedings of the Fifteenth International Conference on Inductive Logic Programming* (pp. 20–37). Bonn, Germany: Springer.

Bacon, P., Harb, J., & Precup, D. (2017). The option-critic architecture. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 1726–1734). San Francisco, CA: AAAI Press.

Chaffin, R., Lisboa, T., Logan, T., & Begosh, K. T. (2010). Preparing for memorized cello performance: The role of performance cues. *Psychology of Music*, *38*, 3–30.

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. *Proceedings of the Thirty-First Annual Conference on Advances in Neural Information Processing Systems* (pp. 4299–4307). Long Beach, CA: Curran Associates.

de la Cruz, G. V., Du, Y., & Taylor, M. E. (2017). Pre-training neural networks with human demonstrations for deep reinforcement learning. *CoRR*, *abs/1709.04083*.

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Boca Raton, FL: Chapman & Hall.

Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, *100*, 363–406.

Eriksen, C. W., & St. James, J. D. (1986). Visual attention within and around the field of focal attention: A zoom lens model. *Perception & Psychophysics*, *40*, 225–240.

Eriksen, C. W., & Yeh, Y.-Y. (1985). Allocation of attention in the visual field. *Journal of Experimental Psychology: Human Perception and Performance*, *11*, 583–597.

Fitts, P. M., & Posner, M. I. (1967). *Human performance*. Oxford, England: Brooks/Cole.

Flegal, K. E., & Anderson, M. C. (2008). Overthinking skilled motor performance: Or why those who teach can't do. *Psychonomic Bulletin & Review*, *15*, 927–932.

Gershman, S. J., & Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annual Review of Psychology*, *68*, 101–128.

Gervasio, M., Myers, K., Yeh, E., & Adkins, B. (2018). Explanation to avert surprise. *Proceedings of the Explainable Smart Systems Workshop at the Twenty-Third International Conference on Intelligent User Interfaces*. Tokyo, Japan: ACM.

Golding, A., Rosenbloom, P. S., & Laird, J. E. (1987). Learning general search control from outside guidance. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence* (pp. 334–337). Milan, Italy: Morgan Kaufmann.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., & Thomaz, A. L. (2013). Policy shaping: Integrating human feedback with reinforcement learning. *Proceedings of the Twenty-Seventh Annual Conference on Advances in Neural Information Processing Systems* (pp. 2625–2633). Lake Tahoe, NV: Curran Associates.

Hagemann, N., Strauss, B., & Cañal-Bruland, R. (2006). Training perceptual skill by orienting visual attention. *Journal of Sport and Exercise Psychology*, *28*, 143–158.

Henke, K. (2010). A model for memory systems based on processing modes rather than consciousness. *Nature Reviews Neuroscience*, *11*, 523.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 1254–1259.

Jiménez, L., & Méndez, C. (2001). Implicit sequence learning with competing explicit cues. *The Quarterly Journal of Experimental Psychology Section A*, *54*, 345–369.

Johnson, M., Hofmann, K., Hutton, T., & Bignell, D. (2016). The Malmo platform for artificial intelligence experimentation. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (pp. 4246–4247). New York: AAAI Press.

Kaushik, S., Scholz, J., Isbell, C. L., & Thomaz., A. L. (2016). Efficient exploration in Monte Carlo tree search using human action abstractions. *Proceedings of the Future of Interactive Learning Machines Workshop at NIPS 2016*. Barcelona, Spain: Curran Associates.

Knox, W. B., & Stone, P. (2009). Interactively shaping agents via human reinforcement: The TAMER framework. *Proceedings of the Fifth International Conference on Knowledge Capture* (pp. 9–16). Redondo Beach, CA: ACM.

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

Kuhlmann, G., Stone, P., Mooney, R., & Shavlik, J. (2004). Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer. *Proceedings of the AAAI-04 Workshop on Supervisory Control of Learning and Adaptive Systems*. San Jose, CA: AAAI Press.

Kulkarni, T. D., Narasimhan, K., Saeedi, A., & Tenenbaum, J. B. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Proceedings of the Thirtieth Annual Conference on Advances in Neural Information Processing Systems* (pp. 3675–3683). Barcelona, Spain: Curran Associates.

Kumaran, D., Hassabis, D., & McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in Cognitive Sciences*, *20*, 512–534.

Lin, L. J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, *8*, 293–321.

Lin, Z., Harrison, B., Keech, A., & Riedl, M. O. (2017). Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3D worlds. *CoRR*, *abs/1709.03969*.

MacGlashan, J., Ho, M. K., Loftin, R. T., Peng, B., Roberts, D. L., Taylor, M. E., & Littman, M. L. (2017). Interactive learning from policy-dependent human feedback. *Proceedings of the Thirty-Fourth International Conference on Machine Learning*. Sydney, Australia: PMLR.

Maclin, R., Shavlik, J. W., Torrey, L., Walker, T., & Wild, E. W. (2005). Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. *Proceedings of the Twentieth National Conference on Artificial Intelligence* (pp. 819–824). Pittsburgh, PA: AAAI Press.

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

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*, 419.

Milner, B., Squire, L. R., & Kandel, E. R. (1998). Cognitive neuroscience and the study of memory. *Neuron*, *20*, 445–468.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. A. (2013). Playing Atari with deep reinforcement learning. *CoRR*, *abs/1312.5602*.

Mostow, D. J. (1983). Machine transformation of advice into a heuristic search procedure. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning*, 367–403. Los Altos, CA: Morgan Kaufmann.

Myers, K. L. (1996). Advisable planning systems. In A. Tate (Ed.), *Advanced planning technology*. Menlo Park, CA: AAAI Press.

Nason, S., & Laird, J. E. (2005). Soar-RL: Integrating reinforcement learning with Soar. *Cognitive Systems Research*, *6*, 51–59.

Ng, A. Y., & Russell, S. (2000). Algorithms for inverse reinforcement learning. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 663–670). Stanford, CA: Morgan Kaufmann.

Petersen, S. E., Van Mier, H., Fiez, J. A., & Raichle, M. E. (1998). The effects of practice on the functional anatomy of task performance. *Proceedings of the National Academy of Sciences*, *95*, 853–860.

Plappert, M. (2016). keras-rl [Source code]. Retrieved from `https://github.com/keras-rl/keras-rl`.

Posner, M., Snyder, C., & Davidson, B. (1980). Attention and the detection of signals. *Journal of Experimental Psychology*, *109*, 160–74.

Reber, P. J., & Squire, L. R. (1998). Encapsulation of implicit and explicit memory in sequence learning. *Journal of Cognitive Neuroscience*, *10*, 248–263.

Rozanov, S., Keren, O., & Karni, A. (2010). The specificity of memory for a highly trained finger movement sequence: Change the ending, change all. *Brain Research*, *1331*, 80–87.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, *3*, 233–242.

Sherry, D. F., & Schacter, D. L. (1987). The evolution of multiple memory systems. *Psychological Review*, *94*, 439–454.

Silver, D., et al. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*, 354–359.

Squire, L. R. (2004). Memory systems of the brain: A brief history and current perspective. *Neurobiology of Learning and Memory*, *82*, 171–177.

Sun, R., Merrill, E., & Peterson, T. (2001). From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*, *25*, 203–244.

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning*. Cambridge, MA: MIT Press.

Taylor, J. A., & Ivry, R. B. (2012). The role of strategies in motor learning. *Annals of the New York Academy of Sciences*, *1251*, 1–12.

Toro Icarte, R., Klassen, T. Q., Valenzano, R., & McIlraith, S. A. (2018). Advice-based exploration in model-based reinforcement learning. *Proceedings of the Thirty-First Canadian Conference in Artificial Intelligence* (pp. 72–83). Toronto, ON, Canada: Springer.

Towell, G. G., & Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, *70*, 119–165.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*, 279–292.

Weber, T., et al. (2017). Imagination-augmented agents for deep reinforcement learning. *CoRR*, *abs/1707.06203*.

Weinberg, R. (2008). Does imagery work? Effects on performance and mental skills. *Journal of Imagery Research in Sport and Physical Activity*, *3*, 1–21.

Wintermute, S. (2010). Using imagery to simplify perceptual abstraction in reinforcement learning agents. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (pp. 1567–1573). Atlanta, GA: AAAI Press.

Wulf, G., Shea, C., & Lewthwaite, R. (2010). Motor skill learning and performance: A review of influential factors. *Medical Education*, *44*, 75–84.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *Proceedings of the Thirty-Second International Conference on Machine Learning* (pp. 2048–2057). Lille, France: International Machine Learning Society.