From Fly Detectors to Action Control: Representations in Reinforcement Learning

Anna-Mari Rusanen, Otto Lappi, Jami Pekkanen, and Jesse Kuokkanen*

According to radical enactivists, cognitive sciences should abandon the representational framework. Perceptuomotor cognition and action control are often provided as paradigmatic examples of nonrepresentational cognitive phenomena. In this article, we illustrate how motor and action control are studied in research that uses reinforcement learning algorithms. Crucially, this approach can be given a representational interpretation. Hence, reinforcement learning provides a way to explicate action-oriented views of cognitive systems in a representational way.

1. Introduction. According to "radical enactivists," the cognitive sciences should abandon the representational framework, for several reasons. For example, enactivists claim that there is no satisfactory, naturalistic account of content at the level of basic cognition. Hence, representationalism faces the "Hard Problem of Content" (Hutto and Myin 2013, 2017). Thus, the cognitive sciences should give up the notion of neurocognitive representations. In addition, enactivists argue, there is no account of how contentful representational states drive action. Again, the conclusion is that cognitive scientists should let go of the assumption of representations (Hutto and Myin 2020). Instead, according to enactivists, action control should be explained without appealing to representations. As Myin and Hutto write, "acts of perceptual, motor, or perceptuomotor cognition—chasing and grasping a swirling leaf—are directed towards worldly objects and states of affairs, or aspects thereof, yet *without representing* them" (2015, 62, italics added).

In this article, however, we question these claims by looking at the contemporary algorithmic research on action control. In what follows, we focus

*To contact the authors, please write to: Anna-Mari Rusanen, PO Box 24, University of Helsinki, 00014 Finland; e-mail: anna-mari.rusanen@helsinki.fi.

Philosophy of Science, 88 (December 2021) pp. 1045–1054. 0031-8248/2021/8805-0025$10.00

https://doi.org/10.1086/715513 Published online by Cambridge University Press

especially on reinforcement learning (RL) algorithms. They are widely used to study various aspects of action and motor control in computational neurosciences, in artificial intelligence (AI), and in robotics.

In RL, agents take actions in an environment in order to maximize cumulative reward. Action control is understood as choosing the right action selection policy for a given environment, so as to maximize future reward. This formulation of the computational problem makes RL-based action control models cognitively more sophisticated than many other models, such as simple proportional feedback control models based on control theory from the 1960s.

Moreover, as the case of action planner systems in RL illustrates, action control can be given a representational interpretation in RL. Thus, RL provides a well-understood, algorithmic way to describe how the manipulation of representations makes a difference to the systems that guide and drive behavior. It provides a way to explicate "action-oriented views" of cognitive systems in a way that is overlooked by recent enactivists (and many other antirepresentationalists).

2. Reinforcement Learning Algorithms. In a nutshell, RL can be described as learning by interacting with an environment. An RL agent learns by trial and error, observing the consequences of its actions, rather than from being explicitly taught what to do. The agent selects its actions on the basis of its past experiences and also by exploring new choices.

Historically, the basic idea of RL—learning as trial and error—was developed by early behaviorists. Thorndike's (1911) "Law of Effect" described how *reinforcing* events (i.e., reward and punishment) affect the tendency to select actions and, hence, how they affect learning. Computer scientists combined this framework with the formalisms of optimal control theory, temporal difference learning, and learning automata and gave a precise formulation in the 1960s and 1970s.[1]

Nowadays, the variants of this algorithmic approach are used in a wide range of applications in AI and robotics. They are used to study various forms of skilled action and motor control in cognitive and computational neurosciences. RL algorithms are also deployed, for example, in learning, decision making, and strategic reasoning tasks, and they have been applied to study attention, procedural memory (for model-free policies or action values), semantic declarative memory (for world maps or models), and episodic memory.[2]

3. The Core Concepts of RL. RL describes how an *agent* learns to interact with the *environment* in a rational way. The algorithms should maximize

[1] For the history of RL algorithms, see Sutton and Barto (2018).

[2] For an overview on RL in cognitive neurosciences, see Niv (2009).

https://doi.org/10.1086/715513 Published online by Cambridge University Press

the cumulative *reward* over time by observing the consequences of the *actions*.[3] In RL, an agent learns from experience to choose actions that lead to greater rewards over time.

When using RL as a theory of brain function, the basic idea is that neural activity reflects a set of operations, which together constitute computations that are specified in the RL framework. One of the key theoretical insights in RL is the way of describing how brains, as computational systems, can learn what to do (see fig. 1). In RL, a (technical) environment is a temporal succession of *states* s_t from a set of environment states S. At each point in time, the environment is in exactly one state. A state encodes "the world" into a number of variables whose values determine the state.[4] Each state has a fixed reward that is observable (in a technical sense) for the agent. Reward is not a complex or multidimensional feature but a simple scalar, which can be negative (punishment) or positive (reward). The agent can act in the environment, performing individual actions from a set of actions A. An action a_t in a state s_t will take the environment at the next time step to a new state $s_{t+}\sigma_t$ according to a *state transition function*.[5] It is part of the world and generally not known to the agent.

At each *time step* t the agent is in a state s_t, where it is possible to choose an action a_t.[6] The agent then receives some amount of reward r_t at a probability $P(r\,|\,s)$. The reward function R(s) is not known to the agent and is considered to be produced by the (technical) environment. This prevents the agent from updating its own reward function—otherwise the agent could trivially maximize the reward by treating whatever happens as maximally rewarding.[7]

However, note that anatomically the reward signal is often generated within the *organism*. Thus, it typically is organism dependent. Moreover, what is rewarding for a particular agent is not a property of the physical world but a property pertaining to the agent. Different agents will have different reward functions even when the physical (or technical) environment is the same.

[3] We present a simplified account of RL in which rewards are tied to states. More generally, the reward is usually associated with state-action *pairs* or state action–next state *tuples*, but this distinction is not relevant for the discussion at hand. For an overview, see Sutton and Barto (2018). See also Dayan and Niv (2008).

[4] For example, shock administered = true, shock administered = false.

[5] The action will result in a some new world state s' according to some state transition probability $P(s'\,|\,s, a)$.

[6] In RL, the Markov assumption holds: these probabilities do not depend on prior history, only on s and a; s can be made to contain information of past history up to some horizon.

[7] This technical solution also allows the reward function to differ between organisms and also to be dependent on the organism's state.

https://doi.org/10.1086/715513 Published online by Cambridge University Press

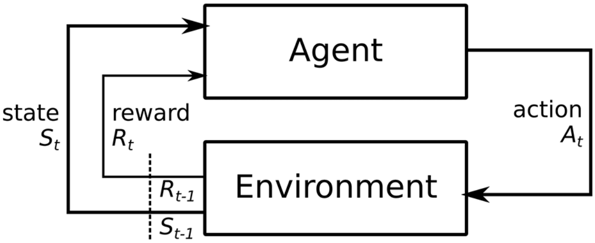

Figure 1. Structure of a reinforcement learning algorithm.

In RL, the agent's task is generally to learn to estimate and maximize the long-term cumulative reward. This means producing an estimation of the *value* of a state and choosing actions that lead to maximally valuable states. The concept of value stands for these *cumulative expected long-term rewards* accruing from a state. Technically, the value V(s) of a state s is the expected temporally discounted sum of rewards—(r_t) observed at time t and future rewards that are *discounted* the further they are in the future.

4. Reinforcement Learning and Action Planning. In experimental work on motor control, actions—such as chasing and reaching a leaf—are typically seen as based on internal predictive or forward models of reaching dynamics (Wolpert, Ghahramani, and Jordan 1995; Miall and Wolpert 1996). Typically, the analyses describe the dynamics as progressing adjustments of internal models to fit with current observations. When the dynamics of action are approached in terms of RL (Doya 2008; Botvinick et al. 2015; Weinstein and Botvinick 2017),[8] the agent is thought to take an action (e.g., reaching a leaf) according to its action policy and then to update the policy after receiving a reward outcome in the form of a signal.[9]

When the goal of the agent changes, the appropriate action becomes different (Doya 2008). In this case, the agent must somehow find a way to handle the new goal. In RL, one possible solution is that the agent uses an *internal model* M to help to update the action policy (Doya 2008). This internal model consist of the learning of the so-called state transition rule P (new state | state, action; Doya 1999; Kawato 1999). If such a model is available, the agent can perform the following inference: if I take action a_t from current state t, what new state (t + 1) will I end up in? In addition, if the reward for

[8] While in many other computational approaches on motor processes the function of the forward model is to predict the sensory consequences of motor commands, in RL the function of the models is to maximize the long-term cumulative reward.

[9] This is known as the basic form of RL.

https://doi.org/10.1086/715513 Published online by Cambridge University Press

each state is also known, the agent can evaluate with this internal model also the “goodness” of any hypothetical action.

This approach assumes the existence of a *forward model* $\hat{M}$ of the environment. This forward model allows an agent to “plan” its actions. It helps to evaluate how the environment will evolve in response to different actions. By using this forward model, the system can select a sequence of actions that will take the agent from its current state to a desired goal state (e.g., which should maximize rewards accruing along the trajectory).[10] Technically, this planning procedure can be described as the maximization of the value up to a time $T : \Sigma Tt = 0rt + 1$, where t indexes discrete time steps up to some maximum T, and r_t is the reward received at each step (Weinstein and Botvinick 2017). Further, based on a particular policy, the system queries the model $\hat{M}$ with a series of state-action pairs (s_t, a_t) and in turn receives an estimated next state $(s_{t+}\sigma_t)$ and reward $(r_{t+}\sigma_t)$. After the planner completes querying $\hat{M}$, it returns an action a, which is executed in M. This results in a new state and a new reward, and the process starts over.

5. Representations in RL-Based Action Control. To characterize the cognitive dynamics of action control in this way is to characterize it in exact abstract and algorithmic terms. This approach makes no mention of the features of the actual environments in which the cognitive processes, or mechanisms, might be deployed. Instead, RL is an exact way to study the cognitive dynamics as forms of algorithmically specified reasoning and learning processes. It explains how cognitive systems control action by planning, selecting, and choosing different options.

In RL-based action control, the algorithms can be taken to operate—at least—with two types of representational states. First, estimations about the values of state-action pairs can be taken to represent the estimated “goodness” of the action in terms of cumulative long-term rewards. What they represent are neither the entities in the real world (say, hands grasping leaves) nor the future trajectories of real world entities (say, the possible future trajectories of hands grasping new leaves). Instead, they represent the goals of action in light of an action policy (e.g., the amount of a reward, if the agent grasps the leaf).

Second, when the algorithm estimates the goodness of future actions, the algorithm uses a forward model. This model can be taken as a representation of future states of the algorithm in a light of its action policy. As a representation, it refers to the estimated, possible future states of the algorithm’s world, not to the states of the real world environment.

[10] In practice, if the actions are discrete, tree search methods can be used to search over different action sequences and evaluate their quality using the forward model.

https://doi.org/10.1086/715513 Published online by Cambridge University Press

Namely, in RL the agent-environment construction is a part of the algorithm specification, and the environment is literally a “synthetic” model of the environment for the algorithm. It is specified in terms of the formalisms, not in terms of real world entities (only).[11] It, or the concept of agent, should not be confused with the notions of “real” environments (e.g., physical stimuli) or “real agents” (e.g., the organism).

Philosophically, these representational states resemble Egan’s (2014, 2020) cognitive states with computational contents. According to Egan’s (2014) distinction, some (computational) contents are about the formal descriptions of the tasks computed by cognitive systems, and some (“cognitive”) contents are about the environment. As Egan (2014, 2020) remarks, computational contents are domain general and environmentally neutral. They can be applied to a variety of different cognitive uses in different contexts, and they make no reference to external environment whatsoever (Egan 2014). Computational states can be assigned a “semantic” content in an appropriate, intentional “gloss.” According to Egan (2014, 2020), it is a pragmatically motivated way to describe the interaction between the organism and its environment as standing in for the objects or properties in the environment. The intentional gloss enables the analysis of cognitive systems to represent the elements of the environment.

In the case of RL, however, the intentional gloss is not addressed in terms of “properties of the external environment.” Namely, the (synthetic) environment is not just a (re)description of the external, real environment. Instead, it is a technical environment for the algorithm, reflecting the computational RL problem and the structure of the algorithm.

In many real world tasks, the sufficient correspondence between the synthetic environment and the real world environment can be crucial. For example, if the goal of a robot hand is, say, to pick a leaf in a real world environment, then, obviously, the leaf’s location, its size, or its configuration with the hand is relevant to the success of performance. To select appropriate policies, the system must take these (and other relevant) external factors into account.

Technically, the degree and the quality of correspondence depends on the details of the specific application, and they can be implemented in many ways. Not all of them are representational, or “contentful” in the radical enactivists’s sense. The real world environment may serve only as a source for feedback information. For example, the parameters of the system can be updated causally by using the feedback information. As Ramsey (2007) remarks, however, mere causal relations do not represent. Thus, the feedback information may play only a causal but not a representational role.

In some cases, systems may receive so-called observations (e.g., an image of the environment) as inputs, parametrize them, and transform them into

[11] Grush (2004) provides an alternative and interesting analysis in terms of emulators.

https://doi.org/10.1086/715513 Published online by Cambridge University Press

hidden states. The hidden states are then updated iteratively by a recurrent process that receives the previous hidden states and hypothetical next actions. At each step the model predicts the policy, value function, and immediate reward.

However, there is no requirement for the hidden states to "match" the states of the external environment or any other such constraints on the semantics of states (Sutton and Barto 2018). Instead, the hidden states may represent states in whatever way is relevant for predicting current and future values. That is, RL algorithms do not only use (current or past) "observations" (about the external environment) to estimate future rewards. They do not track the regularities of the external environments. Instead, they estimate what actions they should take to maximize the reward. Thus, they refer to the future development of (synthetic) environment M, not to the (development of) real world environment as such. Hence, if they stand in for something, they stand in for the entities and states in possible worlds.

6. From Fly Detectors to a Variety of Representations. Obviously, these representations do not fit well with the portrait of neurocognitive representations painted by recent radical enactivists. For example, in their recent work Hutto and Myin (2020) describe representational content as "the property that states of mind possess" (82). It "allows them to represent how things are with the world" (82). The states of the mind are connected with the world via "sensory contact," and the content of representational states is taken to "track" the external environment (Hutto 2015).

This view of representation continues the legacy of so-called fly detectors. In fly detectors, the notion of representation is specified in terms of a relation between the tokening of an internal, neurocognitive state and the external object or property the state represents. Historically, this view is inspired by the receptive field studies on sensory systems in the 1950s and 1960s (Hubel and Wiesel 1959; Lettvin et al. 1959). In Lettvin et al. (1959), the focus was on the signal transformation properties of frog ganglion cells, later known as "fly detectors." These cells were found to respond to small, black, fly-like dots moving in the frog's visual field. Hubel and Wiesel (1962) proposed a way in which "pooling mechanisms" might explain the response properties of these cells in the mammalian primary visual cortex.

This framework affected deeply the neuroscientific and psychological research on sensory processes. They were studied as a bottom-up feature detection for decades. Fly detectors also began to dominate the philosophical intuitions on representations. A great deal of effort was expended in the 1980s to answer the questions of (i) whether a representation of a fly is really about flies, (ii) how to make the leap from the physical signal transformation properties of ganglion cells into semantic properties of fly detectors, or

https://doi.org/10.1086/715513 Published online by Cambridge University Press

(iii) how to specify the content determination of these representations in a satisfactory naturalistic way (Dretske 1981; Millikan 1989; Fodor 1992).

In fly detectors, the activation of representations requires a causal association with preceding stimuli, a ("neural") signal and subsequent behavior, or an activation of a stimulus causing some indicator to fire. Typically, the stimulus is taken as a proximal cause for the activation of the representational state. This requires that a source for the stimulus (e.g., a signal that causes the stimulus) exists somehow in the physical environment. Or, depending on the account, the source can also be taken as a cause that is responsible, for example, for the firing of an indicator.

Obviously, the representations in RL-based action control systems are not specified in such terms. In RL, value refers to a mathematically specified amount of long-term cumulative expected rewards. That is what value representations stand in for. These representations are not "triggered" by the occurrence of a value stimulus or a "value" signal from the (real) external environment. Or, the rewards are not based on what stimulus features of the world neural signals are responding to. Rewards are not "out there," and there are no "reward signals" causing reward stimuli to activate the "reward detectors" or any other mechanisms analogous to how detectors (or "indicators" in teleo- or indicator semantics) have been envisaged in the recent enactivist or classical neurosemantic literature.

Of course, these representations raise very difficult problems of "neural encoding" of such future-oriented, abstract, and organism-dependent entities. They will challenge the intuition that all cognitive states represent as fly detectors or that, generally, sensory representations do. However, this puzzle is not a question answerable to intuition. Instead, it is answerable to the roles that representations play in explaining the action control scientifically.

From a neuro- and cognitive scientific point of view, not all representations are sensory. Instead, there appear to be a variety of representational states. For example, while some sensory states (such as auditory signals) are more directly about external environmental target systems, other representations (such as complex action control representations) may not be. Thus, perhaps we should let go of the assumption that only states that track external environments count as representational and abandon a too-narrow fly-detector-based construal of representations.

7. Conclusion. RL algorithms are used to study the same phenomena (e.g., motor and action control) that are celebrated by enactivists (and many other antirepresentationalists) as paradigmatic examples of nonrepresentational phenomena. And still, as the case of RL illustrates, motor and action control can be given a representational interpretation.

The computational models based on RL do not only use possibly the most powerful algorithms that we have in AI, but they are widely and successfully

https://doi.org/10.1086/715513 Published online by Cambridge University Press

used in many areas of neurocognitive sciences to study biological organisms. One cannot simply ignore this computational and theoretical framework, when assessing the research of action control in current neurocognitive sciences.[12]

Moreover, RL algorithms are theoretically and mathematically well understood. Hence, this framework provides an exact, formal way to analyze, in detail, how action control systems use representations to drive action. It offers a way to explicate action-oriented views of cognitive systems in a way that is overlooked by recent enactivists (and many other antirepresentationalists).

To characterize the cognitive dynamics of action control in this way is to characterize it in abstract and algorithmic terms. This approach makes no mention of the features of the actual environments in which the cognitive processes, or mechanisms, might be deployed. Instead, RL provides an exact way to study the cognitive dynamics as forms of reasoning and learning processes. It helps to explain how cognitive systems control action by planning, selecting, and choosing different options.

Even a simple action—such as grasping a swirling leaf—requires complicated cognitive coordination for an agent in a dynamic, complex, and changing environment. To solve this coordination challenge, cognitive systems learn from observing the consequences of the agent's actions, they select actions on the basis of past results, and explore new strategies. Moreover, when necessary, intelligent cognitive systems change their goals, compare alternative plans, and search for better solutions. When assessing what is the most plausible story of this kind of action control, perhaps we should let go of the assumption that only states that track external environments count as representational, not the whole representational framework.

REFERENCES

Botvinick, Matthew, Ari Weinstein, Alec Solway, and Andrew G. Barto. 2015. "Reinforcement Learning, Efficient Coding, and the Statistics of Natural Tasks." *Current Opinion in Behavioral Sciences* 5:71–77.

Dayan, Peter, and Yael Niv. 2008. "Reinforcement Learning: The Good, the Bad and the Ugly." *Current Opinion in Neurobiology* 18 (2): 185–96.

Doya, Kenji. 1999. "What Are the Computations of the Cerebellum, the Basal Ganglia and the Cerebral Cortex?" *Neural Networks* 12:961–74.

———. 2008. "Modulators of Decision Making." *Nature Neuroscience* 11 (4): 410–16.

Dretske, Fred. 1981. *Knowledge and the Flow of Information*. Cambridge, MA: MIT Press.

Egan, Frances. 2014. "How to Think about Mental Content." *Philosophical Studies* 170:115–35.

———. 2020. "A Deflationary Account of Mental Representation." In *What Are Mental Representations?* ed. Joulia Smortchkova, Krzysztof Dołęga, and Tobias Schlicht, chap. 2. New York: Oxford University Press.

[12] For an overview, see Niv (2009). There is a growing body of neurophysiological evidence suggesting that parts of the midbrain may implement reward prediction encoding and that the prefrontal cortex implements reward-based learning mechanisms in motor control in action selection and in visual attention.

https://doi.org/10.1086/715513 Published online by Cambridge University Press

Fodor, Jerry. 1992. *A Theory of Content and Other Essays*. Cambridge, MA: MIT Press.

Grush, Rick. 2004. "The Emulation Theory of Representation: Motor Control, Imagery, and Perception." *Behavioral and Brain Sciences* 27 (3): 377–96.

Hubel, David H., and Torsten N. Wiesel. 1959. "Receptive Fields of Single Neurones in the Cat's Striate Cortex." *Journal of Physiology* 124 (3): 574–91.

———. 1962. "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex." *Journal of Physiology* 160 (1): 106–54.

Hutto, Daniel. 2015. "Overly Enactive Imagination? Radically Re-imagining Imagining." *Southern Journal of Philosophy* 53:68–89.

Hutto, Daniel, and Erik Myin. 2013. *Radicalizing Enactivism: Basic Minds without Content*. Cambridge, MA: MIT Press.

———. 2017. *Evolving Enactivism: Basic Minds Meet Content*. Cambridge, MA: MIT Press.

———. 2020. "Deflating Deflationism about Mental Representation." In *What Are Mental Representations?* ed. Joulia Smortchkova, Krzysztof Dołęga, and Tobias Schlicht, chap. 4. New York: Oxford University Press.

Kawato, Mitsuo. 1999. "Internal Models for Motor Control and Trajectory Planning." *Current Opinions in Neurobiology* 9 (6): 718–27.

Lettvin, Jerome Ysroael, Humberto Romersín Maturana, Warren Sturgis McCulloch, and Walter Harry Pitts. 1959. "What the Frog's Eye Tells the Frog's Brain." *Proceedings of the IRE* 47:1940–51.

Miall, R. Christopher, and Daniel Wolpert. 1996. Forward Models for Physiological Motor Control." *Neural Networks* 9:1265–79.

Millikan, Ruth. 1989. "Biosemantics." *Journal of Philosophy* 86:281–97.

Myin, Erik, and Daniel Hutto. 2015. "REC: Just Radical Enough." *Studies in Logic, Grammar and Rhetoric* 41 (54): 61–71.

Niv, Yael. 2009. "Reinforcement Learning in the Brain." *Journal of Mathematical Psychology* 53 (3): 139–54.

Ramsey, William. 2007. *Representation Reconsidered*. Cambridge: Cambridge University Press.

Sutton, Richard, and Andrew Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Thorndike, Edward. 1911. *Animal Intelligence*. Darien, CT: Hafner.

Weinstein, Ari, and Matthew Botvinick. 2017. "Structure Learning in Motor Control: A Deep Reinforcement Learning Model." arXiv, Cornell Univeristy. https://arxiv.org/abs/1706.06827.

Wolpert, Daniel, Zoubin Ghahramani, and Michael I. Jordan. 1995. "An Internal Model for Sensorimotor Integration." *Science* 269 (5232): 1880–82.

https://doi.org/10.1086/715513 Published online by Cambridge University Press