A general framework for hierarchical perception-action learning

Tara Carette (tara.carette@gmail.com)
Radboud University Nijmegen
6525GD Nijmegen, Netherlands

Serge Thill (serge.thill@donders.ru.nl)
Donders Institute for Brain, Cognition, and Behaviour
Radboud University Nijmegen
6525GD Nijmegen, Netherlands

Abstract

In hierarchical perception-action (PA) learning, agents discover invariants between percepts and actions that are structured hierarchically, from very basic immediate links to higher-level, more abstract notions. In practice, existing work tends to either focus on the general theory at the expense of details of the proposed mechanisms, or specify a-priori the contents of some layers. Here, we introduce a framework that does without such constraints. We demonstrate the framework in a simple 2D environment using an agent that has minimal perceptual and action abilities. We vary the perceptual abilities of the agent to explore how the specifics of this aspect of the agent's body might affect PA learning and find unexpected consequences. The contribution of this paper is therefore twofold, (1) we add a novel framework to the literature on PA learning, using, in particular curiosity-based reinforcement learning (RL) to implement the necessary learning mechanisms, and (2) we demonstrate that even for very simple agents, the relation between the specifics of an agent's body and its cognitive abilities is not straightforward.

Keywords: Perception Action Learning; Embodiment

Introduction

The common ground in theories of embodiment tends to be an emphasis that cognition does not exist in some abstract vacuum but, rather, in a biological body acting in a physical and social reality (Thill, 2019). What exactly this implies remains a matter of debate, for example whether or not rejecting cognitivism (as such positions tend to do) also entails rejecting computationalism and/or representationalism (Villalobos & Dewhurst, 2017), or what exactly the body actually adds. For example, if the body is just a data source and sink, then this need not have any strong consequences at all: one could still conceive of the mind as computational, with the hard problem being how whatever is being computed is connected to these data sources and sinks. This route tends to lead to the symbol grounding problem (Harnad, 1990), and is popular in areas such as cognitive and developmental robotics (Cangelosi & Schlesinger, 2015; Vernon, 2014).

With stronger notions of embodiment, such conclusions are no longer straightforward (Ziemke & Thill, 2014; Ziemke, 2016) since those must also make stronger commitments to what the body provides beyond the data source and sink. One can thus argue that the body must somehow shape the cognitive mechanisms themselves. In non-representationalist paradigms, for example, much of enactivism and those that group together under the term "4E cognition" (Newen, Bruin, & Gallagher, 2018), this is a straightforward claim. In computationalist paradigms, a possible way such shaping could take place could be by determining either the representations that computations operate on and/or the computational mechanisms themselves (Windridge & Thill, 2018). For example, in the Semantic Pointer Architecture (Eliasmith, 2013), the symbol-like "semantic pointers" are determined by the sensorimotor experience of an agent, and are thus not arbitrary. Among the computational paradigms that are suited for exploring the role of the body, those that explicitly focus on examining the linkage between perception and action are particularly interesting. For the present purposes, we focus on perception-action (PA) learning (Felsberg, Wiklund, & Granlund, 2009), chiefly because it offers a straightforward approach to hierarchical architectures that are organised in terms of levels of abstraction: lower levels of the hierarchy learn very immediate linkages between perception and action while higher levels build on those to learn increasingly abstract notions. Such architectures tend to be subsumptive, that is that higher level subsume the lower ones and consequently, they provide a theory for how low-level subsymbolic cognitive mechanisms (such as sensorimotor cognition) can be linked to higher ones that might operate in a symbol-like manner. However, the actual learning mechanisms of PA learning remain understudied, as a significant proportion of the research is primarily theoretical, with very few actual implementations (Granlund, 2006; Windridge, 2017). Furthermore, such implementations typically do not address all aspects of the architecture. The different layers can, for example, be assigned roles *a priori* instead of allowing the system to self-determine the appropriate level of abstraction for each (Windridge, Felsberg, & Shaukat, 2013; Granlund & Moe, 2004).

The primary contribution of this paper is to address these gaps with a general framework for PA learning. As a secondary contribution, we then use this framework to provide an example of how such architectures can help address the question raised above – how precisely the body might shape cognitive mechanisms. In the case of PA learning, this amounts to what kind of links between percepts and actions can be discovered by the architecture.

In the following, we first give the promised brief additional details of PA learning, followed by a description of our framework. We then describe a very simple simulated agent that

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

implements this framework to learn a two-layer architecture with no *a priori* roles defined from the outside. We vary the perceptual abilities of this agent to demonstrate consequences for the resulting architecture. Lastly, we discuss the implications. The code for the PA learning implementation and results is available as a Github repository[1].

Fundamental principles of PA learning

An agent that acts upon the world can learn to map those actions to the expected percepts (Granlund, 2006). The defining feature of PA learning is then an "action first" approach, as it is actions that are mapped onto expected percepts rather than percepts mapped onto next actions. This therefore breaks with the classic view of cognition as collecting percepts first, reasoning about them second, and acting only as a consequence (or what is known in robotics as the sense-think-act paradigm). In practice, action and perception continuously co-occur, and theories in embodied cognition routinely emphasise that it is not meaningful to clearly delineate perceiving, thinking and acting in the first place. Nonetheless, there is clearly some relationship between percepts and actions and taking an action-first approach has some immediate tangible consequence for how cognition is viewed – for example. since the action space is much smaller than the theoretical perceptual space, the fact that actions are used to gather percepts, rather than the other way around, limits an agent's actual perceptual space to only what it can observe as a result of its actions. In other words, the agent learns to perceive that which it is capable of changing within its environment (Windridge, 2017).

Restricting the learning space to what the agent might feasibly encounter is an important step, but it is insufficient by itself to capture all of the key elements of learning, or to enable the emergence of more complex behaviours. To address this, PA learning makes use of a subsumption hierarchy, in which each layer of the hierarchy works at a different level of abstraction. Higher levels subsume the lower levels, such each layer is defined by the invariants learned by the lower levels of the hierarchy (Shevchenko, Windridge, & Kittler, 2009). As lower levels get more fully developed, the level above can be further abstracted and accomplish more complex tasks.

The general principle behind the development of such a hierarchy involves exploration at each level (Windridge, 2017): agent initially rely on motor babbling to randomly explore their action space. This allows a basic mapping between actions and percepts to be created. The agent can now, through such explorations, learn underlying principles of what preceptual changes stem from what actions. This allows the identification of invariants in the linkage between action and perception. Meaningful perceptual features and action combinations can be identified by the agent, and this forms the initial layer of the hierarchy, from which it is now possible to learn the next layer.

Once again, the agent will begin motor babbling, but this time using the invariants and increased pool of possible action sequences from the layer below, leading eventually to the discovery of more complex invariants. The same process can repeat to learn increasingly complex mappings in successive layers. PA learning thus mimics random exploration of different levels of complex behaviour that can also be observed in humans: babies motor babble in fairly obvious ways when first learning to navigate the world (Oudeyer, Kaplan, Hafner, & Whyte, 2005). Adults do this as well, when they are presented with a new type of problem (Felsberg et al., 2009), albeit in a manner that might appear as meaningful exploration as adults already have many established concepts about the world. Nonetheless, within the new space, the actions can be fairly randomized.

Agents that develop, over time, such a subsumption architecture using PA learning appear to carry out progressively more meaningful actions, even if there is no explicit goal to their behaviour. They simply become more adept at exploring increasingly complex relations between increasingly complex action sequences and percepts in their environment. Shevchenko et al. (2009) demonstrate this using a PA learning agent that eventually solves a shape sorting task even without being directly guided towards that goal.

The above describes the fundamental principles underlying this type of learning: an action-first philosophy and a subsumption architecture (Windridge et al., 2013; Granlund & Moe, 2004). It also illustrates how PA learning is distinct from similar theories, such as (hierarchical) predictive coding (Clark, 2013; Rao & Ballard, 1999; Wacongne et al., 2011) – for example, there is no particular committent to generative models in a Bayesian sense. Beyond this core, however, many specifics are left unarticulated in theoretical work or differ between implementations. How invariants are extracted from the exploration, how new actions are meaningfully found, how perceptual goals are decided upon, when the learning is moved to a new layer, how layers pass commands to each other, if upper layers should have an impact on the learning of lower layers and to what extent (for example, Windridge et al. (2013) improved a PA learning hierarchy implementing a driver assistance system by adding top-down modulation from an added top layer implementing fuzzy logic) and other such questions still need to be asked when designing a PA learning approach. The following section describes how these are addressed here in a manner that is generalisable.

A general framework for PA learning

Scope

Here, we will focus on the necessary aspects for a functioning implementation of PA learning. The important elements are therefore learning the relationships between percepts and actions in an 'action-first' manner, and all the necessary elements to enable learning the subsumption hierarchy. In other words, actions exploring the environment should guide learning, and learning should exist in multiple explicit layers that handle different levels of action and percept complexities.

[1] https://github.com/TaraCarette/PA_Learning_Agent

This means that we do not address secondary aspects, such as top-down feedback (mentioned above) here. Learning will clearly still be possible, albeit possibly less efficient.

We also disregard very formal discoveries of actions. Shevchenko et al. (2009) defines distinct stages of finding perceptual goals, discovering which actions consistently lead to those goals, and then pruning that action sequence down to only what is actually required. Here, we only implement one intrinsic motivation – curiosity – that implicitly leads the agent to explore how to improve its internal models, resulting in an implicit pruning process. As the agent learns to improve its ability to predict what will happen next, if a sequence of actions does not result in any learning, the agent will learn to avoid that sequence.

Lastly, as will be evident in the next subsection, we use reinforcement learning (RL) for the actual learning. There are no particularly strong constraints on how learning is implemented exactly in PA approaches, however, since interaction with the environment is key, RL is a good conceptual fit.

Putting the "learning" into "PA learning"

We implement learning using proximal policy optimisation (PPO; Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017), which is well suited to continuous control problems that one would expect an agent implementing PA learning to have to tackle. PPO can be implemented in actor-critic algorithms, meaning that two elements get trained. The actor, which decides what action the agent should take (in other words, the policy), and the critic, which predicts the value of the state resulting from an action.

The choice of action to be carried out will thus be made in function of the reward signal. The key is then to ensure that rewards are given in a manner that promotes the successful development of an internal model of the PA mappings. The reward should therefore promote exploration of different actions in different areas of the environment as the agent learns the ways it can interact with the world around it.

Exploration-based learning within an RL framework brings us to curiosity-based learning, which has a number of parallels with what PA learning intends to achieve. Both have an intrinsic drive to improve their internal model of the world rather than externally-set goals and achieve this by learning to predict the consequences of their actions upon an external world. In other words, if the actions of the policy are guided by a curiosity-based reward, then the exploration-based elements of PA learning are captured. In addition, as information only follows from the actions the agent takes, the 'action first' philosophy is maintained. However, it is important to note that curiosity-driven RL is not sufficient to implement a full PA learning as it misses the subsumption architecture aspect; we will address this in the next section.

To implement curiosity-based RL, we build on the Intrinsic Curiosity Module (ICM; Pathak, Agrawal, Efros, & Darrell, 2017). ICM uses a forward model to predict the next state given the current state and chosen action by the agent as an input.[2] The reward signal is then inversely related to the quality of the prediction. In other words, if the forward model can perfectly predict the outcome of an action, there is nothing left to learn, so such actions get deprioritised, causing the agent to explore less well understood parts of the environment. This operationalises curiosity.

ICM further implements feature detectors. These turn raw sensory inputs into a lower-dimensional representation that filters out meaningless information about states (in the sense that it is not useful for training the forward model). From a PA perspective, these feature detectors can be understood as picking up on the invariants in the perception-action linkages that the forward model can pick up on.

Subsumption architecture

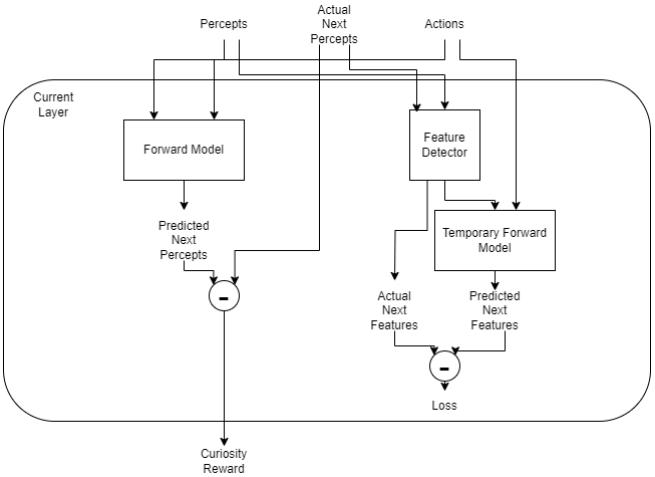

Figure 1: The structure within a single layer of the PA architecture (explained in more detail in the text)

Each layer of our subsumption architecture first implements ICM's forward model to obtain the rewards for the curiosity-driven RL algorithm. The feature detector of each layer will, however, serve as a perceptual input to the next layer given that it will learn accessible invariants the current layer. In other words, unlike the ICM implementation, the forward model of the first layer does actually receive raw perceptual inputs and the feature detector learns to represent the invariants that the agent is able to discover on that basis. The feature detector then serves as the input to the next layer, where a new feature detector will learn the invariants that a new forward model is capable of discovering given those inputs. This creates the subsumption hierarchy: each layer acts given invariants from the layer below as inputs, thus moves into increasingly complex spaces while subsuming the layer below.

A notable difference between this implementation and ICM is that the feature detector and the forward model it re-

[2]We note that the ICM implementation also includes inverse models that help with planning goal-directed actions. We omit these since they are not critical to our framework

lates to exist on different layers and will not co-develop, since layers are trained subsequently in PA architectures. We therefore use a separate "temporary" forward model (in the sense that it is only active while the layer it belongs to is being trained) to train the feature detectors of each layer.

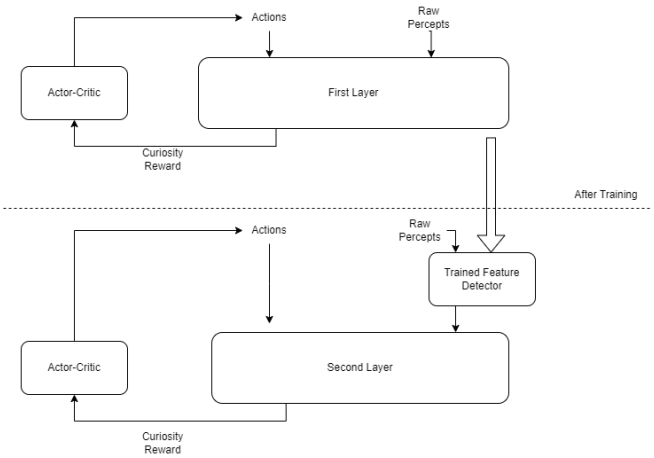

Figure 2: The first and second layer of the PA architecture. Note that, for illustrative purposes, we explicitly depict the trained feature detectors that would belong to the first layer (but, once trained, become available as inputs to the second layer).

This concludes the general framework for PA learning presented here. Figure 1 visualises the implementation of a single layer, while Figure 2 shows how multiple layers connect, and thereby the final architecture. We now move on to the proof of concept implementation.

Proof of concept

Simulation environment

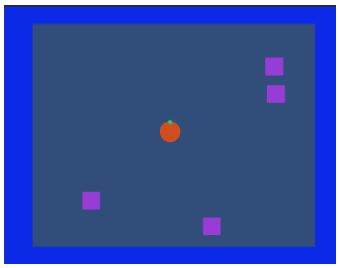

Figure 3: The simple enclosed environment with various objects in purple and the agent in red

A minimalist 2D environment and agent (Figure 3) are created using Unity[3]. As previously mentioned the `mlagents` library[4] is used for an off-the-shelf RL implementation. The environment consists of a bounded arena filled with moveable objects. The agent consists of a simple circular body. In terms of actions, the agent is capable of (1) moving along its 4 cardinal directions, (2) rotating on the spot, and (3) toggling a "sticky" state on and off. If the sticky state is on, objects touched by the agent become attached to it and will move along. In other words, this is a rudimentary grasping ability. In terms of sensing, the agent has proprioceptive abilities, that is, it has information about its speed, whether it is currently "sticky", and current orientation with respect to the environment. It has a touch sensor that informs about whether or not it is currently in contact with something (but not what that thing is) and, most importantly, its visual sensor is implemented as a ray-tracing proximity sensor that for each of the rays, returns the distance and colour of objects (including the walls of the environment) up to some maximum distance. The agent implements the PA architecture described above and has no externally assigned goals, all behaviour being driven by the curiosity-based RL within the architecture.

Experimental design

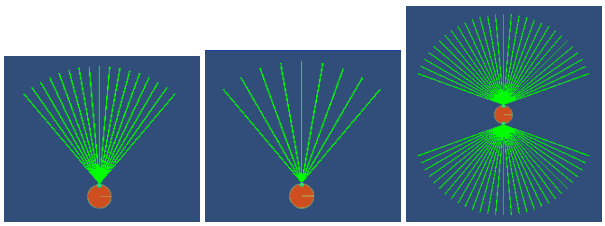

Figure 4: The three perceptual abilities implemented here, varying number of rays and field of view. We term these *middle*, *sparse*, and *wide* agents respectively

We address two things in this environment. First, we demonstrate that the agent is capable of developing the layers of the PA architecture, validating the general framework. Second, we explore what consequences there are to changing some aspects of the agent's body. Given that this is a simple agent and environment, the options are rather limited. We therefore manipulate the agent's visual abilities by varying the number of rays it can cast within a specific field of view and improving its sensing abilities of the environment both by increasing the field of view of the sensor and adding a second one (Figure 4).

There is not that much that one can expect form these kind of agents in this environment. Changes in percepts will be driven by movement, and the sticky function might lead to initially unexpected perceptual consequences. One would expect a PA hierarchy to capture these aspects, but there is no *a priori* reason to expect the variations in sensing abilities to significantly affect what the models and feature detectors in each layer learn. It is also worth repeating that an important aspect of this architecture is that there are no external constraints on what each layer should end up discovering. In other words, the agent will develop its own levels of abstraction that do not necessarily have to map onto what humans would find appropriate. This is a subtle point, but it follows

[3] `wwww.unity.com`
[4] `https://github.com/Unity-Technologies/ml-agents`

from strong notions of embodiment that, if the embodiment is very different from a human one, the same would be expected to apply to the agent's cognitive abilities (Thill, Padó, & Ziemke, 2014).

For the purposes of evaluation, we are primarily interested in qualitative descriptions of the behaviour that the agent displays when driven by each layer of the architecture. Since the architecture is sub-symbolic and no pre-conceived notions of percepts, actions, or links between these are provided, there is no straightforward way to inspect what each layer has learned directly.

Results

PA learning

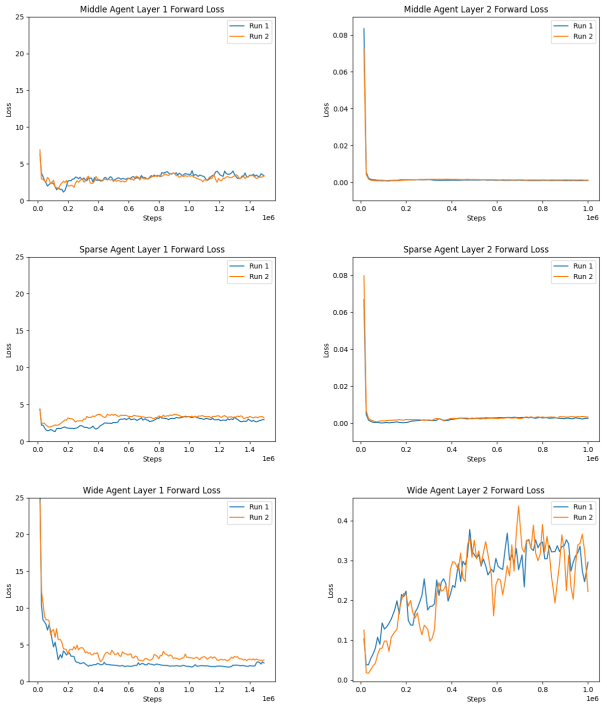

Figure 5: The losses during the development of the forward models for two layers of the *middle*, *sparse*, and *wide* agents' architecture respectively (with the second layer on the right). Two example runs are shown to indicate the variation between runs

Since we are training two separate aspects in each layer, the forward model and the feature detector, we can inspect the evolution of their loss during the lifetime of the agent. Low losses for the forward model indicate learning of linkages between action and percepts, and low losses for the feature detector that the invariants are well captured. Note that this says nothing about the quality of the behaviours discovered, the completeness of the behavioural repertoire of the agent, or similar aspects – just that, given the percepts and actions available at that layer, a decent ability to model their relationship has been achieved.

We find that the losses associated with feature detectors consistently move to near-zero in all cases, indicating that the feature detectors are able to learn invariants of whatever linkages exist between percepts and actions. We omit graphs for these given their trivial nature and lack of space. As far as the losses associated with the forward model are concerned (Fig 5), they drop to low values for all agents on the first layer, indicating that there remain some aspects that the forward model cannot fully capture yet. For two of the agents, the forward loss of the second layer then drops to near-zero, indicating that those have now successfully explored their behavioural repertoire. For the *wide* agent, however, we find an inability to improve the loss of the second layer's forward model. This suggests that this agent requires additional layers to fully map its behavioural repertoire. Allowing it to develop a third layer confirms this, with a loss that now settles at near-zero (with the graph again omitted as it mimics those of the second layer for the other two agents).

Overall, this demonstrates that the framework described here is able to lead to the development of a PA architecture. As noted, it says nothing about what behaviours are learned at each layer, so we investigate these qualitatively.

Behaviours

Although the agent does not have any preconceptions about what its percepts and action spaces are, we do, as previously noted, have some notion of the kind of behaviours an agent can discover in the given setting, and can therefore look at the frequency of certain behaviours at different layers of the hierarchy. We do this for touching walls, or objects, for toggling the sticky function, for being in contact with objects while sticky, and for movement and rotation (Figure 6 shows some examples).

In general, all agents appear to end up with navigation abilities in the first layer of the architecture, with significant exploration through movement but not much usage of the sticky function. If the sticky function is used, this seems to be brief and random, and not explored further. The behaviours learned within the second layer – which now operates on the invariants discovered in the first layer rather than the raw percepts themselves – appear more interesting. The *middle* agent appears to focus primarily on rotation, de-prioritising other types of movement and not making use of its ability in principle to interact with objects, or using its sticky function. In comparison, while it likewise tends to favour rotation over movement, the *sparse* agent explores contact with both objects and the walls. It learns to toggle its sticky function the majority of the time, as it experiments with sticking primarily to a single object. In other words, it acquires a new behaviour, sticking to a single object, that it did not develop in the first layer.

Finally, the *wide* agent also touches objects and walls. In contrast to the *sparse* agent, it learns to toggle the sticky function less, thus staying in contact with an object whether sticky or not. Unlike other agents, we also observe that the *wide* agent tends to be in touch with multiple objects, sometimes

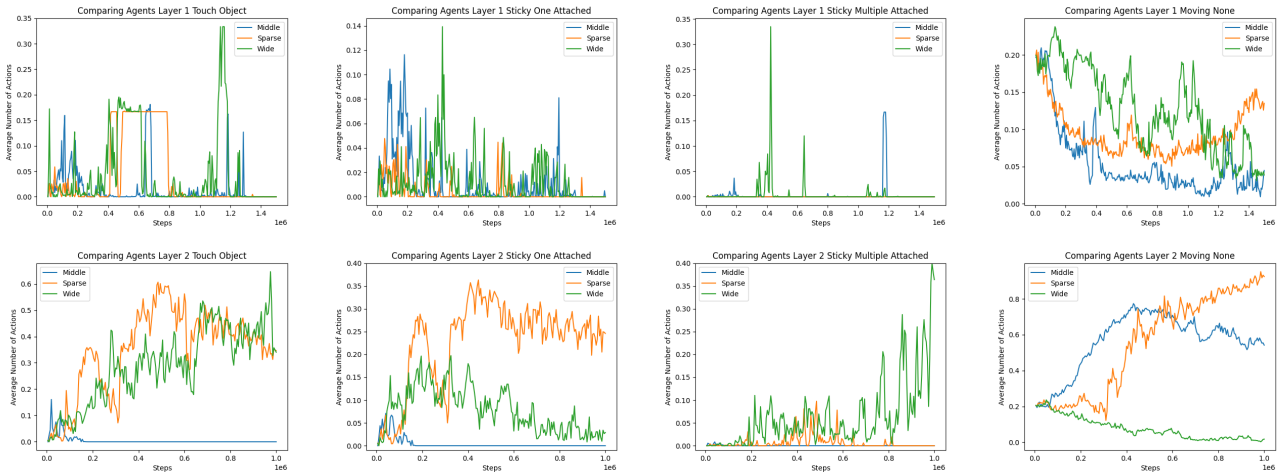

Figure 6: Example frequencies of behaviours that agents show at each layer of the architecture. Figures show the amount of time spent touching walls, using the sticky function on one or multiple objects, and the amount of time spent stationary (though possibly still rotating). Not shown due to lack of space are the amount of time spent toggling the sticky function, touching walls, and rotating.

wedging itself between them, and focuses more on movement than rotation. We noted before that the forward loss for this agent and layer does not decrease. Together, this suggests that the *wide* agent, because of its increased interactions with objects compared to the other two agents, fails to learn to fully predict the consequences of movements in all such situations. Adding a third layer to this agent then reveals behaviour that is similar to the second layer of the *middle* agent, indicating that once more, the agent has discovered everything it is able to discover about its perception-action abilities.

Discussion and conclusion

To summarise the results, our agents were able to develop a PA architecture with different behaviours available at each layer. Manipulating the perceptual abilities also resulted in architectural differences that were somehow unexpected. While all agents essentially develop movement behaviours initially, only the *sparse* and *wide* agent go on to discover object interactions subsequently. There are also qualitative differences to their style of object interaction, with the *sparse* agent more focussed on sticking to one object and the *wide* agent exploring manipulating multiple objects.

It seems straightforward that the *wide* agent discovers more complex object interaction abilities: given its nearly all-encompassing field of view (Figure 4), it is better placed than either other agent to actually perceive perceptual consequences of multiple object interaction. It is less obvious that the *sparse* agent discovers object interaction while the *middle* agent does not. After all, the *sparse* agent has the most restricted perceptual ability of all. There is not too much point in speculating in detail as to why this may be the case. We suspect that the richer perceptual field of the *middle* agent causes more difficulty in learning to predict the consequences of movement, so the curiosity-based RL continues to focus on movement and rotation. For the *sparse* agent, because there is less to predict, this problem might be reduced, leading to increased curiosity about other possible actions. Similarly, this would suggest that the *wide* agent is less concerned with actual object interaction than it is with multiple objects in its field of view causing perceptual changes that are difficult to predict. Thus it seeks out situations involving multiple objects but without really exploring its ability to stick to them.

The curious result is therefore that it is the agent with the poorest perceptual abilities that discovers the most about its ability to move objects. In particular, this means that all other agents also had access to the necessary perceptual information, so their failure to discover their sticky abilities is not a consequence of a lack of information. Even though this is a simplistic setting, it therefore demonstrates how the body can shape cognitive abilities of an agent in ways that are not trivial to expect, and that there are relations between perceptual and action abilities that traditional sense-think-act paradigms cannot capture.

More generally, we have also demonstrated a framework for PA learning that imposes no constraints on what any of the layers can pick up on, demonstrating, in particular, the viability of curiosity-based RL as a concrete learning mechanism in PA architectures. There remains, of course, plenty of scope to develop this further. In particular, the framework still lacks top-down mechanisms that would allow the agent to use its PA architecture for goal-directed (as opposed to merely curiosity-driven) behaviour. Such improvements can be the subject of future work.

References

Cangelosi, A., & Schlesinger, M. (2015). *Developmental robotics: From babies to robots*. The MIT Press. doi: 10.7551/mitpress/9320.001.0001

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*(3), 181–204. (Publisher: Cambridge University Press) doi: 10.1017/S0140525X12000477

Eliasmith, C. (2013). *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford University Press. doi: 10.1093/acprof:oso/9780199794546.001.0001

Felsberg, M., Wiklund, J., & Granlund, G. (2009, 10). Exploratory learning structures in artificial cognitive systems. *Image and Vision Computing*, *27*, 1671-1687. doi: 10.1016/J.IMAVIS.2009.02.012

Granlund, G. H. (2006). Organization of architectures for cognitive vision systems. In *Cognitive vision systems* (pp. 37–55).

Granlund, G. H., & Moe, A. (2004). Unrestricted recognition of 3d objects for robotics using multilevel triplet invariants. *AI Magazine*, *25*, 51-67.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, *42*(1-3), 335–346.

Newen, A., Bruin, L. D., & Gallagher, S. (2018). *The Oxford Handbook of 4e Cognition*. Oxford: Oxford University Press.

Oudeyer, P.-Y., Kaplan, F., Hafner, V. V., & Whyte, A. (2005). The playground experiment: Task-independent development of a curious robot. In *Proceedings of the aaai spring symposium on developmental robotics* (Vol. 42, p. 47).

Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017, 5). Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning* (pp. 2778–2787). Retrieved from `http://arxiv.org/abs/1705.05363`

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1), 79–87. doi: 10.1038/4580

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shevchenko, M., Windridge, D., & Kittler, J. (2009, 10). A linear-complexity reparameterisation strategy for the hierarchical bootstrapping of capabilities within perception-action architectures. *Image and Vision Computing*, *27*, 1702-1714. doi: 10.1016/j.imavis.2008.12.002

Thill, S. (2019). What we need from an embodied cognitive architecture. In M. I. Aldinhas Ferreira, J. Silva Sequeira, & R. Ventura (Eds.), *Cognitive architectures* (pp. 43–57). Springer. doi: 10.1007/978-3-319-97550-4_4

Thill, S., Padó, S., & Ziemke, T. (2014). On the importance of a rich embodiment in the grounding of concepts: Perspectives from embodied cognitive science and computational linguistics. *Topics in Cognitive Science*, *6*(3), 545–558. doi: 10.1111/tops.12093

Vernon, D. (2014). *Artificial Cognitive Systems: A primer*. Cambridge, MA: MIT Press.

Villalobos, M., & Dewhurst, J. (2017). Why post-cognitivism does not (necessarily) entail anti-computationalism. *Adaptive Behavior*, *25*(3), 117–128. (Publisher: SAGE Publications Ltd STM) doi: 10.1177/1059712317710496

Wacongne, C., Labyt, E., van Wassenhove, V., Bekinschtein, T., Naccache, L., & Dehaene, S. (2011). Evidence for a hierarchy of predictions and prediction errors in human cortex. *Proceedings of the National Academy of Sciences*, *108*(51), 20754–20759. doi: 10.1073/pnas.1117807108

Windridge, D. (2017). Emergent intentionality in perception-action subsumption hierarchies. *Frontiers in Robotics and AI*, *4*, 4-38. Retrieved from `www.frontiersin.org` doi: 10.3389/frobt.2017.00038

Windridge, D., Felsberg, M., & Shaukat, A. (2013, 2). A framework for hierarchical perception-action learning utilizing fuzzy reasoning. *IEEE Transactions on Cybernetics*, *43*, 155-169. doi: 10.1109/TSMCB.2012.2202109

Windridge, D., & Thill, S. (2018, 10). Representational fluidity in embodied (artificial) cognition. *BioSystems*, *172*, 9-17. doi: 10.1016/J.BIOSYSTEMS.2018.07.007

Ziemke, T. (2016). The body of knowledge: On the role of the living body in grounding embodied cognition. *Biosystems*, *148*, 4–11. doi: 10.1016/j.biosystems.2016.08.005

Ziemke, T., & Thill, S. (2014). Robots are not embodied! Conceptions of embodiment and their implications for social human-robot interaction. In *Proceedings of RoboPhilosophy 2014: Sociable robots and the future of social relations* (pp. 49–53). Amsterdam, NL: IOS Press BV. doi: 10.3233/978-1-61499-480-0-49