A Comparison of Strategies for Developmental Action Acquisition in QLAP

Jonathan Mugan
Department of Computer Science
The University of Texas at Austin
Austin, TX 78712 USA
jmugan@cs.utexas.edu

Benjamin Kuipers
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109 USA
kuipers@umich.edu

Abstract

An important part of development is acquiring actions to interact with the environment. We have developed a computational model of autonomous action acquisition, called QLAP. In this paper we investigate different strategies for developmental action acquisition within this model. In particular, we introduce a way to actively learn actions and we compare this active action acquisition with passive learning of actions. We also compare curiosity based exploration with random exploration. And finally, we examine the effects of resource restrictions on the agent's ability to learn actions.

1. Introduction

We seek to understand how an agent (human or otherwise) can learn to adapt to its environment through the process of development. Gibson (1988) proposed that human children are endowed with systems to allow them to explore and learn about the world. She emphasized that it was this exploration that enabled cognitive development. One such system appears to be that for learning contingencies. It has been proposed that humans have an innate contingency detection module (Gergely and Watson, 1999). Human infants can detect contingencies in their environment shortly after birth (DeCasper and Carstens, 1981), and they can link these contingencies with observable effects (Adolph and Joh, 2007).

Inspired by this idea that learning can take place through the acquisition of contingencies, we created the Qualitative Learner of Action and Perception (QLAP). QLAP is constructivist in the tradition of Piaget (1952) because the agent constructs representations of the environment. QLAP learns contingencies and actions through autonomous exploration. QLAP learns contingencies by observing events in the environment and looking for correlations (Mugan and Kuipers, 2008, Mugan and Kuipers, 2007). Once a contingency is found that is sufficiently deterministic, QLAP creates a plan to perform an action based on that contingency (Mugan and Kuipers, 2009).

Adolf and Joh (2007) note the importance of action learning in the role of providing agent-centered input to the perceptual systems. Generating agent-centered experience by learning actions requires that the agent autonomously explore its environment. This type of exploration has been characterized as intrinsically motivated learning (Berlyne, 1965) and is essential for autonomous development (Ryan and Deci, 2000). The problem of picking which action to choose has been studied extensively, for example see (Schmidhuber, 1991, Huang and Weng, 2002, Marshall et al., 2004). One promising approach is picking actions that maximize the learning gradient (Oudeyer et al., 2007). However, exploration for learning actions is more than picking which action to choose. The agent must first form the actions.

QLAP assumes that the agent has motor primitives but no initial complex actions. From these motor primitives, QLAP learns actions such as reaching out to hit a block. However, some more complex actions may have to be learned using *active action acquisition*. Active action acquisition involves two steps. First, the agent tunes its search for contingencies related to a desired action to be more sensitive, so that it finds contingencies that it might otherwise overlook. And second, the agent makes it more likely that a found contingency will become a plan to perform the action by lowering the required reliability of the contingency.

The contribution of this paper is to provide an evaluation of exploration strategies for learning actions. We evaluate different exploration strategies in an environment inspired by the sticky mittens experiments (Needham et al., 2002). In these experiments, children wore mittens covered with Velcro that allowed them to more easily grasp objects. They found that infants trained with the sticky mittens exhibited

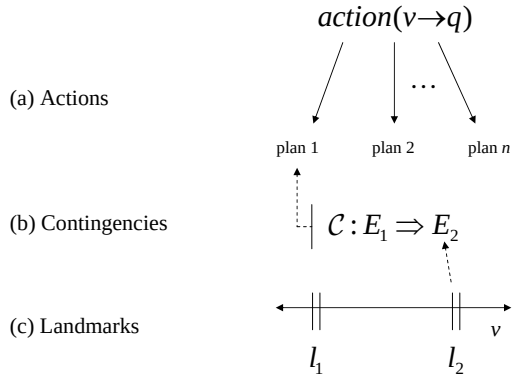


Figure 1: (a) An action brings a qualitative variable v to a desired value q. Each action can have one or more plans. Each plan is a different way to perform the action. (b) Each plan is learned by first learning a contingency. A contingency links an antecedent event E_1 with a consequent event E_2. Associated with each contingency is a probability table that gives the probability of event E_2 following event E_1 for each value of the variables in context $\mathcal{C}$. (c) Each event is able to be perceived because of the discretization created by the landmarks.

more object engagement and more sophisticated object exploration strategies.

We evaluated the effect of using active action acquisition. We found that active acquisition improved the agent's performance on the task of picking up the block with the sticky mitten, but hurt the agent's performance on the easier task of moving the block. We found that using active action acquisition in combination with the exploration method of Intelligent Adaptive Curiosity (IAC) (Oudeyer et al., 2007) worked best in this continuous domain for enabling the agent to develop so that it could learn to pick up the block using the sticky mitten. We also evaluated the use of developmental restrictions and we found that certain developmental restrictions allowed the agent to reduce the number of learned contingencies without hindering learning. And finally, we found that the developmental trajectory allows that agent to progress from actions being used mostly as exploration to actions being used as subactions for other actions.

2. The Qualitative Learner of Action and Perception, QLAP

The Qualitative Learner of Action and Perception (QLAP) is a computational model for learning both important perceptual distinctions and actions (see Figure 1). QLAP assumes that the agent can distinguish objects from the background and track them. QLAP also assumes that the agent can measure distances between objects and that the agent has motor variables for output. The result of these assumptions is that the agent interacts with the world using a set of real-valued variables.

2.1 Qualitative Representation

The distinctions that QLAP learns allows it to represent the state of the world qualitatively. It does this by converting the continuous input and motor variables to qualitative variables (Kuipers, 1994). A qualitative representation allows the agent to focus on important distinctions while ignoring others. The qualitative variables are created by discretizing the continuous variables using *landmarks*. A landmark is a symbolic name for a point on a number line. A variable v with two landmarks l_1 and l_2 would have a set of five possible qualitative (discrete) values $\{(-\infty, l_1), l_1, (l_1, l_2), l_2, (l_2, +\infty)\}$. QLAP must learn these landmarks. For example, QLAP learns a landmark that a force of at least 300 is needed to move the hand to the right. It also learns a landmark that a distance of 0 between the right side of the hand and the left side of the block is important to move the block to the right.

2.2 Landmarks to Events

Landmarks allow the agent to perceive *events*. An event is the change in qualitative value of a variable. We use the notation $E = X_t \rightarrow x$ to denote event E where the value of qualitative variable X changes to x at time t (although the t may be omitted for brevity.) For example, when the distance between the right side of the hand and the left side of the block goes to 0.

2.3 Events to Contingencies

The perception of events allows the agent to learn *contingencies*. Contingencies link an *antecedent event* $E_1 = X \rightarrow x$ with a *consequent event* $E_2 = Y \rightarrow y$ together in time. For each contingency, QLAP learns a context $\mathcal{C}$ that gives the probability of the consequent event following the antecedent event for each value of the variables in $\mathcal{C}$. We call the highest probability of event E_1 leading to event E_2 the *best reliability* of the contingency. Once the best reliability of a contingency exceeds 0.75 the contingency is labeled *sufficiently deterministic*.

New landmarks can be learned by finding new distinctions that make contingencies more reliable. For example, the agent may learn a contingency that states that the event of a positive force on the hand will cause the event of the hand moving to the right. Once this contingency is learned, QLAP can examine the real values of the variables and determine if there is a new distinction that will make this contingency more reliable. In this case, it takes a force of 300 units to move the hand to the right. The agent

can then update the contingency to reflect this new distinction by introducing a landmark. The agent can also learn that the hand will not move to the right if it is already all the way to the right. It can then learn a landmark on the location of the hand to indicate when it is in its rightmost position.

2.4 Contingencies to Plans for Actions

In QLAP, the agent learns *actions* to achieve the qualitative values of variables. Each action sets the qualitative value of a variable to a desired value. In QLAP, actions may be performed in more than one way. Each way to perform the action is called a plan. Each plan is represented as an option (Sutton et al., 1999). Once a contingency is sufficiently deterministic it is converted into a plan. These plans are learned using reinforcement learning (Sutton and Barto, 1998), see (Mugan and Kuipers, 2009) for details.

3. Developmental Learning in QLAP

QLAP is not given a learning objective but learns in a developmental progression. This developmental progression comes from incrementally learning contingencies, actions, and landmarks. In addition, it comes from developmental restrictions that take three forms:

1. **restrictions on learning contingencies**
   In QLAP, a contingency can only be learned if its antecedent event can be reliably predicted by a previously learned contingency.
2. **restrictions on learning plans**
   A contingency can only be converted to a plan if the antecedent event can be reliably achieved using an existing action.
3. **restrictions on cognitive load**
   An agent has limited cognitive resources and an important part of development is freeing up resources. QLAP designates an action as *open*, *full*, or *closed*. An action is closed if it can be achieved 75% of the time; otherwise, it is full if it has 5 plans; and it is open otherwise. Actions that are closed or full do not accept additional plans. When an action is closed, it also affects the learning of contingencies. QLAP does not add a contingency if the action to bring about the consequent event is closed.

   Contingencies can also be deleted. If the contingency does not become a plan after 100,000 timesteps, it is deleted. When an action is closed, all of the related contingencies that are not part of plans for that action are deleted.

   Plans can also be deleted. A plan and its associated contingency are deleted if its associated action is still not closed and the reliability of the plan is less than 5%.

3.1 Choices Made During Exploration

The agent continually makes three types of choices during its exploration. These choices vary in time scale from coarse to fine.

1. The agent chooses an *exploration action*, which is a previously learned action that it can practice. This can be done randomly or by using a version of Intelligent Adaptive Curiosity (IAC) (Oudeyer et al., 2007) which first measures the change in the agent's ability to perform the action over time and then chooses actions where that ability is increasing. For IAC, we use a time window $\tau = 25$ and a smoothing parameter $\theta = 25$ (before the time window of $\tau = 25$ is full, actions are chosen based on the product of probability of success in the current state and the entropy of their overall reliability).
2. The agent chooses the best plan for performing the action. The agent chooses the plan most likely to succeed in the current state with probability 0.95 and chooses a random plan otherwise.
3. The agent chooses the subaction within the plan. This is done using the standard reinforcement learning technique ϵ-greedy that balances exploration with exploitation (Sutton and Barto, 1998).

3.2 Execution

An outline of the execution of QLAP is shown in Algorithm 1. Note that for the first 20,000 timesteps the agent chooses random motor babbling exploration actions. After that point it chooses a motor babbling action with probability 0.1, otherwise it chooses an exploration action and action plan according to (Mugan and Kuipers, 2009).

4. Active Action Acquisition

A plan to perform an action is formed when the contingency is sufficiently deterministic. In the developmental progression just described, the agent learns these plans without paying special attention to what the goal of the associated action is. We call this approach *passive action acquisition*. This method of passive learning may not be sufficient to learn difficult actions. To learn difficult actions, the agent may have to employ active action acquisition. To learn a plan for an action chosen for active action acquisition, QLAP

1. **lowers the threshold needed to learn a contingency.** QLAP learns a contingency linking an event E_1 and an event E_2, if E_2 is more likely to

Algorithm 1 The Qualitative Learning of Action and Perception (QLAP)

1: **for** $t = 1 : \infty$ **do**
2: Sense environment
3: Convert input to qualitative values using current landmarks
4: Update statistics to learn new contingencies
5: Update statistics for each contingency
6: **if** $mod(t, 2000) == 0$ **then**
7: Learn new contingencies
8: Delete unneeded contingencies and plans
9: Learn new landmarks to change qualitative representation
10: Learn new actions
11: **end if**
12: **if** current exploration action is completed **then**
13: Choose new exploration action and action plan
14: **end if**
15: Get low-level motor command based on plan of current exploration action
16: Pass motor command to robot
17: **end for**

soon occur given that E_1 has occurred than otherwise. More formally, if we define a time window with the predicate $soon(t, E)$ that is true if event E occurs within a window if $k = 5$ timesteps starting at time t, then we can say that the contingency is formed if

$$Pr(soon(t, E_2)|E_1(t)) - Pr(soon(t, E_2)) > \theta_p$$

where $\theta_p = 0.05$. If event E_2 is chosen to be the goal of an actively acquired action, we make it more likely that a contingency will by learned by using $\theta_a = 0.02$ instead if $\theta_p = 0.05$.

2. **lowers the threshold needed to learn a plan.** A contingency becomes a plan if its best reliability is greater than 0.75. For a contingency with a consequent event that is chosen to be the goal of an actively acquired action, this threshold is reduced to 0.25.

This leaves the question of when to specify events as goals of actively acquired actions. An event is chosen as a goal for active action acquisition if the probability of being in a state where the event is satisfied is less than 0.05; we call such an event *sufficiently rare*. This is reminiscent of Bonarini et al. (2006). They consider desirable states to be those that are rarely reached or are easily left once reached.

5. Evaluation

We run experiments using the environment shown in Figure 2. The environment is implemented in Breve (Klein, 2003) and has realistic physics. The simulation consists of a robot at a table with a block. The robot has an orthogonal arm that can move in the x, y, and z directions. During learning, the agent chooses exploration actions autonomously. Each time the agent knocks the block out of reach, the block is replaced with a different block and put on the table. The block size varies randomly in length from 1.0 to 3.0 units.

For each experiment we trained 40 agents. We trained each for 250,000 timesteps, which corresponds to about 3.5 hours of physical experience. The robot has a "sticky mitten." If the center of the block touches the bottom of the hand, then the block is "grabbed." For simplicity, there is no ungrab action. Instead, the block has a probability of 0.1 of becoming ungrabbed at each timestep. Then when the block becomes ungrabbed, it falls to the table with probability 0.5 or gets moved to another place on the table with probability 0.5. To make the environment more realistic, there are two distractor objects that float in front of the agent. The agent can perceive the distractor objects and learn contingencies about them, but cannot interact with them.

5.1 Evaluation Tasks

We measure the performance on two tasks. The first task is that of moving the block in a specified direction. The agent is told to move the block either left, right, or forward. The second task is picking up the block using the sticky mitten.

QLAP autonomously learns without being specified a task. We can be confident that it will learn the specified tasks because the number of variables in the environment is small. However, during learning, the agent does not know that it will be evaluated on these tasks.

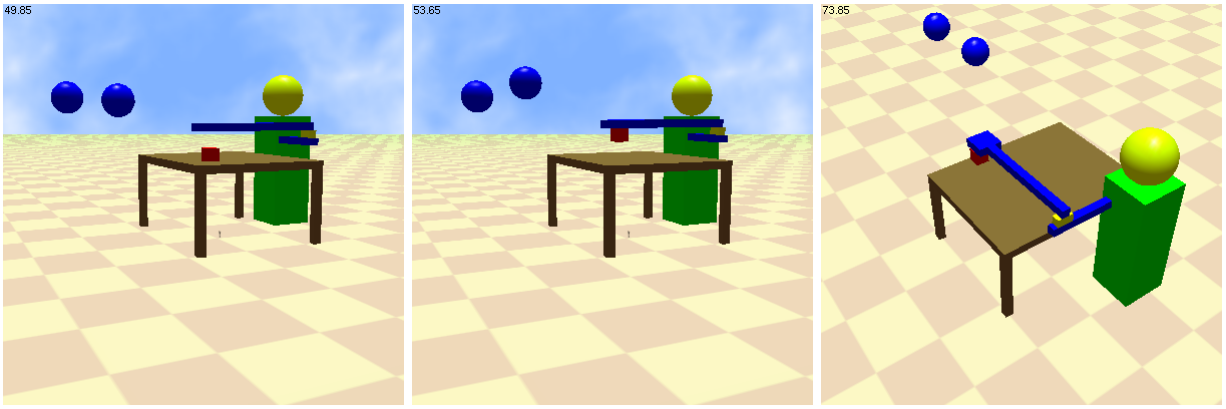
Every 10,000 timesteps (about every 8 minutes of physical experience) we save the state of the agent. We then test how well each can do that task starting from this stored learned state. Each evaluation consisted of 100 episodes. Each episode lasted for 300 timesteps or until the block was moved. The agent received a penalty of -0.01 for each timestep, and it received a reward of 10.0 if it completed the task.

5.2 Experimental Conditions

active random This case used active action acquisition with exploration actions chosen randomly from a uniform distribution.

active IAC This case used active action acquisition with exploration actions chosen using Intelligent Adaptive Curiosity.

passive random This case used passive action acquisition with exploration actions chosen randomly.



(a) Not grasping (b) Grasping (c) Above view

Figure 2: The robot is implemented in Breve; a simulator with realistic physics. The robot has a torso with a 3-dof orthogonal arm and is sitting in front of a table with a block and two floating distractor objects. The robot has three motor variables $\tilde{u}_x$, $\tilde{u}_y$ and $\tilde{u}_z$ that move the hand in the x, y, and z directions, respectively. The location of the hand is given by three time-varying continuous proprioceptive variables $\tilde{h}_x$, $\tilde{h}_y$, $\tilde{h}_z$ that represent the location of the hand in the x, y, and z directions, respectively. The relationship between the hand and the block is represented by the continuous variables $\tilde{x}_{rl}$, $\tilde{x}_{lr}$, $\tilde{y}_{tb}$, $\tilde{y}_{bt}$, and $\tilde{z}_{du}$. The variable $\tilde{x}_{rl}$ is the x value of the location of the right side of the hand in a coordinate system whose origin is centered on the left side of the block (variable $\tilde{x}_{lr}$ is analogous). The variable $\tilde{y}_{tb}$ is the y value of the location of the far (top) side of the hand in a coordinate system whose origin is centered on the bottom (near) side of the block (variable $\tilde{y}_{bt}$ is analogous). And variable $\tilde{z}_{up}$ is the z value of the location of the down side of the hand in a coordinate system whose origin is centered on the up side of the block. Additionally, the variables $\tilde{c}_x$ and $\tilde{c}_y$ represent the two-dimensional coordinates of the center of the hand in the frame of reference of the center of the block. There is also a Boolean touch variable T that is true if the block is colliding with the hand and the center of the top of the block is underneath the bottom of the hand. There are also two distractor floating objects f^1 and f^2. The variables for f^1 are $\tilde{f}_x^1$, $\tilde{f}_y^1$, and $\tilde{f}_z^1$ and the variables for f^2 are analogous. Including the direction of change variables, there are 32 variables total.

passive IAC This case used passive action acquisition with exploration actions chosen using Intelligent Adaptive Curiosity.

active random NDRC This case used active action acquisition with exploration actions chosen randomly, but with no developmental restriction on learning contingencies. This means the antecedent event of a contingency does not have to be sufficiently reliably predicted by another contingency for the contingency to be learned.

active random NDRA This case used active action acquisition with exploration actions chosen randomly, but with no developmental restriction on learning plans for actions. Thus, the agent does not have to be able to achieve the antecedent event of a contingency with sufficient capability before it can become a plan for an action.

all active random This case used active action acquisition with exploration actions chosen randomly with the change that all actions are acquired using active action acquisition.

To make the evaluation fair between active and passive action learning, during evaluation a contingency must be deterministic to be used as a plan.

6. Results

6.1 Ability to Perform Tasks

The results of the move task are shown in Figure 3. On this task passive action acquisition did better. This is likely because moving the block was sufficiently rare and using active acquisition the maximum number of plans was filled up with plans from inferior contingencies.

The results of the pickup task are shown in Figure 4. How the agent was able to do on this task largely depended on its ability to learn a sufficiently deterministic contingency. The method of **active IAC** did the best. It also had the most experience picking up the block (see Figure 7). The method of **all active random** did poorly, most likely because it spent too much time trying to move the distractor objects (see Figure 8).

6.2 Exploration Using Various Actions

We evaluated how often various exploration techniques explored different actions. Figures 5-7 show the cumulative exploratory calls to various types of actions. Figure 5 shows that Intelligent Adaptive Curiosity has the nice property of not continually ex-

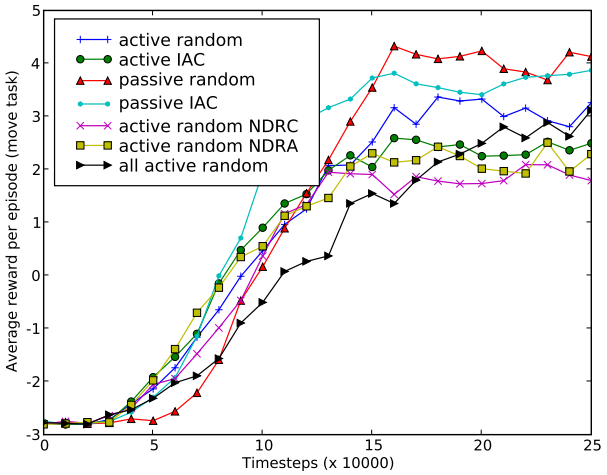


Figure 3: The agent's ability to move the block increases as it develops. Passive action acquisition outperforms active action acquisition.

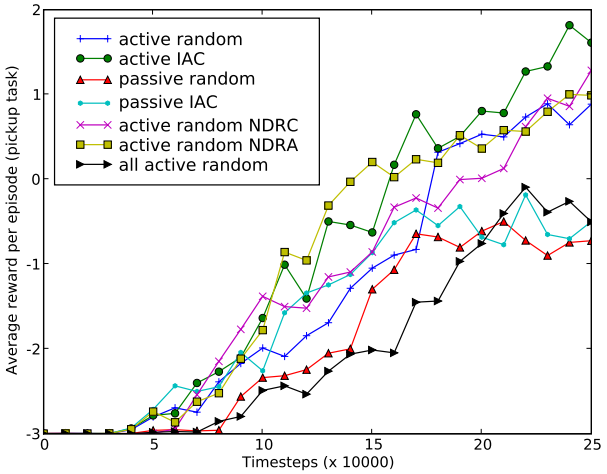


Figure 4: The agent's ability to pickup the block increases as it develops. In this case active acquisition using curiosity-based exploration performs the best.

ploring actions that the agent has already mastered. Figure 6 shows that Intelligent Adaptive Curiosity causes the agent to explore the relatively difficult action of moving the block. We see this behavior as well with Figure 7 for the case of **active IAC**. Figure 8 shows that the agent should not pursue all actions actively. In this case, **all active random** spends time trying to manipulate the distractor objects.

6.3 Developmental Restrictions

We see in Figures 3 and 4 that **active random** does about as well as **active random NDRC**, which has no developmental restriction on learning contingencies, and **active random NDRA**, which has no developmental restrictions on learning plans for actions. However, we see in Figure 9 that during the early course of the agent's development that **active random** has fewer open contingencies, and thus uses fewer resources for those contingencies.

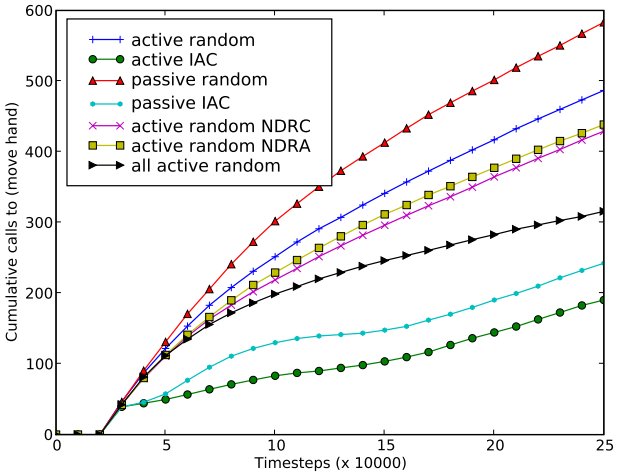


Figure 5: Exploration calls to moving the hand. The curiosity based exploration methods (**active IAC** and **passive IAC**) efficiently use exploration time by making fewer calls to this relatively easy action.

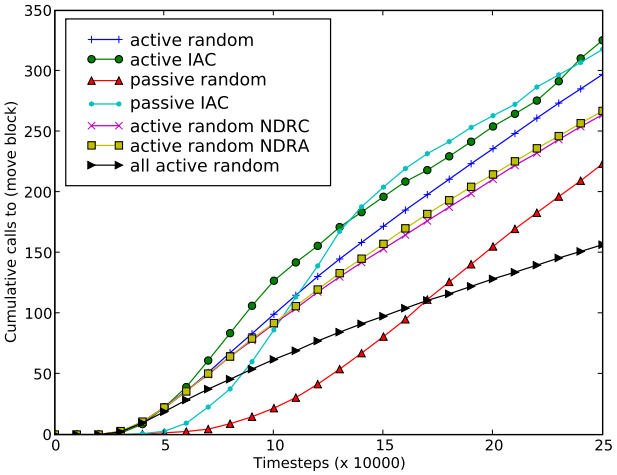


Figure 6: Cumulative exploratory calls to hit the block left, right, or forward.

6.4 Exploration Action to Subaction

When the agent first learns an action it is often called as part of exploration. An interesting part of the developmental progression is that these actions are often later called more often as subactions of other actions. We show graphs from the method **active IAC** that compare exploration calls to subaction calls. Figure 10 shows the calls for moving the hand relative to the block (c_x and c_y in Figure 2). These actions are first used more as exploration actions and then later more as subactions.

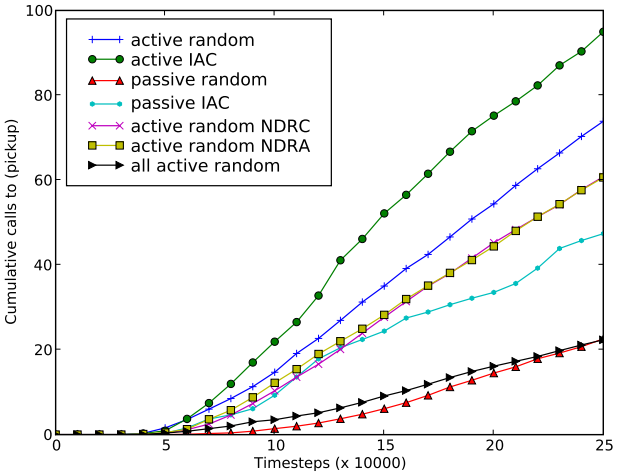


Figure 7: Cumulative exploratory calls to pickup the block. Active acquisition using curiosity-based exploration has the most calls to this complex action.

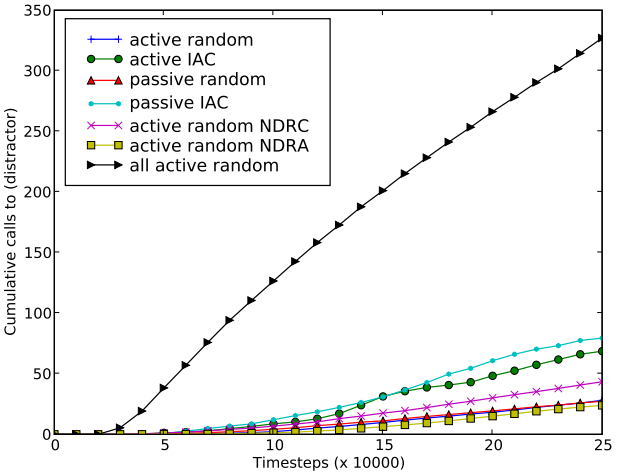


Figure 8: Cumulative exploratory calls to manipulate the floating objects. The method **all active random** has the most calls to this distractor task.

7. Conclusion

In this paper we have presented an evaluation of exploration strategies for learning actions. We found that a combination of active action acquisition and curiosity-based exploration worked best to enable an agent to develop so that it could pick up a block with a sticky mitten. However, we found that active action acquisition was detrimental to the simpler task of moving the block. This is an interesting result that warrants further investigation.

The results indicated that curiosity-based exploration enabled the agent to spend more time exploring the relatively more advanced tasks of moving the block and picking up the block, and enabled the agent to spend less time on the easily masted task of moving the hand. The results also indicated that we could add restrictions on resources without hindering learning.

There are, of course, other approaches that enable agents to learn actions. For example, Metta and Fitzpatrick (2003) focus on learning affordances (Gibson, 1979). However, the focus of QLAP is on enabling an agent to autonomously learn actions from motor primitives. The results presented here will most closely apply to models where the agent picks which action to learn during the process of autonomous development.

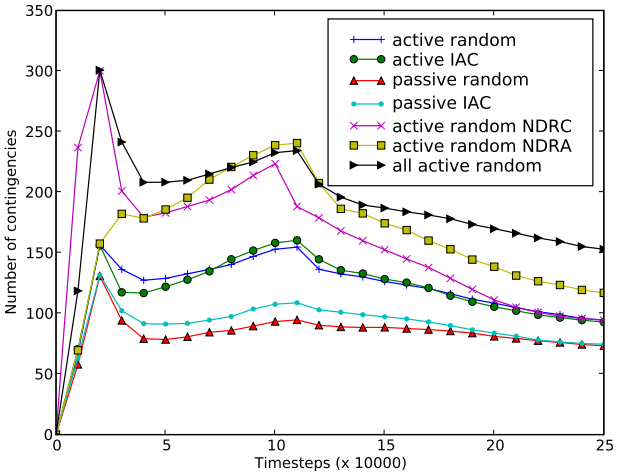


Figure 9: This graph shows that the number of contingencies does not increase without bound. We see two drops in the number of contingencies. The first drop corresponds to learning to move the hand and those actions becoming closed. The second drop corresponds to contingencies being deleted after 100,000 timesteps because they did not become plans to perform actions.

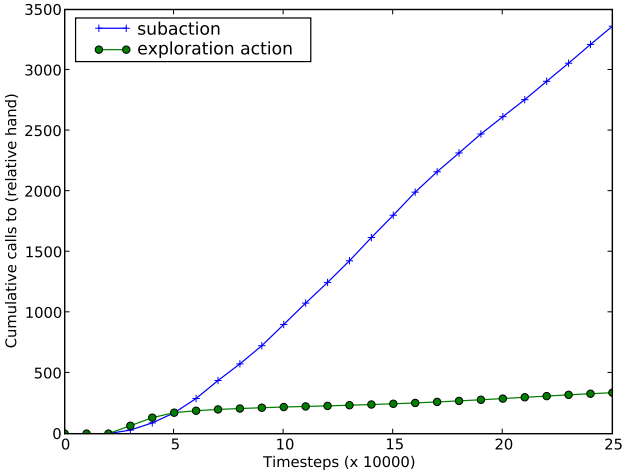


Figure 10: Action calls to moving the hand relative to the block for the method **active IAC**. This task is first called mostly as exploration and then later more as a subaction.

Acknowledgements

This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by grants from the Texas Advanced Research Program (3658-0170-2007), and from the National Science Foundation (IIS-0413257, IIS-0713150, and IIS-0750011). The authors would also like to thank Lewis Fishgold and the anonymous reviewers for helpful comments and suggestions.

References

Adolph, K. E. and Joh, A. S. (2007). Motor development: How infants get into the act. In Slater, A. and Lewis, M., (Eds.), *Introduction to infant development*. Oxford University Press.

Berlyne, D. (1965). *Structure and Direction in Thinking*. John Wiley and Sons, Inc., New York.

Bonarini, A., Lazaric, A., and Restelli, M. (2006). Incremental Skill Acquisition for Self-Motivated Learning Animats. *Lecture Notes in Computer Science*, 4095:357.

DeCasper, A. J. and Carstens, A. (1981). Contingencies of stimulation: Effects of learning and emotions in neonates. *Infant Behavior and Development*, 4:19–35.

Gergely, G. and Watson, J. (1999). Early socio-emotional development: Contingency perception and the social-biofeedback model. *Early social cognition: Understanding others in the first months of life*, pages 101–136.

Gibson, E. (1988). Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge. *Annual review of psychology*, 39(1):1–42.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, New Jersey, USA.

Huang, X. and Weng, J. (2002). Novelty and Reinforcement Learning in the Value System of Developmental Robots. *Proc. 2nd Inter. Workshop on Epigenetic Robotics*.

Klein, J. (2003). Breve: a 3d environment for the simulation of decentralized systems and artificial life. In *Proc. of the Int. Conf. on Artificial Life*.

Kuipers, B. (1994). *Qualitative Reasoning*. The MIT Press, Cambridge, Massachusetts.

Marshall, J., Blank, D., and Meeden, L. (2004). An emergent framework for self-motivation in developmental robotics. *Proc. of the 3rd Int. Conf. on Development and Learning (ICDL 2004)*.

Metta, G. and Fitzpatrick, P. (2003). Early integration of vision and manipulation. *Adaptive Behavior*, 11(2):109–128.

Mugan, J. and Kuipers, B. (2007). Learning to predict the effects of actions: Synergy between rules and landmarks. In *Proc. of the Int. Conf. on Development and Learning*.

Mugan, J. and Kuipers, B. (2008). Towards the application of reinforcement learning to undirected developmental learning. In *Proc. of the Int. Conf. on Epigenetic Robotics*.

Mugan, J. and Kuipers, B. (2009). Autonomously learning an action hierarchy using a learned qualitative state representation. In *Proc. of the Int. Joint Conf. on Artificial Intelligence*.

Needham, A., Barrett, T., and Peterman, K. (2002). A pick-me-up for infants' exploratory skills: Early simulated experiences reaching for objects using 'sticky mittens' enhances young infants' object exploration skills. *Infant Behavior and Development*, 25(3):279–295.

Oudeyer, P., Kaplan, F., and Hafner, V. (2007). Intrinsic Motivation Systems for Autonomous Mental Development. *Evolutionary Computation, IEEE Transactions on*, 11(2):265–286.

Piaget, J. (1952). *The Origins of Intelligence in Children*. Norton, New York.

Ryan, R. M. and Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25:54–67.

Schmidhuber, J. (1991). Curious model-building control systems. In *Proc. Int. Joint Conf. on Neural Networks*.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning*. MIT Press, Cambridge MA.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.