

Learning for Survival

Omid Madani¹[0009–0008–0693–0328], J. Brian Burns, Sean Kugele², Reza Eghbali³, Georgi Georgiev, and Thomas L. Dean³

¹ `omidmadani@yahoo.com`

² Rhodes College, `kugeles@rhodes.edu`

³ UC Berkeley

⁴ Emeritus, Brown University

Abstract. We propose and explore a survival model of learning perceptions and action effects *under change*. The basic goal of our agent is to extend its life as long as possible: this means, in our model, keeping its energy reserve positive. The agent deploys learning in the service of such survival. In a simplest associative task, the agent needs to learn when an action is successful, such as when to bite. An action such as <bite> incurs significant energy cost, but at times succeeds (*e.g.* due to the presence of food), yielding a substantial net gain in agent energy. Thus the agent needs to efficiently discover for an action $\mathcal{A}$ one or more rules, if any, that specify when perceptual cues indicate sufficient likelihood of a successful execution of $\mathcal{A}$. Crucially, these rules can change from time to time (possibly substantially), and the agent is to remain adaptive. We present a formalism, and develop and explore a cost-sensitive learning system, with promising preliminary experiments.

Keywords: Learning for Perceptions and Actions · Perception-Action Cycle · Online (Continual) Learning · Change · Learning Action Models · Sample Efficiency · Cost Sensitive Learning · Supervised Learning · Self-Supervised Learning · Reinforcement Learning · Bandits · Exploration *vs.* Exploitation · Survival

“Everything is impermanent. Strive on with awareness.” (Buddha’s last words)

1 Introduction

Efficiency is paramount for biological organisms. Such agents operate within diverse inter-related constraints (time, space, energy, material, safety...). We model one primitive constraint in this paper: for survival, the energy reserve of the agent is to remain positive at all times.

Further, change is ubiquitous in the life of an organism [8]: change occurs at different time scales, and some changes are permanent (wrt the life of the organism, such as climate change) while others can be temporary and exhibit certain properties such as periodicity (days and seasons). The organism also grows and develops, thus its (internal) capabilities, and needs/priorities/goals change (at some levels).

The relation between change and learning is not simple: if change is continuous and too rapid, learning is rendered impossible because of insufficient learning time. On

the other hand, (moderate) change makes continued learning and adapting necessary: if there was no change, no learning would be required, since evolution could hardwire the optimal behavior.⁵

We seek algorithms and systems supporting efficient continual learning and adaptation that, to the extent possible, can meet the types of challenges that biological agents face. The learnings of an agent should ultimately serve its needs. We break the learning into the following two broad types:

1. Learning actions [2, 21, 10], in particular, learning under which scenario(s) executing an action is likely to be effective (*e.g.* appropriate-context + <action A > leads to increase in energy-level). We develop a novel cost-sensitive rule learning technique in this paper.
2. Learning for perception (from raw low-level sensory input to constructing and detecting higher-level more predictive structures). We use online cumulative self-supervised learning of structures [14].

By executing actions, an organism can influence its environment, and thereby help meet its needs [5, 24]. However, action effects are context dependent, and the agent, ideally, ought to update such relations as necessary during its lifetime. In particular, we propose the *doctrine of impermanence*, guiding our development of agent learning techniques:

If it can be learned, it can change! (may need to be relearned)

Thus, the agent is not done once it accomplishes a task satisfactorily. The process of learning is never complete and any learned aspect of the task may need to be revisited.⁶ We seek techniques and systems that are robust to change as much as possible. Note that this promotes, among other things, modular learning systems (compared to deep feedforward neural networks). However, this ideal could clash at times with other ideals or desiderata, such as learning in a cumulative or hierarchical manner (can the foundations be changed?) and low energy consumption, and we hope to also develop a better understanding of the tradeoffs in our research on agent learning.

A major part of the context dependency of action rules is the **perceptual state** of the agent. The agent’s (raw) sensory input is in general at a low uninformative level (*e.g.* at the low level of pixels, or edges and lines, for the visual modality). However, those raw observations are inherently powerful, and can express higher level structures which can be discovered over time, via perceptual learning. We use learning of higher level structures (compositions or ngrams), in a self-supervised and information-theoretic manner [14, 13]. We assume that effective action learning can be achieved via utilizing such structures (*e.g.* via learning relatively short but accurate rules).

In this work, we begin to study the interaction of an action learning module with a developing perception system, as two separate but highly interacting subsystems in an agent. For action learning, we start by tackling associative learning, in particular, in which contexts, provided by a developing perception system, an action is likely to be successful (increase in agent energy). Our system and algorithms are preliminary, but we expect that both our survival model and our analyses of Sect. 2, and ensuing the

⁵ And indeed, there is a spectrum of possibilities that we observe in nature, from fully hardwired/innate to (mostly) learned.

⁶ For instance, assume the cost of executing an action is learned to be 10 units. We are not done (and cannot treat this as a black box, and move on to learning beyond this) as the cost can change in the future (relearning can be necessary).

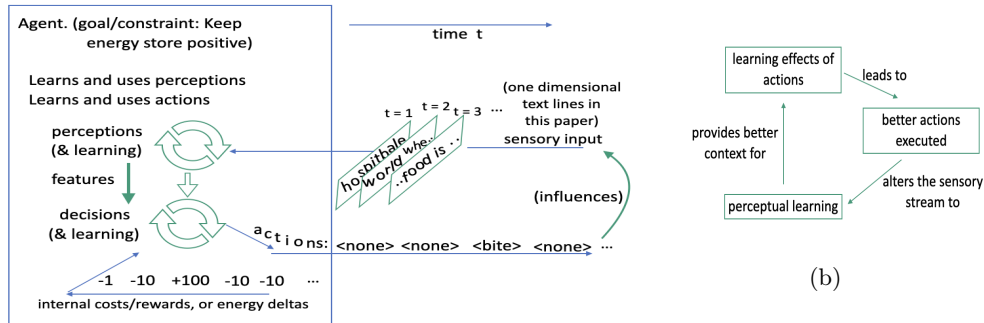


Fig. 1: (left) The agent, to keep its energy store positive, learns, in a tandem online manner, both perceptual structures, as well as action effects, in the form of rules, *i.e.* associations of perceptual structures and actions with (positive) energy change. Perceptual learning consists of making (and detecting) higher-level ngrams, which may eventually lead to more effective rules (than using lower level unigrams). (b) The relation between perceptual (conceptual) learning and action learning: the hope from perceptual learning is that, over time, it can provide more predictive structures (concepts and contexts) for learning when actions are effective. In this paper, we do not address the lower right edge (*i.e.* we have no alterations, in the wake of action executions, to the input to perceptions).

techniques are contributions to better understanding learning actions and perceptions under change (both external and internal or developmental change [15]).

Most successful work in reinforcement learning has used model-free deep neural networks that unfortunately require substantial (pre)training (biologically unrealistic), and do not work well for continual agent learning under change (deep neural networks are non-sequential learners prone to problems such as catastrophic forgetting). Our work has much in common with constructivist learning [2, 10] and action learning in animats [21, 20], but our emphasis on change and separation of perception (concept) learning from action learning sets us apart (see discussions in Sect. 2 and related work, Sect. 6). We begin by describing the agent’s action-learning task next, then present an analysis of (associative) rule learning, in particular when exploration is necessary (Sect. 2.3). We then describe the perception (Sect. 3) and action (rule) learning and use (decision making) modules (Sect. 4), followed by brief experiments (Sect. 5). We conclude with related work and summary.

2 The Setting: Change and Survival

The learnings of a biological organism should ultimately serve its survival needs (constraints). A constraint can be to keep the water amount in the body within a certain lower and upper limits, or that the energy level should always remain positive (otherwise, death). The agent continually meets these constraints via executing actions. We focus on the following two inter-related learning tasks to enable effective action execution (Fig. 1):

1. Learning to predict when executing an action is successful (a subtask of the more general problem of learning action effects) .
2. Learning perceptual structures (to provide better contexts for action learning).

In our approach, we have separated the learning in the perceptual stream(s) (*e.g.* learning perceptual concepts) from the learning of action effects, though they affect one another (Fig. 1). Humans (and other animals) appear to learn much about their world from pure observations (no overt actions), and researchers have noted that actions are costly (take time/energy) and the reward obtained (the need met) is probably too sparse to provide the feedback required for developing deep and vast conceptual structures. Overt actions are of course important in determining what is fed to the perceptual system (the input stream, *e.g.* via change of head direction or moving and focusing the eyes, and so on). We also note that much work in machine learning (unsupervised learning such as clustering, and Bayesian and self-supervised techniques) have the potential to impact perceptual learning.

A major component of biological life is change. Change can be external (eg habitat and seasonal changes) or internal to the agent (addition or loss of capabilities, and normal developmental non-stationarities) (see Sect. 3).⁷

2.1 Cost-Sensitive Learning of (Associative) Action Effects

Time is discrete, and we assume the (primitive) actions available are also discrete and comprise a finite set, and they take the same amount, unit time, to execute and complete.⁸ The agent can perform only one action at a time. We often focus on two actions in this paper, for simplicity: `<none>` (do nothing) and `<bite>` (Fig. 1).

In general, actions have a variety of costs: time cost, energy cost, opportunity cost, and so on. In our simple model, the agent has an energy reserve E , and executing any action incurs immediate positive (energy) cost. We use unit cost of 1 for `<none>`, and a cost of $c = 10$ for others, such as `<bite>` (see Sect. 2.4). We assume `<bite>` can lead to energy increase (when food is present), but `<none>` never does (although the agent may have to discover and always update such information for every action; see Sect. 2.2). Certain actions, in certain contexts, can lead to a reward, *i.e.* a net increase in the energy. Thus, the agent’s goal is to keep alive, which, in our modeling, means keeping its energy (reserve) positive as long as possible, by judiciously executing actions. We assume no upper capacity limit on the energy reserve.

Better decision making is achieved in part by developing one’s perceptions (Fig. 1). At every time tick, or episode, the agent does its perception and then decides and takes an action. By perceiving we mean chunking and mapping the given raw sense data, a low level character string in our system, into internal structures (words or ngrams), or

⁷ An example of what could be viewed as internal change is a right handed person losing her right hand. She needs to now learn to cope using only her left hand, to the extent feasible and useful, to accomplish the tasks previously achieved by her right hand alone or both hands together. The distinction between internal and external is not a hard one, *i.e.* internal to what entity? In general an agent consists of multiple parts performing distributed computation and adaptation, and a change in one part is considered external to other parts needing to adapt.

⁸ Even a seemingly simple action such as `<eat>` is a composite complex action and can involve multiple bitings, chewings and swallowings, in addition to appropriate sensings that trigger those actions (sub-tasks) in appropriate order.

concepts.⁹ We assume perceiving has no cost (but has to complete within the episode, before action execution).

In this work, we only consider learning the immediate effect of an action on the agent’s energy reserve, and we use the immediate perceptions (perceptions derived in the current episode) to do so (previous perceptions and actions are ignored). We do not consider for instance motion actions that do not directly impact internal needs, but do impact the senses reporting from outside the body (such as visual senses & perceptions). Note that some animals are mostly stationary (sea anemone and sea urchins), and the associative problem we study is also a highly simplified form of pouncing or ambush behavior in predation.¹⁰

2.2 Assumptions/Biases: Initial Knowledge on One’s Action Set

We have implied that the `<none>` action could be just like other actions. The following distinction regarding the agent’s associative learning problem is useful:

1. (less general: more bias) The agent knows that `<bite>` can increase energy (despite its high cost), while `<none>` (do nothing) can never increase energy. It only needs to learn and relearn the context(s) that make an action like `<bite>` successful.
2. (more general: no differentiation) The agent only knows its action set: that it can execute each to figure out the (possibly non-stationary) costs and rewards, as a function of context, over time.

A motivating principle behind our work is that *any learned (acquired) knowledge can change* (impermanence). Thus costs/rewards of any action can change under the second assumption: an action that had never yielded a reward before (such as `<none>`) can become productive at some point. Thus, under the second setting (fewer assumptions), the agent from time to time, specially when it doesn’t have an assuring rule (see Sect. 2.5), may repeat an action like `<none>` just to keep estimating its return probability (and not just for the purposes of conserving energy). This exploration may seem counter-productive and wasting valuable time (and samples and energy), and relates to what biases and assumptions we (or the agent) are willing to make. In an engineering setting, this problem choice is application dependent. As another example, and beyond associative learning, the agent may or may not know that certain actions are for motion (for moving its whole body in space), and can never directly lead to energy increase. In general, the primitive actions set may not be uniform, and we may be willing to make certain assumptions (or tolerate ‘biases’) about what the agent knows about certain action subsets (action families). We assume the first setting in the rest of the paper (and focus on rule learning): the agent is deciding between one or more actions and knows it has the option of doing nothing (with unit cost and no reward).

⁹ Another task often considered as part of perception is what could be referred to as *judgment*, *i.e.* assigning values to the concepts that are active, such as danger/threat level or foodiness (or bad, good, ...). Our action learning component addresses some of the value assignment/judgment task. We treat perception as pure conceptual analysis (no judgment).

¹⁰ Under the right timing (perceptual situation), an action could be effective, and otherwise, it is opportunity lost. See ambush predation: https://en.wikipedia.org/wiki/Ambush_predator

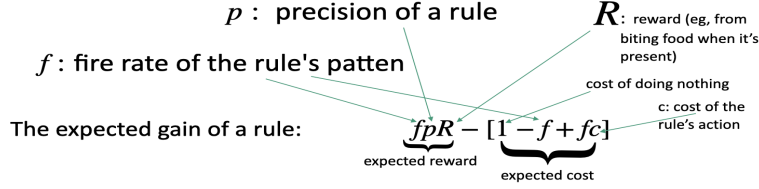


Fig. 2: The expected gain of a rule. The higher the gain, the better: good positive gain rules can prolong life. If the agent has no such, death is foreseeable.

2.3 Triggers (Logical Conjunctions) and Rules

A **trigger** $\mathcal{P}$ (a perceptual pattern) is a conjunction of ngrams in this paper.¹¹ A **rule** is comprised of a trigger $\mathcal{P}$ and an action $\mathcal{A}$, shown as $\mathcal{P} \rightarrow \mathcal{A}$, with the semantics that when its trigger holds true (its context is active), the rule recommends executing its action $\mathcal{A}$ (and otherwise, recommends $\langle \text{none} \rangle$). In this paper a trigger is a conjunction of boolean variables. Each conjunct (boolean variable) corresponds to whether or not an ngram is present (active) in the current episode (Sect. 3).

A trigger has a **fire-rate** f : how often it holds (the fraction of episodes it evaluates to true) (a probability in $[0, 1]$). Whenever the energy is increased, assume it is by a fixed amount R (e.g. reward from a single food type). So we assume an action either succeeds or not (boolean: food is either present or not), and if success, the reward R is obtained. In either scenario of fail or success, the cost c of the action is paid. For this analysis, we assume R and c are the same for all actions except the $\langle \text{none} \rangle$ action, which has unit cost. Then, corresponding to a trigger $\mathcal{P}$ and an action $\mathcal{A}$ (and also the rule $\mathcal{P} \rightarrow \mathcal{A}$) there is a **precision** quantity p (a probability): how often energy is increased (by R), if the agent performs $\mathcal{A}$ given trigger $\mathcal{P}$ is true. In terms of conditional probability, we can write $p := \text{precision}(\mathcal{P} \rightarrow \mathcal{A}) := P(\text{food} \mid \mathcal{P} \text{ and action } \mathcal{A} \text{ is executed})$. For example, a rule such as 'if apple-tree then $\langle \text{bite} \rangle$ ' may have a fire rate of 0.2 and precision of 0.8 initially. If the season changes, and the trees bear no more fruit, the precision may go down to say 0. However, due to say fire, if the trees mostly die, the fire-rate goes down. Fig. 2 provides a summary of the quantities. Our agent keeps online estimates of the fire-rates and precisions for the promising triggers and rules that it keeps track of [15]. A special trigger is the **always-true** trigger, i.e. an empty conjunction, which has a fire-rate of 1.0. The **always-bite** rule uses the always-true trigger (empty context) and the $\langle \text{bite} \rangle$ action (always-true $\rightarrow \langle \text{bite} \rangle$). This rule is useful in easy worlds and initially our agent begins with such rules, for every action (whenever new rules need to be discovered).

The Gain of a Rule (Utility). Assuming unit cost for doing nothing, the expected gain (in energy) of a rule takes a simple form, when estimates of its precision p and fire-rate f are available. The expected cost is $(1 - f)1 + fc$, and the expected reward is $f(pR + (1 - p)0) + (1 - f)0 = fpR$. Therefore, the (expected) gain of a rule (per time tick) is:

$$\text{gain} = fpR - (1 - f + fc) \quad (\text{see also Fig. 2}) \quad (1)$$

¹¹ n-grams are themselves (spatiotemporal) conjunctions of unigrams, but these are learned inside the perception system [14] (and see Sect. 3). Note: we don't support negation in the trigger conjunctions.

However, the agent preferably seeks a rule with a sufficiently large positive gain, otherwise the chance of death could remain too high.

2.4 Different Worlds

In general, the evolution of the agent's energy reserve $E^{(t)}$ is a random walk, that starts with initial reserve $E^{(0)} > 0$, and that would end if $E^{(t)}$ ever hits 0. The agent strives to extend this walk by making it more and more biased towards larger E , via continued discovery of good rules and better rules, as well as rule maintenance (updates). Different agents face different worlds, and the same agent faces a sequence of worlds in general (*i.e.* changes in environmental parameters, tasks, ...). An example of an easy world is one when the always-bite rule is sufficient (no learning is required). The analogy is to breathing and the ubiquity of oxygen for humans in normal circumstances. Let **food-rate** p_{food} be the probability that food is present, or equivalently the precision of the always-bite rule: $p_{food} = P(\text{food} \mid \langle \text{bite} \rangle)$. An easy world is when:

$$\frac{c}{R} \ll p_{food} \quad (\text{and adequate initial energy: } R \ll E^{(0)}) \quad (2)$$

A *hospitable (learning-friendly)* world is one where learning is feasible (but may not be guaranteed). To make things interesting for learning (not too easy, nor impossible), in general we need to have (to provide sufficient time for learning), thus:

$$1 \ll c \ll R \ll E^{(0)} \quad (3)$$

That is, when the agent enters into a new hospitable world, its initial energy reserve should be sufficiently high, so that good rule(s), if they exist, are likely discovered.

Fig. 3 shows what we'd expect optimistically: If there is an assuring rule in terms of the structures that perception eventually provides, given sufficient time, it is discovered by exploration and rule-learning. There is initially a steep cost, *i.e.* fast reduction in energy, due to exploring and executing costly actions (*e.g.* $\langle \text{bite} \rangle$ *vs.* doing nothing), but eventually the exploration pays off, an assuring rule(s) is/are discovered, and energy level climbs.

2.5 Exploration: When to Look for a Better Rule

The agent should keep track of the fire-rates and precisions (which can change from time to time), and to continue to look for a better rule as long as for the best rule discovered so far, the agent continues to have negative or weak gain:

$$p < \frac{c + f^{-1} - 1}{R} \quad (\text{needs to learn better rules}) \quad (4)$$

Or roughly as long as $p \leq R^{-1}(c + f^{-1})$. For instance, with $c = 10$, and $f = 0.1$, if $R = 100$, then as long as $p \leq \frac{20}{100} = 0.2$, the rule does not have positive gain. Since avoiding death is highest priority, the agent needs to require more of its rules. A rule is expected to fire every f^{-1} time ticks, but there is some chance it may not. Further, food is present only a fraction p of such cases. There can be unlucky stretches, multiples of $(pf)^{-1}$, that the trigger is not active or the execution of the action does not bear energy. To be safe, the agent should keep some exploration, for better rules, as long as $p \leq mR^{-1}(c + f^{-1})$ (*i.e.* some multiple m , *e.g.* $m = 5$). A rule with $p \geq mR^{-1}(c + f^{-1})$

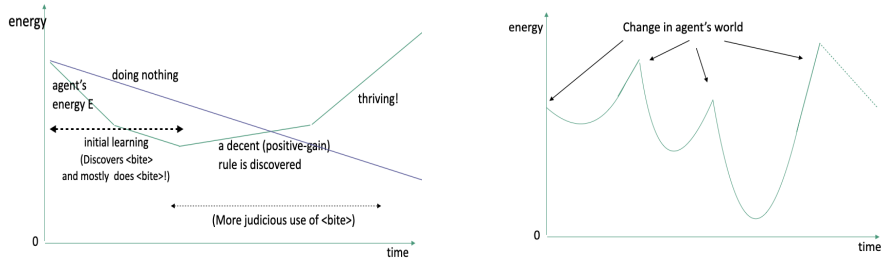


Fig. 3: Left: A plausible evolution of the agent’s energy curve: In a hospitable (learnable) world, there is adequate time to learn (discover $\langle \text{bite} \rangle$, and then when to $\langle \text{bite} \rangle$). Doing nothing initially looks better than costly learning, but leads to slow certain death. Exercising more expensive actions and learning is more beneficial in hospitable (learnable) worlds. Right: the agent’s world can change several times (concatenations of left), and ideally, the agent can adapt in a timely fashion each time.

is called a **assuring rule** (in the previous example, if $f = 0.1$, then, for safety, p should be 1, if we have set $m = 5$).

Note that estimating a fire-rate f is easier than estimating p : it does not require executing a costly action (it is action and thus rule independent), while a precision estimate requires execution of an action, and only the precision value for the active triggers, those that evaluate to true, and under the rule that was followed (if any), should be updated (Sect. 4). The above analysis informs our rule learning. We call our system CLAP, for Cost-sensitive Learning of Actions via Perceptions (Fig. 4). We describe its two components next.

3 Perception: Self-Supervised Feature Extraction

We use Expedition, developed in the Prediction Games framework [13, 14], for the perceptual learning system. In our context, it can be characterized as “learning for the sake of learning”: The structure (concept) discovery is driven by improving the predictions of one another: each concept (ngram) is *both a predictor as well as a predictand* (target of prediction). The raw senses originate from outside the agent (*e.g.* from the audio or visual channels) and do not necessarily reflect internal needs. However, we do expect that some of the higher level structures discovered (*e.g.* the sight of an apple, or a lion) can serve as better predictors (than raw pixels) of how internal needs (food, safety) could be eventually met. This is an important learning bias in our approach.

The systems begins at a low level, a relatively small alphabet of *discrete symbols*, and acquires and detects higher level ngrams (combinations) over time, via practicing *interpretations* [14]. An information theoretic score guides the learning and use of the concepts (ngrams) in interpretations [13]. Note that ngrams correspond to spatiotemporal conjunctions. This work has been on one-dimensional text (1D), though the expectation is that the techniques can generalize to 2D and beyond (*e.g.* images and video). The framework allows for approximate matching and noise in episodes.

Starting at the unigrams (character) level, the agent gradually acquires bigrams and higher order ngrams (not in a strict order, *i.e.* some bigrams are learned after trigrams ...). On each episode, the results of the interpretation, the **active ngrams (features)**,

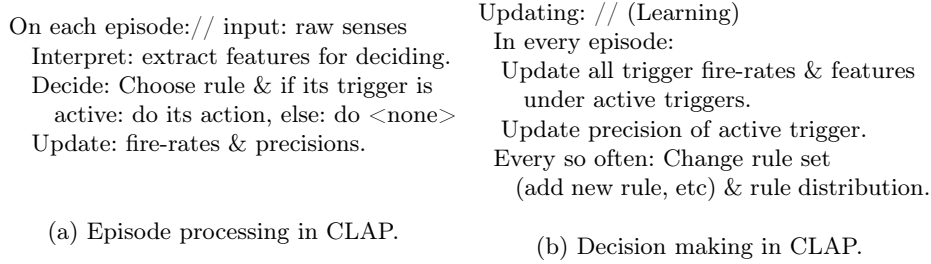


Fig. 4: CLAP at a high level. See Sections 3 (interpretation and feature extraction) and 4 (for rule learning and usage).

is provided to the decision making module (next). For instance, if the episode is the string 'new york', and if the concept 'new york' is in the system and participated in the interpretation, the ngram 'new york', as well as its parts, such as the bigrams 'ne' and 'yo', as well as all the unigrams ('n', 'e', 'w', ' ', ...), are all considered active and are provided for rule learning and use. Note that what the perceptual module provides the action module changes and evolves (develops) over time, for instance from all unigrams to bigrams, trigrams and so on (a form of internal non-stationarity).

4 Decisions: Rule Use and Rule Learning

CLAP is informed by the analysis of the gain related quantities described in Sect. 2.3. The perception module (Sect. 3) provides the ngrams (initially unigrams) or the (active) features that are evaluated over time, and possibly combined, to find positive-gain rules. The rule learning technique grows a conjunctions of features, *i.e.* triggers, and makes new rules (Sect. 2.3). The semantics for **action execution by following a rule** is that when a rule is selected and its trigger is active (evaluates to true) in current episode, then the action is executed and otherwise (inactive trigger), the agent does nothing (<none> is executed, *i.e.* no other rule is tried). Thus, a rule provides a *complete course of action*. We next describe the different parts of CLAP (Fig. 4).

The perception-action cycle. In each episode, after interpretation (getting the active features), the algorithm selects a rule and executes according to it. Then the agent updates the fire-rates and precisions. Every so often, the agent updates the rule selection distribution and may add new triggers and rules or remove some.

Selecting a rule. The agent randomly samples according to a softmax on the gains of current rules: if there are positive-gain rules, it only uses softmax over those. Recomputing probability assignment to rules via softmax is performed every so often. The agent always has some rules to pick from: the agent begins with empty trigger rules, *i.e.* empty conjunctions (trigger) paired with every possible action (*e.g.* 10 actions, there are 10 rules to choose from initially, and uniformly at random).

Updating the rates. The fire-rate of *all triggers* (whether or not active) is updated. In addition, for each active trigger, all active features are added under them, if not already there (akin to a tree, allowing for conditional updates), and then the fire rate of every feature under an active trigger is updated as well. The precision of only the selected rule is updated and only when its trigger is active: when active, the precision is updated according to whether reward was increased (food was found). Furthermore, and with an active trigger, the precision of all active features under the rule

is also updated. It can be verified that we preserve correct semantics with this manner of updating. We use sparse moving averages for the updates [15].

Adding new rules. Every so often (100 episodes), the gain of features under current rules are computed (via feature’s precision and combined fire rate), and if the gain of a feature is above the current average gain (a moving average), a new rule is created and added to list of rules (by combining the feature with the parent trigger).

Brief Discussion. Even if a good rule is discovered, a better rule could be discovered by refining the existing rule via a feature (extending the conjunctive trigger and improving precision). Our next step (not implemented) is to support rule change, *e.g.* use an upper-estimate for rule gains, instead of the plain mean, as well as *aging estimates* (*e.g.* via time-stamps [15]), in order to support exploration when needed.

5 Experiments

For these preliminary experiments we embed a true underlying rule (often a single conjunction, a *trigger* word), with a certain fire-rate, and observe whether the agent discovers that rule or a close approximate sufficiently early, and in particular for how long the agent survives (does the agent’s energy keep climbing after an initial explorative decline?) (Fig. 3). Each episode is a randomly picked line of text, from the NSF abstracts corpus (2 million lines) [3]. Each line is on average 55 characters long, and a character set size of ≈ 100 . The number of concepts that Expedition generates (either primitives corresponding to single characters, or higher level ngrams) grows from just over 60, to over 1k (*i.e.* 1000) by episode 2k, to over 5k by episode 15k.

The user specifies the experimental parameters, *e.g.* the number of actions (10 by default). <none> has cost 1, others have cost 10. One action, <bite>, can yield a reward depending on context (others, such as movement-related actions, only incur cost). We begin the agent with an initial energy (20000), and a food-reward R ranging in 10s to 100s in our experiments. We specify the rough desired fire-rate of the underlying ground-truth rule, with a trigger word or two, (for when food is present). Words such as ‘the’, ‘are’, and ‘be’ are frequent and have a fire-rate ≈ 0.1 or higher. Words such as ‘these’, ‘as’, ‘research’, and ‘project’, have fire-rates ≈ 0.05 or somewhat lower. We specify a desired fire-rate, *eg* 0.05 or 0.1, and a random ground-truth rule is created.

Findings. The agent transitions from trying all actions initially to trying mostly <bite> (the action leading to reward), and after a few 100s of episodes or so, further transitions to a judicious execution of <bite> (as more precise rules are discovered). The agent begins to focus on promising rules early, *e.g.* if the true rule is “that” $\rightarrow$ <bite> (the target word is “that” with fire-rate 0.07), CLAP soon sees “th” as promising. It eventually, by episode 4k, discovers “that” with fire-rate 0.05 and precision of 1.0. R is set to 100 and 300 in these experiments. The energy level goes down in the first few 100s of episodes to 19k, but climbs afterwards (as in Fig. 3). In the above experiments, agent’s energy level keeps climbing, to over 30k, 40k and so on (once an assuring rule is found). If R is lowered to 50, then the average gain of the true rule is low compared to R in 100s and there is little or no climb in the energy level once such a rule is discovered. For instance, with “that” as the trigger word, in one run, the initial energy level goes down to below 16k, but once “that” is discovered, the energy begins its slow climb (compared to previous experiments), to 21k at episode 13k. When R is further lowered to 20, with trigger “on” (fire-rate 0.06), the agent eventually discovers “on” (fire-rate 0.03), as well as later “on the”, but these are insufficient for a positive gain, and the energy continues its gradual slow decrease to 0. Increasing the fire-rate

of the underlying rule (eg to > 0.1), makes the problem easy (even an empty conjunction may work well), while increasing the number of actions (*e.g.* to 50, with only one leading to reward) predictably delays the discovery of a good action and a good rule.

6 Related Work

The present work continues our recent work on agentic learning (navigation under change) [16]. We strive for ‘agent friendly’ learning, *i.e.* learning and development that occurs continually within the agent’s lifetime. Change is a major theme: it is both ubiquitous in the lives of biological organisms (within limits), and is a good vehicle to test the sample efficiency and flexibility of various techniques. Our navigation work made a number of assumptions about what the agent knows or what capabilities are hardwired [16], in particular knowledge of barrier objects for remembering and planning routes. This work is a step towards learning actions combined with richer perception.

Model-free RL with deep networks is very powerful in terms of learning arbitrary functions and requiring little knowledge about the available actions (high flexibility), but this is achieved mostly offline, and sample efficiency and change is considerable challenge. Change, in machine learning and RL, continues to be studied and remains a challenge, and subfields such as continual learning, transfer learning, distribution shift, meta learning, lifelong learning, lifelong RL, and open-ended learning in robotics attempt to address the various aspects of the problem [23, 19, 1, 22, 18]. This work is also related to and informed by cost-sensitive (supervised) learning, active learning, and multiarmed bandits (*e.g.* [4, 25, 12]).

Our work shares much in common with the learning animats work [21]: they model different needs and consider composite actions. Ours separates perceptual learning from action learning, as its own module, and aims for richer perceptual structures. For action learning, we propose cost-sensitivity to inform exploration and exploitation, and analyze a simple model of rule utility. The constructivist approach of Drescher [2] is also solely based on learning via actions. Our work, with its focus on supporting change, takes a *satisficing orientation*, rather than an optimization formulation, toward action learning, consistent with [17, 7] (and in contrast with much work in RL and machine learning). The combination of two modules learning in tandem can be viewed as a simple cognitive architecture [6, 11, 9], and we anticipate that a number of advances need to be made, from the conceptualizing problems down to devising effective techniques, to get a good picture of the space of architectures for autonomous open-ended agents.

7 Summary and Future Directions

We proposed and explored a model of cost-sensitive learning of actions, in particular determining when an action is likely to be successful in increasing one’s energy, as a function of a developing (changing) perceptual context. The approach naturally motivates exploration when it is detected that the existing rule(s) have stopped performing well. Our next step is to complete an implementation to support change in underlying rules. We presented preliminary experiments that show the potential of the approach. We hope to extend the approach beyond the current limitations, such as learning disjunctions of conjunctions, and to further understand and advance the interaction between perception and action as we enrich the perceptual part of the system. In particular, learning how certain actions, such as bodily movements, can affect perception, is another important future direction.

References

1. Botvinick, M., Ritter, S., Wang, J.X., Kurth-Nelson, Z., Blundell, C., Hassabis, D.: Reinforcement learning, fast and slow. *Trends in cognitive sciences* (2019)
2. Drescher, G.L.: *Made-up Minds - a constructivist approach to artificial intelligence*. MIT Press (1991)
3. Dua, D., Graff, C.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2017), <http://archive.ics.uci.edu/ml>, <https://archive.ics.uci.edu/dataset/134/nsf+research+award+abstracts+1990+2003>
4. Elkan, C.: The foundations of cost-sensitive learning. In: *IJCAI* (2001)
5. Franklin, S., Graesser, A.C.: Is it an agent, or just a program?: A taxonomy for autonomous agents. In: *ATAL* (1996)
6. Franklin, S., Madl, T., Strain, S., Faghihi, U., Dong, D., Kugele, S., Snaider, J., Agrawal, P., Chen, S.: A LIDA cognitive model tutorial. In: *Biologically Inspired Cognitive Architectures* (2016)
7. Gigerenzer, G., Selten, R.: *Bounded Rationality*. The MIT Press (2002)
8. Ingold, T.: *The Perception of the Environment: Essays on Livelihood, Dwelling, and Skill*. Routledge (2000), (See Chapter 11 on the temporality of the landscape.)
9. Kotseruba, I., Tsotsos, J.: *The Computational Evolution of Cognitive Architectures* (2025)
10. Kugele, S.: *Constructivist procedural learning for grounded cognitive agents*. Cognitive Systems Research (2025)
11. Langley, P., Laird, J.E., Rogers, S.: *Cognitive architectures: Research issues and challenges*. Cognitive Systems Research (2009)
12. Lizotte, D.J., Madani, O., Greiner, R.: Budgeted learning of naive-bayes classifiers. In: *Uncertainty in AI (UAI)* (2003)
13. Madani, O.: An information theoretic score for learning hierarchical concepts. *Frontiers in Computational Neuroscience* **17** (2023)
14. Madani, O.: Towards understanding and developing open-ended intelligences for infinite worlds. *Bio. Inspired Cognitive Architectures (BICA)* (2025)
15. Madani, O.: Lifelong open-ended probability predictors. *TMLR* (2026)
16. Madani, O., Burns, B., Eghbali, R., Dean, T.: When remembering and planning are worth it: Navigating under change. *ArXiv* (2026), (short version, *BICA* 2025)
17. Mangalam, M.: The myth of optimality in human movement science. *Neurosci Biobehav Rev.* (2025)
18. Santucci, V., Oudeyer, P., Barto, A., Baldassarre, G.: Editorial: Intrinsically motivated open-ended learning in auton. robots. *Front. in Neurorobotics* (2020)
19. Schaul, T., Hasselt, H.V., Modayil, J., White, M., White, A., Bacon, P., Harb, J., Mourad, S., Bellemare, M.G., Precup, D.: The Barbados 2018 list of open issues in continual learning. *ArXiv* (2018)
20. Strannegård, C., Engsner, N., Ulfbäcker, S., Andreasson, S., Endler, J., Nordgren, A.: Survival games for humans and machines. *Cognitive Systems Research* (2024)
21. Strannegård, C., Svängård, N., Lindström, D., Bach, J., Steunebrink, B.R.: Learning and decision-making in artificial animals. *Journal of AGI* (2018)
22. Sutton, R.S., Bowling, M., Pilarski, P.M.: Alberta plan for AI research. *ArXiv* (2022)
23. Thrun, S., Mitchell, T.M.: Lifelong robot learning. *Robotics Auton. Syst.* (1993)
24. Wilson, S.W.: Knowledge growth in an artificial animal. In: *Intrn. Conf. on Genetic Algorithms* (1985)
25. Zhou, L.: A survey on contextual multi-armed bandits (2016)