Accurate and Interpretable Representations of Environments with Anticipatory Learning Classifier Systems

Romain Orhand[1,2], Anne Jeannin-Girardon[1,2], Pierre Parrend[1,3], and Pierre Collet[1,2]
{rorhand, anne.jeannin, pierre.parrend, pierre.collet}@unistra.fr

[1] Icube Laboratory - UMR 7357, 300 bd Sébastien Brant, F-67412, Illkirch, France
[2] University of Strasbourg, 4 rue Blaise Pascal, F-67081, Strasbourg, France
[3] EPITA, 14-16 Rue Voltaire, F-94270 Le Kremlin-Bicêtre, France

Abstract. Anticipatory Learning Classifier Systems (ALCS) are rule-based machine learning algorithms that can simultaneously develop a complete representation of their environment and a decision policy based on this representation to solve their learning tasks. This paper introduces BEACS (Behavioral Enhanced Anticipatory Classifier System) in order to handle non-deterministic partially observable environments and to allow users to better understand the environmental representations issued by the system. BEACS is an ALCS that enhances and merges Probability-Enhanced Predictions and Behavioral Sequences approaches used in ALCS to handle such environments. The Probability-Enhanced Predictions consist in enabling the anticipation of several states, while the Behavioral Sequences permits the construction of sequences of actions. The capabilities of BEACS have been studied on a thorough benchmark of 23 mazes and the results show that BEACS can handle different kinds of non-determinism in partially observable environments, while describing completely and more accurately such environments. BEACS thus provides explanatory insights about created decision policies and environmental representations.

Keywords: Anticipatory Learning Classifier System, Machine Learning, Explainability, Non-determinism, Building Knowledge

1 Introduction

Explainability has now become an important concern for automated decision making in domains such as healthcare, justice, employment or credit scoring, among others. Deep Learning models are widely used in these domains, because of their ability to extract relevant features from complex data. However, they are not intrinsically explainable and require the use of *post-hoc* models to shed light on their decisions. Models exist that are, by design, more explainable, but at the cost of reduced performance in solving tasks: performance is thus balanced with explainability. The approach developed in this paper aims at enhancing the

performance of such models *while* enabling them to provide more explanatory elements regarding their decisions.

We are interested in Reinforcement Learning models. Among intrinsically explainable Reinforcement Learning models, Anticipatory Learning Classifier Systems (ALCS) are rule-based machine learning algorithms that are based on the cognitive mechanism of Anticipatory Behavioral Control [19]. ALCS build their population of rules (called classifiers) by comparing successive perceptions of their environment in {`conditions-actions-effects`} tuples [11]: ALCS try to *anticipate* the consequences of an action according to their environmental situations. ALCS do not depend on a stochastic process to learn new tasks: Anticipatory Behavioral Control enables them to learn new tasks immediately, from the anticipation they built, giving insights to explain the use of the classifiers created by the system.

The work presented in this paper focuses on the ability of ALCS to *deal with non-deterministic environments* used in reinforcement learning problems, while making the classifiers of ALCS more explainable. Non-determinism can take different forms [16]: perceptual sensors can have irrelevant random attributes, be noisy or insufficient to determine the exact state of the environment (referred to as the *Perceptual Aliasing Issue*); the results of actions can be uncertain; rewards from the environment can be noisy. In particular, non-deterministic properties of the perceptual sensors or regarding the results of actions bring about *aliased states*, which are environmental states related to these forms of non-determinism. These aliased states prevent ALCS from achieving their task, if they cannot detect such states and build appropriate classifiers to deal with them.

This paper introduces BEACS (Behavioral Enhanced Anticipatory Classifier System) which aims to strengthen both the performance and the explainability of ALCS in non-deterministic, partially observable environments. The objective of BEACS is to build complete and accurate representations of its environment (*via* its population of classifiers), regardless of the non-deterministic properties of the environment, while being able to efficiently solve the task learned.

In section 2, the main principles of Anticipatory Learning Classifier Systems are presented along with the mechanisms allowing them to handle non-determinism. After an analysis of these principles and mechanisms, BEACS is introduced in section 3. Section 4 presents a study of the capabilities of BEACS through a thorough benchmarking on the different mazes used as test-beds in the literature. The results achieved by BEACS are discussed in section 5, before concluding in section 6.

2 Related Works

2.1 Principles of ALCS

As illustrated in figure 1, ALCS classifiers are mainly made of a $\{C, A, E\}$ tuple (consisting of a condition component C, an action component A and an effect

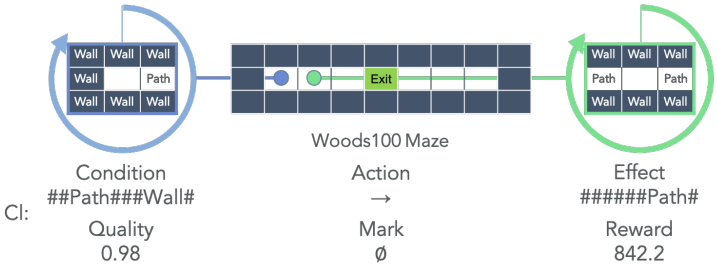

Fig. 1: Illustration of a classifier Cl of an ALCS in a maze environment, if the provided observations are the eight squares adjacent to each position starting from the North and clockwise. The hash is a wildcard that corresponds to all possible items in the condition and indicates there are no changes in the effect.

component E), a mark that specifies states for which the classifier has failed to anticipate, a measurement of the quality of anticipation q, and lastly, a prediction of the expected reward r [18].

Assessment of anticipation quality, and prediction of the expected rewards of classifiers are respectively done using the Anticipatory Learning Process and Reinforcement Learning.

Hence, the Anticipatory Learning Process is used to discover association patterns within the environment by means of the $\{C, A, E\}$ tuples. These tuples are built by comparing perceptions retrieved successively. ALCS classifiers are created or updated based on the differences between these perceptions. Reinforcement Learning is used to compute the expected rewards in order to fit the classifiers to the task being learned, and provide more adaptive capabilities to ALCS.

ALCS manage their population of classifiers by looping between perceiving new sensory inputs, evolving their population of classifiers thanks to both learning paradigms, and interacting with the environment by performing an action from the set of classifiers matching the current perception (for further details, refer to [6]).

ACS2 [6] is an enhanced version of Stolzmann's Anticipatory Classifier System (which was the first ALCS) [18]. It includes a genetic generalization mechanism to remove specialized attributes in the condition components of classifiers [2]. ACS2 also includes new action selection policies [3] to improve the evolution of the most general and accurate classifiers. Different exploration strategies of ACS2 were compared in [12] and an action planning mechanism was added to this model [21] in order to speed up the learning process. [13] replaced the reinforcement component of ACS2 by a mechanism that maximizes the averaged rewards, while [4] used a learning classifier system dedicated to approximating the optimal state values of the transitions learned between states of the environment. In parallel, two other ALCS were developed: YACS implements different heuristics from ACS2 to focus on the most relevant attributes in the condition and effect components of classifiers, as these attributes could be uncorrelated

Woods100 Maze

	Condition	Action	Effect
Cl_1:	##Path###Wall#	→→	##Exit###Path#
Cl_2:	##Path###Path#	→	##{Exit:50%, Path:50%}#####

Fig. 2: Illustration of a classifier Cl_1 having a Behavioral Sequence to bridge the aliased green state from a non-aliased blue state, as well as a classifier Cl_2 enhanced by PEP to represent the environmental transitions from the PAI state, according to the classifiers representation in figure 1.

[9]; MACS implements a new attribute in the effect component of its classifiers to enable the system to discover new patterns from its environment [8].

2.2 ALCS and non-determinism

To allow ALCS to deal with non-deterministic environments, *Behavioral Sequences* (BSeq) [18] and *Probability-Enhanced Predictions* (PEP) [5] have been proposed, as depicted by figure 2. BSeq and PEP have both been integrated to ACS2 in [15] and [16]. They are triggered by classifiers that get both a correct anticipation and an incorrect anticipation in a unique state (*i.e.* in an aliased state).

BSeq enable the ALCS to bridge states that are related to the Perceptual Aliasing Issue using *sequences of actions*. However, BSeq do not enable the ALCS to build a complete and accurate representation of their environments as some states are skipped, and BSeq cannot promote the best decision policies because sub-optimal sequences of actions could be favored [17]. BSeq also imply a finer control of the population of classifiers by the ALCS, because the more these sequences are built, the more the population of classifiers will grow [15].

PEP were introduced in ALCS to deal with aliased states: PEP enable the *prediction of an ensemble of anticipations*, enabling the model to build a complete model of their environment. All items in the effect component of the $\{C, A, E\}$ tuple are replaced by PEP: they consist of an associative array in which keys are symbols related to the expected perceptive attributes, and values represent their probabilities to be anticipated by the classifier. The probabilities p, corresponding to the encountered aliased state, are updated in all the PEP as follows: $p = p + b_p * (1 - p)$, where b_p is an update rate, and all probabilities are then normalized. Nevertheless, both the probabilities computed in PEP and the sets of anticipated states they describe can be incoherent with the environmental settings of the ALCS: nonexistent states can be described by PEP (due to the combination of multiple associative arrays) and the computed probabilities are sensitive to their update parameter and the retrieved perceptions [17].

BEACS (Behavioral Enhanced Anticipatory Classifier System) is hereby introduced, with the goal of improving both the explainability of the system and

Woods100 Maze

	Condition	Action	Effect
PEPACS with PEP	##Path###Path#	→	##{Exit: 48%, Wall: 52%}#####
BEACS with EPE	##Path###Path#	→	{##Exit#####: 71,##Wall#####: 72}

Fig. 3: Illustration of PEPACS and BEACS classifiers enhanced by PEP and EPE in Woods100 to represent the environmental transitions from the PAI state, according to the representation depicted in the figure 1.

its performance in non-deterministic partially observable environments. BEACS couples, for the first time, both Behavioral Sequences and PEP approaches to handle learning tasks in such environments, while each of these approaches is also improved. Because the Perceptual Aliasing Issue (PAI) is a type of aliasing that PEP manage, this coupling is based on the detection of PAI [17]. BEACS is therefore based on the state-of-the-art PEPACS that integrated PEP in ACS2 [16].

3 Behavioral Enhanced Anticipatory Classifier System

3.1 Enhancing PEP into EPE

PEP enhancements resulted in *Enhanced Predictions through Experience* (EPE). Both PEP and EPE have the same goal of allowing classifiers to anticipate several states, but they differ in the representations they employ. As depicted in figure 3, EPE consists in an associative array whose keys are the perceptions anticipated by the classifiers, and whose values are the number of occurrences when these perceptions have been anticipated. As a result, because each anticipated state is explicitly described and counted, EPE provides more detailed information than an effect component with multiple PEP. Moreover, this method does not require a dedicated learning rate to be set up, and the probabilities of each anticipated perceptive item can be retrieved.

BEACS triggers the construction of EPE classifiers (*i.e.* that uses EPE) by using the same aliased state detection mechanism as PEPACS. EPE classifiers are constructed similarly to PEP enhanced classifiers in PEPACS, with two exceptions: they are tagged with the aliased states that trigger their creation; their effect component is merged from those of their two parents, and the number of occurrences of each anticipation are summed. The number of occurrences of a given classifier is only updated when this classifier anticipates several states, otherwise a default value of 1 is set to the unique anticipation. Classifiers that cannot subsume each other and correspond to the same aliased state (due to their condition, mark, and aliased state tag) are used to avoid the generation of useless EPE classifiers.

The anticipation of several states can lead to over-generalization, where enhanced classifiers can be used in non aliased states because of the genetic generalization pressure. To prevent this issue, PEPACS completely replaces the enhanced effects of genetic generalization offspring with the anticipation of the current perception, at the cost of a knowledge loss and a reduced learning speed, as the system may have to gradually rebuild the enhanced effect. BEACS does not replace the enhanced effects. Instead, BEACS exploits the aliased state tag to control the evolution of EPE classifiers if they have been over-generalized: they are specialized from their aliased state tag instead of using the current perception; they can directly learn new anticipations from their aliased state tag only, by adding states they failed to anticipate to their EPE (the related counters are set to 1).

Finally, the mutation operator used in the genetic generalization mechanism was modified to consider the semantics of the wildcard used in ALCS: this wildcard corresponds to all possible perceptive attribute in the condition, and indicates the related attribute from the condition does not change in the effect (*e.g.* figure 1). Hence, a perceptive attribute can be generalized *via* mutation, only if the associated attribute of the effect does not predict both a change and an absence of change. This modification aims at preserving the coherence of BEACS classifiers by preventing the creation of such contradictory classifiers.

3.2 Coupling EPE with Behavioral Sequences

Allowing ALCS to distinguish PAI states (states related to the Perceptual Aliasing Issue) from other aliased states detected by the system is the first step towards coupling EPE and Behavioral Sequences. This differentiation is then used to condition the creation of behavioral classifiers (classifiers with a Behavioral Sequence). It enables BEACS to control more precisely when BSeq should be used and ultimately, the evolution of its population. The Perceptual Aliasing Issue occurs in partially observable environments when the system cannot differentiate states that are truly distinct. The goal is to focus on states reachable from a PAI (and different from it): these reachable states should be more numerous than the number of actions that lead to distinct states (when the same action lead to different states, only the most anticipated state is considered). Both the set of reachable states and the number of actions can be computed with the help of EPE classifiers.

To do so, the most experienced classifiers for each single action are retrieved from the set of classifiers that matches the current perceptive input (with respect to their marks and their aliased state tags), by computing the product of their experience and their cubic quality (this power is used to widen differences between classifier qualities). If these most experienced classifiers exist for all possible single actions and are experienced enough (according to a user defined threshold), the set of reachable states and the number of actions that are related to a perceptive change are computed using the $\{C, A, E\}$ tuple of these classifiers and the current perception.

While the detection of aliased states occurs at a classifier scale, the detection of PAI states should occurs at the scale of a *set* of classifiers. To avoid unneeded computational operations, BEACS does not attempt to detect the PAI states as soon as an aliased state is detected by a classifier: the PAI states detection occurs at the end of the anticipatory learning process. As BEACS also needs time to fit its classifiers to its environment in order to discover transitions between states, the occurrence of the detection of PAI states is similar to the way the genetic generalization takes place (as described by [6]): it depends on θ_{BSeq}, a user parameter representing the delay between two such consecutive detections, and an internal timestamp t_{BSeq} (added to each classifier) measuring the current delay. All states detected as PAI or no longer being PAI are registered in a list.

3.3 Enhancing the Behavioral Sequences

Once the use of Behavioral Sequences is triggered by the detection of PAI states, behavioral classifiers are built thanks to the penultimate classifier selected by BEACS in the state $s(t-1)$ and candidate classifiers in state $s(t)$ that successfully anticipated state $s(t+1)$. These candidate classifiers are temporarily stored by BEACS, until all the classifier anticipations have been considered in the Anticipatory Learning Process. Then, if state $s(t)$ has been registered as a PAI state, the $\{C, A, E\}$ tuples of the behavioral classifiers are made of: the condition component of the penultimate classifier, a fine-tuned effect component from the candidate classifiers and the penultimate classifier, and both action components in a sequence, while other internal parameters are set to default. The fine-tuned effect component is computed by replacing each anticipated change of the penultimate classifier and the candidate classifiers with the related perceptive attributes of state $s(t+1)$, and then by removing all effect attributes that match the condition component.

BEACS also stamps its behavioral classifiers with the PAI state that triggered their construction. If a state previously detected as PAI is no longer related to PAI, all behavioral classifiers related to this state are removed from the population of classifiers. This enables BEACS to adaptively build and delete behavioral classifiers as PAI states are detected, thus avoiding needless population growth.

As opposed to [14, 15], BEACS no longer discriminates behavioral classifiers that lead to a loop between identical states: it is up to the reinforcement component of BEACS to fit the use of these classifiers, instead of decreasing their quality while their anticipations are correct. Thereby, the Anticipatory Learning Process used by BEACS is the same for all classifiers and the mechanism that prevent such loops has been removed.

BEACS genetic generalization mechanism was also extended to prevent the use of behavioral sequences in states unrelated to the Perceptual Aliasing Issue. To do so, behavioral classifiers are indirectly generalized through mutation by comparing the condition components of the offspring: if one perceptive attribute is generalized in one condition component and not in the other, the related perceptive attribute in the other can be generalized. This indirect generalization avoids building useless behavioral classifiers that would match states that do

not lead to PAI states. Because behavioral classifiers can contain EPE, their mutation operator follows the modification introduced in section 3.1 to preserve the meaning of the classifiers.

Finally, BEACS Reinforcement Learning component uses the concept of Double Q-Learning [10] to adapt the rewards predicted by the classifiers to the length of the action sequences in order to promote the usage of the shortest Behavioral Sequences. Each classifier Cl uses two estimators ($Cl.r_A$ and $Cl.r_B$) to compute its reward prediction $Cl.r$, by using the immediate reward ρ, the discounted maximum payoffs predicted in the next time-step $maxP_A$ and $maxP_B$, the discount factor γ, the reinforcement learning rate β, the number of actions in the Behavioral Sequence $\#_{act}$, the maximal length of Behavioral Sequences B_{seq} and a configurable difference ϵ_r. Even if both estimators converged to the same value, ϵ_r allows BEACS to be biased in favor of the shortest sequences. The prediction reward of a classifier is scaled, so that the highest reward of one predictor is given when the sequence is made of a unique action, while the difference between the two estimators is used to decrease the prediction reward according to the length of the sequence of actions. Classifiers prediction rewards of an action set are updated as described by the algorithm 1.

Algorithm 1 BEACS Reinforcement Learning Component

function UPDATEPREDREWARD(Cl, $maxP_A$, $maxP_B$, ρ, γ, β, $\#_{act}$, B_{seq}, ϵ_r)
  if random() < 0.5: **then**
    $Cl.r_A \leftarrow Cl.r_A + \beta(\rho + \gamma maxP_B - Cl.r_A)$
  else
    $Cl.r_B \leftarrow Cl.r_B + \beta(\rho + \gamma maxP_A - Cl.r_B)$
  $max_r \leftarrow max(Cl.r_A, Cl.r_B)$
  $min_r \leftarrow min(Cl.r_A, Cl.r_B)$
  $Cl.r \leftarrow max_r - \frac{(max_r - min_r + \epsilon_r) * (\#_{act} - 1)}{B_{seq}}$

BEACS was tested with a set of mazes, which are widely used as reinforcement learning benchmark [1]: the obtained results are presented in the following section.

4 Performance in maze environments

4.1 Experimental protocol

Using the maze benchmark from [15], the experimental protocol is set in order to address the following questions:

- Can BEACS detect PAI states in order to efficiently build classifiers with Behavioral Sequences?
- Does the coupling of Behavioral Sequences with EPE in BEACS enable the system to build (a) complete representations of its environments, and (b) efficient decision policies?

- To what extent are the probabilities derived from PEP and EPE consistent with the non-deterministic properties of the environments?
- Can BEACS efficiently control the evolution of its populations of classifiers to alleviate the growth of its population due to the use of BSeq?

The benchmark is made up of 23 mazes of different complexities (due to the occurrence of PAI states for instance). To make the learning and the solving of the task more complex, the results of actions have a 25% chance of being uncertain, in which case the system performs a random action without knowing which one.

The goal of BEACS in these mazes is to construct a complete and accurate representation of its environment, by moving one grid-cell at a time, and in either eight adjacent positions, while attempting to reach the exit as fast as possible. Its perceptive capabilities are limited to the eight squares adjacent to each position. Its starting position in the mazes is random (but distinct from the exit).

BEACS is compared with BACS [15] and PEPACS [16] (control experiments) on the maze benchmark, as they are the state-of-the-art ALCS using Behavioral Sequences and PEP, respectively. For each maze of the benchmark, 30 runs were performed using each of these three ALCS. A run firstly consists of a succession of 5000 trials, that are constrained explorations until the exit or the maximal number of actions (100) are reached: ϵ is set to 0.8 for the ϵ-greedy policy used to select actions; the learning rate of the Anticipatory Learning Process and Reinforcement Learning, β, is set to 0.05; the PEP learning rate of PEPACS is set to 0.01; the maximal length of the Behavioral Sequences of BACS and BEACS is set to 3; θ_{BSeq} of BEACS is set to 1000 to manage the PAI states detection. Then, the ALCS are switched to pure exploitation (*i.e.* no Anticipatory Learning Process) and have 500 trials to bootstrap an efficient decision policy ($\epsilon = 0.2$, $\beta = 0.05$) and 500 more trials to stabilize the rewards ($\epsilon = 0$, $\beta = 0.05$), before recording the number of actions required by the ALCS to reach the exit for 500 more trials ($\epsilon = 0$, $\beta = 0.05$). Other ALCS-related parameters not described here are initialized to the default values provided in [6]. A detailed learning parameters analysis could be considered for future work to emphasise their role in the building of environmental representations and decision policies by ALCS.

4.2 Metrics

For each experiment, the following metrics were collected for the 3 ALCS: the size of populations of classifiers along with the ratio of reliable classifiers within the populations, the average number of steps required to reach the exit, the knowledge ratio, and the average EP-accumulated error (EP stands for Enhanced Predictions). The list of states considered as PAI states by BEACS was also collected.

The knowledge ratio is the ratio of correct transitions learned by at least one reliable classifier to all possible transitions. Only transitions that led to environmental changes are included.

The average EP-accumulated error is a new required metric because knowledge ratios only provide information about symbol occurrences in PEP, and not about the environmental fitness of the computed PEP probabilities: for each possible transitions in the maze using a unique action, theoretical probabilities associated with the reachable states are first computed given the non-deterministic properties of the maze.The difference between each PEP item of the effect component of the most experienced and reliable classifier (if one exist, otherwise the most experienced as defined in section 3.2) and the corresponding theoretical probabilities are then computed. These differences are finally accumulated and averaged over the number of possible transitions, before being divided by the length of the effect component to get the average error by EP. In the case of EPE, a normalization of the counters associated to a perceptual attribute of interest can provide probabilities that are equivalent to those obtained by PEP.

All metrics were averaged over the 30 runs, for each environment. The obtained averages were compared with p-values computed by Welch t-test with (Welch-) Satterthwaite degrees of freedom (significance threshold 0.05) [7].

4.3 Performance

Can BEACS detect PAI states in order to efficiently build classifiers with Behavioral Sequences?
According to the environmental properties of the benchmark mazes, there were 1590 PAI states and 9810 non-PAI states throughout all experiments. BEACS has correctly identified 1490 of the 1590 PAI states, missed remaining 100 PAI states and incorrectly categorized 23 states as PAI. Thus, the balanced accuracy of the PAI state detection of BEACS is approximately 99.15%.

Does the coupling of Behavioral Sequences with EPE in BEACS enable the system to build (a) representations of its environments, and (b) efficient decision policies?
Figure 4 illustrates the knowledge ratios achieved by BEACS, BACS and PEPACS: BEACS and PEPACS were globally able to build complete representations of their environments, although PEPACS performed better than BEACS in 7 mazes , equally well as BEACS in 15 mazes (at least $p \geq 0.06$) and worse than BEACS in the remaining maze . BEACS did not achieve full knowledge of its environments in any of the experiments in two environments (MazeE1 and MiyazakiA) where it reached, at most, 97.84% and 99.31% respectively. PEPACS achieved full knowledge of its environments in every maze at least once.
Figure 5 shows the average number of steps required by the three ALCS to reach the exit. BEACS performed better than BACS in 20 mazes , equally well as BACS in the 3 remaining mazes ($p \geq 0.07$). BEACS performed better than PEPACS in 12 mazes , equally well as PEPACS in 3 mazes ($p \geq 0.24$) and worse than PEPACS in the 8 remaining mazes.

To what extent are the probabilities derived from PEP and EPE consistent with the non-deterministic properties of the environments?

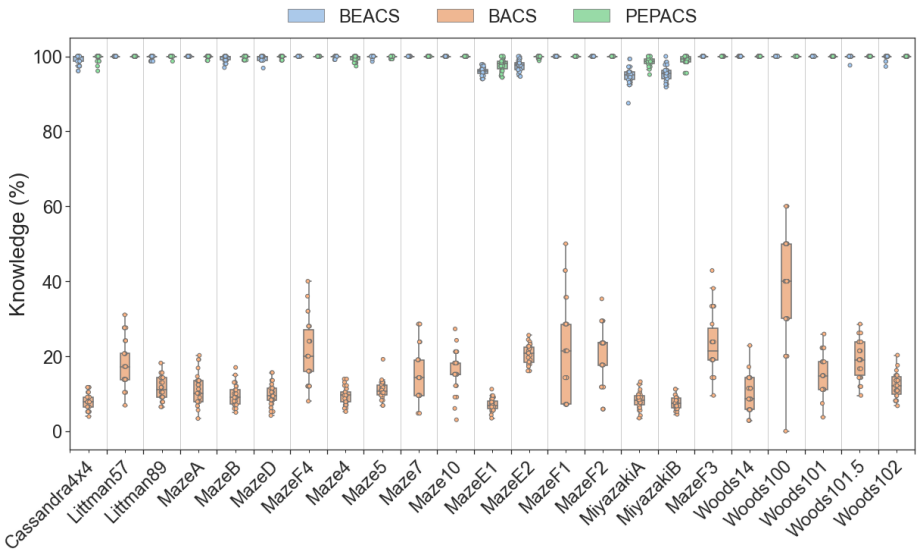

Fig. 4: Knowledge ratio achieved by BEACS, BACS and PEPACS. The higher the knowledge ratios, the better the performance. The use of PEP or EPE respectively permits PEPACS and BEACS to build complete representations of their environments.

The average EP-accumulated errors for each maze and ALCS are depicted in figure 6. BEACS has the lowest average EP-accumulated errors across all environments. BACS obtains, for 21 of the 23 environments, lower errors than PEPACS and larger errors than PEPACS for Cassandra4x4 and MazeE1. Therefore, the probabilities computed from the EPE of BEACS are the most consistent according to the non-deterministic properties of the environments.

Can BEACS efficiently control the evolution of its populations of classifiers to alleviate the growth of its population due to the use of BSeq ?
Figure 7 shows the size of the populations of classifiers created by BEACS, BACS and PEPACS, as well as the ratios of reliable classifiers within these populations. In 21 environments, BEACS populations are smaller than BACS populations while in the remaining two mazes, BEACS populations are larger. BEACS populations are smaller than PEPACS populations in 15 environments and larger for the 8 remaining mazes. BEACS has the highest ratios of reliable classifiers in 19 mazes and shares the highest ratios with PEPACS in Woods101.5 ($p \geq 0.13$). PEPACS has the highest ratios of reliable classifiers in the remaining environments: MazeE1, MiyazakiA and Woods100.

5 Discussion

Some states related to the Perceptual Aliasing Issue are impossible to detect due to the aliasing detection mechanism.

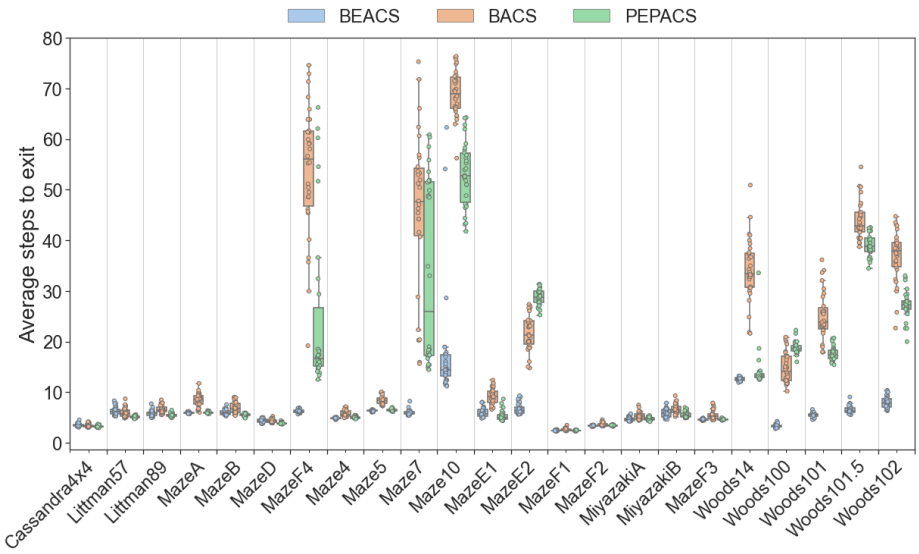

Fig. 5: Average steps to exit achieved by BEACS, BACS and PEPACS. The lower the average number of steps, the better the performance. BEACS is globally more efficient than BACS and PEPACS to reach the exit in non-deterministic mazes.

BEACS is the first ALCS to successfully detect when an aliasing state is related to the Perceptual Aliasing Issue. However, analyzing the 100 states that were not detected as PAI states revealed a limit: 90 of these states were not even detected as aliased. This is explained by the fact that truly distinct states in an environment can yield the same perceptions as well as the exact same environmental transitions to other PAI states. Because ALCS detect aliasing when several states are reachable for a state-action pair, such PAI states cannot currently be detected by BEACS or any ALCS.

BEACS balances performance and explainability

Behavioral Sequences (BSeq) and Enhanced Predictions through Experience (EPE) are used together to improve both performance and explainability of ALCS. BEACS intends to generalize the results reported for both approaches in a range of environments with varying characteristics and non-deterministic properties that have never been tested in previous studies.

First of all, although the number of learning steps used in the experiments is not set up according to the complexity of the benchmark mazes (this complexity refers to, for instance, the size of the maze or its non-deterministic properties), BEACS outperforms BACS and is globally more efficient than PEPACS (as shown in figure 5). However, its performance suggest that its reinforcement component could be further improved to promote the use of the shortest sequences of actions, since BEACS used, at most, one extra step in the environments in which PEPACS performs better than BEACS. Correlating the achieved results

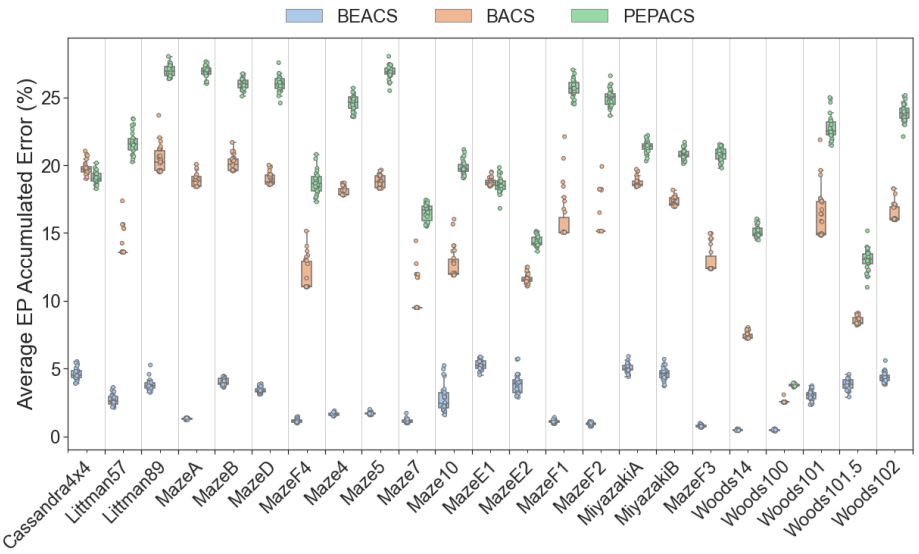

Fig. 6: Average EP-accumulated errors given the non-deterministic mazes, highlighting the gaps of the probabilities computed by PEPACS with the environmental settings. The lower the average EP-accumulated errors, the more accurate the environmental representations. The representations built by BEACS are more accurate than those of BACS and PEPACS.

of the ALCS with the intrinsic properties of each maze could be a direction for future research to improve their reinforcement component.

Then, even if BEACS performs slightly worse than PEPACS overall (up to about 4% worse on average over all environments), it is able to *create complete representations of its environments* (as seen in figure 4). The Behavioral Sequences are responsible for the observed discrepancies between PEPACS and BEACS. Indeed, their use implies that the system explores and fits these sequences to its environment at the expense of representation construction, which is thus slowed as Behavioral Sequences do not build reliable classifiers in PAI states.

However, the probabilities derived from the set of anticipations in BEACS classifiers are closer to the expected theoretical probabilities than those of BACS and PEPACS (as illustrated in figure 6): *BEACS environmental representations are thus much more accurate* than those of PEPACS and BACS. The probabilities computed by PEPACS are worse than those of BACS in 21 of 23 environments, even though PEPACS, as opposed to BACS, includes the PEP-mechanism to build accurate representations. In other words, BACS unreliable classifiers in aliased states may better describe the probability to anticipate next states than PEPACS reliable classifiers. This highlights the sensibility of the probabilities computed by PEPACS with regards to the experience of the system, when the system suffers from uncertainty in its action rather than relying on the experience of the system to make the probabilities converge, as BEACS does.

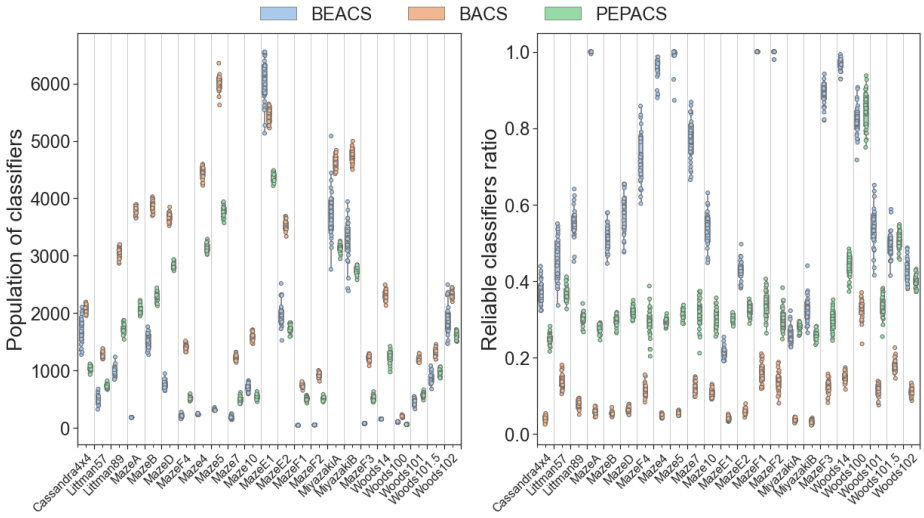

Fig. 7: Size of the populations of classifiers built by BEACS, BACS and PEPACS, along with the ratios of reliable classifiers within these populations. Small classifiers populations are easier to manipulate to extract knowledge. The higher these ratios, the more advanced the learning as the populations converge.

BEACS explainability assessment
BEACS computes its probabilities from EPE through experience, hence the larger the number of learning steps, the more accurate the probabilities. BEACS anticipations therefore provide new insights into classifier explainability by ensuring the reliability of the environmental representations: *BEACS classifiers can reliably be chained to trace the possible causes of a particular event.* However, as environments get more complex due to non-deterministic properties or perceptive inputs related to high-dimensional search spaces, populations of classifiers grow. The smaller the population of classifiers, the more BEACS is explainable. Thus, further works should focus on mechanisms efficiently reducing the populations of classifiers, such as compaction [20], by exploiting the knowledge acquired in BEACS population of classifiers.

Moreover, the representations used by BEACS to describe the classifier conditions and effects are kept unspoiled. This was possible thanks to the new mutation operator introduced in the genetic generalization mechanism, but at the expanse of the building of less general classifiers. Refining these representations to both highlight the changes in ALCS environments and build more general classifiers is a direction for future research.

6 Conclusion

This paper introduced BEACS (Behavioral Enhanced Anticipatory Classifier System) as an alternative machine learning model to solve reinforcement learning tasks, in an effort to increase both the *performance* and *explainability* of

Anticipatory Learning Classifier Systems. BEACS couples *Behavioral Sequences* (BSeq) and *Enhanced Predictions through Experience* (EPE) to handle non-deterministic, partially observable environments, which are common in the real world. While Behavioral Sequences enable ALCS to bridge states which cannot be distinguished by perception alone (known as the Perceptual Aliasing Issue) using *sequences of actions*, Enhanced Predictions through Experience allow ALCS to build multiple anticipations in non-deterministic states (*i.e.* aliased states).

BEACS is the first ALCS integrating a mechanism to distinguish states related to the Perceptual Aliasing Issue from all other aliased states. This allows the system to know when Behavioral Sequences should and should not be used, as BSeq can be used to deal with PAI but not with other types of aliasing. The construction of classifiers using BSeq has been enhanced and now provides a better control of these classifiers. The length of sequences of actions is taken into account to fit these sequences more efficiently to the environment. BEACS uses the EPE mechanism to build more accurate and explainable representations of its environments. The EPE classifiers aim at describing precisely each states encountered by the system, according to the environmental properties. Finally, by adaptively deleting, generalizing, and specializing its classifiers, BEACS can better frame the expansion of its population.

The results of a thorough experimental protocol using maze environments show that BEACS (1) is the only ALCS that builds complete and accurate internal representations of its environment when faced with non-deterministic environmental properties such as the Perceptual Aliasing Issue or uncertain results of action, (2) describes more precisely the states anticipated by the classifiers along with their probabilities to be anticipated, (3) builds efficient decision policies to solve the learning tasks and (4) provides explanatory insights about created decision policies and environmental representations to its user.

Future works should focus on the management of classifiers populations, to ease the interpretation of these populations through compression, visualization or generalization, as well as the assessment of different Reinforcement Learning mechanisms that can be embedded within ALCS.

References

1. Bagnall, A.J., Zatuchna, Z.V.: On the classification of maze problems. In: Foundations of Learning Classifier Systems, pp. 305–316. Springer (2005)
2. Butz, A.M.V., Goldberg, B.D.E., Stolzmann, C.W.: The anticipatory classifier system and genetic generalization. Natural Computing pp. 427–467 (2002)
3. Butz, M.V.: Biasing exploration in an anticipatory learning classifier system. In: International Workshop on Learning Classifier Systems. pp. 3–22 (2001)
4. Butz, M.V., Goldberg, D.E.: Generalized state values in an anticipatory learning classifier system. In: Anticipatory behavior in adaptive learning systems, pp. 282–301. Springer (2003)
5. Butz, M.V., Goldberg, D.E., Stolzmann, W.: Probability-enhanced predictions in the anticipatory classifier system. In: International Workshop on Learning Classifier Systems. pp. 37–51 (2000)

6. Butz, M.V., Stolzmann, W.: An algorithmic description of acs2. In: International Workshop on Learning Classifier Systems. pp. 211–229 (2001)
7. Fagerland, M.W., Sandvik, L.: Performance of five two-sample location tests for skewed distributions with unequal variances. Contemporary Clinical Trials pp. 490 – 496 (2009)
8. Gérard, P., Meyer, J.A., Sigaud, O.: Combining latent learning with dynamic programming in the modular anticipatory classifier system. European Journal of Operational Research pp. 614–637 (2005)
9. Gerard, P., Stolzmann, W., Sigaud, O.: Yacs: a new learning classifier system using anticipation. Soft Computing pp. 216–228 (2002)
10. Hasselt, H.: Double q-learning. Advances in neural information processing systems pp. 2613–2621 (2010)
11. Hoffmann, J.: Anticipatory behavioral control. In: Anticipatory behavior in adaptive learning systems, pp. 44–65. Springer (2003)
12. Kozlowski, N., Unold, O.: Investigating exploration techniques for acs in discretized real-valued environments. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. pp. 1765–1773 (2020)
13. Kozłowski, N., Unold, O.: Anticipatory classifier system with average reward criterion in discretized multi-step environments. Applied Sciences **11**(3), 1098 (2021)
14. Métivier, M., Lattaud, C.: Anticipatory classifier system using behavioral sequences in non-markov environments. In: International Workshop on Learning Classifier Systems. pp. 143–162 (2002)
15. Orhand, R., Jeannin-Girardon, A., Parrend, P., Collet, P.: Bacs: A thorough study of using behavioral sequences in acs2. In: International Conference on Parallel Problem Solving from Nature. pp. 524–538. Springer (2020)
16. Orhand, R., Jeannin-Girardon, A., Parrend, P., Collet, P.: Pepacs: Integrating probability-enhanced predictions to acs2. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. p. 1774–1781 (2020)
17. Orhand, R., Jeannin-Girardon, A., Parrend, P., Collet, P.: Explainability and performance of anticipatory learning classifier systems in non-deterministic environments. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 163–164 (2021)
18. Stolzmann, W.: An introduction to anticipatory classifier systems. In: International Workshop on Learning Classifier Systems. pp. 175–194 (1999)
19. Stolzmann, W., Butz, M., Hoffmann, J., Goldberg, D.: First cognitive capabilities in the anticipatory classifier system (02 2000)
20. Tan, J., Moore, J., Urbanowicz, R.: Rapid rule compaction strategies for global knowledge discovery in a supervised learning classifier system. In: Artificial Life Conference Proceedings 13. pp. 110–117 (2013)
21. Unold, O., Rogula, E., Kozłowski, N.: Introducing action planning to the anticipatory classifier system acs2. In: International Conference on Computer Recognition Systems. pp. 264–275. Springer (2019)