METAL: A Framework for Mixture-of-Experts Task and Attention Learning

MARYAM S. MIRIAN[1], BABAK N. ARAABI[1,2], MAJID NILI AHMADABADI[1,2] and ROLAND R. SIEGWART[3]

[1] *Control and Intelligent Processing Centre of Excellence, School of Electrical and Computer Eng, University of Tehran*
[2] *School of Cognitive Sciences, IPM, Tehran, Iran*
[3] *ASL, ETHZ, Switzerland*
{mmirian, araabi, mnili}@ut.ac.ir, rsiegwart@ethz.ch

Abstract

Rapid increase in the size and complexity of sensory systems demands for attention control in real world robotic tasks. However, attention control and the task are often highly interlaced which demands for interactive learning. In this paper, a framework called METAL (mixture-of-experts task and attention learning) is proposed to cope with this complex learning problem. METAL consists of three consecutive learning phases, where the first two phases provide an initial knowledge about the task, while in the third phase the attention control is learned concurrently with the task. The mind of the robot is composed of a set of tiny agents learning and acting in parallel in addition to an attention control learning (ACL) agent. Each tiny agent provides the ACL agent with some partial knowledge about the task in the form of its decision preference- called policy as well. The ACL agent in the third phase learns how to make the final decision by attending the least possible number of tiny agents. It acts on a continuous decision space which gives METAL the ability to integrate different sources of knowledge with ease. A Bayesian continuous RL method is utilized at both levels of learning on perceptual and decision spaces. Implementation of METAL on an E-puck robot in a miniature highway driving task along with farther simulation studies in Webots™ environment verify the applicability and effectiveness of the proposed framework, where a smooth driving behavior is shaped. It is also shown that even though the robot has learned to discard some sensory data,

probability of raising aliasing in the decision space is very low, which means that the robot can learn the task as well as attention control simultaneously.

Keywords: Attention Control Learning, Decision Space, Perceptual Space, Bayesian Continuous RL, Learning to Drive

1. INTRODUCTION

Learning is a key challenge in real world robotic tasks, since it is hardly possible to find a working hand coded strategy for complex tasks. Moreover, a human designer and the robot do not look at the world in the same way. A hand coded strategy seems more inappropriate particularly when the robot tries to learn a task while its dynamics is not fully known or changes over time. While learning in the field of robotics has been addressed by many researchers, the problem of proper information selection for decision making at each state has not been studied enough yet. Finding an appropriate answer to the later problem seems to be crucial since robots have different input sensors with a diverse value range, which cannot (and even may not need to) be used all together at each decision making step.

The first answer to this problem is "to choose a fixed subset (or a static combination) of sensors for all states". Such a static subset may work for simple tasks; however, in general it seems more appropriate and cost effective to select a different subset of sensory space information (based on the task's requirements) at each state for decision making.

As a result, a twofold learning problem arises at each robot learning task:

- To learn when to use a piece of sensory information, that is, to learn which piece of sensory information should be used at each state.
- To learn which action to perform at each state (defined based on the answer given to the above question)

The first question is the important question of top down attention control. In fact these two problems are not separate and are attached together tightly [1]. This means that attention strategy has to be learned per task. Here, we assume that a bottom-up path [2] for attention control is hand designed or shaped or already

learned [3] and this is the top down attention which has to be learned. Hereafter "attention" refers to "top-down attention".

In real world applications, solution to this coupled problem is not known at the design time and the robot should learn to solve it while interacting with the environment. From all possible candidates for online and interactive methods of learning, we have selected Reinforcement Learning (RL). This selection has two main reasons. First, usually the only available level of knowledge about the proper solution is a reinforcement that we can propose to the learning agent. Second, RL enjoys a large pool of powerful mathematical tools and properties, which helps us to formulate the problems that we have to solve. In fact, our problem –which is actually two concurrent learning problems coupled together- is a complex problem with a big state space. Such state space results in the so-called curse of dimensionality and slows down all interactive learning methods and RL methods are not exceptions.

It is shown in [1] that even for very simple tasks, starting to learn attention and task from scratch is not wise and impedes the learning speed of the coupled problem; which is not acceptable in robotic applications. In addition, selection of state-space of the attention control learning agent is a big challenge. These observations encourage us to propose a unified framework for learning task and attention control in consecutive learning phases for robotic applications. To evaluate the usefulness of the proposed framework, we selected a task to be learned in which we definitely need our attention to be controlled: *Driving.* As we briefly mentioned in [4], shaping a complex behavior such as driving, naturally bares a huge multi-dimensional sensory space. Selective attention is thought to be necessary because there are too many things in the environment to perceive and respond to at once considering the demand for fast response in face of limited computational resources in the driving task [5]. It is tried to keep the main concerns of driving however; we have simplified them to be implementable on a miniature driving simulator.

The paper is organized as follows: we continue with a review section on attention control learning. Then, the proposed framework for task and attention control learning is presented. After that the task, the robot, and the simulation environment as well as the real experiment are explained. Then, obtained results are presented and discussed. Finally, the paper is concluded with some final discussions, summary, and future works.

2. REVIEW OF RELATED WORKS

We all implicitly know what we mean by attention. But, a psychological definition may be a good starting point: focusing mind control in a clear manner on one of many subjects or objects that may simultaneously stimulates the mind [6]. As a definition from engineering perspective, it can be considered a filtering process to trim down the input sensory space such that we focus on something which is more valuable to be processed. Let us look at the attention problem from action perspective. This means using active perception instead of processing the entire sensory space. This is the viewpoint we have adopted and tried to realize through learning.

Because of the diversity of biological basics of attention, the review is done with more emphasis on the learning aspects of attention from engineering perspective. There are few researches on learning and formation of attention control; rather most are related to the attention modeling.[7] presents an RL based approach in which visual, cognitive and motor processes are integrated to help an agent learn how to move its eyes in order to generate an efficient behavior of a human expert while reading. Using two spatial and temporal modeling parameters (fixation location of eyes as well as their fixation time) the optimal behavior is learned. The idea of this work is "learning the eye movement behavior in concert with learning the reading task" which is similar to the idea of our paper. In [1] a framework for attention control is presented which acts actively in high level cognitive tasks. It contains three phases: the first phase is learning attention control as in active perception. Then in the second phase, it extracts those concepts learned previously and at last using mirror neurons it abstracts the learned knowledge to some higher level concepts. One of our main motivations is to continue the work in [1], but we are interested here in learning in the decision space rather than perceptual space for many reasons given next. In [8] a 3-step-architecture is presented which extracts attention center according to information theoretic saliency measures. Then, by searching in pre-specified areas found from first step decides whether the object is available in the image or not and lastly a shift for attention will be suggested. The final step is done using Q-Learning with the goal of finding the best perceptual action according to the search task. The interactive learning method and its step-by-step design are close to the idea of our proposed framework. In [9] some approaches based on hidden states in reinforcement learning are proposed for active perception in human gesture recognition. This work proposes some solutions for perceptual aliasing. This problem is realized when there is a many to many correspondence among environment's state and agent's state. In such a situation, the agent's

decision making has ambiguity and in order to reduce it, the agents decide to perform perceptual actions. This problem can be handled by merging similar (from utility perspective) states or splitting one state due to non-homogeneity in utility measure. The approaches for merging/splitting states presented in [9] are called Utile Distinction Memory and Perceptual Distinction Approach. Moreover, in order to handle the problem of requiring more than one shot observation, an approach called Nearest Sequence Matching is proposed which uses a chain of recent observations (state/action) to declare current state. The results show that by learning, they can find more informative set of features to attend for gesture recognition rather than just selecting them in a pre-specified manner. Unfortunately, it is mentioned that the computation load of these approaches are very high and can be problematic in real complex applications. In papers reviewed till now, the attention control policy selects something from spatial sources. In [10] biological evidences are presented which show that attention can also be directed to particular visual features, such as a color, an orientation or a direction of motion. They showed effects of shifting attention between feature dimensions, rather than specific values of a given feature. In one condition the monkey was required to attend to the orientation of a stimulus in a distant location. In a second condition it was required to attend to the color of an un-oriented stimulus in the distant location. Finally, inspired from *Mirror Neuron* idea in [11], there is an indirect biological support for the action-based representation in the decision space -what we proposed in this paper. It can be assumed that for each stimulus in perceptual space, there is a corresponding action-based representation in the decision space and we have proposed an approach for learning attention control in this alternate space.

3. PROPOSED FRAMEWORK: MIXTURE OF EXPERT TASK AND ATTENTION LEARNING

Learning attention control along with the task poses as a highly coupled complex learning task. Therefore, starting to learn attention and task from scratch does not seem to be feasible in real world robotic applications, where the number of learning trials should be kept as small as possible. Inspired by human's approach to learning complicated and attention demanding tasks [12] we propose a three-phase learning approach with the following successive distinct learning phases:

Demonstration-based passive learning: In the first phase, our learning agent just watches an expert advisor who performs the task. Similar to sitting next to a driver, the learning agent observes the world,

sense the advisor's action as well as the effect of the actions. This is similar to what occurs in imitative/demonstrative learning [13][14] [15]. However, there is an extra assumption here, that is, the possibility of sensing the mentor's action. Passive learning phase is designed to provide a minimum knowledge about the task to be learned. This phase may not cover all the aspects of task learning due to inherent difference in the perceptual state of mentor and the learning agent, and the very short availability of the mentor. As a result, subsequent learning phases are inevitable.

Initial active learning: In the second phase, the learning agent takes the control of the task but in a controlled environment or in an environment with slow speed movements. It is similar to training a beginner at a driving school, in an engineered environment. The learning agent has sufficient resources and can fully observe its sensory space and needs no attention control mechanism. In this phase, the critic only gives reinforcement, unlike previous phase where corrective actions are presented as well.

Attention control learning: In the last phase, the learning agent tries to learn attention control along with the task. Given the limited time that the learning agent has for decision making, the learning agent cannot observe its entire sensory space. As a result, shift of attention becomes inevitable. Previous two learning phases provide an initial knowledge about the task. As a result, when the learning agent starts to learn attention control, it does not have to start everything from scratch.

Now we propose a unified architecture for learning attention control along with the task which integrates above three phases of learning. Attention control as an adaptive filter on input space may operate on a variety of entities, which includes but not limited to objects, events, tasks, visual fields and operating parameters [16]. However, in our framework we realize the attention control at the *decision level*.

We assume there are some internal Tiny Agents (TAs) in the mind of our learning agent (or robot) that propose their decisions on the request of Attention Control Learning (ACL) agent. Each TA observes a part of the sensory space. TAs learn the task in the first and the second phases of learning. In the third phase of learning, TAs continue to learn along with ACL agent in order to improve the overall performance. The ACL agent observes TAs' decisions and learns to make the final decision (action) by attending to an appropriate subset of TAs. A Full Observer Agent (FOA) is considered only in first two

phases as well. The FOA is added to the proposed architecture only to facilitate the learning of TAs. Details of the FOA's role in learning is explained in Section 3.4.

A simple view of the proposed framework including expert advisor, FOA, TAs and the ACL agent is shown in Fig. 1. We call this framework Mixture of Expert Task and Attention Learning (METAL).

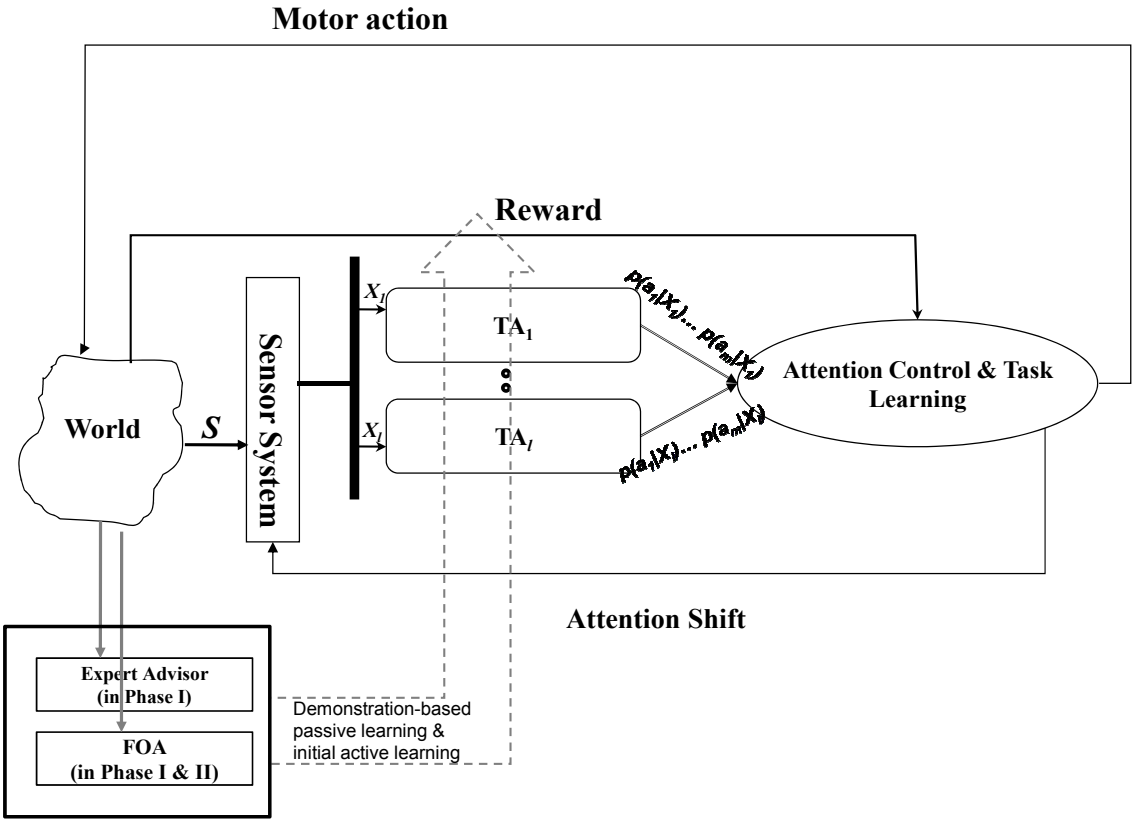


Fig. 1- A simple view of the collaboration among expert advisor (Phase I), FOA (Phase I & II), TAs (Phase I, II & III) and ACL agent (Phase III) in the proposed framework. X_i is the partial state of TA_i and $P(a_j|X_i)$ is the degree of support of a_j according to TA_i.

3.1. Learning in the Decision Space

Decision Space (DS) can be realized at three levels of best action, ranking of all actions, or Degree of Support (DoS) [17] for all actions. In this research we opt the last one. That is, our DS is formed by the DoS for all actions, according to all TAs at all states. There are some benefits associated with learning in the DS as compared with learning in the Perceptual Space (PS): (1) The DS can handle non-homogeneity in the learning and decision making algorithms of TAs. By using the DS one may utilize different sources of decision information ranging from naive classifier to expert advisor. (2) In the DS, one may form a compact and homogenous definition for state, while this is not the case in the PS. In the DS, the observed

state for the ACL agent is the output of TAs, as a result the size of the state space equals the number of actions multiplied by the number of TAs –both of which are manageable design parameters. While the size of the state space in the PS is proportional to the number of sensor readings, which is not necessarily controlled by the designer, varies a lot, and may result in a large state space.

Nevertheless, it is not always the case that the ACL on the DS is preferred to the ACL on the PS [18]. Particularly, when the dimension of action space is large, working on the DS may not be the preferred choice. A comprehensive comparison between the natures of these two spaces is given in [18].

As mentioned, we used DoS for the DS formation. The DoS is formed by continuous quantity of probability thus the ACL agent should learn in a continuous state space. The TAs learn directly on the sensory space, which is a continuous state space as well. As an efficient method for handling continuity, in our proposed framework TAs and ACL agent employ a Bayesian Continuous RL, which is introduced in [19].

3.2. An overview of the proposed framework

The proposed framework supporting the above mentioned requirements is depicted in Fig. 2, which consists of three basic entities: (1) different learning phases, (2) a number of key structural components utilized in learning phases and (3) evaluation measures. In the next three subsections, three learning phases are described in detail. Explanation of structural components (expert advisor, FOA, TAs, and ACL agent) are integrated into the description of learning phases. The last subsection describes different structural and behavioral evaluation measures used in this paper.

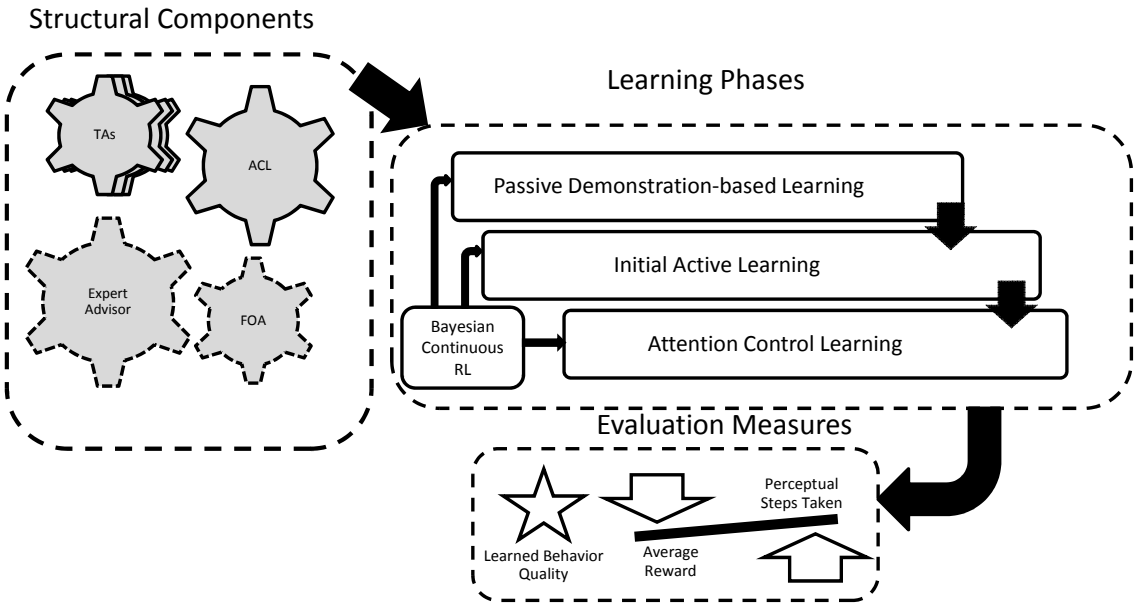


Fig. 2 – A schematic view of the basic entities of the proposed framework

In the first phase of learning, there is no attention control and all TAs along with the FOA learn based on the actions of the expert advisor as well as the received reinforcement signal. In the second phase, the expert advisor's role is restricted to a critic and the control of the task performance is transferred to the learning agents. In this phase, there is no ACL agent. Instead, a simple and fixed decision fuser is used to fuse the FOA's and all TAs' decisions. In the third phase, the FOA is removed as well. Thus, attention shift and decision fusion to find the final motor action are concurrently learned by the ACL agent. TAs continue to learn in this phase.

3.3.First Phase: Demonstration-based Passive Learning

In the first phase, the expert advisor acts both as mentor and critic. The TAs and the FOA recognize the state, partially and fully, respectively, sense the mentor's action, and receive the feedback from the environment. Then, TAs and FOA reinforce the association. *Expert advisor* has one particular role in the current phase along with two permanent roles in all three phases. Only in this phase the expert advisor proposes the correct action and directs the learning of TAs and FOA. Since the actions of expert advisor are assumed flawless, TAs and FOA learn with full learning rate. The other two roles are: to define the PS partitioning criterion and to design the reward function while acting as the critic. The complete description

of the two latter roles is given in Sections 4.2, 4.3, where the task is introduced. Fig. 3 shows a simple view of the components and the process of learning in the first phase.

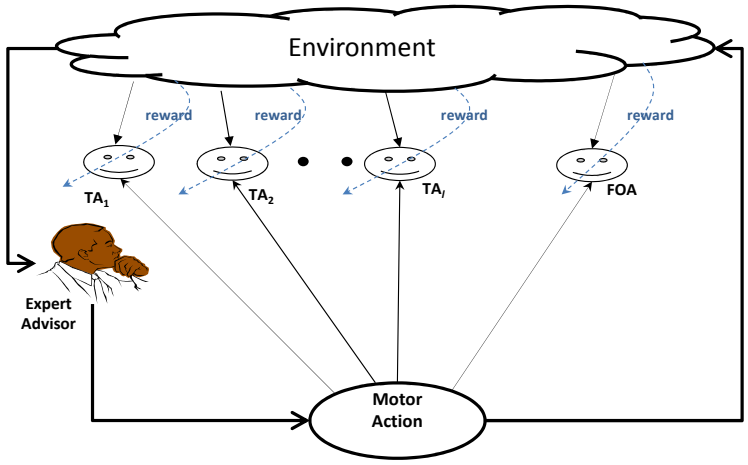


Fig. 3- A schematic diagram of first phase: Demonstration-based Passive learning

Now, let define the state of TAs. s_{TA_i} is the state of *i*th TA which is defined as

$$s_{TA_i} = [O_{1i}, O_{2i}, ..., O_{c_i i}],\ i = 1,2,...,l \qquad (1)$$

where O_{ji} is the *j*th feature of the *i*th TA, $j = 1,2,...,c_i$, c_i is the number of observable features for the *i*th TA, and *l* is the number of TAs. Using the definition of s_{TA_i}, the FOA's state (s_{FOA}) is defined as the concatenation of TAs' states

$$s_{FOA} = [s_{TA_1}, s_{TA_2}, ..., s_{TA_l}] \qquad (2)$$

After sensing the expert advisor's action a^*_{EA}, each TA updates $Q_{TA_i}(s_{TA_i}, a^*_{EA})$ and the FOA updates $Q_{FOA}(s_{FOA}, a^*_{EA})$ accordingly, where Q_{TA_i} and Q_{FOA} are *Q*-values in a *Q*-learning framework [20]. The general updating formulas in this framework are

$$Q_{TA_i}(s_{TA_i}, a^*_{EA}) = Q_{TA_i}(s_{TA_i}, a^*_{EA}) + TD^{TA_i}_{error} \quad (3)$$

$$Q_{FOA}(s_{FOA}, a^*_{EA}) = Q_{FOA}(s_{FOA}, a^*_{EA}) + TD^{FOA}_{error} \quad (4)$$

where $TD_{error}^{TA_i}$ and TD_{error}^{FOA} are Temporal Difference Errors. Among different possible choices for computation of $TD_{error}^{TA_i}$ and TD_{error}^{FOA} we opt the Baysian learning framework introduced in [19], for its action-based partitioning of state space; another method for action-based partitioning is proposed in [21]. It should be emphasized, however, that the Baysian learning framework utilized here can be replaced by any other continuous RL approach such as[22] [23] [24].

3.4. Second Phase: Initial Active Learning

In this phase, the expert advisor's role is restricted to a critic. A fusion center is also placed at the output of TAs and FOA, Fig. 4. TAs and FOA make their greedy decisions as follows

$$a_{TA_i} = \arg\max_k (Q(s_{TA_i}, a_k)), k = 1,2,...,|A| \qquad (5)$$

$$a_{FOA} = \arg\max_k (Q(s_{FOA}, a_k)), k = 1,2,...,|A| \qquad (6)$$

and provide the fusion center with their decisions (a_{TA_i} and a_{FOA}) along with their DoSs -defined by the probabilities $p(a_{TA_i} | s_{TA_i})$ and $p(a_{FOA} | s_{FOA})$. $|A|$ is the dimensionality of the action space. The final decision is made by the fuser based on maximizing a measure of expertness [25] [26]. Here conditional probability of actions to the observations is used as a measure of expertness.

$$a_{fusion} = \arg\max_a (p(a_{TA_1} | s_{TA_1}), p(a_{TA_2} | s_{TA_2}),..., p(a_{TA_l} | s_{TA_l}), p(a_{FOA} | s_{FOA})) \qquad (7)$$

Now a_{fusion} is performed, the reinforcement signal is received, and $Q_{TA_i}(s_{TA_i}, a_{fusion})$ and $Q_{FOA}(s_{FOA}, a_{fusion})$ are updated accordingly. The updating mechanism is similar to Eq (3), (4) in the first phase. The role of the FOA in learning of TAs is over at the end of the second phase. When the learning of TAs is converged, the FOA is removed and TAs become our local decision experts.

Let us explain the role of FOA in the learning of TAs. The FOA is able to fully observe the environment without any limitation in time and processing power. The learning problem of FOA is assumed to be MDP. It has a spatially unlimited access to the PS. Parallel learning of TAs along with learning of FOA accelerates the learning speed of both TAs and FOA. On the other hand, at early learning stages, although

TAs has not optimally learned the task, they can speed up the learning process of FOA due to their more abstract form of knowledge. The partial observation characteristic of TAs brings some spatial generalization ability to their knowledge formation -from observed states to un-observed ones. As soon as the learning of FOA becomes mature, it directs the TAs to the best action with more accurate and often higher DoS, i.e. $p(a_{FOA} \mid s_{FOA})$. Thus, interacting with FOA can implicitly adjusts TAs' DoSs by proposing the most probably correct actions that are not as often recognizable by TAs due to their inherent aliasing.

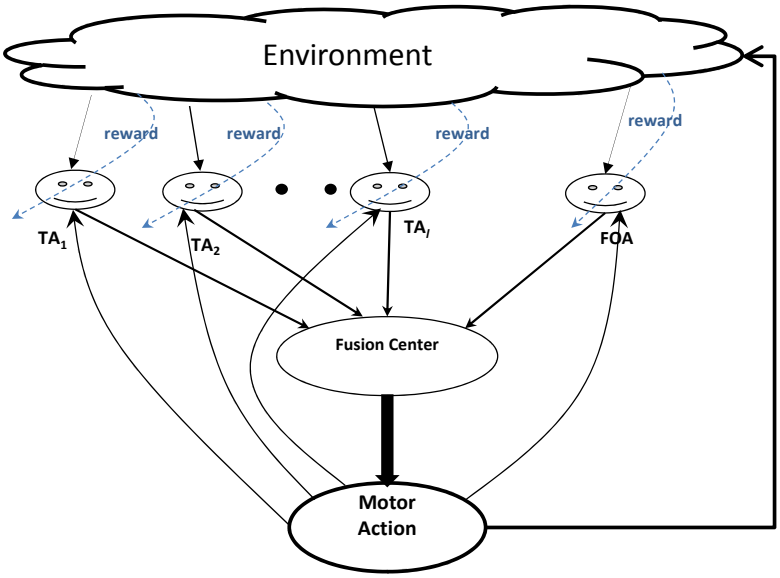


Fig. 4- A schematic diagram of second phase: Initial active learning

3.5. Third Phase: Attention Control Learning

Let us start the description of our idea about attention control learning in the DS with an analogy with the traditional form of ACL question in the PS: "If we have at most *n* sensors (or in general case *n* spatial locations/objects –for example inside a modality like vision) to perceive the state, which n' ones are more informative yet more cost effective to be utilized at each situation?" In the DS we have *l* TAs instead of *n* sensors. So, at each step of learning, our ACL agent tries to answer: Which l' (out of *l*) TAs to consult with in order to find the most rewarding action while paying the least total cost (i.e. the average of motor cost and perceptual cost)? This is our notion of attention control in the DS. Here, attending to -or consulting with- a TA means asking the TA to process its own partial sensory input, and then pass its decision to the ACL agent. Fig. 5 illustrates the third phase.

The ACL agent has two broad types of actions to select from: (1) to activate a TA and consult with it, which is equivalent to performing a perceptual action, or (2) to perform a motor action. Therefore, we can define its actions as

$$A_{ACL} = A_{Motor} \cup A_{Perceptual} \quad (8)$$

where A_{ACL} is the ACL agent's action set, consisting of motor and perceptual actions. The motor actions are those affecting the environment, and the perceptual actions are defined by

$$A_{perceptual} = \{ConsultTA_1, ConsultTA_2, ..., ConsultTA_l\} \quad (9)$$

The state of ACL is shaped by the DoSs proposed by the attended TAs, defined by

$$s_{ACL} = [(D_{TA_1} \,||\, null)...(D_{TA_i} \,||\, null)...(D_{TA_l} \,||\, null)] \quad (10)$$

"||" is the logical OR operator and D_{TA_i} is the DoS of all actions according to *i*th TA, that is

$$D_{TA_i} = \left[P(action_j \mid s_{TA_i})\right]_{j=1}^{|A_{Motor}|} \quad (11)$$

As seen in Eq. (10), the state of ACL agent is formed by concatenating the attended TAs' decision vectors. In fact, null replaces D_{TA_i} if i^{th} TA is not attended by the ACL agent. In this phase, $Q_{ACL}(s_{ACL}, a_{ACL})$ is updated as soon as ACL agent makes a decision. In addition, the Q-values of the attended TAs are also updated after every motor action taken by the ACL agent ($Q_{TA_i}(s_{TA_i}, a_{ACL_{motor}}), i \in$ attended TAs). It means that attended TAs continue to learn in this phase. The whole updating mechanism happens in the Bayesian Q-Learning core [19]. The ACL agent employs a soft-max [20] method for the action selection. Section 6 provides more discussions on the soundness of the proposed DS representation.

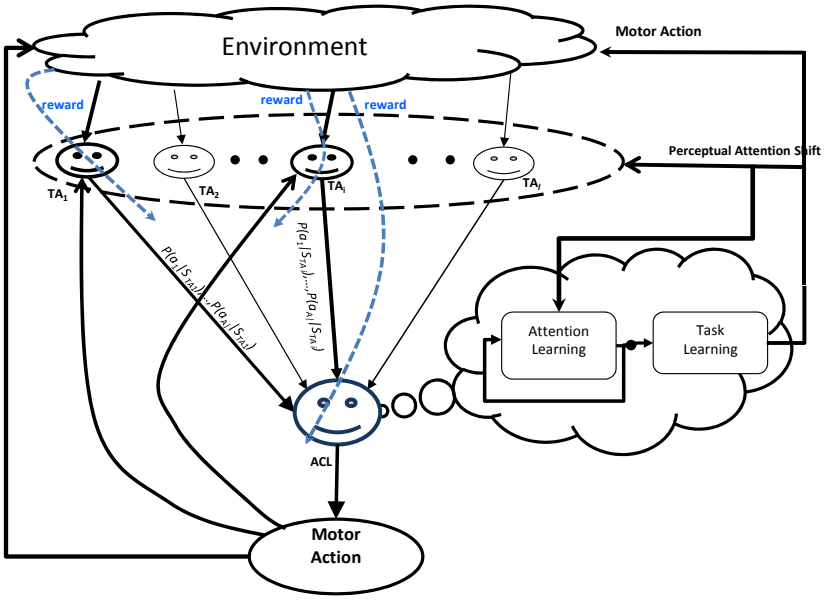


Fig. 5- A schematic diagram of third phase: attention control learning. Here, as an example, TA_1 and TA_i are attended while other TAs are inactive, at a particular state.

3.6. Evaluation Measures

One may categorize the evaluation measures into two main groups: structural and behavioral [27]. In the first group, the learning progress is measured in terms of the average reward [20] that the learning agent gains during learning. Accumulated average reward during test is another measure to evaluate the quality of learned knowledge disregarding the nature of the task to be learned. Moreover, the average number of perceptual actions taken at each state can be considered as a measure. The latter in-average-decreasing quantity illustrates that the learning agent has learned the task and found more knowledgeable TAs to consult with, at each state. In the second group, i.e. behavioral measures, the measure is supposed to evaluate the quality of the learned behavior. Obviously, these measures are task-dependent. For example, during learning driving and especially after that, it is important to consider the number of accidents, the smoothness of the driving path (or whether it does jerky motions), the speed smoothness (not selecting too many different speed values) and other related metrics [12]. In this paper, measures from both categories are utilized to evaluate the results.

4. Simulation Environment and the Results

To examine the effectiveness of METAL, this framework is realized on a learning driving task. In driving, not only the role of attention control is crucial but also the main decision making task is rather challenging.

The METAL framework is examined in both simulation studies and experiments with real robot where the results of the latter experiments are given in section 5.

4.1. The Robot

The E-puck robot (see Fig. 6.a) is selected as our miniature vehicle [28] i.e. the learning agent. It is facilitated with a color camera and eight infra-red sensors around it. The E-puck camera captures at the resolution of 39×52 . One sample image is shown in Fig. 6.c. The simulation is done in a 3D environment created in Webots™ [29] containing an irregular shaped road (as a highway with two lanes, see Fig. 6.b) plus multiple miniature vehicles of simulated E-puck robots (two other cars are shown in red). These mobile robots are FOAs that have learned how to drive during Phase I and II. All three robots try to be careful not to collide with red moving obstacles. In the simulation, these red cars are running in the greedy mode (with no learning) just to play the role of other drivers. The learner is shown as an un-colored E-puck. In the simulation, the mind of the learning agent is realized in MATLAB® while its body is located in Webots™.

4.2. The Design of TAs and FOA

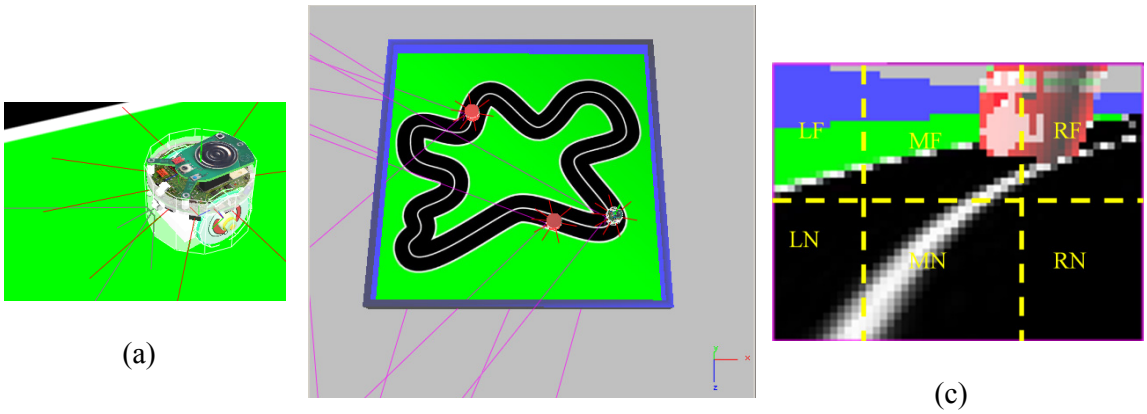
The PS of the E-puck robot is multi-dimensional. In general, a learning agent is neither able nor needs to simultaneously attend to its entire PS. Moreover, learning in such an over-sized non-homogenous space seems inefficient. As a result, in our approach we opt to partition the multi-dimensional PS, and assign separate TAs to each partition. In fact, by inter- and intra-sensor partitioning, we can find more homogeneous PS. Intra-sensor partitioning is straightforward: each physical sensor is individually assigned to a TA for exploration and learning. On the other hand, inter-sensor partitioning seems to be a more demanding task. Inter-sensor partitioning can be performed automatically by an optimization method [30] or based on methods such as agglomerative clustering [31] . The other possible solution is to hand-design the partitioning. Here, we opt for a simple heuristic hand-design since optimal partitioning is not the focus of this research. We define TAs of vision on each of six local areas shown in Fig. 6.c: Middle-Near (MN), Left-Near (LN), Right-Near (RN), Middle-Far (MF), Left-Far (LF), and Right-Far (RF) part of the scene. Furthermore, One TA is assigned to the space generated by eight Infra-Red (IR) sensors. Eight IR sensors of IR space's TA is visualized in Fig. 6.a. The FOA of vision observes all six areas at once but has no IR sensor. As explained in Section 3.4, the FOA is utilized merely to facilitate the learning of TAs. Adding an

extra input from the IR sensor to the FOA is avoided to keep the state representation homogenous, which results in a faster learning.

The state space of vision's TAs is composed of four features corresponding to four colors associated with four dominant objects: black for road, white for middle and side lines of the road, red for obstacles, and green for side-road area. Four features are defined to express the percentage of each observed color in the TA's visible scene. Therefore, in Eq. (1) $o_{ji} = color_{ji}$, $i = 1,2,...,6$, $j = 1,2,3,4$, and $o_{j7} = ir_j$, $j = 1,2,...,8$. Moreover, The s_{FOA} in Eq. (2) consists of s_{TA_i} for $i = 1,2,...,6$.

Five motor actions are considered for the robot

$$A_{Motor} = \{Go\ Fast,\ Go\ with\ Medium\ Speed,\ Slow\ Down,\ Turn\ Right,\ Turn\ Left\} \quad (12)$$



(a) (b) (c)

Fig. 6. (a) The E-puck robot close view (modeled in Webots™), (b) The top view of the road used for learning driving, and (c) The image captured by the E-puck camera while it is in the same position as it is at the above snapshot: spatial areas of interests are marked with related names.

4.3. Reward Function Design

As the Reward function is a high-level representation of the desired behavior, the learning process can be seen as the process of translating such a representation into a low-level control program. In our framework, the reward function consists of motor and perpetual parts.

4.3.1. Motor Reward

According to [12] there are two critical issues in driving: (1) the path of travel which is essential in keeping a car within the lane, and (2) the line of sight that allows the driver to see far enough ahead to have the time and space needed to make speed and position adjustments. Fig. 7.a shows a synthesis of reward

function (motor part) based on important driving skills. At each state, when the relevant portion of sensory signal is computed, it is normalized accordingly and amplified by an appropriate weight. These weights represent the value of each sub-behavior from expert advisor's point of view. In this paper, we adjust weights by simple heuristic assumptions, like $w_1 > w_2 > w_3 > w_4$, as well as trial and error to achieve a safer and smoother driving while observing the rules. The weights we used are $w_1 = 50, w_2 = 30, w_3 = 20, w_4 = 2$.

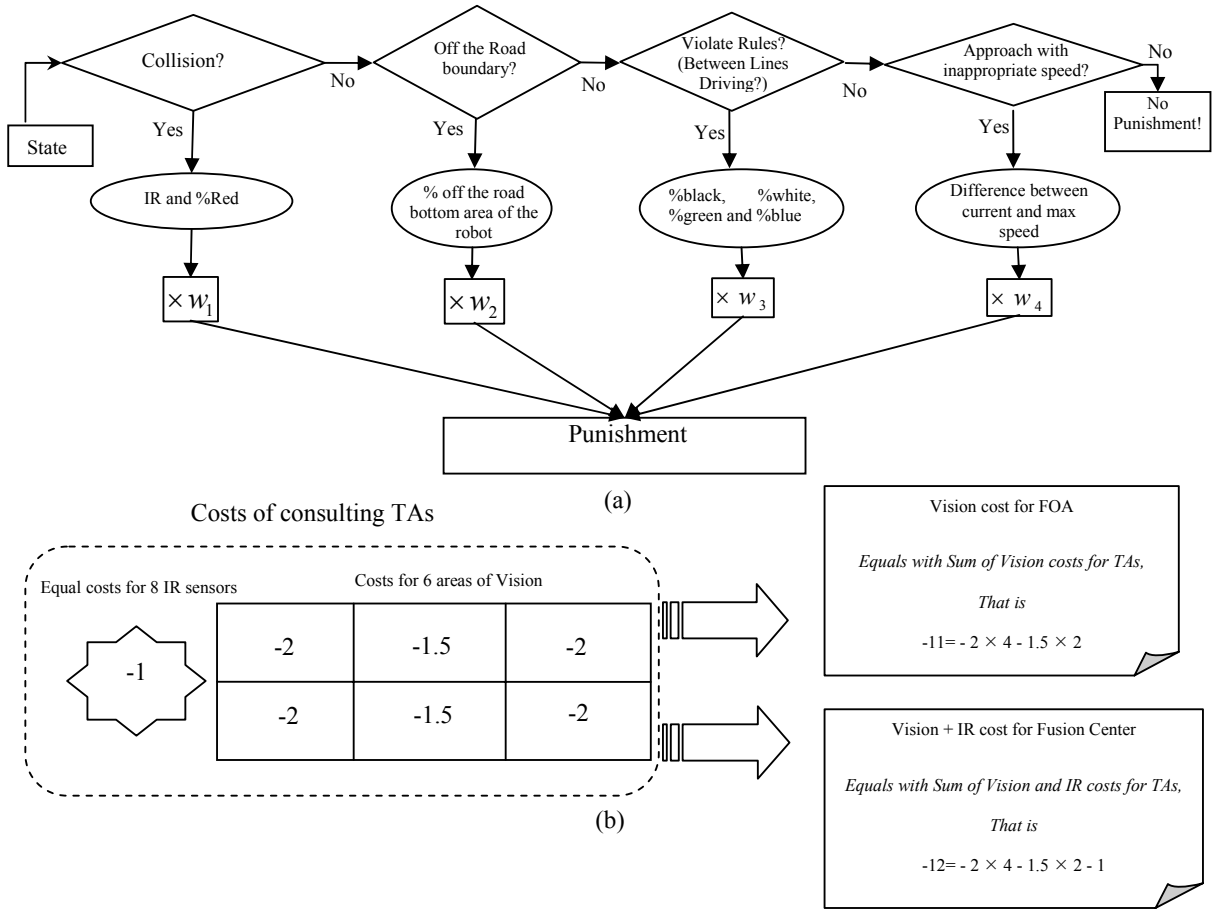


Fig. 7. (a) A hierarchical approach for Motor Reward Synthesis: Cost of motor actions for TAs in the first and second phases and ACL in the third phase, (b) Perceptual Reward for the ACL agent in the third phase, cost of perception for FOA, and fusion center in the second phase.

The reason behind the hierarchical design of the reward function is that by prioritizing the factors, the robot is implicitly helped to find the outcomes of its actions thus not to become confused.

4.3.2. Perceptual Reward

Taking each perceptual action imposes a cost to the robot. This is to make the robot take the least possible number of perceptual actions. As it is shown in Fig. 7.b the cost of processing spatial parts of vision is considered a bit higher than that of perceiving IR. This is to make the robot to prefer utilization of lower cost sensor unless not available. Besides, the cost of processing for the middle parts of vision (either far or near) is considered a bit lower than that of the other four lateral parts to make the robot prefers straight looking. The sum of vision costs for TAs is considered as the cost of vision for the FOA and the sum of vision and IR costs is considered as the cost for the fusion center in Phase II.

4.4. Simulation Description and the Results

In the first phase, the TAs and the FOA are trained by the driving demonstration of the expert advisor. In order to realize the action selection in the first phase by the expert advisor, we use three of the keyboard's arrow keys plus two other ones to select among five possible actions. Since the driving behavior of the expert advisor is assumed flawless, TAs and FOA learn these samples with full learning rate. This training phase lasts for 200 steps which is short enough in contrast with the duration of learning in the second and the third phases. At the end of this initial learning phase, not only the FOA but also all TAs have gained a minimum level of knowledge about the expected driving behavior.

After that, training of FOA is allowed to be continued individually till convergence. FOA is considered as the benchmark because the learning agent learns how to drive by applying the continuous state Q-Learning approach without the attention control. The accumulated reward by the FOA is shown in Fig. 8 with black solid lines, where the vision cost for FOA is -11. The learning is stopped after 1200 episodes in which the driving behavior seems satisfactory and the action selection becomes greedy.

In the second phase, the motor action is the result of fusion of all learning agents' decisions. The result of learning in the second phase is shown in Fig. 8 with dashed red line where the vision and IR cost for the fusion center is -12. An important point to note is that the number of TAs does not change the number of required learning trials, since all TAs observe the state in parallel, sense the executed action concurrently, get the reinforcement signal equally and as a result learn in a parallel manner.

When the learning of TAs in the second phase is converged, we remove the FOA and start ACL in the third phase. Each step of ACL starts with an action selection and ends after performing a motor action and

transferring to a new physical state. If the action selection results in a perceptual action the current step will continue till finally a motor action is decided.

The result of ACL in the third phase is shown in Fig. 8 with bold blue line, which confirms that the robot learns attention control as well as the driving task in the DS. The robot in the third phase consults with those TAs that their opinions are found more helpful in specific states. When it decides to stop consultation, it is time to decide which motor action to perform to gain more rewards. Note that selection of TAs by ACL agent is done one at a time.

As seen in Fig. 8, none of the approaches could reach to zero-punishment level. This is because there are some risky cases that the robot could not perfectly pass. One reason is that the red E-puck robots (shown in Fig. 6.b) do not necessarily care about colliding with the learning robot. They have the role of moving obstacles and only the learning robot cares for collision and collisions from back are also punished. In addition, the learning robot cannot predict the movement of the other robots using its current sensory information thus a few sudden front side collisions could not be avoided as well. In contrast to the FOA and the fusion of decisions –with fixed number of perceptions- the number of perceptions that ACL performs decreases as along the learning progress. The final average of perceptions is as low as 2.1 out of 7. As it is shown in Fig. 9, when the learning progresses, the percentage of selecting fewer perceptual steps increases and the robot does not pay to look at every part. Rather, it selects more beneficial perceptual actions. This is a good sign of learning which shows that the robot has learned which experts are wise to be consulted in a situation.

There are found situations in which ACL agent has preferred to pay more to look adequately by elaborating more on these situations -where several cases are shown in Table 1. It is seen that since attending to TAs of MN and MF parts impose lower costs while provide more information, these two TAs are attended more often; specially at the beginning of learning steps.

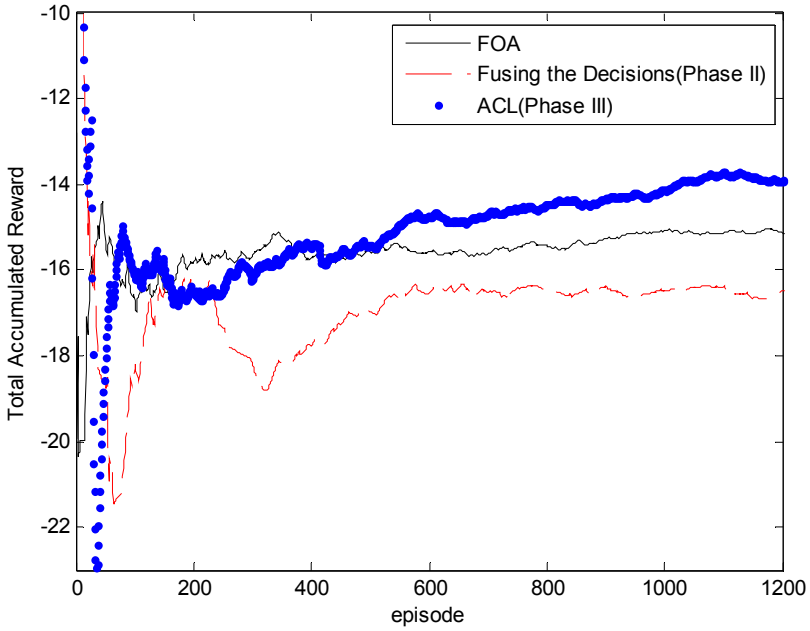


Fig. 8- The total accumulated reward gained during learning: the result shown is for the average of five runs with random initial position and orientation for the robot in the road. The robot starts learning from a random position but always inside the road.

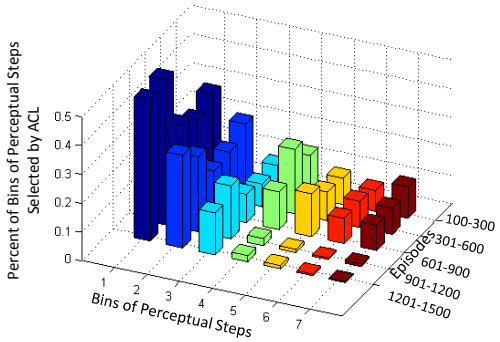
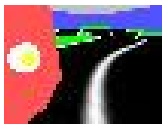
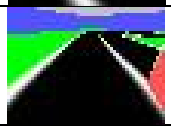


Fig. 9- The histogram for bins of perceptual steps during episodes of learning

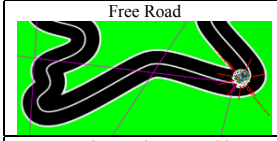
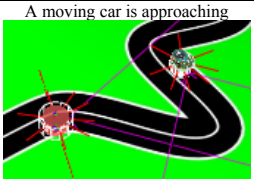
Table 1- Sample test situations (from the robot's point of view) and the perceptual steps taken

Situation	Scene (from the robot's point of view)	# Perceptual Actions Taken	Sequence of Performed Perceptual Actions	Intuitive Justification
Perpendicular view to free road		5	MN, RN, RF, MF, LF	This is a hard situation. In this case the robot decides to attend to nearly all TAs to find the best motor action.

Looking at a free curve from the roadside		4	MN, RF, MF, LF	This is a moderately hard situation and the robot takes four attention shifts to decide the motor action.
An obstacle at the left		3	MN, IR, LF	The middle near part contains a few red pixels. So, the robot decides to attend to IR and one more local spatial area to check the proximity and location of the obstacle.
The road is blocked with an obstacle		3	MN, MF, IR	The middle near part contains a few red pixels. This case can be justified like the above situation.
Free road (middle)		2	MN, MF	MN and MF are used to check the curvature as white pixels are seen in MN.
Free Road (left)		1	MN	All pixels in MN are black. So no obstacle is close ahead and the robot can move forward just by this observation.

When the learning in each of the three phases is finished, we tested the behavior of the learning agent in the same test scenarios. Test scenarios include free road and two other potentially risky situations shown in Table **2**. Every case is run ten times with random initial positions and orientations of the robot inside the road and the resulted total cost is averaged over all runs. The total cost is equal to the motor reward/punishment plus the perceptual cost. Perceptual cost in the first two columns is explicitly underlined but in ACL it is dynamically encoded in the computation of total punishment because it varies from state to state. The result of each run with/without collision is reported separately. The number of collisions is also reported (Accident/No Accident). The average of ten runs is shown at the bottom of each cell.

Table 2- Analysis of the after learning behavior: total punishment for 50 steps.

<table>
<tr><th>Policies
Test Scenarios</th><th colspan="6">Average Motor Cost + Average Perceptual Cost</th></tr>
<tr><td></td><td colspan="2">FOA</td><td colspan="2">Fusion of TAs' Decisions</td><td colspan="2">Attention Control Learning</td></tr>
<tr><td>Free Road
  </td><td colspan="2">-4.6 – 11 = -15.6</td><td colspan="2">-11.3 – 12 = -23.3</td><td colspan="2">-12.3</td></tr>
<tr><td rowspan="3">A moving car is approaching
  </td><td>Accident: 4</td><td>No Accident: 6</td><td>Accident: 7</td><td>No Accident: 3</td><td>Accident: 3</td><td>No Accident: 7</td></tr>
<tr><td>-23 –11 = -34</td><td>-9.9 – 11 = -20.9</td><td>-16.5 – 12 = -28.5</td><td>-11.5 – 12 = -23.5</td><td>-21.2</td><td>-13</td></tr>
<tr><td colspan="2">-26.1</td><td colspan="2">-27.0</td><td colspan="2">-15.4</td></tr>
</table>

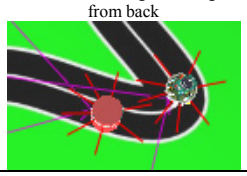
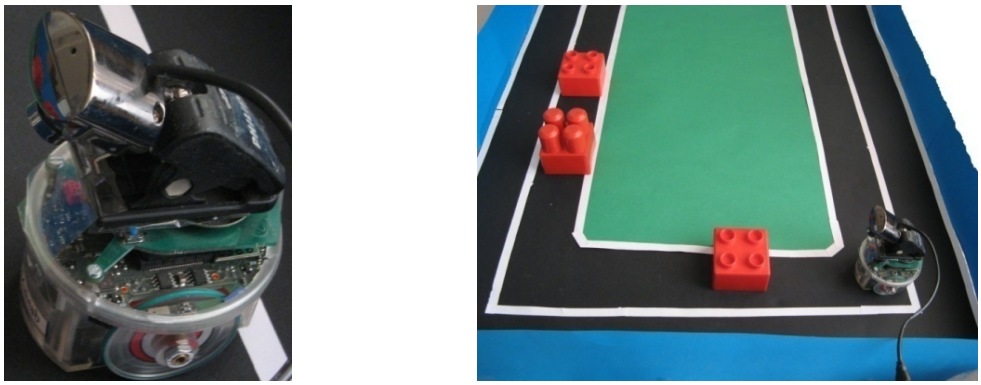
The robot is nearing a moving car from back 						
	Accident: 3	No Accident: 7	Accident: 5	No Accident: 5	Accident: 3	No Accident: 7
	-19.7 – 11 = -30.7	-7.3 – 11 = -18.3	-15.5 -12 = -27.5	-10.3 – 12 = -22.3	-21.1	-15.4
	-22.0		-24.9		-17.1	

Table 2 demonstrates that ACL performs quite better than two other opponents in terms of total cost. It depicts an acceptable driving behavior when the learning is finished as well. A movie of the driving behavior after learning in simulation can be found at [32]. Detailed inspection of the robot behavior revealed that the robot sometimes goes off the road in order not to collide with the other robots. It is because the reward function gives the first priority to accident avoidance over driving in the road boundaries. This can be activated by increasing the punishment of driving off the road.

5. EXPERIMENT WITH THE REAL E-PUCK ROBOT

To realize the proposed framework on a real robot, an E-puck robot is used. It is facilitated with an extra camera mounted on top of the robot as shown in Fig. 10.a due to the technical problem of communicating via Bluetooth to receive the built-in camera's image. A road with different shape but same dominant colors and static red objects is also designed to test the learned behavior of the robot. The road is shown in Fig. 10.b. When learning in simulation is converged (as shown in Fig. 8), we start training the robot in the real road in the phase I. This phase lasts for 20 episodes and contains driving in free road and avoiding collision. After that, the test in the designed road, Fig. 10.b, is started. The average reward gained by the robot in test on this road is shown in Fig. 11. The robot's behavior demonstrates smooth driving in free road and acceptable maneuvers to bypass the obstacles. A movie of the driving behavior in this experiment can be found in [33].



(a) (b)

Fig. 10. (a) The E-puck robot with a camera mounted on top, (b) A part of the real road designed to evaluate the proposed approach for learning driving

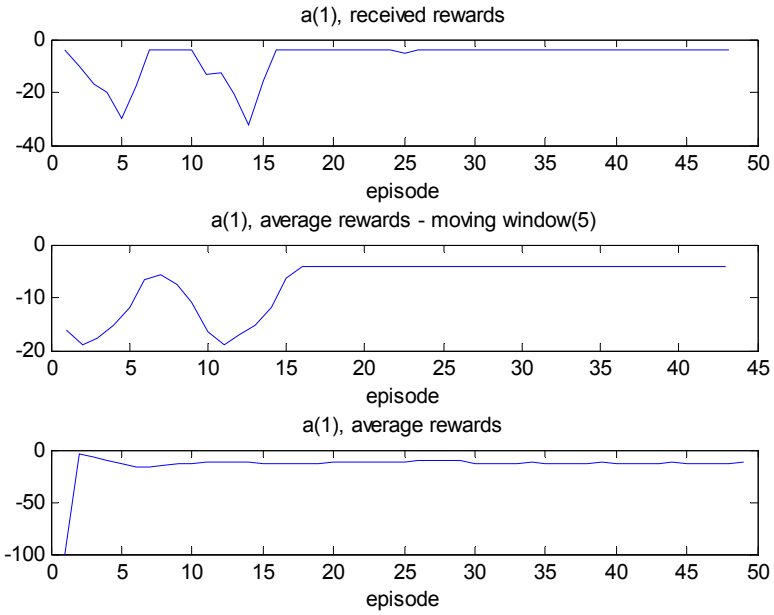


Fig. 11- Received reward on top, average reward in moving window (5) in the middle and accumulated reward at the bottom for the ACL agent's driving in the real miniature road

Simulation results and the experiment justified that the proposed framework works well in learning how to drive. In general, the proposed framework can explain how an existing source of knowledge about a task may be integrated to facilitate the learning process. Moreover, it can be claimed that the proposed METAL framework is actually a learning agent's design methodology. In fact, it can potentially facilitate the

process of learning a general robotic task by giving some hints on several important aspects of learning; such as

- a design-time method for decomposing the big PS of a problem at hand into manageable partitions
- a methodology of putting together the outcomes of learning by individual learning agents through an optimization method
- a policy for designing a hierarchical complex reward function
- a policy for learning which partition is more processing-worth which is the most important challenge of attention control learning

As a result, one may easily think about employing the METAL framework to other robot learning tasks.

6. A DISCUSSIONS ON ALIASING IN DS

One may argue that the appropriateness of ACL's policy –and actually that of our METAL framework- can be measured in terms of the level of perceptual aliasing that ACL agent may be faced in DS in comparison with the level of aliasing that the FOA is faced in PS. To do so, we check the amount of aliasing that may occur in the DS. In order to show that the learning agent can learn the task in the DS, we need to show that each physical state may not be mapped to more than one point in the DS and if this happens, either it is desirable (due to generalization ability) or it is very unlikely.

Let us start with computing the dimension of the PS. If we have *l* TAs, each observing *k* sub-dimensions of the world, the cardinality of the perceptual state space is

$$| S_{per} | = \left(q_{per} \right)^{l.k} \qquad (13)$$

where q_{per} is the quantization level of continuous values making the state of the FOA in the PS. This is the perceptual dimension, if and only if $s_{TA_i} \subset s_{FOA}$. Otherwise, the power of q_{per} is less than $l.k$ in Eq. (13). For example, in an n^2 maze, we can assume that there are two TAs (one for observing X, one for observing Y). So, we have

$$| S_{per} | = \left(q_{per} \right)^{l.k} = n^2 \qquad (14)$$

This computation can be repeated for the DS as

$$| S_{dec} |= \left(q_{dec}\right)^{l.|A|} \qquad (15)$$

where q_{dec} is the quantization level of continuous values making the state of the ACL agent in the DS and $|A|$ is the dimensionality of the motor action space. Now, if we want to have the least ambiguity in representation, the necessary condition is

$$\left(q_{per}\right)^{l.k} = (q_{dec})^{l|A|} \Rightarrow lk \ln(q_{per}) = l|A|\ln(q_{dec}) \Rightarrow \ln q_{dec} = \frac{k}{|A|}\ln(q_{per}) \Rightarrow q_{dec} = e^{\frac{k\times\ln(q_{per})}{|A|}} \qquad (16)$$

Now, we find the upper bound for the probability of arising aliasing in the DS. First, let us formally define the case arising no aliasing: For every two distinct physical states (s_i and s_j) if we can find two different messages (m_i and m_j) that are different at least in the value of one of the actions, there is no aliasing in the DS.

$$\forall s_i, s_j \in S, s_i \neq s_j, \nexists m_i, m_j \quad m_i \neq m_j \,(17)$$

$$\exists a_k \in A, p(a_k \mid m_i) \neq p(a_k \mid m_j)$$

Consider two physically distinct states in just one dimension. This is the worst case as it defines the upper bound of the probability of aliasing. Let us compute the probability of arising this ambiguity. We may assume that

$$\begin{aligned} s_1 &= [m_{11}, m_{12}, ..., m_{1l}] \\ s_2 &= [m_{21}, m_{22}, ..., m_{2l}] \end{aligned} \qquad (18)$$

It means that s_1 and s_2 contain totally different messages for TAs. The worst case is that they are the same in all but just one dimension. This means that $m_{1i} = m_{2i}$ for every i but $i = j$. Thus we have: $m_{1j} \neq m_{2j}$ .This case is shown in Fig. 12 for a 4x4 maze.

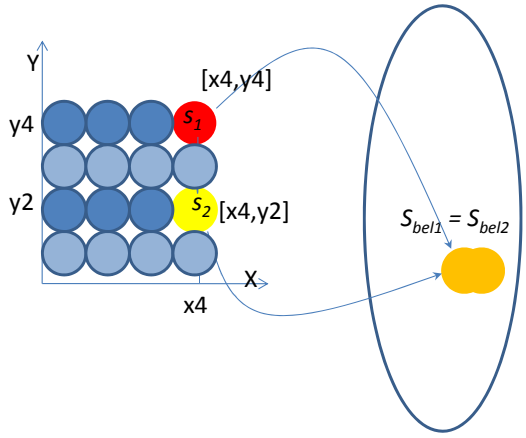


Fig. 12 – s_1 and s_2 are aliased in dimension X. They are distinct just in dimension Y.

We have aliasing in the belief space if $s_{bel_1} = s_{bel_2}$, where

$$s_{bel_k} = \begin{bmatrix} p(a_1 \mid m_{k1}) & p(a_2 \mid m_{k1}) & . & . & p(a_{|A|} \mid m_{k1}) \\ . & . & . & . & . \\ p(a_1 \mid m_{kj}) & p(a_2 \mid m_{kj}) & . & . & p(a_{|A|} \mid m_{kj}) \\ . & . & . & . & . \\ p(a_1 \mid m_{kL}) & p(a_2 \mid m_{kL}) & . & . & p(a_{|A|} \mid m_{kL}) \end{bmatrix}_{l|A|} \qquad k = 1,2 \quad (19)$$

It means we have $|A| \times l$ different equations which should be simultaneously satisfied. The worst case happens when $|A| \times (l-1)$ equations are already perceptually satisfied with probability of 1.0 and the DoSs are different in just one action. Therefore, the problem reduces to $|A|$ remaining equations, which should be satisfied. Then, we have

$$P_{aliasing} = \prod_{i=1}^{A} \prod_{j=1}^{l} p(p(a_i \mid m_{1j}) = p(a_i \mid m_{2j})) \quad (20)$$

Let us elaborate on one of the equations

$$p(a_1 \mid m_{1j}) = p(a_1 \mid m_{2j}) \Rightarrow \sum_{j=1}^{AliasedNo_{1j}} Q(a_1 \mid m_{1j}) = \sum_{j=1}^{AliasedNo_{2j}} Q(a_1 \mid m_{2j}) = C_1 \qquad (21)$$

where $AliasedNo_{1j}$ (or $AliasedNo_{2j}$) is the resolution of the dimension they are different in. It is equivalent to the number of physically distinct states that are all projected to m_{1j} (or m_{2j}) in the TA_1's eyes due to its inherent aliasing. In fact, an averaging mechanism on their Q-values has been occurred in the mind of TA_1 that forms each side of the above equation. Now, our goal is to evaluate the probability of satisfying Eq. (23). Each side of the above equation is equivalent to the combinatory problem expressed below:

We have $N = C_1$ balls (i.e. $k(q_{bel} - 1)$) and we want to distribute them in k places (where $k = AliasedNo_{1j} = AliasedNo_{2j}$) constrained to the condition that each place contains no more than q_{dec} (quantization level of each Q-value) balls. What is the probability of such placement?

This is the probability in the form of a recursive function

$$p_0 = \sum_{n=0}^{k(q_{dec}-1)} \left(\frac{r(n, q_{dec}, k)}{(q_{dec})^k}\right)^2 \qquad (22)$$

$$r(n, q_{dec}, k) = \sum_{i=0}^{\min(q_{dec}-1, n)} r(n-i, q_{dec}, k-1) \qquad (23)$$

with the following termination conditions

$$r(n, q_{dec}, k \le 1) = 1, r(n = 0, q_{dec}, k) = 0 \qquad (24)$$

As stated above, the number of aliased states is not fixed. Furthermore, C_1 is a continuous constant. But, we can make a discrete number out of C_1 considering the quantization level of DS i.e. q_{dec}. Therefore

$$C_1 \le (q_{bel} - 1) \times max(AliasedNo_{1l}, AliasedNo_{2l}) \qquad (25)$$

The final point is that after computing p_0, we have to raise it to the power of number of actions to find the upper bound of aliasing

$$P_{aliasing} = p_0^{|A|} \qquad (26)$$

Table 3 shows the maximum aliasing probability for various settings of the factors it depends upon. As seen, the aliasing probability is very low and it will become much lower if aliasing in multiple states is calculated. This justifies that the current representation for DS is sound and an appropriate.

Table 3- Maximum probability of aliasing in one state in terms of all affecting factors

$p_{aliasing}$	p_0	k	q_{bel}	$\|A\|$
0.1234	0.1234	4	4	1
2.32×10^{-4}	0.1234	4	4	4
5.1×10^{-7}	0.055	3	10	5
9.71×10^{-8}	0.0396	6	10	5
1.07×10^{-5}	0.10	6	4	5

So, we have demonstrated that transferring to the DS does not impose any further aliasing to the problem. Rather, it brings advantages already discussed.

7. CONCLUSIONS AND FUTURE WORKS

In this paper, a unified framework called METAL (Mixture-of-Experts Task and Attention Learning) with three consecutive learning phases was proposed to facilitate the complicated learning problem araised by combination of task and attention control. The role of the first two phases is providing an initial knowledge about the task, while in the third phase attention control was learned along with the task. Attention control was learned in the decision space instead of perceptual space. Utilization of decision space accounts for several advantages namely, ability to integrate different sources of knowledge at decision level, ability to keep the dimensionality of the space to be processed manageable, and possibility of using different learning methodologies for tiny agents before integration. Moreover, it was shown that utilization of decision space is unlikely to bring extra aliasing. The proposed framework works in a continuous state space which along with aforementioned benefits makes METAL suitable for complicated real world tasks.

The METAL framework was studied in Webots™ simulation environment and further implemented on an E-puck robot. The simulation results demonstrated that the robot (with attention control in mind) could learn the task with a few attention shifts. It means that the attention control learning agent can purposefully overlook those irrelevant pieces of information to the decision it should make at each state. Thus, our learning agent can gain a proper level of rationality in terms of time and computation. Experiments with

real robot in a miniature highway driving task also verified the applicability and appropriate performance of the METAL framework.

To pursue the ongoing research on attention control learning, there are some theoretical steps to work on. The first one is trying to find a more compact yet informative decision space within which attention control is learned. This space may possibly bring the higher advantages such as faster learning speed, lower cost and maybe more robustness. The other step is to try automatic partitioning of the perceptual space using an optimization approach in the demonstration-based learning phase. Furthermore, expanding attention control to the past observations in addition to the current ones should be considered. Moreover, application of the developed framework to more complicated tasks are in order.

ACKNOWLEDGMENT

This research was supported by University of Tehran and has been realized in close collaboration with the BACS project supported by EC-contract number FP6-IST-02´140, Action line: Cognitive Systems. The first author would like to acknowledge several of her colleagues in Robotics and AI Lab: Mr. Mohammad Afshar for the development of a powerful Webots™ interface to MATLAB® which facilitates the simulation of our idea, Mr. Mohammad M. Ajallooeian for the useful discussions we had together, and Mr. Hadi Firouzi for the efficient implementation of the Bayesian Continuous RL framework which is used in this research.

REFERENCES

[1] F. Shariatpanahi and Nili Ahmadabadi, "Biologically Inspired Framework for Learning and Abstract Representation of Attention Control," *Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint*, 2008, p. 324.

[2] L. Itti, C. Koch, E. Niebur, and others, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, 1998, pp. 1254–1259.

[3] A. Borji, M.N. Ahmadabadi, B.N. Araabi, and M. Hamidi, "Online learning of task-driven object-based visual attention control," *Under Press in Image and Vision Computing*, 2009.

[4] M.S. Mirian, H. Firouzi, M.N. Ahmadabadi, and B.N. Araabi, "Concurrent Learning of Task and Attention Control in the Decision Space," *Proceedings of IEEE/ASME Advanced Intelligent Mechatronics*, Singapore: 2009, pp. 1353-1358.

[5] A.K. McCallum, "Reinforcement learning with selective perception and hidden state," University of Rochester, 1996.

[6] W. James, "The principles ofpsychology," *New York: Holt*, 1890.

[7] E.D. Reichle and P.A. Laurent, "Using reinforcement learning to understand the emergence of" intelligent" eye-movement behavior during reading," *Psychological review*, vol. 113, 2006, pp. 390–408.

[8] L. Paletta, G. Fritz, and C. Seifert, "Cascaded Sequential Attention for Object Recognition with Informative Local Descriptors and Q-learning of Grouping Strategies," *Proceedings of the IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
[9] T. Darrell, "Reinforcement learning of active recognition behaviors," *Advances in Neural Information Processing Systems*, vol. 8, 1995, pp. 858–864.
[10] J.H. Maunsell and S. Treue, "Feature-based attention in visual cortex," *TRENDS in Neurosciences*, vol. 29, 2006, pp. 317–322.
[11] G. Rizzolatti, L. Fogassi, and V. Gallese, "Neurophysiological mechanisms underlying the understanding and imitation of action," *Nature Reviews Neuroscience*, vol. 2, 2001, pp. 661–670.
[12] T. Kline, "Developing Divided Attention Tasks: Dealing with Distractions," *Annual Spring Conference*, Myrtle Beach: South-east Region American Driver And Traffic Safety Education Association, 2006.
[13] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," *International symposium on robotics research*, 2004.
[14] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*, 2008.
[15] H. Mobahi, M.N. Ahmadabadi, and B.N. Araabi, "A Biologically Inspired Method for Conceptual Imitation using Reinforcement Learning," *Applied Artificial Intelligence*, vol. 21, 2007, pp. 155–183.
[16] A.L. Rothenstein and J.K. Tsotsos, "Attention links sensing to recognition," *Image and Vision Computing*, vol. 26, 2008, pp. 114–126.
[17] L.I. Kuncheva, J.C. Bezdek, and R.P. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, vol. 34, 2001, pp. 299–314.
[18] M.S. Mirian, M.N. Ahmadabadi, B.N. Araabi, and R.R. Siegwart, "Comparing Learning Attention Control in Perceptual and Decision Space," *Lecture Notes In Artificial Intelligence*, 2009, pp. 242–256.
[19] B.N. Firouzi, M.N. Ahmadabadi, and B.N. Araabi, "A probabilistic reinforcement-based approach to conceptualization," *International Journal of Intelligent Technology*, vol. 3, 2008, pp. 48–55.
[20] R.S. Sutton and A.G. Barto, "Introduction to reinforcement learning," 1998.
[21] M. Asada, S. Ichinoda, and K. Hosoda, "Action-based sensor space segmentation for soccer robot learning," *Applied Artificial Intelligence*, vol. 12, 1998, pp. 149–164.
[22] S. Whiteson, M.E. Taylor, and P. Stone, "Adaptive tile coding for value function approximation," *learning*, 2007.
[23] K. Doya, "Reinforcement learning in continuous time and space," *Neural Computation*, vol. 12, 2000, pp. 219–245.
[24] H.R. Berenji, "Fuzzy Q-learning for generalization of reinforcement learning," *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, 1996.
[25] M.N. Ahmadabadi, A. Imanipour, B.N. Araabi, M. Asadpour, and R. Siegwart, "Knowledge-based extraction of area of expertise for cooperation in learning," *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3700–3705.
[26] M.N. Ahmadabadi and M. Asadpour, "Expertness based cooperative Q-learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, 2002, pp. 66–76.
[27] J. Kleer and J.S. Brown, "A qualitative physics based on confluences," *Artificial Intelligence*, vol. 24, 1984, pp. 7–83.
[28] "E-puck, EPFL Education Robot, http://www.E-puck.org."
[29] "Professional Mobile robot Simulator, http://www.cyberbotics.com."
[30] N. Noori, M. Nili, M.S. Mirian, and B.N.Araabi, "Speeding up Top-Down Attention Control Learning by Using Full Observation Knowledge," *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2009)*, Daejeon, Korea: 2009, pp. 369-374.
[31] S. Theodoridis and K. Koutroumbas, *Pattern recognition. 2003*, Academic Press, New York, .
[32] *After Learning Driving: Simulation, http://khorshid.ut.ac.ir/~mirian/ACL/acl_simulation.mp4*.
[33] *After Learning Driving: Experiment, http://khorshid.ut.ac.ir/~mirian/ACL/acl_experiment.mp4*.