Conceptual Imitation Learning Based on Perceptual and Functional Characteristics of Action

Hossein Hajimirsadeghi, Majid Nili Ahmadabadi, and Babak Nadjar Araabi

***Abstract*—This paper presents a conceptual model for imitation learning to abstract spatio-temporal demonstrations based on their perceptual and functional characteristics. To this end, the concepts are represented by prototypes irregularly scattered in the perceptual space but sharing the same functionality. functional similarity between demonstrations is understood by reinforcements of the teacher or recognizing the effects of actions. Abstraction, concept acquisition, and self-organization of prototypes are performed through incremental and gradual learning algorithms. In these algorithms, hidden Markov models are used to prototype perceptually similar demonstrations. In addition to above, a mechanism is introduced to integrate perceptions of different modalities for multimodal concept recognition. Performance of the proposed model is evaluated in two different tasks. The first one is imitation learning of some hand gestures through interaction with the teachers. In this task, the perceptions from different modalities, including vision, motor, and audition, are used in a variety of experiments. The second task is to learn a set of actions by recognizing their emotional effects. Results of the experiments on a humanoid robot show the efficacy of our model for conceptual imitation learning.**

***Index Terms*—Imitation, Abstraction, Concept Learning, Incremental Learning, Hidden Markov Model.**

I. Introduction

THE robots capable of imitation can enter the human society by learning the human skills and social interactions [1]. However, true imitation is discriminated from other types of social learning (e.g., mimicking and sampling) by abstraction, conceptualization and symbolization [2]–[4]. In fact, perfect imitation is accompanied by comprehension and generalization which are attained by abstraction. Hence, skills can be represented in a generalized symbolic level which is desired for high level cognitive tasks [5]. In addition, abstraction helps for efficient memory management, handling the huge real world search spaces [4], and quick knowledge transfer from an agent to another agent or from a situation to another situation [6].

Recently, symbolization and conceptualization has drawn attention in robot learning by imitation [4], [6]–[10]. However, the majority of previous works are dedicated to form concepts based on similarity in perceptual space, and there is not enough work to find abstract concepts which share functional characteristics. Perceptual characteristics of a concept are physical or appearance-based features which show what the concept looks like while functional characteristics are those which specify what the concept is used for. We think that although perceptual categorization is necessary to abstract demonstrations in imitation, however, there exist skills or knowledge which cannot be transferred merely from perceptual information, like functional meaning or effect of the action. According to [11], [12], if a robot is to interact socially with a human, it should convey intentionality, i.e., it should express its beliefs, desires, and intentions. Human displays these internal states by social cues like gestures, postures, vocal and facial expressions. Indeed, children gradually learn to recognize, reproduce, and respond to these social cues in order to attribute beliefs, goals, and desires to others, engage in social interactions, and respond appropriately to situations [11]. However, intentionality is usually determined by functionality. It means that it is the function of social cues that specifies the intention of individuals. Thus, functional categorization of action is critically important for development of social skills.

The concepts which consider both perceptual and functional characteristics are called relational concepts [8], [13]. In this paper, we aim to develop a model for learning relational concepts through imitation.

H. Hajimirsadeghi is with School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran. He is also with School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada. E-mail: hosseinh@sfu.ca

M. Nili Ahmadabadi and B. Nadjar Araabi are with Cognitive Robotics Lab., Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran. They are also with School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. (e-mail: {mnili, araabi}@ut.ac.ir)

II. Related Works

In the last two decades many researchers have tackled the problem of *imitation* and *abstraction*. Most of these works perform perceptual abstraction of demonstrations and do not address functional associations between actions. Friedrich et al. [14] proposed one of the earliest frameworks of robot programming by demonstration with abstraction and generalization capabilities. In this model, abstraction is performed in symbolic level by encoding the task into a generalized sequence of basic operations. However, each task or skill is learned separately, and there is no *functional* conceptualization or abstraction of the skills.

Wermter et al. [15], [16] proposed an architecture for learning by demonstration and imitation based on multimodal learning, integration, and association of motor actions, vision, and language. In this model, the action sensor readings are abstracted and encoded by selforganizing networks trained to associate the actions to appropriate body parts and regions. In the next level, the motor action codes are associated to vision and language counterparts by a Helmholtz mechaine. This system is based on the neurocognitive hypothesis that mirror neurons fire in response to an action being performed, observed or verbally referred to [2], [16], [17]. Breazeal et al. [18] proposed a bio-inspired model for imitation learning and building socially intelligent robots. This model is inspired by the early facial imitation in human infants. Using the model, a robot was trained which could engage in both imitation and social referencing. However, this model does not approach abstraction of dynamics of action. Also, it does not capture functional relations between the learned actions.

Kadone and Nakamura [6] introduced an incremental algorithm to learn human motion primitives. Their model is able to automatically segment, abstract, memorize, and recognize demonstrated motions, using associative neural networks. However, like previous works, symbolization and categorization of the motion primitives is perceptual.

Hidden Markov Models (HMMs) have been extensively used for development of imitation models in the last decade [4], [7], [19]–[24]. Inamura et al. [4] proposed an HMM-based imitation model, inspired by mirror neurons and mimesis theory [25]. In this mathematical mimesis model, HMMs are used to abstract body motions as symbols, and these symbols are used for motion pattern recognition and generation. This model is the core of most subsequent papers which uses HMMs to symbolize motion patterns. In the mimesis model, demonstrated motions of different behaviors are grouped manually and next encoded into distinct HMMs in a batch training approach. So, the number of HMMs representing different behaviors is also determined a priori. Calinon et al. [19] proposed a modified HMM-based imitation framework based on dimensionality reduction. In this model, demonstrations are captured by visual and motor sensors and then projected into a latent space of reduced dimensionality. Next, the resulting trajectories are stochastically encoded by HMMs. This model performs well for both motion recognition and generalized motion reconstruction, and also it is robust against noise and needs less number of parameters for encoding. However, it is not incremental, and also the concept of each trajectory should be known a priori. Considering these shortcomings into account, successive algorithms were proposed for *incremental* and *autonomous* acquisition and learning of human motion patterns from continuous demonstrations [20], [26], [27]. For example, Kulic et al. [20] developed an algorithm for incremental and autonomous learning, symbolization, recognition, clustering and hierarchical organization of whole body motion patterns, using Factorial HMMs. However, in all of these works, abstraction and symbolization are based on similarity in perceptual space, and the proposed approaches do not address abstraction of relational concepts, which consider both perceptual and functional characteristics.

On the other hand, there are very few papers on imitation and abstraction based on similarity in functional space. One of the main frameworks for conceptual imitation which considers both perceptual and functional properties of action was proposed by Mobahi et al. [8], [28], who introduced a bio-inspired model to acquire abstract relational concepts from imitation, using reinforcement learning. However, unlike our algorithm which is suitable for sequence of observations (e.g., human motion), their proposed algorithm is only applicable for concept acquisition from single observations. Moreover, in the Mobahi et al. [8] model, only one motor action is considered for all perceptual variants of a concept.

The present work largely extends (in both theory and applications) our previously proposed model for imitation learning of relational concepts [9], [10], [29]. In this model, each concept is formed by a number of HMM prototypes which represent different perceptual variants of the concept but have the same functionality. First, we formulate conceptual imitation learning as a maximum likelihood optimization problem in order to provide mathematical justifications for the proposed algorithms. Next, two learning algorithms are presented: 1) Reinforcement-Based Conceptual Imitation Learning Algorithm (RBCIL) and 2) Effect-Based Conceptual Imitation Learning Algorithm (EBCIL). RBCIL, which works based on reinforcements of the teacher, is the modified version of the algorithm in [9], [29] and leads to increase of recognition accuracy with few training demonstrations. This improvement is obtained by a new kind of prototype, namely conceptual prototype, which fairly approximates the overall perceptual features of a concept until emergence of the perceptual prototypes which thoroughly represent the perceptual variants of the concept. RBCIL algorithm is evaluated in a variety of experiments with different types of data (perceived by different sensory systems), including visual, motor, and auditory data. It is also compared with some base-line batch training algorithms. In addition, we show how our model can be extended for multimodal concept representation and recognition. The multimodal integration in imitation is basically inspired by the well-known postulation that mirror neurons are multimodal [2], [16], [17]. The second algorithm, EBCIL, aims for imitation learning through recognition of functional effects of action. This algorithm identifies functional similarity between demonstrations through recognition of the effects of actions, which is more natural, implicit, and user-friendly (compared to explicit interaction with the teacher in RBCIL).

III. Conceptual Imitation

A. Concepts

Concepts are categorized into three levels of abstraction, namely, *perceptual*, *relational*, and *associative* [13]. Perceptual concepts are formed based on similarity of instances in perceptual space. In relational concepts perceptual similarity still contributes, but some external information, which specify the functionalities, should be also provided. This information integrates perceptually scattered categories into one concept. For example, consider the two gestures “kneeling” and “removing hat”, which are used for expressing respect in some cultures. On one hand, we have different instances of kneeling performed by different people, which are perceptually similar to each other. On the other hand, these samples are perceptually different from the samples of removing hat. So, these two gestures are categorized separately in perceptual space. But the external information (e.g., the situation) which describes the function of these gestures can link them into the right concept, which is “respect”. Finally, in associative concepts, instances of each concept has not any obvious physical similarity, but shared functional characteristics put them into one concept.

An important issue with a concept is how to represent it. Two of the main theories for concept representation are *exemplar* and *prototype* theories [30]. In exemplar theory, all instances of a concept are memorized. In prototype theory, a summary of instances are derived to represent various instances of a concept. This theory is more abstract and efficient to come up with limitations in memory.

B. Problem Description

In this work, we want to devise an algorithm for autonomous extraction and learning of relational concepts from imitation. In this way, demonstrated spatio-temporal actions are abstracted based on similarity in both perceptual and functional space. To this end, we favour to represent concepts by prototypes. Actually, the ideal situation is when we have the least number but the most comprehensive prototypes to understand a concept. Consequently, in the face of new demonstrations, the previously learned concepts can be recognized using generated prototypes, and there is no need of learning the action (motor commands to perform the action) from scratch. Also, actions which are associated with the same concept can be used alternatively in place of each other according to robot's comfort or affordance.

C. How to Teach Relational Concepts?

As described in Section III-A, relational concepts cannot form merely from perceptual observations, and external information should be also provided. This information can unify perceptually scattered prototypes which represent the same concept. We think that the external information can be obtained through *interaction* with the teacher or *recognizing the effects* of demonstrated actions in the environment. For the former, it is desired to have a simple process to transfer information from the naive teacher to the robot. One solution to this problem is same/different judgement [31]. According to [13], it is one of the simplest methods for learning relational concepts, and it has been experimented to be effective in teaching abstract concepts to some animals like pigeons and chimpanzees [31]. In this work, a similar method is used. First, the learning agent observes the teacher's demonstration. In response to the teacher, it guesses the concept of the demonstration. Next, it executes an action which is linked to that concept in its mind. Now, the teacher issues a reward or punishment signal according to correctness or incorrectness of the learning agent's response. So, the agent understands if the observed action and its executed action are associated with the same or different concept. In this way, it gradually learns and develops abstract concepts to increase its reward.

However, if the agent is able to recognize the effect of demonstrated actions in the environment, there will be no need of interaction with the teacher. In this case, the agent classifies the demonstrations with similar effects in the same concept.

IV. Proposed Conceptual Imitation Learning Model

In this section, we present a mathematical formulation of the problem and propose two algorithms for conceptual imitation learning of incremental demonstrations. In these algorithms, HMMs are used for abstraction and symbolization of spatio-temporal perceptions. People unfamiliar with HMM should refer to [32]. Also, to find the algorithms for motion generation through HMM, one might see [4], [20], [21].

A. Model Formulation

In conceptual imitation, the agent recognizes the concept of perceived demonstrations and then realizes the concepts by performing appropriate actions. Now, we want to formulate the model for concept recognition procedure. Let $\mathbf{x} = x_1 x_2 \cdots x_T$ be a perceived demonstration with the concept label $y \in \mathcal{Y}$. This demonstration has also a perceptual label $h \in \mathcal{H}_y$, which represents one of the perceptual variants of the concept y. In addition, for each perceptual variant of each concept, we consider an HMM denoted by λ_{hy}. Our goal is to learn all these HMMs (i.e., $\mathbf{\Lambda} = \{\lambda_{yh}\}_{y\in\mathcal{Y}, h\in\mathcal{H}_y}$) in order to conceptualize the perceived demonstrations. To this end, the probability of a demonstration $\mathbf{x}$ with concept label y and perceptual label h is defined as follows:

$$\begin{aligned} P\left(\mathbf{x}|y,h,\mathbf{\Lambda}\right) &= \sum_{a\in\mathcal{Y}}\sum_{b\in\mathcal{H}_a} P\left(\mathbf{x}|\lambda_{ab}\right)\mathbb{1}(y=a,h=b) \\ &= P\left(\mathbf{x}|\lambda_{yh}\right). \end{aligned} \tag{1}$$

Using this formulation, we try to solve two problems: 1) *inference*, which is to find (recognize) the concept of demonstrations given the learned model; and 2) *learning*, which is to learn the model parameters (i.e., $\mathbf{\Lambda}$).

1) Inference: Inference aims to predict the concept of a demonstration. First, we define the probability of a demonstration $\mathbf{x}$ given a concept label y by:

$$P\left(\mathbf{x}|y,\mathbf{\Lambda}\right) = \max_{h\in\mathcal{H}_y} P\left(\mathbf{x}|y,h,\mathbf{\Lambda}\right). \tag{2}$$

This problem can be easily solved by enumerating all possible perceptual labels $h \in \mathcal{H}_y$. Next, the concept label $y^\star$ is recognized by:

$$y^\star = \arg\max_{y\in\mathcal{Y}} P\left(\mathbf{x}|y,\mathbf{\Lambda}\right) \tag{3}$$

2) Learning: For now, we assume we are given a training set of demonstrations and concept labels $\mathcal{T} = \{(\mathbf{x}^n, y^n)\}_{n=1}^N$. The goal is to learn the the model parameters Λ. To this end, we propose the following maximum likelihood optimization problem:

$$\begin{aligned} \Lambda^\star &= \arg\max_{\Lambda}\prod_{n=1}^{N} P\left(\mathbf{x}^n|y^n,\mathbf{\Lambda}\right) \\ &= \arg\max_{\Lambda}\prod_{n=1}^{N}\max_{h\in\mathcal{H}} P\left(\mathbf{x}^n|y^n,h,\mathbf{\Lambda}\right). \end{aligned} \tag{4}$$

This is a difficult optimization problem. Hence, inspired by the popular approach for solving similar problems in latent models, we use an iterative algorithm of alternating between estimation of the latent variables and optimization of the model parameters:

- *Estimation of latent variables*: Hold Λ fix, and find the latent variable h^n of each training demonstration by the following inference problem:

$$h^n = \underset{h \in \mathcal{H}_{y^n}}{\arg\max}\, P\left(\mathbf{x}|y^n, h, \mathbf{\Lambda}\right). \tag{5}$$

  This can be solved identical to what proposed for (2).
- *Optimization of the model parameters*: Hold h^n fixed and solve the following optimization problem:

$$\Lambda^\star = \underset{\Lambda}{\arg\max} \prod_{n=1}^{N} P\left(\mathbf{x}^n|y^n, h^n, \mathbf{\Lambda}\right) \tag{6}$$

$$= \underset{\Lambda}{\arg\max} \prod_{n=1}^{N} P\left(\mathbf{x}^n|\lambda_{y^n h^n}\right) \tag{7}$$

$$= \underset{\Lambda}{\arg\max} \prod_{a \in \mathcal{Y}, b \in \mathcal{H}_a} \prod_{n: y^n = a, h^n = b} P\left(\mathbf{x}^n|\lambda_{ab}\right), \tag{8}$$

  which can be simplified as follows:

$$\lambda_{ab}^\star = \underset{\lambda_{ab}}{\arg\max} \prod_{n: y^n = a, h^n = b} P\left(\mathbf{x}^n|\lambda_{ab}\right), \forall a \in \mathcal{Y}, b \in \mathcal{H}_a. \tag{9}$$

  This optimization problem is exactly the same as maximum likelihood optimization problem for training HMMs, which is a very classic problem.

In summary, the first step categorize the demonstrations into the perceptual variants (i.e. HMMs), and the second step trains the HMMs with the corresponding demonstrations.

So far, we have assumed that we are given a full set of labelled demonstrations for supervised learning. However, in our conceptual imitation learning problem, the demonstration are provided incrementally, and also the concept labels are not explicitly known. In addition, we do not know the number of perceptual variants of a concept a priori. Inspired by the associative memory hypothesis of mirror neurons in computational models of imitation [33] and what described in Section III, we will modify the formulation and propose two algorithms to come up with these problems in the next sections. It will be shown how the agent can retrieve the true concepts of demonstrated actions by the auxiliary reinforcement signal of the teacher or recognizing the effects of actions during learning process. Moreover, the agent will be able to abstract incrementally perceived demonstrations into HMM exemplars and prototypes in order to form the concepts gradually. In a nutshell, the proposed algorithms use the two above-mentioned steps for learning. However, due to the incremental nature of the problem, the demonstrations are incrementally assigned to the HMMs (equivalent to the first step), and after each assignment, the corresponding HMM is updated with the current demonstration (equivalent to the second step). Moreover, since the number of perceptual variants are not given, the algorithms start by training an HMM exemplar for each demonstration. But, after receiving more demonstrations, perceptually similar HMM exemplars of the same concept are clustered into consolidated and compact HMM prototypes.

B. Reinforcement-Based Conceptual Imitation Learning Algorithm (RBCIL)

The learning algorithm is a procedure where a cycle is repeated whenever a new demonstration is perceived. To ease explanation of the learning algorithm, assume we are at the middle of execution where some concepts have been formed, and some prototypes and exemplars have been stored in the agent's memory. In our algorithm, an exemplar is an HMM made up of only one demonstration. However, prototypes are HMMs formed by unifying and consolidating some exemplars in the memory. Accordingly, we store the exemplars and prototypes in two different sets, namely Working Memory (WM) and Long-Term Memory (LTM), respectively. WM stores the exemplars temporarily to manipulate and organize the perceived information in the mind. LTM stores the prototypes produced by abstraction and consolidation of categorized information in the WM and forms the concepts. As a result, each concept is defined as a set of prototypes, and all the concepts together make the set of concepts Q:

$$WM = \bigcup_e \lambda_e ; \lambda_e \text{ is an HMM exemplar} \tag{10}$$

$$LTM = \bigcup_p \lambda_p ; \lambda_p \text{ is an HMM prototype} \tag{11}$$

$$Q = \bigcup_y q_y ; q_y \text{ is a set of HMM prototypes} \tag{12}$$

However, to have better understanding of the algorithm, the concepts have been embodied into symbolic units in Fig. 1, and the exemplars and prototypes have been connected to them. In fact, in the proposed algorithm, the exemplars and prototypes membership in the concepts are described by two functions c^W and c^L which associate exemplar and prototype indexes with concept labels, respectively:

$$c^W : \mathbf{N}_{|WM|} \to \mathbf{N}_{|Q|} \tag{13}$$

$$c^L : \mathbf{N}_{|LTM|} \to \mathbf{N}_{|Q|} \tag{14}$$

$$\mathbf{N}_{\mathcal{I}} := \{1, 2, \cdots, \mathcal{I}\} ; \mathcal{I} \in \mathbb{N} \tag{15}$$

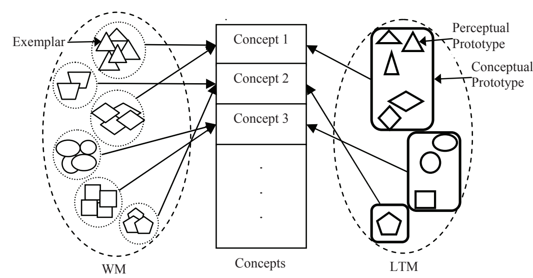


Fig. 1: Associative memory of exemplars, prototypes, and concepts.

As shown in Fig. 1, the prototypes in the LTM are divided in two types: conceptual and perceptual. Each concept has only one conceptual prototype, but it might have several perceptual prototypes. A conceptual prototype is an overall prototype (HMM) made out of all perceptual variants of a concept, while perceptual prototypes are HMMs trained by perceptually similar exemplars in the WM. So, perceptual prototypes represent different perceptual variants of a concept and can be employed to both recognize and generate these variants. However, the conceptual prototypes can be solely used for recognition purposes, and their generated sequences might be dissimilar to any of the perceptual variants of a concept. Furthermore, conceptual prototypes are more general prototypes which show less recognition power in the face of more specific perceptions.

Considering above definitions and assumptions, the pseudo-code of the algorithm accompanied by its schematic illustration are provided in Algorithm 1 and Fig. 2. Detailed explanation of this algorithm is presented as follows.

Algorithm 1 Reinforcement-Based Conceptual Imitaion Learning Algorithm

Input: Observed sequence, $\mathbf{x} = \mathbf{Sense}()$.
$Y_{tried} = \emptyset$
1: $i = \arg\max_{m,\lambda_m \in LTM, c_m^L \notin Y_{tried}} P(\mathbf{x}|\lambda_m)$.
2: **if** i is not Null **then**
3: $y = c_i^L, \mathbf{m} = a_y, Y_{tried} = Y_{tried} \cup \{y\}$
4: $\mathbf{Perform}(\mathbf{m})$
5: $R = \mathbf{Get\_Reinforcement}()$
6: **if** $R > 0$ and λ_i is a perceptual prototype **then**
7: **if** $\log P(\mathbf{x}|\lambda_i) \geq ll\_min_i$ **then**
8: Update λ_i with $\mathbf{x}$
9: **else if** $\log P(\mathbf{x}|\lambda_i) < ll\_min_i$ **then**
10: $\lambda_e = \mathbf{New\_W}(\mathbf{x}), c_e^W = y$
11: Update the conceptual prototype of q_y with $\mathbf{x}$
12: $\mathbf{Try\_Clustering}(q_y)$
13: **end if**
14: **else if** $R > 0$ and λ_i is a conceptual prototype **then**
15: $i' = \arg\max_{m \neq i, \lambda_m \in LTM, c_m^L = y} P(\mathbf{x}|\lambda_m)$.
16: $i = i'$
17: Do the lines 7-13
18: **else if** $R < 0$ **then**
19: Go to line 1 and repeat the steps
20: **end if**
21: **else if** i is Null **then**
22: Find the motor action $\mathbf{m}^\star$ such that $\mathbf{Perform}(\mathbf{m}^\star) = \mathbf{x}$
23: $\lambda_e = \mathbf{New\_W}(\mathbf{x}), \lambda_p = \mathbf{New\_L}(\lambda_e)$,
24: $q_y = \mathbf{New\_C}(\lambda_p), c_e^W = y, a_y = m^\star$
25: **end if**
Output: The concepts: $Q = \bigcup_y q_y$.

Try_Clustering(q_y):
1: **if** number of exemplars in $q_y > Num_{th}$ **then**
2: Cluster the elements in q_y
3: **for** clusters which are satisfying the criteria for making new prototypes **do**
4: $\lambda_p = \mathbf{New\_L}$(cluster elements), $c_p^L = y$
5: **end for**
6: **end if**

A New Demonstration is Perceived (Lines #1 to #5 in the pseudo-code): Now, assume that a novel demonstration is perceived by the robot. First, Likelihood of this perception sequence ($\mathbf{x} = x_1 x_2 \cdots x_T$) is computed against the HMM prototypes in the LTM, using forward algorithm. Next, the HMM prototype with the highest likelihood (we call this prototype as the winning prototype) is found according to (16), and the concept of this prototype is obtained according to (17):

$$i = \underset{m,\lambda_m \in LTM}{\arg\max} \; P(\mathbf{x}|\lambda_m), \tag{16}$$

$$y = c_i^L, \tag{17}$$

where, c_i^L is a simple function that maps a prototype index (e.g., i) to a concept label (e.g., y). Then, the motor action for that concept (i.e., $\mathbf{m} = a_y$) is performed. Afterwards, reinforcement signal (reward or punishment) from the teacher is issued. Now, it is crucial to specify three processes of concept acquisition in the learning algorithm [34]: when to make a new concept, when to modify a concept, and how to modify a concept. The description of these procedures are as follows.

Reinforcement Signal is Positive and the Winning Prototype is a Perceptual Prototype (Lines #6 to #13 in the pseudo-code): In this situation, it is firstly checked whether the new perception sequence can be absorbed by this prototype or not. It is defined that if the likelihood of the prototype is high enough, the prototype can absorb the perception. To evaluate whether the likelihood is high enough or not, the following criteria is used. If the log likelihood of the new perception given the winning HMM prototype is greater than the minimum log likelihood of that HMM's contents (i.e., the perceptions previously encoded in that HMM), the likelihood of this prototype is said to be high enough. We call the aforementioned minimum log likelihood value ll_min which is adjusted whenever a new HMM prototype is generated or modified. Next, the instructions in the following items are performed according to ll_min.

- 1. *The Log Likelihood of the Winning Prototype is greater than ll_min (Lines #7 and #8 in the pseudo-code):*
  In this case, the only thing to do is to strengthen the winning prototype by the new spatio-temporal perception. To this end, a modified form of re-estimation formulas suited for multiple observation sequences can be used [32]. The algorithm works by over-weighting HMM prototypes in order to consider the fact that they are built from multiple sequences.
- 2. *The Log Likelihood of the Winning Prototype is less than ll_min (Lines #9 to #13 in the pseudo-code):*
  This is the case when the log likelihood of the winning HMM prototype is not high enough to absorb the new perceived demonstration. In this situation, the perception sequence is encoded as a new HMM exemplar, stored in the WM, and linked to the rewarding concept (i.e., c^W is modified). Moreover, the conceptual prototype of that concept is updated by the new perception sequence. The reason to separate this item from the previous item is that there might be no true perceptual prototype for the new demonstration in the LTM, and so the winning prototype does not truly represent the perceive sequence.

Reinforcement Signal is Positive and the Winning Prototype is a Conceptual Prototype (Lines #14 to #17 in the pseudo-code): In this case, first the most probable perceptual prototype of the rewarding concept is found, and the absorbing criteria (i.e., $\log P(\mathbf{x}|\lambda) > ll\_min$) explained above is probed for this prototype. If the criteria is satisfied, the procedure described in item 1 is followed; Otherwise, the instructions in item 2 are performed.

Reinforcement Signal is Negative, But there are Still Untried Concepts in the Memory (Lines #18 to #20 in the pseudo-code): In this case, the untried concepts are tried in an order based on the likelihood of their HMM prototypes in the LTM. Whenever a concept is tried, its label is stored in a set of tried concept labels, namely Y_{tried}. This process repeats until the reinforcement signal of the teacher becomes positive. It means that the perceived demonstration belongs to the concept which receives reward from the teacher. Then, the agent modifies this concept exactly the same as the previously explained instructions, by updating the perceptual HMM prototype (if log likelihood is greater than ll_min) or making a new HMM exemplar and updating the conceptual prototype (if log likelihood is less than ll_min).

Reinforcement Signal is Negative, But there is no Untried Concepts in the Memory (Lines #21 to #24 in the pseudo-code): This situation means that all the concepts have been tried, but no reward has been issued by the teacher. In this case, a new concept should be generated. For this purpose, the perceived signal is encoded into an HMM, stored as a conceptual prototype in the LTM, and make a new concept. This prototype is also copied in the WM as an exemplar and connects to the new born concept. The reason that we make conceptual prototypes is to always have a representative of the perceived concepts in the LTM. The number of these prototypes is equal to the number of available concepts which is usually less than the number of exemplars in the WM. The stored exemplars in the WM are used in future to make perceptual prototypes which lead to perceptual organization, abstraction, and classification of information in the mind.

As previously noted, conceptual prototypes cannot be used to regenerate the sequences appropriately. So, whenever a new concept is created, the agent makes the motor action for the perceived demonstration by the body inverse models and stores this motor information in the memory until a perceptual prototype emerges for that concept. Perceptual prototypes can successfully generate appropriate generalized motion patterns which are transformed to motor commands, using inverse models.

*Making New Prototypes by Clustering Exemplars and Perceptual Prototypes of a Concept (***Try_Clustering** *function in the pseudo-code):* Following the procedure explained so far, the WM is overpopulated with exemplars after a short time. So, we must have an abstraction and consolidation mechanism to merge HMM exemplars and make HMM prototypes, which are stored in the LTM. For this purpose, whenever an exemplar is stored in the WM of a concept and the number of exemplars of that concept exceeds a threshold number (Num_{th}[1]), then

[1]It is a fixed parameter which is set before the algorithm starts. Low value of this parameter speeds down the processing time of the algorithm (due to increase of calling clustering process), but it advances production of prototypes which results in less number of trial-and-error interactions required to train the system (due to more accurate recognition of first demonstrations). In the experiments of this paper, Num_{th} is always set to 3.

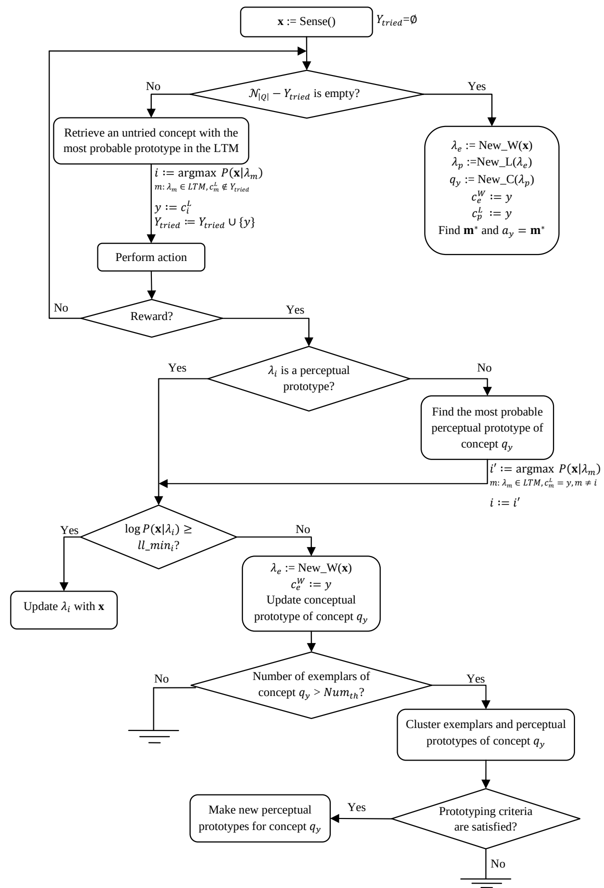


Fig. 2: Schematic diagram of RBCIL algorithm.

a clustering process gets started on both the HMM exemplars and perceptual prototypes of that concept. In this work, we use the algorithm proposed by Kulic et al. [20] to cluster HMMs based on the pseudo-distance:

$$D(\lambda_1, \lambda_2) = \frac{1}{T}\left[\log P\left(\mathbf{x}^1|\lambda_1\right) - \log P\left(\mathbf{x}^1|\lambda_2\right)\right], \quad (18)$$

where, λ_1 and λ_2 are two HMM models, $\mathbf{x}^1$ is a perception sequence generated by λ_1, and T is the length of $\mathbf{x}^1$. Finally a symmetric distance is defined as:

$$D_s = \frac{D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)}{2}. \quad (19)$$

Now that the distances between HMMs are specified, an agglomerative algorithm which performs a complete link hierarchical clustering is used to construct new prototypes. Final clusters are selected based on two criteria, i.e., surpassing the minimum number of elements and falling behind the maximum distance measure. Maximum distance measure is defined according to mean (μ_D) and standard deviation (σ_D) of the distances between all the exemplars and perceptual prototypes in the concept:

$$D_{cutoff} = \mu_D - K_{cutoff} \cdot \sigma_D. \quad (20)$$

where, K_{cutoff} is an algorithm parameter which controls the number and granularity of clusters. After this operation, if new clusters are produced, corresponding HMM prototypes are trained with their associated elements in the clusters, using Baum-Welch algorithm or modified re-estimation formulas explained before. These HMMs are stored as perceptual prototypes in the LTM. For example, if there are three exemplars in a produced cluster, perception sequences from these exemplars are trained into one HMM. In addition, the likelihood of each of the three sequences is computed against the new HMM prototype, and the minimum value is set as ll_min of this perceptual prototype. An illustration for the process of clustering and producing perceptual prototypes is depicted in Fig. 3. Note that high values of K_{cutoff} lead to more number of prototypes with more specific and tighter encoding, while low values of K_{cutoff} lead to less number of prototypes with more general and looser encoding. In our experiments, this parameter is set based on some prior knowledge of the problems (e.g., perceptual similarity between different perceptual variants of a concepts) and some trial and error.

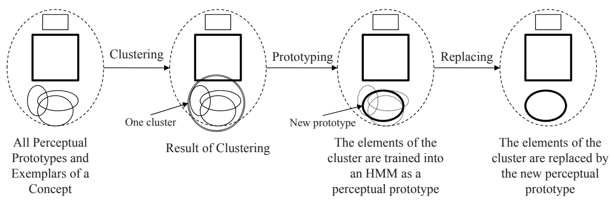


Fig. 3: An illustration for the process of clustering and producing perceptual prototypes.

In the pseudo-code of the algorithm, there are also some other functions, namely **New_W**, **New_L**, and **New_C** to make new exemplars, prototypes, and concepts, respectively:

$$\mathbf{New\_W} : \Omega(\mathbf{x}) \rightarrow WM \tag{21}$$

$$\mathbf{New\_L} : WM \cup LTM \rightarrow LTM \tag{22}$$

$$\mathbf{New\_C} := LTM \rightarrow Q \tag{23}$$

where, $\Omega(\mathbf{x})$ denotes the set of all finite sequences (or spatio-temporal signals). Hence, in **New_W** function, a new HMM is trained with the perceived sequence and stored in the WM.

C. Effect-Based Conceptual Imitation Learning Algorithm

Another possible approach for conceptual imitation learning is to use functional effects of actions. More specifically, the functional similarities between demonstrations are identified by their common effects. In this case, there is no need of interaction with the teacher (and consequently the reinforcement signal) to understand the function of demonstrations. The only thing to do is to recognize the effects and then encode, organize, and categorize the perceived demonstrations in the memory by formation of appropriate prototypes and exemplars (similar to the procedures explained for RBCIL algorithm in the previous section).

The schematic illustration of EBCIL algorithm is shown in Fig. 4. Note that in this algorithm the robot should have the prior knowledge of classifying effects of action, and subsequently this categorization of effects is transformed to conceptualization of actions through the learning process. Thus, at the end of the learning phase, the agent can recognize and predict concept (effect) of novel demonstrations. In addition, the agent will be capable of performing appropriate actions to realize the learned concepts (effects).

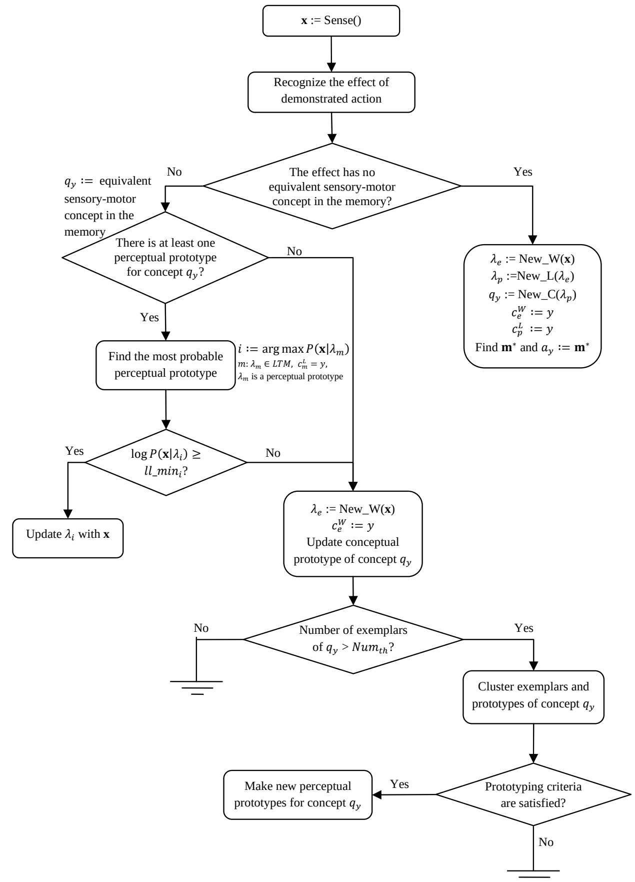


Fig. 4: Schematic diagram of EBCIL algorithm.

D. Recall Phase

After learning and in the recall phase, there is no external information to retrieve the concepts. So, the robot should use the acquired knowledge in the learning phase to classify concept of each novel demonstration and produce appropriate motor actions to realize that concept. For this purpose, we follow the inference method in Section IV-A1, using only the HMM prototypes in the LTM. After inferring the concept and corresponding HMM prototype, a generalized motion pattern is generated by that HMM (if it is a perceptual prototype) and transformed into motor commands through the robot's inverse models. However, if it is a conceptual HMM prototype and there is no perceptual prototype for that concept, the motor program of the concept stored in the memory is used to retrieve appropriate motor commands.

V. Experimental Scenario 1: Conceptual Hand Gesture Imitation through Interaction with the Teacher

To test the proposed RBCIL algorithm in a human-robot interaction task, we set up an experiment, which can be called conceptual hand gesture imitation. In this experiment, five people are asked to draw six signs by moving their hands in the air. Signs are “Heart”, “Rectangle”, “Infinity”, “Tick”, “Arc”, and “Eight”. Each sign might be produced with perceptually different hand trajectories. For example, the Tick sign might be sketched from left to right or from right to left, but the functional meaning of both sketches is the same for the subjects. In our experiment, we have one perceptual representation for Rectangle and one perceptual representation for Infinity but two representations for each remaining sign. Samples of demonstrated hand gestures are provided in Fig. 5.

Fig. 5: Samples of demonstrated signs by the subjects. The arrows show the direction of hand movement.

The demonstrations are incrementally provided to the robot. The robot is the Aldebaran Robotics® Nao humanoid robot. The task is to learn conceptual imitation of the hand gestures. It means that each sign is considered as a distinctive concept which could have irregularly scattered representations in the robot’s perceptual space. So, we are dealing with relational concepts in this problem.

In the following sections, the details of each experiment are explained and the results are reported and analysed comprehensively. In addition, we try to compare the proposed algorithm with some other methods. However, there are a few methods which are conceptually comparable to our algorithm. In fact, our proposed algorithm aims to recognize and regenerate relational concepts, but other methods usually considers perceptual concepts. As explained in Section II, RBCIL is the modified version of the algorithm in [9], [29]. The main distinction is learning and using conceptual prototypes (besides the perceptual prototypes) in RBCIL algorithm. Thus, we provide a detailed comparison between RBCIL algorithm and the algorithm without conceptual prototypes in all experiments of this section. Moreover, we compare the final recognition results of RBCIL algorithm with some base-line methods for motion abstraction and recognition. But, we should note that our algorithm has an inherent superiority over these methods, which is autonomous formation of perceptual variants of a concept.

A. Learning through Visual and Motor Representations of Demonstrations

This section explains the experiments conducted by visual and motor representations of the demonstrated gestures. The visual data represents the hand path trajectory in the robot’s visual space. However, instead of the absolute values of hand location in the image coordinates, we use the relative displacements (i.e., $\{d_x, d_y\}$) to form the perception sequences entering the learning algorithm. So, the results would be invariant to the translational and rotational transformations in the camera coordinate. On the other hand, the motor data represents the transformed trajectory from resulting visual sequence to the joint angles of the Nao’s arm. This transformation is performed by using the Nao’s built-in module of inverse kinematics, which also considers the robot’s affordances. Throughout this paper, the trajectory of roll and pitch angles of the Nao’s elbow and shoulder joints recorded during the robot’s performance is called as motor data. To understand the procedure of visual hand motion extraction and tracking, interested reader is referred to [9], [35].

210 demonstrations were incrementally provided for the robot, including 43 demonstrations for Heart (22 for the 1st perceptual representation and 21 for the second one), 23 demonstrations for Rectangle, 20 demonstrations for Infinity, 42 demonstrations for Tick (21 for each perceptual representation), 42 demonstrations for Arc (21 for each perceptual representation), and 40 demonstrations for Eight (20 for each perceptual representation). We employed our proposed algorithm to learn the concept of demonstrated hand gestures. In the concept learning algorithm, we set $K_{cutoff} = 0.5$, $Num_{th} = 3$, and the number of states for HMMs was set to 10. In the learning algorithm, minimum number of elements to form a new perceptual prototype was set based on the following rule. There should be at least one perceptual prototype and one exemplar or three exemplars in a candidate cluster to make a new perceptual prototype. We used k-fold cross-validation with $k = 5$ to evaluate the performance of our algorithm. So, the experiment was repeated five times with different combinations of demonstrations for training and test.

1) Results of the Experiments with Visual Data: The reinforcement signals (average of five experiments) issued by the teacher during the learning phase for the training demonstrations are depicted in Fig. 6(a). More precisely, this plot shows the first reinforcement of the teacher for each demonstration. Thus, this plot illustrates the performance of the proposed algorithm for incremental recognition. Note that due to the discrete nature of the reinforcements (+1 for reward, and -1 for punishment), the results in the figure has been smoothed with a window length of 10 to clearly reflect the expected behavior. In addition, this figure provides a comparison between the RBCIL algorithm and the algorithm without conceptual prototypes. It can be observed that using conceptual prototypes helps for better performance at the first demonstrations. The reason is that the conceptual prototypes of RBCIL algorithm make the LTM keep at least one representative for each perceived concept throughout the learning. However, the algorithm with no conceptual prototypes should wait for emergence of perceptual prototypes. Hence, the contents of the LTM are limited in this case, and consequently the LTM is not capable of recognizing the previously seen concepts at first demonstrations. But, by increasing number of demonstrations, there will be no considerable difference between the two methods since the equivalent perceptual prototypes of the concepts are produced. Nevertheless, note that higher correct recognition rate at the initial stages of learning is crucial for the real world applications.

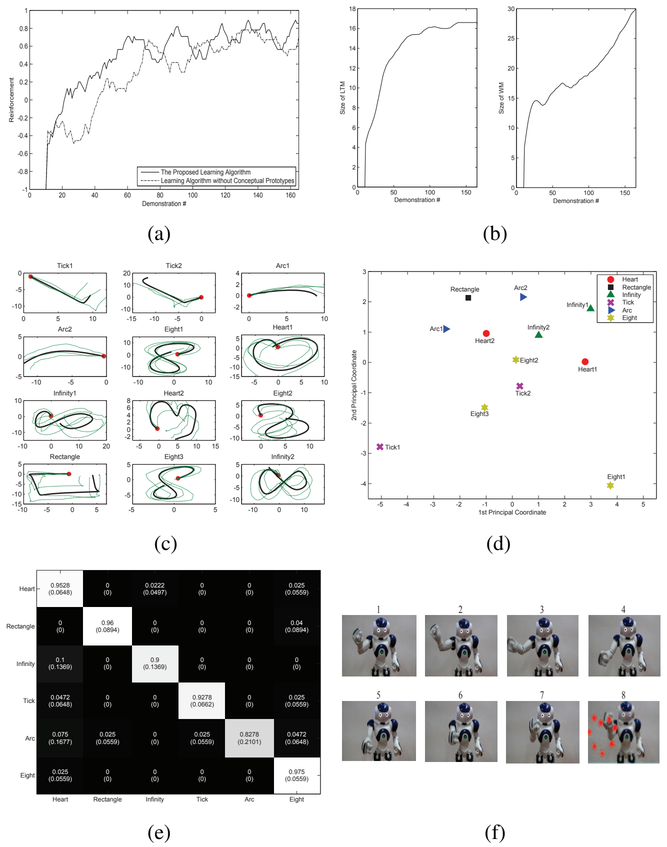


Fig. 6: Results of the experiments with RBCIL algorithm on visual data. (a) Average reinforcement over training demonstrations with and without using conceptual prototypes. (b) Load in long-term memory and working memory during learning process. (c) Generated motion trajectories (solid lines) by the resulting perceptual prototypes of the 2nd fold in the experiments with the visual data, accompanied by some of their training samples, denoted by dashed lines (the red circles show the starting points). (d) Proto-symbol space of the resulting perceptual prototypes of the 2nd fold. (e) Average normalized confusion matrix. (f) An example of the signs (the Heart sign) regenerated by the robot

Fig. 6(b) shows the average smoothed size of the LTM and WM during learning. Number of perceptual prototypes produced at the end of the learning process of each experiment is also provided in Table I. In most cases, the proposed algorithm finds the same number of perceptual prototypes as the number of perceptual variants of each sign shown in Fig. 5. In sum, however, there might be one or two prototypes more than what is expected. For example, in the second fold, three prototypes have been produced for the Eight sign, but two perceptual representations were considered for this sign in the task. The reason is that the teachers are free to sketch the signs, and so their demonstrations are not completely the same. As a result, some samples of the same type might be more similar to one another than the others, and consequently they are clustered into one prototype when they are incrementally entering the learning algorithm in a special order. Particularly, since the visual features making perceptions out of demonstrations are not scale invariant, motion samples of different scales could make different perceptual prototypes. To illustrate this fact, motion trajectories generated by the perceptual prototypes of the second fold are demonstrated in Fig. 6(c). It can be seen that although the generated motion sequences for the Eight1 and Eight3 are perceptually similar at the first glance, but they are different in scale.

TABLE I: Number of Perceptual Prototypes Generated for Each Concept in the Experiments with the Visual Data

Fold no.	Heart	Rectangle	Infinity	Tick	Arc	Eight	Total
1	2	1	1	2	2	2	10
2	2	1	2	2	2	3	12
3	2	1	2	2	2	2	11
4	2	1	2	2	2	2	11
5	2	1	1	2	2	2	10

We also illustrate the proto-symbol space [7] of the perceptual HMM prototypes of the second fold in Fig. 6(d). This space is constructed based on distances between all pairs of the perceptual prototypes in the LTM, using multidimensional scaling [36]. Distance between each pair of HMMs is obtained according to (18) and (19). In Fig. 6(d), the first two principal coordinates of multidimensional scaling have been used to visualize dissimilarity of perceptual prototypes in the proto-symbol space. In our work, this figure illustrates categorization of relational concepts in perceptual space, where the prototypes of a concept might be represented differently.

After the learning phase is completed, we use the prototypes in the LTM to classify the concept of test demonstrations, following the instructions in Section IV-D. The average correct classification ratio for the five folds of the test data is 0.9235 with standard deviation of 0.0458. The average normalized confusion matrix obtained in this experiment is also illustrated in Fig. 6(e). It can be observed that recognition accuracy of the Arc concept is the lowest. It is because we used the same number of states (i.e., 10) to train all the HMMs. However, the appropriate number of states varies according to complexity of each sign. In fact, to have a better performance for recognition of the Arc sign which has a simple waveform (shape), less number of states should be used. But, it is not easy to automatically estimate the optimal number of states a priori in our incremental algorithm. One possible solution is to use factorial hidden Markov models (FHMM) according to the explanations in [20]. Indeed, FHMMs can be employed to adaptively change complexity of the models. Hence, simple waveforms can be encoded with less number of states, while more complex waveforms or perceptually similar waveforms which require more discriminative models can be trained with more complex structures.

Finally, an example of the signs (the Heart sign), regenerated by the Nao through the learned perceptual prototype of the proposed learning algorithm, is demonstrated in Fig. 6(f).

2) Results of the Experiments with Motor Data: In this section, we use the 4-dimensional sequences of arm joint angles to learn the concepts. This approach is motivated by a postulation about mirror neurons (located in the F5 area of the macaque's brain) that suggests gesture recognition/imitation is performed in motor terms [37]. The results of this experiment are summarized as follows. Fig. 7(a) shows the average reinforcement over demonstrations during the learning phase. Again, it is noticeable that the proposed algorithm outperforms the algorithm without conceptual prototypes for the first demonstrations.

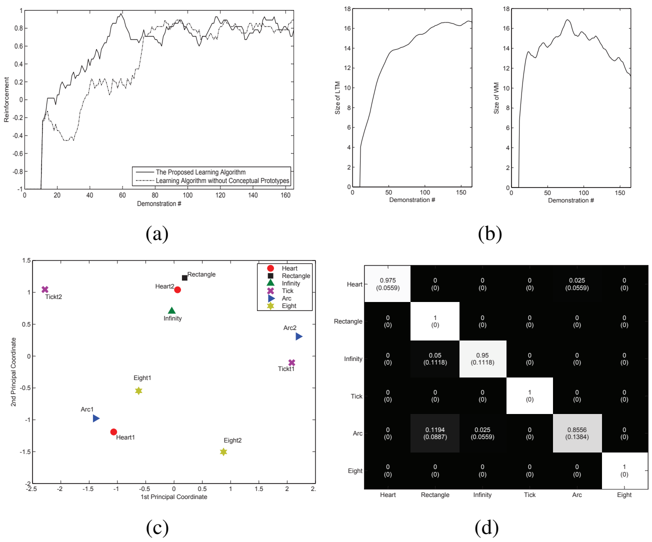


Fig. 7: Experimental results with RBCIL algorithm on motor data. (a) Average reinforcement over training demonstrations with and without using conceptual prototypes. (b) Load in long-term memory and working memory during learning. (c) Proto-symbol space of the resulting perceptual prototypes of the 1st fold. (d) Average normalized confusion matrix.

Fig. 7(b) shows the average smoothed size of the LTM and WM throughout the learning phase. Moreover, the number of perceptual prototypes produced after the learning are provided in Table II. Also, Fig. 7(c) shows the proto-symbol space of these prototypes.

TABLE II: Number of Perceptual Prototypes Generated for Each Concept in the Experiments with Motor Data

Fold no.	Heart	Rectangle	Infinity	Tick	Arc	Eight	Total
1	2	1	1	2	2	2	10
2	2	1	2	2	2	2	11
3	2	1	2	2	2	2	11
4	2	1	1	2	2	2	10
5	2	1	2	2	2	2	11

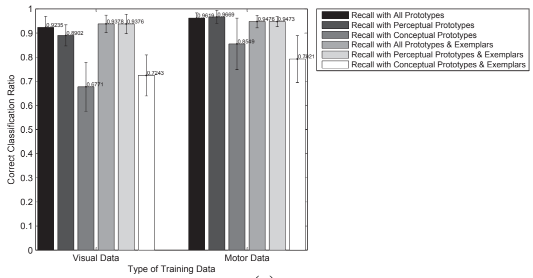
In recall phase, the algorithm achieves average correct classification ratio of 0.9619 with standard deviation of 0.0213 on the test data. Note that the recognition rate by motor data is about 4% better than the recognition rate by the visual data reported in Section V-A1. This is because the motor representations (which are 4D signals) contain more information than the visual representations (which are 2D signals). Finally, the average normalized confusion matrix for this experiment is illustrated in Fig. 7(d).

3) Comparison Between the Proposed Algorithm and Some Alternative Algorithms: In this section, we justify our proposed algorithm experimentally by comparing with some alternative algorithms. First, performance of the algorithm is evaluated with different types of trained HMMs, and next the algorithm is compared with standard HMM-based batch algorithms.

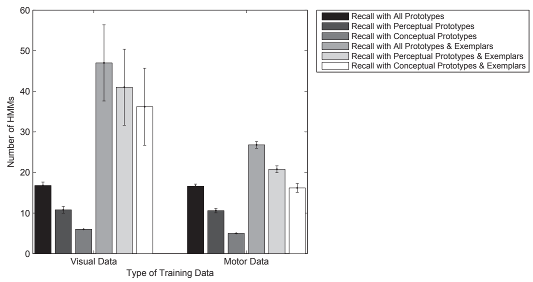
Comparison of Recognition Results with Different Types of HMMs: As described in Section IV-D, perceived demonstrations are classified based on all the prototypes in the LTM. But, there are also some other possibilities. For example, one might use only perceptual prototypes or both prototypes and exemplars. A comparison of recognition results on test data with different types of HMMs is demonstrated in Fig. 8(a). The number of HMMs used for each option is also provided in Fig. 8(b) to have an estimation about the computational costs. It can be seen that recognition with both prototypes (all prototypes or only perceptual prototypes) and exemplars have the highest recognition rate for the visual data (marginally better than the algorithms without exemplars). But, the computational cost of these algorithms are much more than the computational cost of the algorithms using only prototypes (which are about three times less costly). On the other hand, for the motor data, recognition accuracy without exemplars is even better than recognition accuracy with exemplars. In fact, the exemplars in the proposed learning algorithm encode the outlying samples which could not put into clusters with other samples. Consequently, it can be declared that the exemplars usually represent the special cases of a concept. So, it seems even better to separate these samples in order to have more unified and general models.

It is also noticeable in Fig. 8(a) and Fig. 8(b) that using only perceptual prototypes leads to less computational cost than using all the prototypes, while both the approaches have almost the same recognition rate. However, as shown in the previous sections, using both conceptual and perceptual prototypes have better recognition results at the first demonstrations. Indeed, conceptual prototypes can help to fairly represent the concepts until the perceptual prototypes emerge. Note that in the real-life applications, the agent should be capable of extracting the maximum performance out of the learning knowledge while it is living. Thus, to have a consistent approach throughout the algorithm and satisfying the so-called learning by living criteria, we suggest to use all the prototypes for concept recognition in the recall phase.

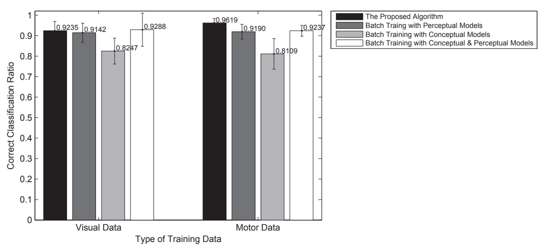
Comparison between the Proposed Algorithm and Standard HMM-Based Batch Algorithms: Now, we compare our proposed algorithm with the algorithms using HMMs in a batch training process. In the batch training, all the demonstrations should be prepared, grouped, and labelled a priori before learning gets started. We consider two algorithms: 1) batch training with conceptual models and 2) batch training with perceptual models. In the former, all samples of one concept



(a)



(b)



(c)

Fig. 8: Comparison of the algorithm performance with different types of HMMs in terms of (a) recognition accuracy on test data, and (b) number of HMMs.
(c) Comparison between the proposed algorithm and the batch algorithms in terms of recognition accuracy on test data.

are trained by one HMM. Hence, the number of HMMs is exactly the same as the number of concepts. In this algorithm, the resulting HMMs can be used for concept recognition, but they cannot be used independently for regeneration. Actually, since different perceptual variants of a concept are pushed into one model, their perceptual features are mixed up, and the regenerated trajectory might be perceptually similar to none of the variants. In the second algorithm, samples of the same perceptual type (variant) are trained separately with one HMM, and consequently, each concept is made up of the HMMs representing its perceptual variants. For example, in our experimental scenario, there are two HMMs for the Heart, Tick, Arc, and Eight signs and one HMM for the Rectangle and Infinity signs (summing up to 10 HMMs). Contrary to the first algorithm, this algorithm can be used for both recognition and regeneration purposes. Note that in the experiments of this section, we used the same parameters and initializing process for the HMMs as those described in Section V-A.

Fig. 8(c) shows the comparison between the recognition results of our proposed learning algorithm and the batch training algorithms on test demonstrations. Since our algorithm uses both conceptual and perceptual prototypes for concept recognition, we have also included the recognition results obtained by batch-trained conceptual and perceptual models together (although it is not a standard batch training algorithm). It can be observed that although our proposed algorithm is incremental and the perceived demonstrations are not labelled a priori in this algorithm, it outperforms the batch training algorithm with conceptual models, and also it is slightly better than batch training algorithm with perceptual models. In addition, our incremental algorithm is comparable to the batch training algorithm with both conceptual and perceptual models using the visual data, and it is even better than this algorithm using motor data. There are two reasons for this outcome. First, our incremental learning algorithm detects the perceptual variants of a concept automatically according to the robot's perceptual space. For example, the algorithm makes different prototypes for samples of different scales, considering the robot's perceptual space which is scale variant. The second reason is that the proposed algorithm inhibits the outlying samples from being trained into the HMM prototypes. Thus, the resulting prototypes are more general and unified. It can be also understood from Fig. 8(c) that batch training with perceptual models have higher recognition accuracy than batch training with conceptual models. Actually, it is an expectable outcome since the perceptual models lead to a more strong and comprehensive representation of the concepts (because the perceptual features of demonstrations are better consolidated in perceptual models). However, it should be considered that the perceptual variants of the concepts should be manually guessed in this algorithm (i.e., batch training algorithm with perceptual models), which is not very straightforward.

B. Learning through Auditory Representations of Demonstrations

Execution of some actions is accompanied by sounds, e.g. knocking the door, flipping the papers, breaking a peanut, etc. Actually, we can recognize these actions from their sounds. In this experiment, we simulate attribution of some auditory signals to the previously explained hand gestures. It means that the teacher utters a word while he is demonstrating a gesture. The uttered words are Persian numbers from 0 to 9, i.e., 0 and 1 for the two perceptual representations of the Heart, 3 for the Rectangle, 6 for the Infinity, 2 and 5 for the two representations of the Tick, 4 and 7 for the two representations of the Arc, and finally, 8 and 9 for the two representations of the Eight. The robot should hear the uttered word and observe the demonstration performing alongside. Next, the speech signal is provided to the learning algorithm as the perception entry, and the motor representation of the observed demonstration is saved as the motor action of the associated concept. As a result, the robot learns to recognize the concept of each demonstration through perception of its accompanied sound, but employs the stored motor action of that concept to regenerate the concept-equivalent gesture.

In this experiment, first the robot should extract features out of perceived pure speech signals. For this purpose, we use the Mel-frequency cepstral coefficients (MFCCs), which are popular and well-known spectral features for automatic speech recognition [38]. Through this process, 13 features can be extracted for each speech frame. However, in our experiments we employ only the first four features. Consequently, the resulting signal is a 4 dimensional sequence of MFCCs. For the learning algorithm, we used the same settings as explained in Section V-A except that $K_{cutoff} = 0.9$ and the number of HMM states is equal to 5.

1) Results of the Experiments with Auditory Data: Fig. 9(a) shows the average reinforcement over demonstrations during the learning phase. Again, it is evident that using conceptual prototypes leads to better performance at first demonstrations.

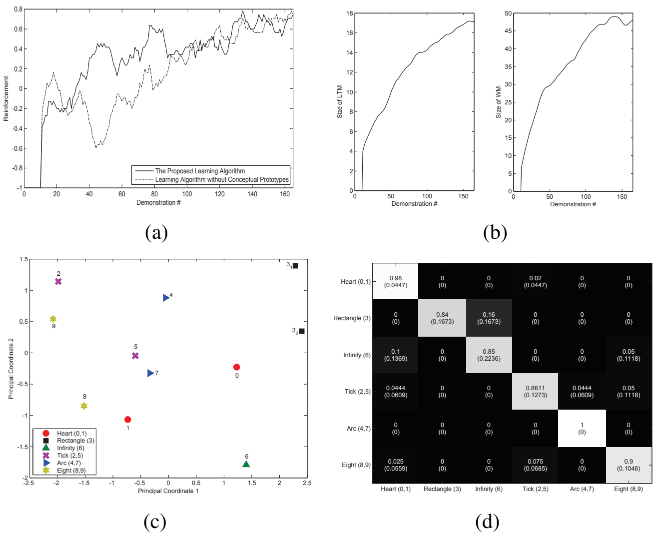


Fig. 9: Experimental results with RBCIL algorithm on auditory data. (a) Average reinforcement over training demonstrations with and without using conceptual prototypes. (b) Load in long-term memory and working memory during learning. (c) Proto-symbol space of the resulting perceptual prototypes of the 1st fold. (d) Average normalized confusion matrix.

Fig. 9(b) shows the average smoothed size of the LTM and WM throughout the learning phase. Moreover, the number of perceptual prototypes produced at the end of the learning phase are provided in Table III. Accordingly, the proto-symbol space of the perceptual prototypes in the first fold are illustrated in Fig. 9(c).

TABLE III: Number of Perceptual Prototypes Generated for Each Concept in the Experiments with Auditory Data

Fold no.	Heart	Rectangle	Infinity	Tick	Arc	Eight	Total
1	2	2	1	2	2	2	11
2	3	1	1	3	2	2	12
3	3	1	2	2	2	2	12
4	2	1	1	2	2	3	11
5	1	1	1	2	3	2	10

After accomplishment of the learning phase, the obtained prototypes are used to recognize the concept of test demonstrations. the average recognition accuracy on test data is 0.9147 with standard deviation of 0.0391. Finally, the average normalized confusion matrix for this experiment is illustrated in Fig. 9(d).

C. Concept Recognition by Multimodal Integration of Heterogeneous Perception Sequences

As shown in the previous sections, a concept can be represented by perceptual entries from different modalities. For example, a concept can be understood by the perceived information from visual, auditory or kinesthetic sensory systems. There is also a well-known postulation that mirror neurons which map perception to action are multimodal, i.e., they respond to actions perceived from multiple modalities [17].

In this section, we aim to propose a solution to integrate heterogeneous perceptions from distinct modalities to improve concept recognition. Multimodal integration helps to compensate ambiguity or lack of information in one modality with information from other modalities. So, the robustness is increased, and the proposed imitation model becomes more practical. To this end, a multimodal concept is defined as a set of prototypes obtained from different modalities. In this case, whenever a new demonstration is perceived only through one modality, it is compared against prototypes of the same modality, and the most probable prototype retrieves the concept. However, the problem arises when heterogeneous perceptions of different modalities are sensed simultaneously. So, there should be a mechanism to integrate all this information and make final decision. Thanks to the stochastic representation of concepts endowed by HMMs, multimodal integration is facilitated in our model. In fact, one of the main challenges in multimodal signal processing is that the measures in different modalities are not comparable. However, probability is a per-unit value without dimension which makes comparison meaningful between heterogeneous modalities. So, with the independence assumption between different modalities, the probability of heterogeneous perception sequences can be multiplied by each other. According to what explained above, the likelihood of heterogeneous perception sequences is computed against available multimodal concepts, using the following formula:

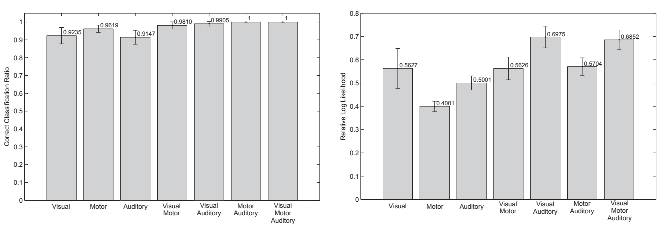
$$\log P\left(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L | q_y\right) = \max_{\lambda_{i1} \in q_y} \frac{\log P\left(\mathbf{x}_1 | \lambda_{i1}\right)}{T_1} + \max_{\lambda_{i2} \in q_y} \frac{\log P\left(\mathbf{x}_2 | \lambda_{i2}\right)}{T_2} + \cdots + \max_{\lambda_{iL} \in q_y} \frac{\log P\left(\mathbf{x}_L | \lambda_{iL}\right)}{T_L}, \tag{24}$$

where, L is the number of modalities, $\mathbf{x}_l$ $(l = 1, 2, \cdots, L)$ is the perception sequence of the lth modality with length T_l, and λ_{il} is the ith prototype of the lth modality. The reason for division by T is to normalize the weight of each modality to become invariant to the sequence length. Finally, the most probable concept is retrieved according to (25).

$$y^{\star} = \underset{y \in \mathbf{N}_{|Q|}}{\arg\max} \log P\left(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L | q_y\right). \tag{25}$$

1) Results of Concept Recognition by Multimodal Integration of Heterogeneous Perception Sequences: In this section, we use the resulting prototypes of the learning algorithm in Sections V-A1, V-A2, and V-B1 to recognize the concept of test demonstrations by the method presented in Section V-C.

Note that the experiment is evaluated with a 5-fold cross-validation. Fig. 10(a) reports the recognition accuracy on test data for each type of data, including visual, motor, auditory, visual-motor, visual-auditory, and visual-motor-auditory data. It can be observed that integration of different modalities improves the recognition performance. It is also noticeable that the motor-auditory and visual-motor-auditory integrations lead to recognition rate of 100%. In addition, it can be seen that integration of audition with the other modalities has better results than visual-motor integration in this experiment (although single usage of auditory data has the worst result). The reason is that visual and motor data have been extracted form the same source of information, but auditory data is from a completely different source. Hence, in this experiment auditory integration could better compensate the ambiguity in the two other modalities.



(a) (b)

Fig. 10: Multimodal concept recognition. (a) Comparison of the recognition rates for different modalities. (b) Comparison of the recognition confidence in terms of log likelihood for different modalities.

In addition, to compare the confidence of recognizing the true concept with different modalities, we show the recognition results of test data (in terms of log likelihood) in Fig. 10(b). The vertical axis of this plot is defined by:

$$\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \frac{1}{L} \Bigg(\log P\left(\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_L^i | q^{true}\right) - \max_{y: q_y \neq q^{true}} \log P\left(\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_L^i | q_y\right) \Bigg), \tag{26}$$

where, N_{test} indicates the number of test demonstrations. This figure shows that integration of the modalities even increases the true recognition confidence.

VI. Experimental Scenario 2: Conceptual Hand Gesture Imitation through Recognition of Emotional Effects

In this section, we evaluate the performance of EBCIL algorithm in an experimental scenario. In this experiment, the concept of demonstrations are specified by their emotional effects. The experiment is as follows. There are three participants: a robot, a teacher, and a human affected by the robot's action (we call him the third agent). The robot is the Nao humanoid robot introduced in Section V. Demonstrations are provided for the robot by kinesthetic teaching. It means that the teacher grabs the robot's arm and performs an action, e.g. strikes at the third agent. Next, the third agent shows an emotional response (reaction) to the the teacher's action, e.g. gets angry. The robot perceives its arm joint variables during performance of the action. Indeed, the perceived signal is a 4 dimensional sequence of roll and pitch angles of the elbow and shoulder joints in the robot's right arm. The robot also recognizes the emotional response of the third agent after action execution. So, it can understand functional similarities between different demonstrations based on the sameness of their emotional effects. For example, whether the teacher strikes the third agent from right or left, emotional response of the third agent will be the angry facial expression.

All the concepts accompanied by their actions and emotional responses are summarized in Table IV In all the concepts, the third agent's response is identified by his facial expression except for the Love concept. For this concept, the response is to stimulate the tactile sensor on the Nao's head by caressing. For facial expression detection, the simple algorithm[2] of *Eigenfaces* [39] is employed after cropping the face image using the algorithm of Pai et al. [40]. Total number of demonstrations in this experiment is 120, i.e., 15 demonstrations for each action. For the concept learning algorithm, we use the same settings as explained in Section V-A, i.e., $K_{cutoff} = 0.5$, $Num_{th} = 3$, and 10 states for each HMM. Like the previous experiments, the results are evaluated with a 5-fold cross-validation.

A. Results

Results of the experiments are summarized as follows. To show the progress of recognition accuracy during learning (like the previous experiments), a scoring system is used. In this system, whenever a demonstration is perceived, first it is classified by the previously learned concepts before entering to the learning algorithm. If the perception sequence is classified correctly, a positive score (+1) is issued, else a negative score (-1) is recorded. The smoothed average scores (over the five folds) recorded during the learning process on the training data is illustrated in Fig. 11(a). It can be observed in the figure that the robot successfully recognizes the concept of almost all new perception sequences after being trained by a few demonstrations. In addition, the average smoothed size of the LTM and WM throughout the learning phase is shown in Fig. 11(b).

The Number of perceptual prototypes produced at the end of learning by each fold is reported in Table V. In most cases, EBCIL algorithm finds the same number of perceptual prototypes as the number of actions which represent perceptual variants of a concept. We also illustrate the proto-symbol space of perceptual prototypes of the first fold in Fig. 11(c).

After termination of the learning phase, we use the resulting prototypes of the LTM to recognize concept of the test demonstrations (according to the explanations in Section IV-D). The outcome is that the algorithm can correctly classify all the test demonstrations. Finally, the action execution results are presented in Fig. 11(d), which demonstrates samples of

[2]Since in our experiments facial expression classifier is trained and test on the same person, this algorithm can be accurate enough.

TABLE IV: All the Concepts Accompanied by their Actions and Emotional Responses

#	Concept	Robot's Action 1	Robot's Action 2	Robot's Action 3	Third Agent's Response
1	Anger	Striking from Right	Striking from Left	-	Angry Face
2	Unhappiness	Hitting on Head	Hitting on Chest	-	Unhappy Face
3	Happiness	Throwing Fist Up & Down	-	-	Happy Face
4	Love	Sketching Heart Sign	Air Kiss	Caressing the Face	Caressing the Robot's Tactile Sensor

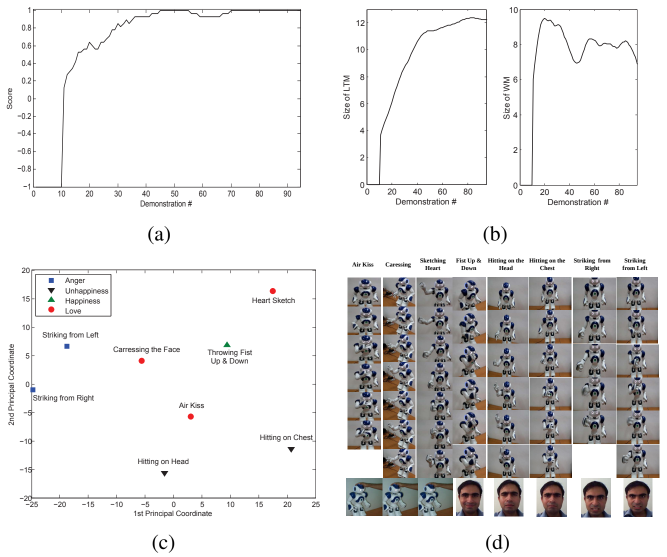


Fig. 11: Results for the experimental scenario 2. (a) Average recorded score over demonstrations during learning process using EBCIL algorithm. (b) Average load in long-term memory and working memory during learning. (c) Proto-symbol space of the resulting perceptual prototypes of the 1st fold. (d) Samples of regenerated actions by the Nao humanoid robot.

TABLE V: Number of Perceptual Prototypes Generated for Each Concept in the Experiments of the Experimental Scenario 2

Fold no.	Anger	Unhappiness	Happiness	Love	Total
1	2	2	1	3	8
2	2	2	2	3	9
3	2	2	1	3	8
4	2	2	1	3	8
5	2	2	2	3	9

reproduced actions by the Nao humanoid robot accompanied by their equivalent emotional responses.

VII. Conclusion

In this paper, we introduced a model for conceptual imitation learning. Conceptual imitation tries to model true imitation in human and mammals, which is said to be fulfilled by abstraction, generalization, recognition, and regeneration of action. Due to abstraction and generalization, the learning agent can even recognize novel demonstrations of previously learned concepts and regenerate generalized motion patterns for these concepts. But, imitation learning based on relational concepts leads to abstraction in both perceptual and functional space and consequently leads to less number of concepts (since perceptual variants of a function are all put into one concept). As a result, the agent will have a smaller representation of the world in its mind, and so it can interact with the world more simply and search in its memory more easily. In addition, the functional abstraction causes ease of knowledge transfer and flexibility of choice between action alternatives.

The main contribution of this paper was to devise incremental and gradual learning algorithms for autonomous learning and acquisition of relational concepts from spatio-temporal demonstrations, using perceptual and functional characteristics of action. Functional similarities between demonstrations were identified by interaction with the teacher (RBCIL algorithm) or recognizing the effects of actions (EBCIL algorithm). HMMs were used to abstract perception sequences into stochastic prototypes and exemplars. Consequently, relational concepts formed as a set of irregularly scattered HMMs unified based on their functionalities. In addition to above, a modified conceptual representation was proposed for learning multimodal concepts. In fact, the original model is modified to a multimodal model which can recognize multimodal concepts by integration of heterogeneous perceptions. We also showed how recognizing functional effects of action can remove the need of interaction, and how the knowledge of classifying the effects leads to the knowledge of conceptualizing the actions.

We evaluated the proposed RBCIL algorithm in an experimental scenario, namely conceptual hand gesture imitation through interaction with the teachers. The experiment was conducted on the Nao humanoid robot. Results showed that the proposed model is successful for parallel acquisition of concepts, emergence and self-organization of prototypes, recognition, and regeneration of demonstrated gestures. It was also shown that our incremental learning algorithm can even outperform the base-line batch algorithms. This successful outcome was due to autonomous formation of prototypes proportionate to the agent's perceptual space and also inhibition of prototyping the outliers by separating them into exemplars. In addition, an experiment was performed for multimodal concept recognition. Simulation results showed that our model can be successfully fit to the multimodal concepts. As a result of the proposed multimodal integration, both recognition accuracy and recognition confidence were increased in this experiment.

Another experimental scenario was also carried out to evaluate performance of the proposed EBCIL algorithm for conceptual imitation learning based on functional effects of action. In this experiment, the robot conceptually imitates a number of moving hand gestures by recognition of their emotional effects. The experiments, conducted on the Nao, showed the successful results of our model for learning and acquisition of all the concepts and regeneration of their equivalent actions. As a result, the robot transforms to an effective agent which is capable of predicting effects (concepts) of novel demonstrations and also realizing these effects by execution of appropriate actions.

ACKNOWLEDGMENT

The authors wish to thank Behrouz Haji Soleimani[3] for providing us with the speech signals of Persian numbers.

REFERENCES

[1] F. Chersi, "Learning through imitation: a biological approach to robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 3, 2012.

[2] M. Arbib, "The Mirror System, Imitation, and the Evolution of Language," *Imitation in animals and artifacts*, p. 229, 2002.

[3] C. Breazeal and B. Scassellati, "Challenges in Building Robots That Imitate People," *Imitation in animals and artifacts*, pp. 363–390, 2002.

[4] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 363–377, 2004.

[5] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," *Handbook of robotics*, pp. 1371–1394, 2008.

[6] H. Kadone and Y. Nakamura, "Segmentation, memorization, recognition and abstraction of humanoid motions based on correlations and associative memory," in *6th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS*, 2006, pp. 1–6.

[7] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," in *6th IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS*, 2006, pp. 425–431.

[8] H. Mobahi, M. Nili Ahmadabadi, and B. Nadjar Araabi, "A biologically inspired method for conceptual imitation using reinforcement learning," *Applied Artificial Intelligence*, vol. 21, no. 3, pp. 155–183, 2007.

[9] H. Hajimirsadeghi, M. Nili Ahmadabadi, B. Nadjar Araabi, and H. Moradi, "Conceptual imitation learning in a human-robot interaction paradigm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 2, p. 28, 2012.

[10] H. Hajimirsadeghi, "Conceptual imitation learning based on functional effects of action," in *EUROCON-International Conference on Computer as a Tool*. IEEE, 2011, pp. 1–6.

[11] C. Breazeal and B. Scassellati, "How to Build Robots that Make Friends and Influence People," in *RSJ International Conference on Intelligent Robots and Systems (IROS-99)*, 1999.

[12] R. Kelley, A. Tavakkoli, C. King, A. Ambardekar, and M. Nicolescu, "Context-based bayesian intent recognition," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 3, 2012.

[13] T. R. Zentall, M. Galizio, and T. S. Critchfied, "Categorization, concept learning, and behavior analysis: an introduction." *Journal of the experimental analysis of behavior*, vol. 78, no. 3, p. 237, 2002.

[14] H. Friedrich, S. Muench, R. Dillmann, S. Bocionek, and M. Sassin, "Robot programming by demonstration (rpd): Supporting the induction by human interaction," *Machine Learning*, vol. 23, no. 2-3, pp. 163–189, 1996.

[15] S. Wermter, M. Elshaw, C. Weber, C. Panchev, and H. Erwin, "Towards integrating learning by demonstration and learning by instruction in a multimodal robot," in *Proceedings of the IROS-2003 Workshop on Robot Learning by Demonstration*, 2003, pp. 72–79.

[16] S. Wermter, C. Weber, M. Elshaw, C. Panchev, H. Erwin, and F. Pulvermüller, "Towards multimodal neural robot learning," *Robotics and Autonomous Systems*, vol. 47, no. 2, pp. 171–175, 2004.

[17] E. Kohler, C. Keysers, M. A. Umilta, L. Fogassi, V. Gallese, and G. Rizzolatti, "Hearing sounds, understanding actions: action representation in mirror neurons," *Science*, vol. 297, no. 5582, p. 846, 2002.

[18] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg, "Learning from and about others: Towards using imitation to bootstrap the social understanding of others by robots," *Artificial life*, vol. 11, no. 1-2, pp. 31–62, 2005.

[19] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 105–112.

[20] D. Kulic, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *The International Journal of Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008.

[21] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[22] S. Calinon and A. Billard, "Stochastic gesture production and recognition model for a humanoid robot," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2769–2774.

[23] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, 2005, pp. 299–304.

[24] D. Lee, D. Kulic, and Y. Nakamura, "Missing motion data recovery using factorial hidden markov models," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2008, pp. 1722–1728.

[25] M. Donald, *Origins of the modern mind*. Harvard University Press, 1991.

[26] D. Kulic, W. Takano, and Y. Nakamura, "Incremental on-line hierarchical clustering of whole body motion patterns," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2007, pp. 1016–1021.

[27] ——, "Online segmentation and clustering from continuous observation of whole body motions," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 1158–1166, 2009.

[28] H. Mobahi, M. Nili Ahmadabadi, and B. Nadjar Araabi, "Concept oriented imitation towards verbal human-robot interaction," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005*, 2005, pp. 1495–1500.

[29] H. Hajimirsadeghi, M. Nili Ahmadabadi, M. Ajallooeian, B. Nadjar Araabi, and H. Moradi, "Conceptual imitation learning: An application to human-robot interaction," *Journal of Machine Learning Research: Workshop and Conference Proceedings*, vol. 13, pp. 341–356, 2010.

[30] J. K. Kruschke, "Category learning," *The handbook of cognition*, pp. 183–201, 2005.

[31] R. Cook, D. Kelly, and J. Katz, "Successive two-item same-different discrimination and concept learning by pigeons," *Behavioural Processes*, vol. 62, no. 1-3, pp. 125–144, 2003.

[32] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Readings in speech recognition*, vol. 53, no. 3, pp. 267–296, 1990.

[33] E. Oztop, M. Kawato, and M. Arbib, "Mirror neurons and imitation: A computationally guided review," *Neural Networks*, vol. 19, no. 3, pp. 254–271, 2006.

[34] R. C. Schank, G. C. Collins, and L. E. Hunter, "Transcending inductive category formation in learning." *Behavioral and Brain Sciences*, vol. 9, no. 4, pp. 639–651, 1986.

[35] M. Ajallooeian, A. Borji, B. Nadjar Araabi, M. Nili Ahmadabadi, and H. Moradi, "Fast hand gesture recognition based on saliency maps: An application to interactive robotic marionette playing," in *Proceeding of IEEE ROMAN 2009*, Osaka, Japan, 2009.

[36] I. Borg and P. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Verlag, 2005.

[37] L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti, "Visuomotor neurons: ambiguity of the discharge or 'motor' perception?" *International Journal of Psychophysiology*, vol. 35, no. 2-3, pp. 165–177, 2000.

[38] D. O'shaughnessy, *Speech communications: human and machine*. Universities Press, 1998.

[39] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.

[40] Y. Pai, S. Ruan, M. Shie, and Y. Liu, "A simple and accurate color face detection algorithm in complex background," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2006, pp. 1545–1548.

[3]behrouzhaji@ut.ac.ir