An Intrinsic Value System for Developing Multiple Invariant Representations with Incremental Slowness Learning

Matthew Luciw†, Varun Kompella∗†, Sohrob Kazerounian∗, and Juergen Schmidhuber*
*IDSIA, Galleria 2, Lugano-Manno, 6928, Switzerland
† Joint First Authors

Abstract—Curiosity Driven Modular Incremental Slow Feature Analysis (CD-MISFA; [21]) is a recently introduced model of intrinsically-motivated invariance learning. Artificial curiosity enables the orderly formation of multiple stable sensory representations to simplify the agent’s complex sensory input. We discuss computational properties of the CD-MISFA model itself as well as neurophysiological analogs fulfilling similar functional roles. CD-MISFA combines 1. unsupervised representation learning through the slowness principle, 2. generation of an intrinsic reward signal through learning progress of the developing features, and 3. balancing of exploration and exploitation to maximize learning progress and quickly learn multiple feature sets for perceptual simplification. Experimental results on synthetic observations and on the iCub robot show that the intrinsic value system is essential for representation learning. Representations are typically explored and learned in order from least to most costly, as predicted by the theory of curiosity.

I. INTRODUCTION

We describe a model called CURIOUSity-DRiven, Modular, Incremental Slow Feature Analysis (Curious Dr. MISFA), which autonomously explores various action contexts, learning low-dimensional encodings from the high-dimensional sensory inputs (i.e., video) that result from each such context. Autonomous behavior in this regard requires the coordinated interaction between a number of subsystems which enable an agent to balance exploration-exploitation, to engage in useful contexts while disengaging from others, and to organize representations such that newly learned representations do not overwrite previously learned ones. Ultimately, an agent making use of CD-MISFA learns to seek out and engage contexts wherein it expects to make the quickest progress, learns an appropriate compact, context-dependent representation, and upon fully learning such a representation, disengages from that context to enable further exploration of the environment, and learning of subsequent representations. The goal of such an agent is to maximize intrinsic reward accumulation, and as a byproduct learn all such representations that are learnable given the contexts available to it. We not only show why the interacting subsystems of CD-MISFA are necessary for the kind of unsupervised learning it undertakes, but moreover, we show how the subsystems that enable the model to autonomously explore and acquire new sensory representations, mirror the functional roles of some of the underlying cortical and neuromodulatory systems responsible for unsupervised learning, intrinsic motivation, task engagement and task switching.

Although difficult, attempts to integrate such disparate functional subsystems are not only helpful in understanding the brain, but are increasingly necessary for building autonomous artificial and robotic systems. It does not suffice, for example, to know the cortical mechanisms responsible for unsupervised learning of sensory representations, if these mechanisms aren’t linked to the systems responsible for exploring one’s environment. In the absence of external rewards, how should an agent decide which actions and contexts to explore, in order to determine which representations are relevant and learnable? If a sensory representation is deemed overly complex or even unlearnable, what are the mechanisms by which the agent can disengage from exploring its current context, in order to allow it to explore others? Although CD-MISFA is an algorithmic approach to developmental robotics, and does not explicitly model the neural mechanisms by which these functions are realized in the brain, it is notable that the functional roles of the various subsystems in CD-MISFA find counterparts in neurophysiology.

In the following, we first discuss background on CD-MISFA. Artificial Curiosity, and developmental learning, then provide a detailed computational description of how the various sub-systems in CD-MISFA operate and interact, followed by a description of the neurophysiological correlates whose functional roles mirror those of CD-MISFA; namely, the interactions between the neuromodulatory systems involved in intrinsic motivation, task engagement, task switching, and value approximation. CD-MISFA is implemented in two situations: an environment composed of synthetic high-dimensional visual contexts, and a real-world environment, with an actively exploring humanoid iCub robot. A method for measuring the learning cost in the different contexts is introduced, and it is shown that the model is most likely to engage within the context where it can learn an as yet unlearned representation, where the cost is least among all possible contexts; this type of behavior is predicted by the theory of curiosity, and may be a general principle of development. The second result shows that IM-based exploration enables the embodied agent to learn interesting sensory representations, again in the predicted order, all while operating on high-dimensional video streams as sensory input.
II. CURIOUS DR. MISFA

A. Background

Artificial Curiosity. Consider a setting in which an agent operates without a teacher or any other type of external motivation, such as external reward. In this case, an agent needs to be self-motivated, or curious. The Formal Theory of Fun and Creativity [21], [54], [55] mathematically formalizes driving forces behind curious and creative behaviors. This theory requires that a curious agent have two learning components: an adaptive predictor/compressor of the agent’s growing history of perceptions and actions, and a reinforcement learner [62]. The learning progress or expected improvement of the compressor becomes an intrinsic reward for the reinforcement learner. To maximize intrinsic reward accumulation, the reinforcement learner is motivated to create new experiences such that the compressor makes quick progress.

Curiosity and Development. Such a creative agent produces a sequence of self-generated tasks and their solutions, each task still unsolvable before learning, yet becoming solvable after learning. Further, there is an expected order in task-learning. Since the value function of the intrinsic reward contains the cost of learning, in the sense of an estimation of what type of progress it can expect, a task with the lowest cost of learning is preferentially learned next, among all possible tasks.

An orderly acquisition of competence can be seen as a developmental process. An important aspect to development is the gradual emergence of more and more types of skills, knowledge, etc [31], [52], [53]. Such emergence, referred to as developmental stages, can observed through behavioral competence [25]. More specifically, by developmental stages we mean that certain competencies are always seen to precede later ones, although the earlier competencies are not necessarily prerequisites for those learned later (which would be the case in continual learning [45]).

It has been shown in an n-armed bandit scenario that a system based on Artificial Curiosity undergoes developmental stages [35]. Further, when the goal is to maximize expected improvement of the predictor or other world model, it was shown that it is optimal to concentrate on the current easiest to learn task that has not yet been learned [26] (also in a bandit scenario).

However the bandit setting involves initial knowledge of the number of possible tasks, in which case the learner can initially reserve learning resources for each task. This is unrealistic for open-ended autonomous development, in which the number of different tasks is initially unknown. What is learned in one part of the environment could apply to another part of the environment. To enable open-ended learning, CD-MISFA learns one module at a time, and if it finds a context that is represented well by one of its already stored modules, it will not need to assign learning resources or time to that context.

Developmental Robotics. Developmental Robotics aims to discover and implement mechanisms that can lead to emergence of mind in a embodied agent [29]. The underlying developmental program has several general requirements.

- Not Task Specific. The task(s) that the robot will handle, i.e., the skills that it can learn, are not explicitly coded in the program. In CD-MISFA, we have such a situation, as the perceptual representations that emerge are dependent on the statistics of the image sequences that are generated from autonomous exploration of the different contexts.
- Environmental Openness. Can the system handle a wide variety of possibly uncontrolled environments that the designers might not have explicitly thought of? Currently, CD-MISFA specifically requires a designer to define the environment contexts that the robot can explore over, and so this is a drawback of the system.
- Raw Information Processing. Learning is on raw (low-level) information, such as pixels and motor activation values. CD-MISFA slow features are updated directly from pixels, not symbolic inputs or hand-designed feature outputs. On the motor end, the active joints are extremely constrained, but this aspect is low-level as well.
- Online Learning. Batch data collection is avoided completely through the incremental slow feature analysis (IncSFA) technique, IncSFA [20], with which a perceptual representation is updated after each image.
- Continual Learning [25]. For scaling up the machine’s intelligent capabilities, it is necessary that learned skills lead to (or are combined to create) more complex skills. CD-MISFA has not yet demonstrated this, but skill development (albeit in a limited sense) has been shown [21] to be enabled by its representation learning (i.e., exploiting the learned representations for external reward).
tially, a framework for continual learning can be built in here; this is to be explored in future work.

With respect to development, a main contribution of this paper is to show how non-task-specific and low-level visuo-motor interactions can give rise to emergent behaviors, which are at a higher (but not yet conceptual) level. In a set-up environmental context, an agent’s randomly moving effectors (motor babbling) lead to observable consequences involving interactions with the environment not directly controllable by the agent. Slowness learning leads to the emergent “higher-level” representations, since the learning is forced to pay attention to the events that occur on the slower time-scale instead of the regular, but more quickly changing parts of the sensorimotor data stream. For example, a robot that watches its arm can learn causality between its joint controls and the images quite quickly, since there is an abundance of data in babbling — in a sense, this is highly salient. But for the robot to learn about something external to it (i.e., an object), that it interacts with more infrequently, without human supervision, is more difficult, but nonetheless enabled by slowness learning.

**Unsupervised Visuomotor Representation Learning.** There are many works on representation learning, but we are specifically interested in representation learning from high-dimensional image sequences where the sequence results from an agent’s actions. Slow Feature Analysis (SFA; [70]), is well-suited to this case. SFA applies to image sequences, and it provides *invariant* representations, unlike e.g., Principal Components Analysis (PCA; [16]), which provides a compressed representation, but not invariance. SFA is also an appearance-based approach [34], [65]. Appearance-based approaches learn analog world properties (object identity, person identity, pose estimation, etc.) from a set of views. In the setting of a developmental embodied agent, SFA provides emergent invariant representations that resemble symbolic world knowledge; IncSFA provides this autonomously. When an agent is placed in the loop, such that its input sequence is caused by its selected actions, the emergent slow features have been shown to be useful decompositions of the environment [30], [60], specifically for reinforcement learning [62].

**Related Works.** Related to CD-MISFA in terms of having similar motivations and being based in developmental principles [67] are the biologically-constrained intrinsic motivation model, and robotic implementation, of Baldassarre et al. [31] and the Qualitative Learner of Action and Perception (QLAP; [33]). Powerplay [50], [61] was also important in terms of motivating CD-MISFA.

The Qualitative Learner of Action and Perception (QLAP) is a developmental robotics system designed to learn simplified predictable knowledge (potentially useful for skills) from autonomous and curiosity-driven exploration. It discretizes low-level sensorimotor experience through defining landmarks in the variables and observing contingencies between landmarks. It builds predictive models on the low-level experience, which it can use to generate plans of action later. It either selects its actions randomly or such that it expects to make fast progress in the performance of the predictive models (a form of artificial curiosity). A major difference between this system and ours is that we operate upon the raw pixels directly, rather than assuming the existence of a low-level sensory model. In QLAP, for example, the sensory channels are preprocessed so that the input variables track the positions of the objects in the scene. Through IncSFA, features emerge for raw visual processing, and this feature development is tightly coupled with the curiosity-driven learning.

The recently formulated **PowerPlay** can be viewed as a greedy variant of the Formal Theory of Creativity. In PowerPlay, an increasingly general problem solver is improved by searching for the easiest to solve, still not yet known, task, while ensuring all previously solved tasks remain solved. By its formulation, PowerPlay has no problems with forgetting, which can easily occur in an open-ended learning setup [38], [50]. In CD-MISFA, when each new representation is learned well enough to be internally predictable (low error), it is frozen and added to a long-term memory storage, and therefore there will be no destruction of already learned representations. Further, CD-MISFA searches for the context corresponding to the easiest to encode new representation, thereby acting in a PowerPlay-esque manner.

**B. CD-MISFA Overview**


The agent autonomously explores among $m$ contexts, and builds a representation library, denoted as

$$
\Phi^C = \{ \Phi^C_1, \Phi^C_2, ..., \Phi^C_n \}.
$$

There are, lets say $n(\leq m)$ different representations to learn in the environment (but the agent does not know $n$). So, one representation can suit more than one context. Learning resources are not assigned to each context individually. Instead, the agent learns one representation at a time.

Each representation $\Phi^C$ is composed of two subsequent mappings. The first takes a sensory input vector (e.g., pixels) $x(t)$ (where $t$ indicates discrete time), and encodes it via slow features. A straightforward example is linear SFA, which projects $x$ from $I$ to $J$ dimensions ($J << I$) via matrix $W = (w_1, ..., w_J)$, composed of $J$ column vectors which are the slow features. In this case, for the $i$-th representation,

$$
y^i(t) = x^T(t)W^i.
$$

The second mapping produces internal state $s^i$ from slow feature output $y^i$. To this end, it has a set of cluster centers $C = (c_1, ..., c_k)$ in the slow feature output space, and assigns the current state as the one with the smallest error from the current output:

$$
s^i(t) = \arg\min_j \|y^i(t) - c_j^i\|.
$$

These mappings provide a simplification of the raw sensory data expected to be perceived when the agent is within the context. The first provides invariance, suppressing irrelevant
information. The second provides specificity in the remaining (relevant) information.

**Contexts.** The agent explores different contexts, by switching between them. Example contexts are rooms to explore, objects to interact with, or types of videos to perceive. As a specific example context, see Fig. 1. We do not specifically define context, but note the following. 1. For convenience, a context can be thought of as having some state and action space, that is known to the agent. Thus, each context involves a set of states, a set of actions, and transition probabilities, from one state to another, given some action. 2. There is some exploration policy, internal to the context, by which the agent interacts with this environment. Exploration policies define how the randomized exploration (i.e., motor babbling) will occur on the given states and actions, e.g., Brownian motion on a mobile robot’s wheel velocities with an innate reflex to turn away from obstacles sensed through the distance sensors. 3. There is a potentially unobservable world state that defines the high-dimensional observations that will be the input to IncSFA. In Fig. 1 is defined an example of a robot perched over an object. Here, a state space is a discretization of the right arm shoulder joint angles to 20 states, while the actions are 1. increase or 2. decrease the joint angle enough to move to an adjacent state. The world state includes the condition of the object, which is not known to the agent initially. But this becomes “known” through the slow feature encoding, after it is learned. Another example context is a simple grid world where the agent explores via random selection of one of four actions (up, down, left, and right). Its state space is given by the grid with observations of high-dimensional images showing the grid and the agent as viewed from above.

Each interaction with a context is an episode. There is a start condition to the episode, and an ending condition, which must occur at some point in the random exploration. After the ending condition, the agent must decide whether to continue exploration of this context, or to move to another.

The agent uses curiosity to explore among multiple contexts. Rewards and motivation are intrinsic to the agent, and this intrinsic reward is calculated from representation learning progress. The agent can choose to remain engaged in its current context (exploitation), or seek to engage in another context (exploration). These decisions are due to utility judgements, where the utility is an estimate of expected learning progress of remaining engaged in the current context versus the expected learning progress of another contexts. If the former is higher, remaining engaged within the current context is most valuable, and, if the latter is higher, disengagement and switching is the more valuable choice.

Fig. 2 shows the architecture of CD-MISFA. The “adaptive module” encompasses the unsupervised learning part, which involves a combination of IncSFA and Robust Online Clustering. The representation library is shown by the “trained modules”. Estimation errors are denoted by \( \epsilon \), while intrinsic reward is denoted by \( \dot{\epsilon} \). The intrinsic reward signal feeds into the value function estimation module. The possible environmental contexts are shown at the bottom, the current context is the “state” (with respect to the higher-level value function), while the “action” is either to remain engaged in that context, or to disengage and go to another.

Next, in Sec. 4.C we discuss learning of a single representation. Specifically, we use IncSFA and the Robust Online Clustering (ROC) method, respectively. Some details of these algorithms are described below, but more thorough descriptions can be found elsewhere.

C. Unsupervised Representation Learning: IncSFA

SFA is concerned with the following optimization problem: Given an \( I \)-dimensional input signal \( x(t) = [x_1(t), \ldots, x_I(t)]^T \), find a set of \( J \) instantaneous real-valued functions \( g(x) = [g_1(x), \ldots, g_J(x)]^T \), which together generate a \( J \)-dimensional output signal \( y(t) = [y_1(t), \ldots, y_J(t)]^T \) with \( y_j(t) := g_j(x(t)) \), such that for each \( j \in \{1, \ldots, J\} \)

\[
\Delta_j := \Delta(y_j) := \langle \dot{y}_j^2 \rangle
\]

is minimal under the constraints

\[
\langle y_j^2 \rangle = 0 \quad \text{(zero mean)}, \quad \langle y_j^2 \rangle = 1 \quad \text{(unit variance)}, \quad \forall i < j : \langle y_i y_j \rangle = 0 \quad \text{(decoration and order)},
\]

with \( \langle \cdot \rangle \) and \( \dot{\cdot} \) indicating temporal averaging and the derivative of \( y \), respectively.

The goal is to find instantaneous functions \( g_j \) generating different output signals that are as slowly varying as possible.\(^2\)

\(^2\)We note that there are some similarities with the options framework here. One could link the start condition idea to an initiation state, and the end condition would correspond to having a termination probability of one at some state, and zero elsewhere. Indeed, one can view the problem of representation learning as analogous to abstraction learning in options, which remains an important open problem. However, we do not need to formalize contexts as options in this paper.

Fig. 2. Architecture of CD-MISFA.
Algorithm 1: IncSFA(J,K,θ)

```
//Incremental update of J slow features from samples x ∈ R^t
// V: K columns: PCs of x
// W: J columns: SFs
// v^\gamma: First PC in z-space
// x: Mean of x
1 {V, W, v^\gamma, x} ← INITIALIZE()
2 for t ← 1 to ∞ do
3   \bar{x} ← SENSE(worldstate)
4   {\eta_t^{PCA}, \eta_t^{MCA}} ← LRNSCHEDULE(θ, t)
5   x ← (1 - \eta_t^{MCA}) x + \eta_t^{PCA} x // Update mean
6   u ← (\bar{x} - x) //Centering
7   //Candid Covariance-Free Incremental PCA
8   V ← CCIPCA-UPDATE (V, K, u, \eta_t^{PCA})
9   S ← CONSTRUCTWHITENINGMATRIX(V)
10  if t > 1 then (z_{prev} ← z_{curr}) //Store prev.
11     //Whitening and dim. reduction
12     z_{curr} ← S^T u
13     if t > 1 then
14        \dot{z} ← (z_{curr} - z_{prev}) //Approx. derivative
15        //For seq. addition (γ)
16        v^\gamma ← CCIPCA-UPDATE (V^\gamma, 1, \dot{z}, \eta_t^{PCA})
17        γ ← \|v^\gamma\|
18        //Covariance-free Incremental MCA
19        W ← CIMCA-UPDATE (W, J, \dot{z}, γ, \eta_t^{MCA})
20  end
21  end
22  y ← z_{curr}^T W //Slow feature output
```

The decorrelation constraint (7) ensures that different functions \(g_i\) do not code for the same features. The other constraints (5) and (6) avoid trivial constant output solutions.

In a linear sense, the optimization problem can be solved through an eigenvector approach, involving two uses of principal component analysis (PCA) — first, of the covariance matrix of the inputs (for whitening) and, second, of the covariance matrix of the whitened approximate derivative measurements [70]. IncSFA uses incremental algorithms for the two required PCAs. For the first, Candid Covariance-Free Incremental PCA [68], [72], is used, which can also reduce the dimensionality by only computing the \(K\) highest eigenvectors. For the second, Minor Components Analysis (MCA; [36], [39], [40]) updates the \(J\) slowest features.

The overall framework of IncSFA is shown in Algorithm 1. IncSFA needs to update the signal mean (learned incrementally by simple online average estimation), the \(K\) principal components, and the \(J\) slow features. In general \(K < I\) and \(J < K\) — and \(K\) and \(J\) are parameters of the algorithm. The learning methods use “amnesic” learning rate schedule, so they are potentially suited to nonstationary input sequences.

Algorithm 2: CCIPCA-UPDATE(V, K, u, η)

```
//Candid Covariance-Free Incremental PCA
1 u_1 ← u
2 for i ← 1 to K do
3   v_i ← (1 - η) v_i + η \left(\frac{u_i \cdot v_i}{\|v_i\|^2}\right)
4   //Residual
5   u_{i+1} = u_i - \left(\frac{\|v_i\|^2}{\|v_i\|}\right) v_i
6 end
7 return V
```

IncSFA uses Hebbian (CCIPCA) and anti-Hebbian (CIMCA) update rules [9] to compute slow-features from a time-varying input signal.

CCIPCA updates estimates of eigenvalues and eigenvectors from each centered observation. CCIPCA combines a statistically efficient Hebbian update with the residual method [22], [48] to generate observations in a complementary space in order to update components besides the first, dealing with the requirement that any component must also be orthogonal to all higher-order components. The CCIPCA algorithm is presented in Alg. 2. The principal component estimates are used to construct a whitening matrix. After whitening, the signal is (approximately) normalized and decorrelated.

Minor Components Analysis preferentially learns the least significant principal components. The update for each slow-feature vector \(w_i\) from 1 to \(J\), is

\[
w_i ← (1 - η^{MCA}) w_i - η^{MCA} \left(\frac{\dot{z} \cdot w_i}{\|w_i\|^2} \right) \sum_{j=1}^{i-1} (w_j \cdot w_i) w_j\]

(8)

where \(η^{MCA}\) is a learning rate. This update is based on anti-Hebbian learning with an additional Gram-Schmidt term inside the summation that enforces different features to be orthogonal. After updating, a feature is normalized for stability.

The feature output is an instantaneous function,

\[
y(t) = z(t)^T w(t).
\]

D. Adapting the States with Robust Online Clustering

In a context’s pre-defined state space, each state has its own instance of an online clustering algorithm. Clustering is done in an associative space that combines this pre-defined state space with the slow feature output space. These clusters, once learned, act as augmented internal states, potentially providing information about invariants captured with IncSFA.

As an example, consider again the robot viewing its arm move eventually toppling an object in the scene. The state space here is a quantization of the joint angles of the shoulder into 20 bins, thereby providing 20 states, leading to 20 instances of the clustering algorithm. A developed slow feature output here is a step function, e.g., when the object is not
toppled, the feature output equals zero, and when the object is toppled the feature output equals one. Upon convergence of, first, IncSFA and, second, the clustering, each joint-angle state will be replaced by two internal states, which inform whether the object is or is not toppled.

Learning these clusters is not as straightforward as the above example makes it seem, since the signal is highly nonstationary during the early learning phases, due to its input being a function of adapting slow features. The slow feature outputs can change rapidly during the training phase. The estimator therefore has to be able to change its estimates to this non-stationary input, while converging to a good estimate when the input becomes stable. To this end, we use a clustering algorithm to specifically handle non-stationary data, called Robust Online Clustering (ROC) [3], [7].

ROC is similar to an incremental K-means algorithm — a set of cluster centers is maintained, and with each new input, the most similar cluster center (the winner) is adapted to become more like the input. Unlike K-means, with each input, it follows the adaptation step by merging the two most similar cluster centers, and creating a new cluster center at the latest input. In this way, ROC can quickly adjust to non-stationary input distributions by directly adding a new cluster for the input. In this way, ROC can quickly adjust to non-stationary input distributions by directly adding a new cluster for the newest input sample, which may mark the beginning of a new input process.

But is this plasticity at the cost of stability? No. In order to enforce stability, clusters maintain a weight, which increases faster for more similar (to the cluster center) inputs. A large weight prevents a cluster center from changing that much. When two clusters are merged, their weights are also combined.

A sketch of the ROC per-sample update is in Alg. 3. The ROC algorithm repeatedly iterates through the following steps. For every input sample, the algorithm finds the closest cluster winner and updates the center cwinner towards it, also increasing the weighting parameter awinner. Next, the closest two clusters are merged into one cluster. Then, a new cluster is created around sample y. Finally, all clusters weights decrease slightly. Parameters required are ξ, the maximum number of clusters, an amnesic parameter φ to prevent convergence, and the response function for similarity measurement.

Algorithm 3: ROC-AMNESIC(y, s, ξ, φ)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{C, a} ← GETCLUSTERINGINSTANCE (s)</td>
</tr>
<tr>
<td>2</td>
<td>if</td>
</tr>
<tr>
<td>3</td>
<td>{C, a} ← ADDNEWCLUSTER (y, C, a)</td>
</tr>
<tr>
<td>4</td>
<td>else</td>
</tr>
<tr>
<td>5</td>
<td>winner ← arg max RESPONSE(y, c_i)</td>
</tr>
<tr>
<td>6</td>
<td>c_winner ← c_winner + y - c_winner</td>
</tr>
<tr>
<td>7</td>
<td>a_winner ← a_winner + RESPONSE(y, c_winner)</td>
</tr>
<tr>
<td>8</td>
<td>{γ, δ} ← arg max RESPONSE(c_γ, c_δ)</td>
</tr>
<tr>
<td>9</td>
<td>c_γ ← c_γ a_γ + c_δ a_δ</td>
</tr>
<tr>
<td>10</td>
<td>a_γ ← a_γ + a_δ</td>
</tr>
<tr>
<td>11</td>
<td>c_δ ← y</td>
</tr>
<tr>
<td>12</td>
<td>a_δ ← 0</td>
</tr>
<tr>
<td>13</td>
<td>for i ← 1 to ξ do</td>
</tr>
<tr>
<td>14</td>
<td>a_i ← a_i (1 - φ)</td>
</tr>
<tr>
<td>15</td>
<td>end</td>
</tr>
</tbody>
</table>

εest(t) = \sum_{i=1}^{M} εest(t), (11)

and the intrinsic reward is the derivative of the total estimation error εest = εest(t) - εest(t-1). Figure 3 shows an example with a 20-state estimator.

E. Intrinsic Reward

The intrinsic reward is expected learning progress. Learning progress is approximated as the decrease in context-specific cumulative estimation error. Each context state i has an associated error εest_i. These errors are updated whenever the agent visits that state —

εest_i(t) = \min_j ||y(t) - c_j|| (10)

where y(t) is the slow-feature output vector and c_j is the j-th cluster center associated with this state. The context’s current estimation error is the sum of stored errors, over all M context states.

F. Module Storage and Gating

Once the slow feature outputs stabilize, the estimator clusters converge and the error will become very low. Next, estimator clusters with small weights a_i are eliminated, to avoid having spurious internal states. Finally, this overall representation module is frozen, considered learned, and placed in long-term memory.

The already trained set of modules are the abstraction library Φ^C (Eq. 1). If one of these module’s estimation error within a context is below a threshold, that context is assigned that module’s representation and the adaptive training module will be prevented from learning, by this gating signal. There will no intrinsic reward in this case. On the other hand, if the
est function is updated at every engage-disengage from context is the expected change in estimation error when transitioning on the two internal-actions) is given by:

$$P_{i,j}^{\text{engage}} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \quad (12)$$

$$P_{i,j}^{\text{disengage}} = \begin{cases} 0, & \text{if } i = j \\ \frac{1}{N-1}, & \text{if } i \neq j \end{cases} \quad (13)$$

\(\forall i, j \in [1, \ldots, N]\).

**H. Reward and Value Function**

The agent maintains an estimated reward function, which is the expected change in estimation error when transitioning from context \(o\) to context \(o'\) (and \(o = o'\) is possible). The agent’s reward function is updated at every engage-disengage decision, from the intrinsic rewards, as a sample average:

$$R_{a}^{o,o'} := (1 - \eta) R_{a}^{o,o'} + \eta \sum_{t=0}^{T} -\epsilon_{\text{est}}(t) \quad (14)$$

where \(0 < \eta < 1\) is a learning rate, \(T\) was the duration of the previous context until its termination, \((o, o') \in \{O_{1}, \ldots, O_{n}\}\) and \(a \in \{\text{engage}, \text{disengage}\}\).

Using the current updated model of the reward function \(R\) and the internal-state transition-probability model \(P\), the agent’s policy \((O \times A^o \rightarrow [0, 1])\) is updated.

It is important that the policy adapts quickly enough to adapt to the quickly changing reward function. Intrinsic rewards can change quickly as learning progresses, and the RL must adapt quicker than the underlying representation learner. We used model-based Least Squares Policy Iteration [23], which is an efficient value-estimation technique, although in principle the more biologically-plausible temporal-difference methods could also work.

**Epsilon-greedy.** The agent cannot take the value-maximizing decision from the very beginning, since it needs time to build its value estimates to a more accurate level. Early on, it can make decisions more or less randomly so that it can gather experience in the different contexts, and to learn good estimates of value over all contexts. Given good value estimates, it can choose to engage within the context where it should learn quickly, in other words, make the fastest learning progress, and to lead to a quick learning of the next representation. To this end, the model augments its internal action selection with decaying \(\epsilon\)-greedy exploration.

**III. NEURAL CORRELATES TO CD-MISFA**

**A. SFA and Competitive Learning — Entorhinal Cortex and Hippocampus**

Slow Feature Analysis variants have been used to simulate representation learning in a number of biological scenarios. Based on the general principle that underlying driving forces manifest through slow changes in sensory observations, the features that emerge from SFA often encode important invariants. Hierarchical SFA has been shown to develop grid cells from high-dimensional visual input streams [11]. Grid cells, found in entorhinal cortex (EC) [14], have a pattern of firing that effectively represent hexagonal codes of any two-dimensional environment. As such, grid cells are effective general representations for spatial navigation in typical environments.
A competitive learning layer, over the top-layer of slow features, leads to features acting as place cells or head-direction cells, depending on what changes more slowly from the observation sequences. A place cell will fire when the animal is in a specific location in the environment, typically invariant to its heading direction. Head-direction cells fire when the animal faces a certain direction, no matter what coordinate position it is in. Place cells and head-direction cells are found in hippocampus [37], [54], which has input from EC. It’s been hypothesized that hippocampus acts as a relatively fast encoder of specific, episodic information, on top of cortex, which learns general structure from lots of data over a long period [7] — “It has been proposed that this universal spatial representation might be recoded onto a context-specific code in hippocampal networks, and that this interplay might be crucial for successful storage of episodic memories [12].”

SFA’s biological plausibility was furthered by IncSFA, which avoids batch processing and has Hebbian and anti-Hebbian updating equations. Hierarchical SFA [11] and IncSFA [27], with competitive learning on top, was shown to develop place and head-direction cell representations. For the representations learned in CD-MISFA, we use the basic structure suggested by these results: A slow feature learner (possibly hierarchical) for global features (IncSFA), inputs into a competitive learner for development of local features (ROC).

B. Neuromodulatory Subsystems for Intrinsic Reward and Context Switching

1) Intrinsic Rewards: Dopamine and Learning Progress: 
Dopaminergic projections originate from the ventral tegmental area (VTA). Dopamine has been implicated in reward prediction [59], leading to plausible relation to the theory of reinforcement learning [62] — specifically, dopamine may be acting as a temporal-difference (TD) error signal. However, this account remains controversial [17], [44]. A major deviation from the dopamine as TD-error theory comes from data implicating dopamine in responding to novel salient stimuli [43], [57], even for stimuli that are not predictive of reward. Dopaminergic responses to such stimuli fade over subsequent trials. It has been proposed that this characteristic serves the purpose of a “novelty bonus” — e.g., a reward addendum serving as a “optimistic initialization”.

These data present intriguing correlations to the curiosity theory. Dopamine release in response to novel stimuli could potentially signal a predicted intrinsic reward — an expectation of learning progress. Could DA in some situations signal the intrinsic reward? Dopamine’s potential role in intrinsic motivation has been discussed before [19], [43], but not with respect to the formal theory of curiosity [55], which predicts that intrinsic reward should be proportional to compression progress. Computational models in neuroscience often treat intrinsic reward as resulting from the novelty of a stimulus. If intrinsic reward really does result from novelty, we would expect persistent high levels of dopamine in response to unpredictable noisy stimuli (as it remains novel from moment to moment). On the other hand, if intrinsic rewards encode compression progress, we would expect decreases in the level of dopamine as the predictive model becomes unable to learn anything more about the structure of the noise.

2) Engagement and Disengagement (and Switching): Norepinephrine: 
Neurons of the locus coeruleus (LC), in the brainstem, are the sole source of norepinephrine (NE). NE is linked to arousal, uncertainty, vigilance, attention, motivation, and task-engagement. The LC-NE system is more traditionally thought to affect levels of arousal, but more recently has been implicated in optimization of behavioral performance [2], [49], [66].

In that context, the activity of the LC-NE system can be understood as modulation of exploration-exploitation. The tonic differences in LC-NE response are associated with levels of arousal. Tonic NE response is correlated with task performance levels [66]. Low tonic activity coincides with low attentiveness and alertness [1], while high tonic activity coincides with agitation and distractibility [2]. Good task performance coincides with an intermediate tonic level during which phasic bursts of activity are observed, while poor task performance due to distraction is associated with high tonic activity. In phasic mode during periods of intermediate tonic NE activity, NE is released in response to task-relevant events [10]. As suggested by Usher, et al. and others [2], [66], the phasic modes might correspond to exploration, whereas high tonic states of NE activity might correspond to exploration.

When it is beneficial for the agent to remain engaged in the current task, the tonic NE level stays moderate, and only relevant task stimuli will be salient. However, when it is not beneficial to remain engaged in the current task, the NE level raises and task-irrelevant stimuli become more salient. This drives the agent to distractibility, and task performance suffers. Attending to some distractor stimuli could have the effect of causing the agent to switch to another task in which this distractor becomes relevant, ostensibly with the purpose of exploring among available tasks (i.e., it “throws the ball in the air so another team can take it” [2]).

In CD-MISFA, the agent’s two internal-actions, (engage or disengage), and the reasons they are taken, links to the NE-driven task engagement/disengagement model. Boredom (low NE) indicates that a good representation already has been learned, leading to low estimation error, and thereby low potential intrinsic reward. Distractibility (high NE) indicates that the errors are too high, not decreasing quickly enough, or they cannot be reduced. In this case, it becomes valuable to disengage and find some other context, where learning may progress faster (or at all). When the agent has found a good context, the estimation errors decrease regularly, providing intrinsic reward that leads to a high value estimate (and a desire to remain engaged in that context).

C. Frontal Cortex: Value Function and Representation Selection

The NE and DA neuromodulatory systems each have reciprocal connectivity with the prefrontal cortex — executive areas, which deal with cognitive aspects such as decision making, and top-down control of other functions, such as

\textsuperscript{1}To our knowledge, this has not been tested yet.
selective attention [32]. If the LC-NE system is handling task-engagement and disengagement based on some value judgement, then this system needs to be controlled by another system that is estimating these values. The prefrontal cortex (PFC) plausibly plays a role in value estimation, and might use the utility information to provide top-down regulation of the activities of the LC neurons [15].

PFC and nearby structures, specifically the anterior cingulate cortex (ACC) and the orbital frontal cortex (OFC), are implicated in value-based judgements. The ACC is involved in error detection (i.e., recognizing a prediction error) and estimating the costs of these errors [5]. OFC is thought to be of import in motivational control of goal-directed behaviors [46] — OFC damage leads to responses to objects which are no longer rewarding [31], [47]. The dorsolateral pre-frontal cortex (DLPF) is implicated in value-based working memory [42]. Thus, these structures could possibly work together to estimate a value function, in the RL sense [15].

Another important property of PFC is to maintain an appropriate task representation, i.e., imposing internal representations that guide subsequent performance, and switching these for another when it is no longer appropriate [6], [32]. This property requires mechanisms to keep goal-relevant information (i.e., what should be considered salient and what should be considered a distractor) enabled in resonance with lower structures. Further, it requires a mechanism to maintain a context despite bottom-up disturbances, and a mechanism to switch the context. The PFC has connections from and to higher-order associative cortices, so it is in a good position to impose task-relevant representations from the top-down. Such “executive attention” enables memory representations to be “maintained in a highly accessible state in the presence of interference, and these representation may reflect action plans, goal states or task-relevant stimuli in the environment [18].

IV. EXPERIMENTS AND RESULTS
A. Synthetic Signals

In other works, we have studied the types of representations uncovered by IncSFA, and their applicability [20], [28]. The experiments here will focus more on the curiosity-driven behavior, especially in comparison to what the formal theory of curiosity predicts. We also explore the potential link of CD-MISFA to neuromodulatory task-switching — what quantities in our experimental results might be analogous to associated neuromodulators dopamine and norepinephrine?

CD-MISFA’s typical behavior involves cycles of exploitation, exploration, and module storage. Exploration involves context switching, enabling accumulation of learning progress estimates about each context. The exploitation period has it settle into a single context where progress is easiest, until the representation is stored in long-term memory. Based on the formal theory of curiosity, we expect CD-MISFA to learn the representations in inverse order of their learning difficulty. Further, it will not waste time on anything unlearnable, corresponding to noise — which we note is novel and surprising in the traditional sense of Shannon [59], however uninteresting since no learning progress can be made.

To this end, the first experiment involves a synthetic learning environment, with four types of sources — also known as driving forces [69]. The simple driving forces are the fundamental “causes” of the complex observations. For example, an observation sequence given by an onboard camera of a mobile robot is “caused” by the robot’s position, orientation, and camera angle. One cannot reconstruct the observations from the driving forces alone, of course, but tasks and rewarding conditions are often associated with the driving forces, and knowledge of the driving forces leads to useful (potentially rewarding) predictive power.

At any time the agent is experiencing one of five contexts. Two contexts are generated based on the same driving force, while the other three each have a different driving force. In Fig. 4(a), the $2 \times 1000$ (dimension by time steps) signal sources can be seen (S-A, S-B, S-C, S-D), ordered via learning difficulty, with the easiest signal at the top. The blue curve shows the first dimension, while the red dotted curve shows the second dimension. At the bottom, we have a highly nonstationary source, which changes irregularly, so as to be unlearnable to IncSFA. We want to hide each of these sources within a different high-dimensional process, albeit linearly, so that linear IncSFA will be able to extract them and it will take enough effort to do so. A high-dimensional observation is generated from a source by multiplication with one of four $400 \times 2$ matrices, which are randomly generated before each experiment. The 400 resulting values are rearranged into a $20 \times 20$ and value-normalized from zero to one to be pixel values for each image. Each input observation $x(t)$ is an image of $20 \times 20$ pixels. In Fig. 4(b), one can see a few sample observations. The task for CD-MISFA is to extract all three learnable driving force signals from a single stream of high-dimensional observations.

Fig. 4(c) shows the CD-MISFA agent’s environment, which contains the five contexts (C1-C5; which can be considered states in the RL sense), and has two actions — stay (engage) or switch (disengage). Each time a context is entered, 100 steps of observations are fed to IncSFA. Each context has a local clock, so that the local time step will pick up where it left off if the agent returns from another. At the end of the 1000 time steps, the local time step resets.

Measuring Learning Difficulty. In order to test predictions of the Formal Theory of Curiosity, we need to analytically establish a definition of learning cost for slow features, by which we will measure the relative complexity of the signals within each context. We introduce here a measure denoted as $\Omega$, to quantify the learning progress of IncSFA.

$$\Omega(x) = \left[ 1 - \frac{\eta mna(\lambda_n - 1 - \lambda_n)}{1 - \eta mna - \eta mne \lambda_n} \right]$$

(15)

where $\lambda_n$ and $\lambda_{n-1}$ are the eigenvalues corresponding to the lowest-order and second-lowest-order (respectively) principal components in the whitened derivative space. We will discuss the origins of $\Omega$ further in Sec. IV-D with full derivation.

4We use “time step” in a local sense — it refers to one of 1000 steps that make up each context. We will use “time” to refer to global time (over all contexts).
In this experiment, the three learnable signals are quantified as $\Omega^A = 0.9933$ (for S-A), $\Omega^B = 0.9988$ (for S-B), and $\Omega^C = 0.9997$ (for S-C). They are quite close due to the similarities of the last and second to last eigenvalues in each distribution, however there is a nonlinear relationship between $\Omega$ and learning time. For S-A, about 2-3 epochs are needed. For S-B, about 15 epochs are needed. For S-C, about 40 epochs are needed.

Experiment Setup. The experiment setup is as follows. Since there are 1000 different time-steps, we use 1000 states for the clustering. Thus, there will be 1000 different clusterers, each with maximum number of clusters set as 2. The estimator error is measured as an average of the estimation errors after each episode — an interaction with a single context of 100 time steps. The intrinsic reward estimates and policy are updated after each episode. Once the estimation error gets below the threshold 0.3, the module is frozen, and a new module created. The initial setting for $\epsilon$-greedy is 0.6, which decreases via multiplication with 0.995 after every episode, and is reset when a new module is created. The learning rates for IncSFA: for CCIPCA, a $1/t$ learning rate is used, with amnesic parameter $l = 2$ (68), the MCA learning rate is a constant $\beta_{MCA} = 0.05$. We collected results over 25 different runs. Each run has a different initialization of all aspects, wherein CD-MISFA operates for time $3.5 \times 10^5$.

We note here some implementation details about the gating system. The gating system prevents corruption of the adapting IncSFA with samples from an already known/learned representation. This is implemented as a buffer that fills during each episode, at the end of which the 100 observations are sent to all feature sets, from which the output is calculated. That output is then sent to the clusters in each SF output space, enabling error calculation for all modules. If the minimum module error is less than 0.3, the previous 100 samples are not used for learning, and a negative reward of -100 is given. Otherwise, the samples are fed into IncSFA for learning. In this case, the intrinsic reward is calculated as the difference between the current estimation error of the adaptive learner and the same context’s previously measured estimation error. The negative reward serves only to speed up the learning process. If it were removed, each run would simply take longer to complete.

Behavior. In all 25 runs behavior of CD-MISFA involved alternating phases of mostly exploration among all contexts, and exploitation once it settles on a context where it expects to make the most progress. We call this exploitation-exploration process a cycle. Exploration is caused by the initial high amount of change in the adapting slow features, so that the estimator, which is on top of the slow features, cannot make progress. Once CD-MISFA remains within a context for enough time, the features become predictable enough so that an advantageous intrinsic reward can result. Due to $\epsilon$-greedy, it continues to switch between contexts, allowing it to accumulate good estimates for all (the previous intrinsic reward accumulations are captured in the reward function). As $\epsilon$ decays, the policy converges to the simple but optimal strategy to disengage from all contexts except the easiest to learn new context.

**Results.** Results are shown in Fig. 4(d)-(g). Part (d) shows the average cumulative estimator error (a single run is also plotted for perspective). In each cycle, the error starts high, then trends down as representations are learned; finally a module is created. Within each run, this is a rather noisy signal, as the agent jumps from context to context. The end of each run has only the unlearnable context remaining, so the error cannot reduce enough to store another module.

Part (e) shows the run-averaged and temporally-averaged (for smoothness) intrinsic reward. Each cycle (notably except the first) involves a rise and fall. Relatively low intrinsic reward that trends higher is associated with disengagement behavior. Relatively high intrinsic reward that trends lower is associated with engagement behavior. The high punishment for boring experiences within a learned context tends to drag the values down, moreso later in each run, when more representations have been learned. The first cycle seems to lack the typical rise, which we posit is due to the simplicity of the signal.

Part (f) shows average learning times and standard deviations for the three learnable signals. The ordering tends to be as predicted, but not always: A module for S-A emerges first all 25 times, S-B’s module occurs second 18 times, and third 7 times, while S-A’s module is mostly third (18/25). Due to the 7 runs when S-B and S-C were learned opposite as expected, the average learning time for S-B is higher than the average time when the second module is typically learned (as can be seen in Fig. 4(d)), and the average learning time for S-C is lower than when the third module is learned.

Part (g) illustrates the reward function for run number 15, which is a fairly typical run. C1 and C2 are associated with initial rising reward. Once the shared source (S-A) is learned, both have their expectations of reward drop. We see C3 subsequently rise, followed by C4, then C5 (unlearnable).

On Invariance. There are two independent dimensions to each source, which together generate the observations. The corresponding representation thereby also contains two parts. One part of the driving force is (trivially) invariant to the other part, and, after learning, the invariance property is observable at the representation outputs. For example, if (after learning) the first dimension of our source is held constant but the second allowed to change, then the observations will change, but the output of the first feature of the corresponding component will be constant, while the second changes. Fig. 5 illustrates this concept. As a real world example, consider place cells and head-direction cells. The output of the place cells are invariant to changes in orientation, and vice versa.

On Neuromodulators. The estimation error profile observed in Fig. 4(d) and associated behavior mirrors the findings regarding the LC-NE system and the “inverted U”. High levels of estimation error correspond with predominantly disengagement and switching (“agitation”), while low levels of estimation error correspond with switching (“boredom”). There is a “sweet spot” of error, where the agent mostly engages in a single context. In this sweet spot, the intrinsic reward, representing learning progress, is at its relative peak. The intrinsic reward signal could link to dopamine, although, as we mentioned, there is no conclusive evidence about this.
B. Emergent Representation from Sensorimotor Loops — an iCub Experiment

This experiment uses an embodied agent (iCub) with real high-dimensional images (grayscale $75 \times 100$), from the robot’s eyes. There are two contexts here. In each, the iCub explores via random movement of its shoulder joint, causing the outstretched hand to eventually displace the single object in its field of view. It then observes the outcome while the hand continues to move. It is not given any prior knowledge about the objects, itself, or any concepts at all. It merely observes the pixel values, and uses CD-MISFA for learning and decision making. In one context, the object is a cup, which topples over upon contact with very predictable outcome. In the other, the object will roll in different directions. About 70 episodes of image sequences were collected for each context. The eventual slow features, emergent from the holistic images, will code for
reduction by calculating how many eigenvalues would be useful. The learning rate was 0.01. CCIPCA did variable size dimension reduction to only about 10 dimensions.

When the input signals from widely different clusters; the feature here encodes the context identity, which is to be expected when the input signals from widely different clusters; in a sense this is similar to a multiple rooms case, where the features code for room ID. In order to prevent learning progress from continual switching, the following rule was implemented: when the agent decided to remain in its current context, it experienced two subsequent episodes, but when it decided to switch to the other, it only experienced one. In other words, the agent is given more time to learn by staying rather than by switching.

**Results.** 15 experimental runs were performed. Fig. 6(b)-(d) show results. Part (b) shows the average estimation error during the first module’s learning, while part (c) shows average estimation error for the second. Part (b) has a higher error, with more fluctuation than part (c), which mostly involves learning in a single context, since it will learn to quickly disengage away from the already learned context due to boredom punishment. In part (d), one can see the easier representation was indeed mostly learned first (in 14 of the 15 runs, this was the case).

Examples of the context-specific representations over time are shown in Fig. 7. Both representations eventually encode whether the object is displaced or not. Most of the information in the image sequences can be broken down into three components: a baseline (the background), the object, and the arm. The object changes slower than the arm, so it is preferentially extracted by SFA. Moreover, the object-based features are invariant to the arm’s position. Generalization is also possible, in a limited sense. If the arm were replaced by some other object (e.g., a stick), the feature output would not be perturbed. For more robust generalization, a better preprocessing is probably needed, as is typical with appearance-based vision techniques.

Once the features are learned, the feature output space creates a reduced-dimension state space for reinforcement learning techniques, if an external reward is in play. For examples, see our illustrative video of state-space simplification [20], and Kompella et al., 2012 [21] for an example of using RL to maximize external reward upon the previously learned features.

**On Concepts.** How do the learned representations relate to concepts? CD-MISFA could be the basis for something more substantial in the direction of concept learning, but, by itself, it is limited.

The representations learned by CD-MISFA correspond to compressed descriptions of image feeds, emerging from an eigendecomposition of the covariance of temporally subsequent image differences. In some cases the resulting representations loosely resemble concepts, as when the slowest feature is shown to invariantly capture the state of some object in the images. But we are hesitant to explicitly refer to these representations as concepts, for a number of reasons. First, the notion of concept is itself up for debate. Arguments about what constitutes a concept will necessarily jump disciplinary boundaries, including philosophy, linguistics, and artificial intelligence. We do not wish to wade into this debate however, and we instead concern ourselves with the manner in which an agent or robot, starting with little prior knowledge, might direct its own behavior so as to increase what it knows about the world around it. Second, the types of representations learned by CD-MISFA are generally too low-level to be considered conceptual. For example, if CD-MISFA used intrinsic rewards to guide it to areas which enabled it to develop low-level

![Diagram](http://www.idsia.ch/~luciw/videos/IncSFAArm.mp4)
feature detectors, such as edge detectors (which SFA can learn from a moving fovea [4]), would we want to refer to the edge detectors or the edges themselves as concepts? Likely not, despite the fact that it could develop from the same learning mechanisms that led to a representation for a toppling event.

C. Comparison

Baldassarre et al. [3] recently presented a biologically-constrained model of IM, which is also applicable to developmental robotics. Although the Baldassarre model (TBM) is more closely tied to neuroanatomical function than CD-MISFA, we argue that a number of functional and theoretical drawbacks of TBM make CD-MISFA a superior choice.

TBM was implemented on an iCub robot, and tested in an environment motivated by psychological studies, which includes a box with 3 buttons, 3 doors, and a series of lights. The robot can take 6 oculomotor actions (eyes fixating at either of the 3 buttons or 3 boxes) and 3 arm-motor actions (‘reach and press’, ‘reach and point’, ‘reach and wave’). The IM reward function is modeled based on illumination change, considered as an automatically extracted salient event, and is a value that decays with recurrence of the salient event. In a learning phase, the model is allowed to explore by selecting any of its oculomotor/arm-motor actions, and observing the result (i.e., the opening of a door). In a test phase, external rewards are ‘hidden’ inside one of the doors, and the goal of the agent becomes: press the correct button to retrieve the reward.

A primary drawback of TBM and its experimental validation is, although it makes use of IM, it is not clear to what extent (if any) IM is necessary for appropriate learning to occur. The model is not tested without an IM reward function, and in principle, the task undertaken would be learnable simply through random exploration without any IM reward whatsoever. Conversely, the role of IM in CD-MISFA and its associated experiments is essential, since if CD-MISFA is not presented with intrinsic reward, the model will not stay in any particular context long enough to learn the underlying representations. If CD-MISFA simply explores its environments in a random fashion, it is incapable of learning any meaningful representation.

A major advantage of CD-MISFA over TBM is the former’s grounding in the Formal Theory of Fun and Creativity. Whereas the decay of the intrinsic reward value in TBM
arbitrarily depends on the number of times the agent repeats a given action, CD-MISFA makes use of the more appropriate learning progress measure. In CD-MISFA, information ceases to be intrinsically rewarding as a function of how and when those visits lose informational value.

Lastly (and perhaps most importantly), TBM does not operate on realistic sensory/motor spaces. Whereas CD-MISFA explicitly shows how IM can operate in a model learning from high-dimensional input streams, and how action selection can operate on low-level motor outputs, TBM only shows how a model of IM can learn a small subset of predefined actions, operating on abstract representations of visual input.

\[ w^{mca}(k) = (1 - \eta^{mca}) \ w^{mca}(k - 1) \]
\[ - \eta^{mca} \ (x(k) \cdot w^{mca}(k - 1)) \ x(k) \]

To analyze the "average" dynamics of Eq. 16 we reformulate it to a deterministic discrete time (DDT) system by taking the conditional expected value

\[ E[w^{mca}(k + 1) | w^{mca}(0), x(t), i < k] \]

at each iteration:

\[ w^{mca}(k) = (1 - \eta^{mca}) \ w^{mca}(k - 1) \]
\[ - \eta^{mca} \ E[x(k) | x(k)]^T w^{mca}(k - 1) \]

Here, \( E[xx^T] \) is the correlation matrix (\( R \)) of the input data (\( x \in \mathbb{R}^n \)). The weight vector \( w^{mca}(k) \) is shown to converge to minor component of input data \( \Omega \), if the following conditions are satisfied:

\[ \eta^{mca} \lambda_1 < 0.5, \quad ||w^{mca}(0)||^2 = 1, \]
\[ 0 < \eta^{mca} \leq 0.5, \quad w^{mca}(0)^T w^{mca} \neq 0 \]

where \( \lambda_1 \) is the largest eigenvalue of \( R \), \( w^{mca}(0) \) is the initial weight vector and \( w^{mca} \) is the eigenvector with the smallest eigenvalue of \( R \). Since the correlation matrix \( R \) is a symmetric nonnegative definite matrix, it can be factorized into \( QDQ^{-1} \), where \( Q \) is the eigenvector matrix (columns representing unit-eigenvectors \( v_i \)) and \( D \) is a diagonal matrix.

D. Quantifying the Learning Cost

We discuss here the measure denoted as \( \Omega \), which is used to quantify the learning cost of various types of signals for IncSFA. For simplicity, we consider here signals with similar input-variance but that have a different temporal structure. This assumption allows CCIPCA to approximately have a similar progress for the signals. Therefore, our focus remains here only on the progress of the CIMCA algorithm.

In an approach similar to the proof provided by Peng et al. [40] for the convergence of MCA, we present here an analysis on quantifying the learning progress of the CIMCA algorithm. For the sake of simplicity, we just consider here only the first output component, but this can trivially be extended for higher output components.

The weight-update rule of CIMCA is given by:

\[ w^{mca}(k) = \]
\[ (1 - \eta^{mca}) \ w^{mca}(k - 1) \]
\[ - \eta^{mca} \ (x(k) \cdot w^{mca}(k - 1)) \ x(k) \]

Fig. 7. Example emergence of object-centric slow features for both contexts. The final result encodes two states of each object — upright or displaced.
with corresponding eigenvalues ($\lambda_i$). In addition, the eigenvectors \{\(v_i|i = 1, 2, ..., n\}\} form an orthonormal basis spanning \(\mathcal{R}^n\). The weight vector \(w^{mca}\) can then be represented as

\[
aw^{mca}(k) = \sum_{i=1}^{n} a_i(k)v_i
\]

where \(a_i(k)\) are some constant coefficients.

**Definition 1:** Given a stationary input distribution \(x \in \mathcal{R}^n\) and its eigendecomposition: \(\{v_i, \lambda_i\}, \forall i \in \{1, ..., n\}\), where \(v\) denotes the set of eigenvectors and \(\lambda\) their corresponding eigenvalues (such that \(\lambda_1 > ... > \lambda_n \geq 0\)). Then, we define \(\Omega(x)\) as a measure to indicate the learning progress of CIMCA for the input distribution \(x\).

The following lemmas are useful to derive an analytical expression for \(\Omega\). Note that for all the following lemmas to hold true, the convergence conditions in (20) have to be satisfied.

**Lemma 1:** Let \(V_i\) be denoted as

\[
V_i = [1 - \eta^{mca} - \eta^{mca}\lambda_i]
\]

then,

\[
a_i(k) = \frac{V_i a_i(0)}{\sqrt{\sum_j V_j^2 a_j^2(0)}} \quad \forall i \in \{1, 2, ..., n\}
\]

**Proof:** We prove the result by mathematical induction.  

\(k = 1:\) Substituting (21) in (19) for \(k=0\), we get

\[
a_i(1) = V_i a_i(0) \quad \forall i \in \{1, 2, ..., n\}
\]

At each update, the weight vector \(w^{mca}(k)\) is normalized according to (17).

\[
a_i(1) = \frac{V_i a_i(0)}{\sqrt{\sum_j V_j^2 a_j^2(0)}} \quad \forall i \in \{1, 2, ..., n\}
\]

Therefore, (23) is true for \(k=1\).

\(k = m:\) Assuming the result to be true for some \(k = m > 1\)

\[
a_i(m) = \frac{V_i^{m} a_i(0)}{\sqrt{\sum_j V_j^2 a_j^2(0)}} \quad \forall i \in \{1, 2, ..., n\}
\]

let \(P\) denote

\[
P = \sqrt{\sum_j V_j^{2m} a_j^2(0)}
\]

\(k = m+1:\) Substituting (21) in (19) for \(k=m\), we get

\[
a_i(m+1) = V_i a_i(m) = \frac{V_i^{m+1} a_i(0)}{P}
\]

Upon normalizing,

\[
a_i(m+1) = \frac{V_i^{m+1} a_i(0)}{\sqrt{\sum_j V_j^{2m+2} a_j^2(0)}} = \frac{V_i^{m+1} a_i(0)}{\sqrt{\sum_j V_j^{2m+2} a_j^2(0)}} \quad \forall i \in \{1, 2, ..., n\}
\]

which is same as substituting \(k=m+1\) in (23). Therefore, by the principle of mathematical induction the result (23) is true for any \(k > 1\).

**Lemma 2:** Let \(\sigma_i\) be denoted as

\[
\sigma_i = \left[1 - \frac{\eta^{mca}(\lambda_i - \lambda_n)}{1 - \eta^{mca} - \eta^{mca}\lambda_n}\right]
\]

then,

\[
0 < \sigma_1 < ... < \sigma_{n-1} < 1
\]

**Proof:** If we show that

\[
0 < \frac{\eta^{mca}(\lambda_i - \lambda_n)}{1 - \eta^{mca} - \eta^{mca}\lambda_n} < 1
\]

then the condition (27) is straightforward.

We first prove the left inequality. Clearly, since \(\lambda_1 > ... > \lambda_n \geq 0\) and \(0 < \eta^{mca} \leq 0.5\), the numerator

\[
\eta^{mca}(\lambda_i - \lambda_n) > 0, \quad \forall i \in \{1, ..., n-1\}
\]

and the denominator

\[
1 - \eta^{mca} - \eta^{mca}\lambda_i > 1 - \eta^{mca} - \eta^{mca}\lambda_n
\]

\(> 0.5 - \eta^{mca}\lambda_1 \quad \therefore \eta^{mca} < 0.5 \quad \therefore \eta^{mca}\lambda_1 < 0.5 \quad (30)
\]

To prove the right inequality, it holds

\[
\text{iff,} \quad \eta^{mca}(\lambda_i - \lambda_n) < 1 - \eta^{mca} - \eta^{mca}\lambda_n
\]

\[
\text{iff,} \quad \eta^{mca}(\lambda_1 - \lambda_n) < 1 - \eta^{mca} - \eta^{mca}\lambda_n
\]

\[
\text{iff,} \quad \eta^{mca}\lambda_1 < 1 - \eta^{mca}
\]

\[
\text{iff,} \quad 0.5 < 1 - \eta^{mca} \quad \text{which is true}
\]

**Lemma 3:** Let \(C_i = \left[\begin{array}{c}a_i(0) \\ a_i(n)\end{array}\right]\) then,

\[
a_i(k) = C_i a_i^T a_i(k), \quad \forall i \in \{1, ..., n-1\}
\]

\[
a_i(k) = \frac{1}{\sqrt{\sum_j^{n-1} a_j^2(0) C_j^2 + 1}}
\]
Proof: Using Eq. (23) and the condition (30), we get
\[
\begin{align*}
    a_i(k+1) &= \left[ 1 - \eta^{mca} - \eta^{mca} \frac{1}{\lambda_i} \right] \frac{a_i(k)}{a_n(k)}, \\
    \forall i &\in \{1, ..., n-1\}
\end{align*}
\]

\[
\begin{align*}
    &= \left[ 1 - \eta^{mca} (\lambda_i - \lambda_n) \right] \frac{a_i(k)}{a_n(k)} \\
    &= \sigma_i \cdot \left[ \frac{a_i(0)}{a_n(0)} \right] \\
    &= \sigma_i^{k+1}, \left[ \frac{a_i(0)}{a_n(0)} \right]
\end{align*}
\]

This implies,
\[
a_i(k) = C_i \sigma_i^k a_n(k), \quad \forall i \in \{1, ..., n-1\}
\]

Using the result from Lemma 1 and substituting for i=n, we get
\[
a_n(k) = \frac{V_n^k a_n(0)}{\sqrt{2n} V_n^k a_n(0) / \sqrt{2n} a_n(0)^2 + 1}
\]

\[
\lim_{k \to \infty} a_i(k) = 0, \quad \forall i \in \{1, ..., n-1\}
\]

Therefore, from (21) \( w^{mca}(k) \) converges to the minor-component vector \( v_{n-1} \).

**Theorem 1:** Let \( \tau_{i/2} \) denote the half-life period of \( a_i(k) \), then the following inequality holds:
\[
\tau_{i/2}^{1/2} < \ldots < \tau_{n-1/2}^{1/2}
\]

**Proof:** Since \( a_n(k) \) is bounded (0 < \( a_n(k) < 1 \), \( a_i(k) \)'s \( \forall i \in \{1, ..., n-1\} \) belong to a family of exponential-decay functions: \( C_i a_i(k) e^{-k ln(1/\sigma_i)} \).

Therefore,
\[
\begin{align*}
    C_i a_i(k) e^{-k ln(1/\sigma_i)}
\end{align*}
\]

Using Lemma 3 and simplifying we get,
\[
k = -\frac{ln(2)}{ln \sigma_i} + \frac{0.5}{ln \sigma_i} \sum_{i=1}^{n-1} a_i^2 C_i^2 + 1 = \frac{1}{2} \sum_{j=1}^{n-1} C_j^2 + 1
\]

Let us denote the term
\[
\sum_{j=1}^{n-1} C_j^2 + 1
\]

as \( \xi \). It is clearly evident by using Lemma 3 that for \( k > 0 \), 0 < \( \xi < 1 \) and \( \xi \) is a monotonically decreasing function w.r.t. \( k \). However, for larger values of \( k \) and for consecutive \( \sigma_i \)'s, \( \xi \) can be assumed to be a constant. Substituting the term \( \xi \) in Eq. (36), we get
\[
\tau_{i/2}^{1/2} = \frac{ln(2) - 0.5 \cdot ln(\xi)}{ln \sigma_i} = \frac{ln(2) - 0.5 \cdot ln(\xi)}{ln(1/\sigma_i)}
\]

Therefore, from Eq. (37) and Lemma 2 we have,
\[
\tau_{j/2}^{1/2} < \tau_{j/2}^{1/2}, \quad \forall j \in \{2, ..., n-1\}
\]

**Theorem 2:** Given two input distributions \( x^1, x^2 \in \mathbb{R}^n \) and the eigendecomposition of their corresponding expected correlation-matrix: \( \{ v_i^1, \lambda_i^1 \}, \{ v_i^2, \lambda_i^2 \}, \forall i \in \{1, ..., n\} \), where \( v \) denotes the set of eigenvectors and \( \lambda \) their corresponding eigen-values \( \lambda_1 > \ldots > \lambda_n \geq 0 \). For an \( \eta^{mca} \) that satisfies,
\[
\eta^{mca} \lambda_1 < 0.5, \quad \eta^{mca} \lambda_2 < 0.5
\]

Then, the signal with a lower \( \sigma_{n-1} \) will have quicker convergence and therefore quicker learning progress.

**Proof:** From Theorem 1 it is clear that, the weight-vector \( w^{mca}(k) \) converges to the minor component \( v_n \) when the penultimate coefficient \( a_{n-1}(k) \) tends to 0. Therefore, a signal with lower \( \sigma_{n-1} \) will have a lower half-life period \( \tau_{n-1/2}^{1/2} \) and hence the weight-vector \( w^{mca}(k) \) converges quicker.

**Definition 2:** We therefore define \( \Omega(x) \) as a measure to indicate the learning progress of CIMCA for an input-distribution \( x \) equal to \( \sigma_{n-1} \) value, that is,
\[
\Omega(x) = \left[ 1 - \frac{1 - \eta^{mca}(\lambda_{n-1} - \lambda_n)}{1 - \eta^{mca} - \eta^{mca} \eta n} \right]
\]

**V. CONCLUSIONS**

A CD-MISFA agent autonomously explores multi-context environments. Compact context representations are learned from high-dimensional inputs through incremental slow feature analysis. Intrinsinc rewards for measurable learning progress tell the agent which context is temporarily "interesting," and when to actively engage in/disengage from a context or task. Such mechanisms are necessary from a computational perspective, and biological systems have evolved methods of achieving similar functional roles. In particular, while cortical regions of the brain are involved in unsupervised learning from sensory data (among other things), neuromodulatory systems are responsible for providing intrinsic rewards through dopamine, and regulating levels of attention to allow for task engagement and disengagement through norepinephrine. As artificial and robotic agents become increasingly sophisticated, they will not only look to biological solutions for inspiration, but may begin to resemble those solutions simply through the pressure of computational constraints.

**Acknowledgments.** We thank the IDSAni for many stimulating discussions. This work was funded through the 7th framework program of the EU under grants #231722 (IM-CLeVeR


