The final version of this article has been published in Neural Computation, 28(8):1599-1662 (2016) published by The MIT Press. This version does not differ significantly from the final version.

Optimal Curiosity-Driven Modular Incremental Slow Feature Analysis

Varun Raj Kompella, **Matthew Luciw**, **Marijn Frederik Stollenga**, **Juergen Schmidhuber**

IDSIA, SUPSI, USI, Galleria 2, Manno-Lugano 6928, Switzerland

{varun, matthew, marijn, juergen}@idsia.ch

Keywords: IncSFA (Incremental Slow Feature Analysis), Artificial Curiosity, Reinforcement Learning, Dimensionality Reduction, Online Learning.

Abstract

Consider a self-motivated artificial agent, who is exploring a complex environment. Part of the complexity is due to the raw high-dimensional sensory input streams, which the agent needs to make sense of. Such inputs can be compactly encoded through a variety of means - one of these is Slow Feature Analysis (SFA). Slow features encode spatio-temporal regularities, which are information-rich explanatory factors (*i.e.*, latent variables) underlying the high-dimensional input streams. In our previous work, we have shown how slow features can be learned incrementally, while the agent explores its world, and modularly, such that different sets of features are learned for different parts of the environment (since a single set of regularities does not explain everything). In what order should the agent explore the different parts of the environment? Following Schmidhuber's theory of Artificial Curiosity, the agent should always concentrate on the area where it can learn the *easiest to learn* set of features, which it has not already learned. We formalize this learning problem and theoretically show that, using our

model, called Curiosity-Driven Modular Incremental Slow Feature Analysis, the agent on an average will learn slow feature representations in order of increasing learning difficulty, under certain mild conditions. We provide experimental results to support the theoretical analysis.

1 Introduction

Consider a playroom setting for a baby humanoid robot, as shown in Figure 1. The robot is placed at a table with three objects. At any time, the robot holds its gaze on one of the objects. The figure illustrates three such "perspectives". Each perspective provides a stream of high-dimensional images to the robot, and each image stream will be compressible and predictable in some way. This robot is *intrinsically motivated* to learn the underlying *regularities* of the image streams as quickly as it can. In other words, it wants to maximize its learning progress. How should it direct its gaze to accomplish this? The robot receives no external supervision or external reward signal, nor any information about the *learning difficultly* of each perspective. If it wants to act *optimally*, it might first try to learn how difficult each perspective is to learn. This paper presents a mathematical formulation of this problem, a model that solves it, and a proof that shows this solution is optimal.

This work is based on the theory of *Artificial Curiosity* [AC; Schmidhuber, 2006b, 2010b]. An artificial agent driven by curiosity receives *intrinsic rewards* for its actions. Intrinsic rewards are proportional to the improvement of the agent's internal world model. Reinforcement learning (RL) is used to decide which actions lead to the highest intrinsic rewards. The curious agent is motivated to go to places where it expects to maximize learning progress [Schmidhuber, 1991, Storck et al., 1995, Schmidhuber, 1999a, 2006a, 2010a, Pape et al., 2012]. The improving world model of the robot is a set of *abstractions* or *latent variables*. The abstraction learning method we use is *Slow Feature Analysis* [SFA; Wiskott and Sejnowski, 2002, Franzius et al., 2007, Legenstein et al., 2010], an unsupervised learning algorithm that can extract spatio-temporal regularities from rapidly changing raw sensory inputs. SFA is based on the Slowness Principle [Földiák, 1991, Mitchison, 1991, Wallis and Rolls, 1997], which states that the underlying causes of changing signals vary more slowly than the primary sensory

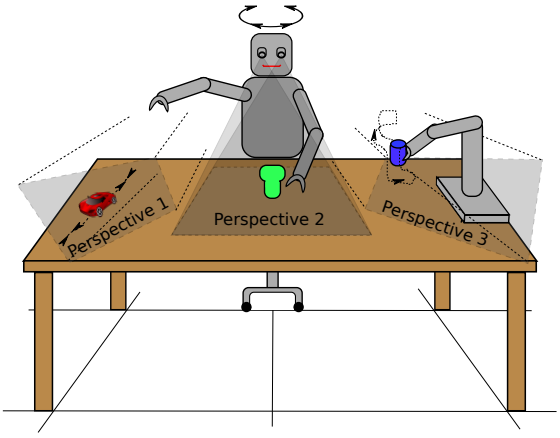

Figure 1: A playroom setting for a baby humanoid robot. There are three interesting areas for the robot to look at, but each area is not equally interesting. Perspective three shows another robot moving a block in a complex pattern, while perspective one shows a robot vehicle moving in a simple back and forth pattern. What sequence of perspectives should the robot choose to maximize learning progress?

stimulus. For example, individual retinal receptor responses or gray-scale pixel values of video will change much faster compared to the latent variables, such as the position of a moving object. SFA has achieved success in many problems, such as extraction of driving forces of a dynamical system [Wiskott, 2003], nonlinear blind source separation [Sprekeler et al., 2014], preprocessing for reinforcement learning [Legenstein et al., 2010, Kompella et al., 2011b], learning of place-cells, head-direction cells, grid-cells, and spatial view cells from high-dimensional visual input [Franzius et al., 2007], dynamic scene classification [Theriault et al., 2013], recognition of postures of a biped humanoid robot [Höfer et al., 2012], and learning human action sequences [Zhang and Tao, 2012, Sun et al., 2014].

In our previous work, we derived a low complexity, online implementation of SFA, called Incremental Slow Feature Analysis [IncSFA; Kompella et al., 2011a, 2012a]. IncSFA extracts slow features without estimating costly covariance matrices. Standard SFA techniques are not readily applicable to vision-based curiosity-driven online learning agents, as they estimate covariance matrices from the data via batch processing. But IncSFA is suitable to use in online learning applications with high-dimensional inputs.

IncSFA, like most online learning approaches, gradually forgets previously learned

representations whenever the statistics of the input change, e.g.,when the robot shifts its gaze to another perspective (see Figure 1). To prevent forgetting, we developed a modular version of IncSFA, called Curiosity-Driven Modular Incremental Slow Feature Analysis (Curious Dr. MISFA). Curious Dr. MISFA retains previously learned abstractions in the form of *expert modules* [Ring, 1994], and actively learns multiple expert modules in order of increasing learning difficulty. Each abstraction learned can be used by a reinforcement learner to map the potentially high-dimensional visual inputs to useful action sequences [Legenstein et al., 2010, Kompella et al., 2011b, 2014]. The next section discusses the novel contributions of this paper.

1.1 Contributions

Our previous work includes an initial implementation of Curious Dr. MISFA [Kompella et al., 2012b], a discussion on its neurophysiological correlates [Luciw et al., 2013] and its application to high-dimensional image streams captured from the camera-eyes of a humanoid iCub robot [Luciw et al., 2013, Kompella et al., 2014, 2015]. The novel contributions of this paper are as follows.

- An improved Curious Dr. MISFA algorithm with a new intrinsic reward function that is crucial to ensure its stability.
- A theoretical formulation of the learning problem associated with Curious Dr. MISFA.
- A formal analysis of the average dynamics of Curious Dr. MISFA, where we show that the abstractions are learned in the order of increasing learning difficulty, under certain mild conditions. Although not explicitly shown in this paper, this analysis is not limited to IncSFA and can be extended to an other abstraction learning algorithm.
- An experimental validation conducted on test signals to support the analysis.
- Design extensions of the method to make it applicable to maze environments.

The rest of paper is organized as follows. Section 2 presents an overview of the Curious Dr. MISFA algorithm along with a theoretical formulation of the learning problem. Section 3 discusses the crucial parts of the algorithm in detail. A formal analysis

of the dynamics of the algorithm is presented in Section 4. Section 5 summarizes the algorithm with a pseudocode and a discussion on tuning hyper-parameters. Section 6 presents experimental results. Design extensions to make the algorithm applicable to specific maze domains are discussed in Section 7. Section 8 concludes the paper.

2 Overview

In this section, we present an overview of the Curious Dr. MISFA. See Figure 2 for a schematic diagram of the algorithm.

Environment. The input to the algorithm is a set of pre-defined observation streams $X = \{\mathbf{x}_1, ..., \mathbf{x}_n : \mathbf{x}_i(t) = \left(x_i^1(t), x_i^2(t), ..., x_i^I(t)\right) \in \mathbb{R}^{I \in \mathbb{N}}\}$, which may or may not be unique. Each stream $\mathbf{x}_i$ could potentially represent a video sequence of images observed for a particular head rotation angle (perspective) of a camera-equipped humanoid robot. It can also represent a sequence of images observed while the agent is executing a particular task. In addition to the streams, the algorithm may also receive a finite, low-dimensional, *digital signal*[1] $\mathbf{u}, \mathbf{u}(t) \in U, U \subset \mathbb{R}^{P \in \mathbb{N}}$, for *e.g.*, a sequence of discretized joint-angle observations (proprioception) of the humanoid robot. At any time t, the agent receives an observation $\mathbf{x}(t) \in \{\mathbf{x}_1(t), ..., \mathbf{x}_n(t)\}$ from only one of the streams. The agent explores the streams with two actions: {stay, switch}. When the agent takes the *stay* action, the current stream $\mathbf{x}_i$ remains the same and it receives a set of τ observations from that stream. We denote the τ observation set as $\mathbf{x}(t; \tau) = [\mathbf{x}_i(t), ..., \mathbf{x}_i(t+\tau)]$. When it takes the action *switch*, the agent selects a stream $\mathbf{x}_{j \neq i}$ randomly from one of the other $n - 1$ streams and it receives τ number of observations from the new stream $\mathbf{x}(t'; \tau) = [\mathbf{x}_j(t'), ..., \mathbf{x}_j(t' + \tau)]$ (see Section 3.2 for details on why this is crucial). Next, we discuss the goal of the algorithm.

Goal. The desired goal of the algorithm is to learn a sequence of abstractions $\Phi = \{\phi_1, ..., \phi_m; m \leq n\}$, where each abstraction is unique and encodes some underlying regularity within one or more observation streams. The order of the sequence is such that, the first abstraction learned corresponds to the easiest encodable observation stream. Since the learning difficulty of the observation streams is not known *a priori*, the learning process involves estimating not just the abstractions, but also the

[1] A digital signal is a discrete-time signal for which both time and amplitude have discrete values.

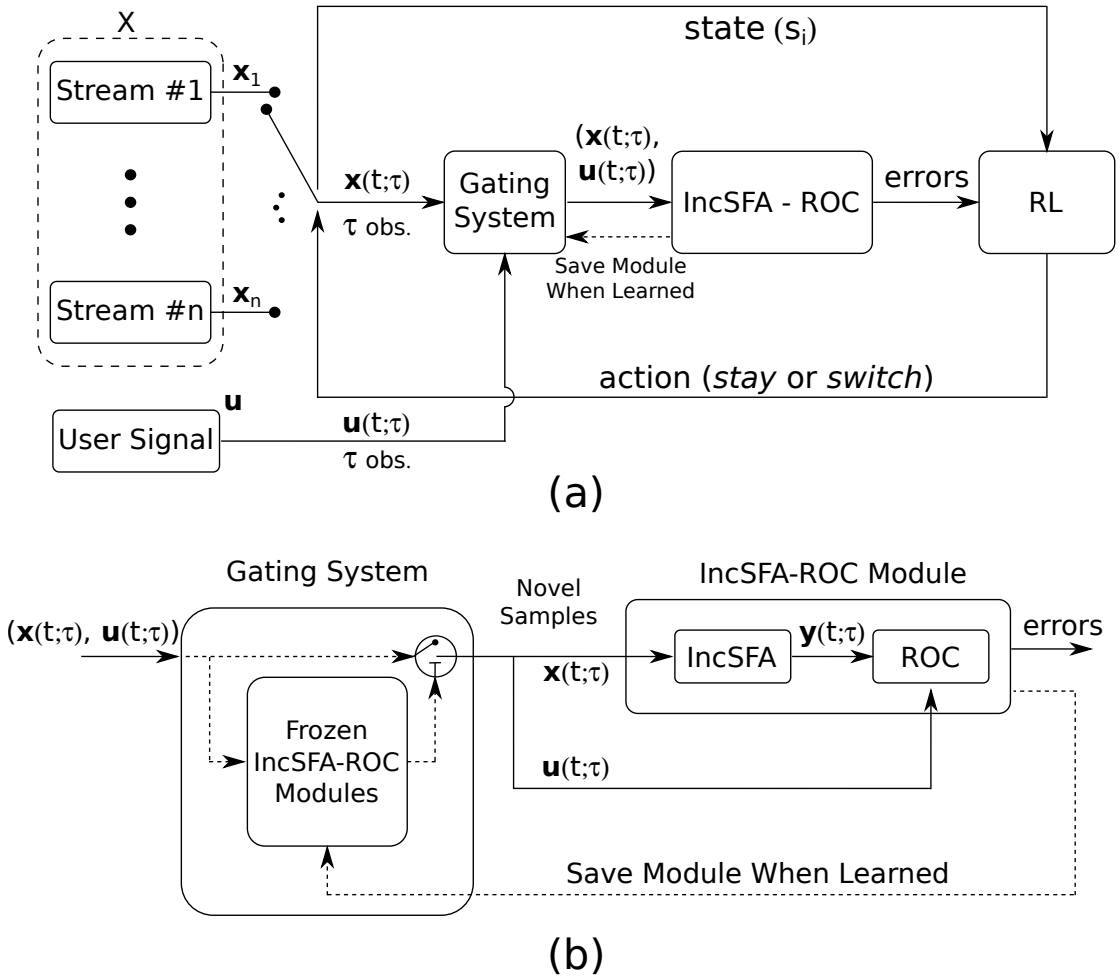

Figure 2: **Schematic Diagrams of Curious Dr. MISFA.** (a) The overall architecture includes a Reinforcement Learner (RL), a gating system and an adaptive IncSFA-ROC module. The input to the algorithm is a set of observation streams, where each stream could potentially represent a video sequence of images observed for a particular head rotation angle of a humanoid robot. An additional input to the algorithm is a user-defined time-varying signal, for *e.g.*, the joint-angle observations of a humanoid robot. The RL decides which stream to select using actions *stay* or *switch*, to make a set of τ observations. (b) Inner details of the gating system and the IncSFA-ROC module. Gating system uses the learned modules to detect if the observed inputs are novel. IncSFA updates slow features from the novel inputs and the ROC estimates the slow feature outputs with respect to the user-signal observations. The errors generated through the module are used to update the RL. When the IncSFA-ROC module is learned, it is saved in the gating system and a new module is created. See text for more details.

order in which the observation streams need to be encoded.

Architecture. The overall architecture of the Curious Dr. MISFA agent that achieves this goal includes the following (see Figure 2(a)):

1. A gating system that detects if the input observations are novel.

2. An adaptive Incremental Slow Feature Analysis [IncSFA; Kompella et al., 2011a, 2012a] coupled with a Robust Online Clustering [ROC; Guedalia et al., 1999, Zhang et al., 2005] module (denoted by Θ) that updates an IncSFA-ROC abstraction based on the novel input observations.

3. A Reinforcement Learner (RL) that finds the stream that is the easiest to learn based on the learning progress made by the adaptive IncSFA-ROC module.

We discuss next how the control flows within this architecture.

Control Flow. The algorithm begins with no previously learned modules. The agent receives τ observations each from the current stream and the user-signal: $(\mathbf{x}(t;\tau), \mathbf{u}(t;\tau))$, where $\mathbf{u}(t;\tau) = [\mathbf{u}(t), ..., \mathbf{u}(t+\tau)]$. Since there are no previously learned modules, the tuple of τ-observations set $(\mathbf{x}(t;\tau), \mathbf{u}(t;\tau))$ is novel. The novel tuple is an input to the adaptive IncSFA-ROC module (Figure 2(b)). IncSFA updates a slow feature matrix ($\widehat{\phi}^{\text{sfa}}$; a real-valued matrix of size $I \times J$) based on the input $\mathbf{x}(t;\tau)$. The slow feature matrix $\widehat{\phi}^{\text{sfa}} : \mathbf{x}(t) \mapsto \mathbf{y}(t)$ maps the observations $\mathbf{x}(t;\tau)$ to a lower-dimensional output $\mathbf{y}(t;\tau) = [\mathbf{y}(t), ..., \mathbf{y}(t+\tau)], \mathbf{y}(t) \in \mathbb{R}^{J \in \mathbb{N}}, J \ll I$, such that, $\mathbf{y}(t;\tau) = \widehat{\phi}^{\text{sfa}} \cdot \mathbf{x}(t;\tau)$. The ROC algorithm updates discrete cluster centers $\widehat{\phi}^{\text{roc}}$ improving its estimates of the slow feature output $\mathbf{y}(t;\tau)$ with respect to the user-signal observations $\mathbf{u}(t;\tau)$, such that, $\widehat{\phi}^{\text{roc}} : (\mathbf{y}(t), \mathbf{u}(t)) \mapsto \widehat{\mathbf{y}}(t)$. The user-signal observations $\mathbf{u}(t;\tau)$ (defined earlier) act as meta-class variables for clustering the data. $\widehat{\mathbf{y}}(t;\tau)$ are discrete estimates of $\mathbf{y}(t;\tau)$ (see Section 3.1 for more details). The adaptive abstraction $\widehat{\phi}$ is a tuple of real-valued matrices $\widehat{\phi} = (\widehat{\phi}^{\text{sfa}}, \widehat{\phi}^{\text{roc}})$. The overall IncSFA-ROC update step can be summarized as:

$$\widehat{\phi} \leftarrow \Theta\left((\mathbf{x}(t;\tau), \mathbf{u}(t;\tau)), \widehat{\phi}\right). \tag{1}$$

After each update the learning error of the IncSFA-ROC module is computed as a tuple $\xi(t) = \left(\xi^{\text{sfa}}(t), \xi^{\text{roc}}(t)\right)$. $\xi^{\text{sfa}}(t)$ is the change of slow feature matrix over time $\left(\xi^{\text{sfa}}(t) = \|\widehat{\phi}^{\text{sfa}}(t) - \widehat{\phi}^{\text{sfa}}(t+1)\|\right)$, where $\|.\|$ denotes the Frobenius matrix norm. $\xi^{\text{roc}}(t)$ is the

estimation error of the ROC. An expression for the ROC estimation error is discussed later in Section 3.1. The IncSFA-ROC error $\xi(t)$ is then used to update the RL.

The RL is used to accomplish two tasks (a) find the easiest novel stream to encode, and (b) stick to that stream long enough to learn an expert IncSFA-ROC module. To this end, it learns a stream selection policy that outputs an action (*stay* or *switch*) for each stream. The state and action space of the RL are defined as $\mathcal{S} = \{s_1, ..., s_n\}$, where s_i denotes the stream identification number, and $\mathcal{A} = \{\text{stay}, \text{switch}\}$. RL learns the deterministic stream selection policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ by updating a reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ based on the IncSFA-ROC error. The agent then uses a decaying ϵ-greedy strategy [Sutton and Barto, 1998] to follow the learned policy with a probability of $1 - \epsilon$, and with ϵ probability it follows a policy that outputs actions *stay* and *switch* uniform randomly for each state. Let this stochastic policy be denoted by $\pi^b : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, such that, $\pi^b = \epsilon\text{-greedy}(\pi)$. The agent takes a new action based on π^b and the process repeats. When $\xi^{\text{roc}}(t) < \delta(\approx 0)$, the adaptive module is saved $\phi = (\phi^{\text{sfa}}, \phi^{\text{roc}}) \leftarrow (\widehat{\phi}^{\text{sfa}}, \widehat{\phi}^{\text{roc}})$ and added to the abstraction library:

$$\Phi \leftarrow \Phi \cup \phi. \tag{2}$$

Once the abstraction is saved, it is never updated again (frozen) and a new adaptive module $\widehat{\phi}$ is created. The gating system uses the stored frozen IncSFA-ROC modules to generate a gating signal that filters known or similar input observations. To this end, it first computes the ROC estimation errors of all the frozen modules for the new input $(\mathbf{x}(t;\tau), \mathbf{u}(t;\tau))$ and filters the input if any of the estimation errors is less than δ. Otherwise, the input is forwarded to the adaptive IncSFA-ROC module. Therefore, ROC plays a crucial role in the algorithm as it contributes in deciding (a) when to stop updating the adaptive module by checking if its estimation error falls below δ, and (b) if the new inputs are novel by checking the errors of the frozen modules. Additionally, ROC also biases the algorithm to learn slow features whose outputs are predictable with respect to the user-signal. For example, if the user-signal observations are discretized joint angles of a humanoid robot, then the slow features learned by IncSFA-ROC would encode environment variations that are correlated to the joint-angle transitions (*e.g.*, when the robot grasps a cup). Other independent environment variations like people moving in the background, illumination, *etc.*, are not considered because the estimation

error would never go below the threshold δ. This slow feature learning bias of the ROC is especially useful for humanoid applications. In applications where such a signal is unavailable or difficult to provide, the algorithm assumes $\mathbf{u}(t)$ to indicate a time-index of a fixed arbitrary period, for *e.g.*, $U = [0, T]$ and $\mathbf{u}(t) = t\%(T+1)$, where $\%$ denotes the modulo operator. Next, we formalize the problem of learning modular abstractions in the order of increasing learning difficulty.

Learning Problem Formalized. The underlying learning problem associated with Curious Dr. MISFA can be formulated as an *optimization* problem. Simply put, the problem states that for a given set of time-varying observation streams, an abstraction corresponding to the most easily learnable yet unknown observation stream is learned first. The optimization problem is not specific to learning modular slow features, therefore, Θ denotes here any general abstraction-estimator that converges to a fixed-point. To keep it simple, we skip the user-signal $\mathbf{u}$. Later in Section 4, we will show that the Curious Dr. MISFA algorithm converges to the *optimal* solution of the proposed problem. We introduce here some additional notation required for the formulation:

Encoded Streams: Let $X^{\phi_i} \subseteq X$ denote the set of observation streams such that the average abstraction-estimation error $\langle \|\Theta(\mathbf{x}_j(t), \phi_i) - \phi_i\| \rangle_t^\tau \leq \delta, \forall \mathbf{x}_j \in X^{\phi_i}$. X^{ϕ_i} represents the streams encoded by the abstraction ϕ_i. $\langle . \rangle_t$ indicates averaging over time, $\langle . \rangle_t^\tau$ indicates windowed-average over time with a fixed window size τ and $\forall$ indicates *for all*.

Curiosity Function: Let $\Omega : X \rightarrow [0, 1)$ denote a function indicating the speed of learning an abstraction by the abstraction-estimator Θ. Easily learnable inputs have lower values of Ω. Ω induces a total ordering among the observation streams making them comparable in terms of the learning difficulty (see Section 4 for a proof on the existence of such a function).

The optimization problem is formulated as follows: Given the input X, find a set of m abstractions $[\phi_1, ..., \phi_{m \leq n}]$, such that, for each $i \in \{1, ..., m\}$

$$\Omega\big(\mathbf{x}_i \in X^{\phi_i}\big) \text{ is minimal,}$$

under the constraints,

$$\langle y_i^j \rangle_t = 0, \;\; \langle \left(y_i^j\right)^2 \rangle_t = 1, \quad \forall j \in \{1, ..., J\} \qquad \text{(std. normal stats)} \tag{3}$$

$$\left(\begin{array}{c} \forall \phi_i \in \Phi, \exists \mathbf{x}_j \in X, \\ \text{and} \;\; \forall \phi_{k<i} \in \Phi \end{array} \right) : \begin{array}{l} \langle \|\Theta(\mathbf{x}_j(t), \phi_i) - \phi_i\| \rangle_t^\tau \leq \delta \;\; \text{(at least one stream encoded)} \\ \langle \|\Theta(\mathbf{x}_j(t), \phi_k) - \phi_k\| \rangle_t^\tau > \delta \;\; \text{(unique abstraction learned)} \end{array} \tag{4}$$

The goal here is to find abstractions that encode the top easiest to learn observation streams. Constraint (3) requires that the abstraction-output components have zero mean and unit variance. This constraint enables the abstractions to be non-zero and avoids learning features for constant observation streams. Constraint (4) requires that a *unique* abstraction be learned that encodes *at least* one of the streams, avoiding redundancy. This constraint also induces an order in the abstractions learned, such that, $\Omega(\mathbf{x}_i \in X^{\phi_1}) < \Omega(\mathbf{x}_i \in X^{\phi_2}) < ... < \Omega(\mathbf{x}_i \in X^{\phi_m})$. This is explained as follows. When learning ϕ_1, only the first part of Constraint (4) applies and therefore the objective is minimal when ϕ_1 encodes the observation stream ($\in X$) with the lowest Ω value. When learning ϕ_2, the second part of Constraint (4) ensures that ϕ_2 does not encode any $\mathbf{x}_i \in X^{\phi_1}$. Therefore, ϕ_2 encodes the observation stream ($\in \{\mathbf{x}_i \in X | \mathbf{x}_i \notin X^{\phi_1}\}$) with the lowest Ω value. The same reasoning follows for the rest resulting in a set of abstractions ordered according to the increasing Ω-values of the corresponding observation streams that they encode.

Finding the optimal solution to the above problem is straightforward when the curiosity function values for each observation stream are known *a priori*. However, this is generally not the case. One possible approach to address this is to find (a) an analytical expression of Ω for the particular abstraction-estimator Θ and (b) an input sampling technique that can estimate the Ω values for each observation stream. However, this approach is dependent on the Θ used. A more general approach is to use the *learning progress* of Θ, while exploring using reinforcement learning (RL) to estimate the Ω values in the form of *curiosity rewards* for each observation stream. This approach is independent of the abstraction-estimator used. However, it requires learning an observation stream selection policy π, and at the same time the abstraction from the incoming observations based on the (imperfect) policy π. Curious Dr. MISFA employs the later approach to address this problem. Next, we discuss details of some crucial design aspects of the Curious Dr. MISFA algorithm.

3 Method Description

Two crucial learning blocks of the Curious Dr. MISFA algorithm are the unsupervised abstraction learning block and the reinforcement policy learning block. Curious Dr. MISFA updates both these online learning methods simultaneously such that the overall system converges to the optimal solution discussed in the previous section. We discuss here in detail how these learning updates are carried out.

3.1 Learning Abstractions using IncSFA-ROC

Curious Dr. MISFA's abstraction-estimator is the IncSFA-ROC algorithm. IncSFA is used to learn real-valued slow features of the input observations whereas the ROC is used to learn a discrete model mapping the slow feature outputs with respect to the user-signal. We discuss here more details on the individual algorithms.

IncSFA: IncSFA is an unsupervised learning technique that extracts features from an input stream with the objective of maintaining an informative but slowly-changing feature response over time. IncSFA is an online implementation of the original batch Slow Feature Analysis (SFA; Wiskott and Sejnowski [2002]) whose optimization problem is as follows: Given an I-dimensional input stream $\mathbf{x}(t) = (x^1(t), ..., x^I(t))$, find a set of J instantaneous real-valued functions $\mathbf{g} = [g^1, ..., g^J]$, which together generate a J-dimensional output stream $\mathbf{y}(t) = (y^1(t), ..., y^J(t))$ with $y^i(t) = g^i(\mathbf{x}(t))$, such that for each $i \in \{1, ..., J\}$

$$\Delta_i = \Delta(y^i) = \langle (\dot{y}^i)^2 \rangle_t \quad \text{is minimal} \tag{5}$$

under the constraints

$$\langle y^i \rangle_t = 0 \quad \text{(zero mean)}, \tag{6}$$

$$\langle (y^i)^2 \rangle_t = 1 \quad \text{(unit variance)}, \tag{7}$$

$$\forall j < i : \langle y^i y^j \rangle_t = 0 \quad \text{(decorrelation and order)}, \tag{8}$$

where $\dot{y}$ denotes the time derivative of y. The goal is to find instantaneous functions g^j generating different output streams that are as *slowly varying* as possible. The decorrelation constraint (8) ensures that different functions g^j do not code for the same features. The other constraints (6) and (7) avoid trivial constant output solutions. IncSFA,

like SFA, uses a simpler eigenvector based approach to find a linear-approximate solution to the problem (ϕ^{sfa} = linear-approximation (g)). The input is first incrementally *whitened* using a Candid Covariance-Free Incremental PCA (CCIPCA; [Weng et al., 2003, Zhang and Weng, 2001]), such that, the whitened input has unit covariance. Then, the eigenvectors (minor components) with the smallest eigenvalues of the derivative of the whitened input are extracted using Minor Component Analysis (MCA; [Oja, 1992, Peng and Yi, 2006, Peng et al., 2007]). Slow feature vectors are an inner product of the CCIPCA and MCA weight vectors. CCIPCA requires a dynamic learning rate scheduling that is automatically set, whereas MCA requires a constant learning rate (η^{mca}) that needs to be hand-set for each experiment. In this paper, we refer to η^{mca} as the learning rate of the IncSFA η^{sfa}.

To handle quadratic non-linearities, the input $\mathbf{x}(t) = [x^1(t), ..., x^I(t)]$ is expanded over a quadratic space $[(x^1(t))^2, (x^2(t))^2, ..., (x^1(t)x^2(t)), ..., x^1(t), ..., x^I(t)]$ using a quadratic kernel and the linear IncSFA is applied on the expanded input [Kompella et al., 2011a]. For extracting higher non-linearities, quadratic IncSFAs can be applied in a deep converging hierarchy [Luciw et al., 2012], or a linear IncSFA can be combined with a non-linear auto-associative neural network [Kompella et al., 2011b].

IncSFA learns *instantaneous* features from sequential data. Relevance cannot be *uncovered* without taking time into account, but once it is known, each input frame in most cases can be encoded on its own. Due to this, IncSFA differs from both (1) many well-known unsupervised feature extractors [Abut, 1990, Jolliffe, 1986, Comon, 1994, Lee and Seung, 1999, Kohonen, 2001, Hinton, 2002] that ignore dynamics, and (2) other UL systems that both learn and apply features to sequences [Schmidhuber, 1992a,c,b, Lindstädt, 1993, Klapper-Rybicka et al., 2001, Jenkins and Matarić, 2004, Lee et al., Gisslén et al., 2011], thereby assuming that the state of the system itself can depend on past information.

The compact relevant encodings uncovered by IncSFA reduce the search space for downstream goal-directed learning procedures [Schmidhuber, 1999b, Barlow, 2001], especially reinforcement learning. As an example, consider a robot sensing with an onboard camera. Reinforcement learning algorithms applied directly to pixels can be quite inefficient due to the size of the search space. Slow features can encode each image into a small set of useful state variables, and the robot can use these few state

variables to quickly develop useful control policies. The state variables from SFA are approximations of low-order eigenvectors of the graph Laplacian [Sprekeler, 2011], *i.e.*, proto-value functions [Mahadevan and Maggioni, 2007]. This is why they are typically more useful as features in reinforcement learning in comparison with other types of features, such as principal components. More details on IncSFA and its applications to high-dimensional image inputs can be found in our previous work [Kompella et al., 2012a, 2011b]. Next, we discuss briefly how the ROC is coupled to the IncSFA.

ROC: The 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 either *merging* the two most similar cluster centers or by *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 newest input sample, which may mark the beginning of a new input process. ROC has two main hyper parameters: maximum number of clusters centers N^{roc} and amnesic rate η^{roc}. N^{roc} limits the algorithm to learn at most N^{roc} clusters and η^{roc} is a memory parameter that biases the algorithm to adapt clusters based on the recent history of observations.

In Curious Dr. MISFA, the ROC algorithm is coupled to the IncSFA algorithm to learn discrete cluster centers $\widehat{\phi}^{\text{roc}}$ estimating the slow feature output $\mathbf{y}(t;\tau)$ with respect to the user-signal observations $\mathbf{u}(t;\tau)$, such that, $\widehat{\phi}^{\text{roc}} : (\mathbf{u}(t), \mathbf{y}(t)) \mapsto \widehat{\mathbf{y}}(t)$. $\widehat{\mathbf{y}}(t;\tau)$ are discrete estimates of $\mathbf{y}(t;\tau)$. Learning $\widehat{\phi}^{\text{roc}}$ can be computationally intensive. We simplify the learning process by using a *lookup* table approach since U is a discrete bounded set (say U has p elements $= \{\mathbf{u}_1, ..., \mathbf{u}_p\}$). For each $\mathbf{u}_i \in U$, we associate an ROC instantiation (node). Therefore, $\widehat{\phi}^{\text{roc}} = \{\widehat{\phi}^{\text{roc}}_{\mathbf{u}_1}, ..., \widehat{\phi}^{\text{roc}}_{\mathbf{u}_p}\}$. For each tuple $(\mathbf{u}(t), \mathbf{y}(t))$, $\widehat{\phi}^{\text{roc}}_{\mathbf{u}_i=\mathbf{u}(t)}$ is updated with the slow feature output $\mathbf{y}(t)$ and the estimation error is computed and stored $\xi^{\text{roc}}_{\mathbf{u}_i=\mathbf{u}(t)} = \|\widehat{\mathbf{y}}(t) - \mathbf{y}(t)\|$. Therefore, at any time t, only one ROC node is updated. The total ROC estimation error (Figure 3) is the sum of stored errors of all the ROC nodes:

$$\xi^{\text{roc}}(t) = \sum_{i=1}^{p} \xi^{\text{roc}}_{\mathbf{u}_i}(t). \tag{9}$$

To illustrate the working of IncSFA-ROC algorithm, consider an example from our previous work [Luciw et al., 2013]. A robot is placed next to a table with a plastic cup in

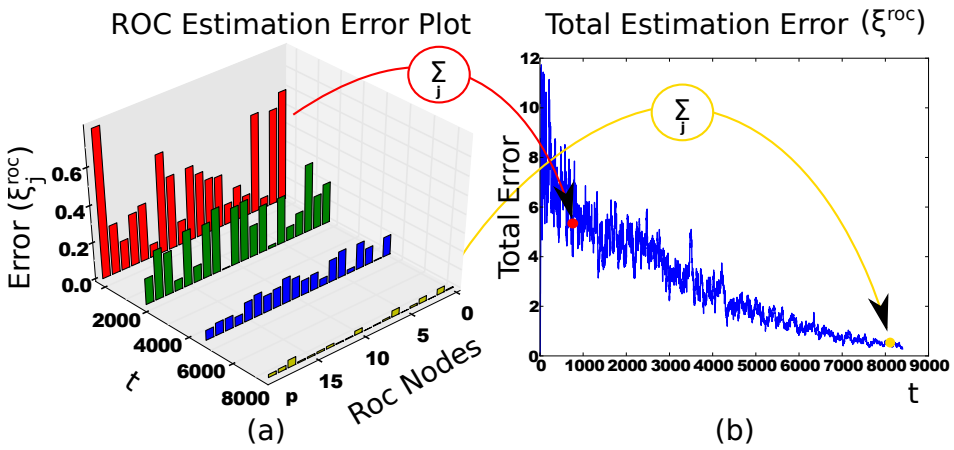

Figure 3: An example estimation error over time of the ROC-algorithm. (a) The estimation error of 20 instances of ROC nodes to estimate the IncSFA output to 20 values. (b) The total estimation-error $\xi^{\text{roc}}(t)$ is the sum of stored errors of all the nodes.

its reach. It explores by moving its right hand through a random-walk along its shoulder joint. As a consequence of its exploration it topples the cup (Figures 4(b)-(c)). After a few time steps the cup is put back in its standing position and the toppling event repeats. IncSFA-ROC receives a continuous sequence of gray-scaled image observations ($\mathbf{x}(t)$; downscaled to 100x80 pixels) and the user-signal observations which are the shoulder-joint angles discretized into *p=20* bins ($\mathbf{u}(t) \in U = \{\mathbf{u}_1, ..., \mathbf{u}_{20}\}$). For each $\mathbf{u}_i$, there is an associated instance of the ROC algorithm, resulting to *p=20* instances of the ROC algorithm. A developing slow feature output here is a step function (Figure 4(d)), e.g., when the object is not toppled the feature output equals ≈ -1.5, and when the object is toppled the feature output equals ≈ 0.5, invariant to the robot's arm position and other variations in the image sequence. Upon convergence of the IncSFA first and the ROC second, each joint angle will be mapped to two cluster centers (Figure 4(e)), (except for the joint angles $\mathbf{u}_{15} - \mathbf{u}_{20}$, where the robot's hand is to the left of the object's position and the object cannot be in a not-toppled position) providing information about invariants captured with IncSFA. The slow feature step function output along with the learned model (in the form of cluster center estimates) can easily drive a subsequent planning algorithm to learn a policy that efficiently topples the cup. Next we discuss, how the RL in Curious Dr. MISFA uses the updating IncSFA-ROC abstraction algorithm to guide learning the stream selection policy.

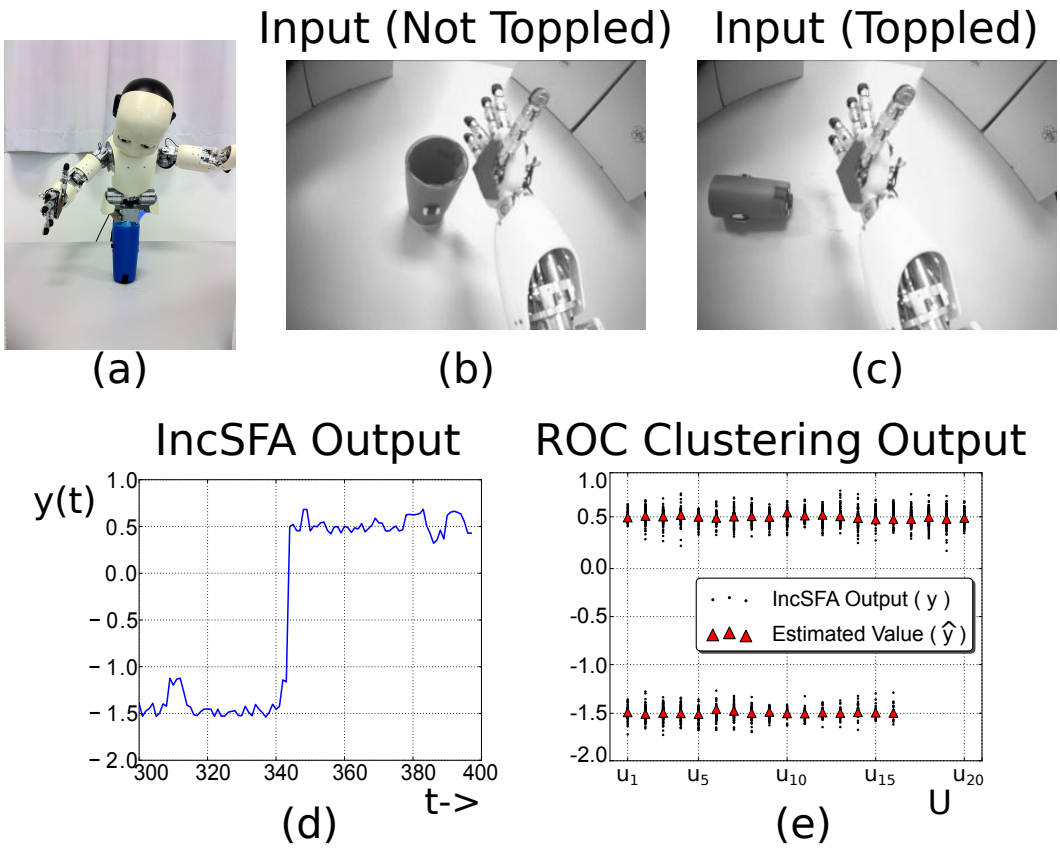

Figure 4: IncSFA-ROC example experiment. (a) A robot is placed next to a table with a plastic cup in its reach. It explores by moving its right hand through a random-walk along its shoulder joint. As a consequence of its exploration it topples the cup. After a few time steps the cup is put back in its standing position and the toppling event repeats. IncSFA-ROC receives a continuous sequence of gray-scaled image observations ($\mathbf{x}(t)$; downscaled to 100x80 pixels) and the shoulder-joint angles ($\mathbf{u}(t) \in U = \{\mathbf{u}_1, ..., \mathbf{u}_{20}\}$). (b) A sample input image when the cup is not toppled. (c) A sample input image when the cup is toppled. After a few 100 time steps of random walk, IncSFA-ROC converges. (d) 100 time steps of IncSFA output plotted against time after convergence. The output is a step function indicating whether the cup is toppled or not, invariant to the position of its hand visible in the image. (e) ROC learns cluster centers estimating different slow feature output values (blue dots) for each joint angle $\mathbf{u}_i \in U$. Each joint-angle has two estimates (red triangles) denoting two states of the cup (toppled or not).

3.2 Learning Stream Selection Policy Using RL

Section 2 presented an overview of the basic functioning of the RL agent. Here, we discuss more details of the RL algorithm and how it learns the stream selection policy.

One crucial design aspect of the RL used is the novel action space = $\{$*stay*, *switch*$\}$, where the action *stay* makes the agent's state to be the same as the previous state, while *switch* randomly shifts the agent's state to one of the other neighboring states with equal probability. The random nature of the switching forces the IncSFA-ROC algorithm to not encode any regularity while switching between the states. This is crucial to ensure the stability of the method by avoiding combinatorial possibilities of generating a coherent stream of data through deterministically switching between a few states at different times.

The goal of the RL agent is to learn the deterministic stream selection policy π that (a) finds the easiest novel stream to encode, and (b) sticks to that stream long enough to learn an expert IncSFA-ROC module. To this end, the RL optimizes the following cost function:

$$\mathcal{J} = \min_{\pi} \ (\dot{\xi}^{\text{sfa}}(t), \xi^{\text{roc}}(t)) \tag{10}$$

Eq. (10) is a multi-objective reinforcement learning problem (MORL; [Gábor et al., 1998, Vamplew et al., 2011]). Minimization of the first objective would result in a policy that will shift the agent to states where the error decreases sharply ($\dot{\xi}^{\text{sfa}}(t) < 0$), indicating faster learning progress of the IncSFA. While, minimization of the second objective would result in a policy that will improve the developing IncSFA-ROC abstraction to satisfy the Constraint (4). A cost function with only one of these terms may not be sufficient to solve the problem since using only the first objective, the agent would never be motivated to learn an expert module. While, using only the second objective, the agent would be motivated to learn constant streams or streams that change slowly (near constant streams or difficult-to-learn streams).

Optimizing the two objectives simultaneously is not straightforward since they are correlated and partially conflicting: optimizing the first objective aids in optimizing the second, however, optimizing the second objective would result in an increasing error-gradient $\dot{\xi}^{\text{sfa}}(t)$ (from a negative value to 0), which conflicts with the first. Therefore, there does not exist a single policy that simultaneously optimizes each objective. We instead use an approach to find a dynamically changing *pareto-optimal policy* [Vamplew et al., 2011] by prioritizing each objective based on the error $\xi^{\text{roc}}(t)$. To this end, we scalarize the cost in terms of a scalar *reward* r that evaluates the current τ-samples input

$(\mathbf{x}(t;\tau), \mathbf{u}(t;\tau))$ received for the tuple (current state s, current action a, future state s') as follows:

$$r_a^{ss'} = \left(- \int_\tau \dot{\xi}^{\text{sfa}} dt + \beta Z^{\delta,\sigma}(\langle \xi^{\text{roc}} \rangle_t^\tau) \right) \tag{11}$$

where Z represents a Gaussian function $Z^{\delta,\sigma}(x) = e^{-\frac{(x-\delta)^2}{2\sigma^2}}$. δ, σ and β are scalar constants. We refer to $-\int_\tau \dot{\xi}^{\text{sfa}} dt$ as the curiosity-reward term and $Z^{\delta,\sigma}(\langle \xi^{\text{roc}} \rangle_t^\tau)$ as the expert-reward term. A fast reliable approximation of these terms are computed from τ observations as follows:

$$\langle \xi^{\text{roc}} \rangle_t^\tau = \frac{1}{\tau} \sum_t^{t+\tau-1} \xi^{\text{roc}}(t), \tag{12}$$

$$\int_\tau \dot{\xi}^{\text{sfa}} dt = \sum_t^{t+\tau-2} \left(\|\widehat{\phi}^{\text{sfa}}(t+2) - \widehat{\phi}^{\text{sfa}}(t+1)\| - \|\widehat{\phi}^{\text{sfa}}(t+1) - \widehat{\phi}^{\text{sfa}}(t)\| \right) \tag{13}$$

An intuition behind using the Gaussian function to represent the expert-reward term is as follows: (a) to compensate for the exponential decrease of the curiosity-reward term (discussed in detail in Section 4) and (b) the expert-reward needs to increase monotonously as the error gets closer to the threshold δ. δ is usually selected to be a small value close to zero. σ determines the contribution of the expert-rewards in the total reward. We discuss later in Section 5 how to tune these parameters.

A model $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ of these instantaneous rewards is updated as:

$$\widetilde{R}_a^{ss'} \leftarrow \alpha \, r_a^{ss'} + \, (1-\alpha)\widetilde{R}_a^{ss'}; \; R \leftarrow \widetilde{R}/\|\widetilde{R}\|, \tag{14}$$

where α is a constant smoothing coefficient and $\|\widetilde{R}\|$ is a scalar.

The transition model $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ of the environment dynamics resembles that of a *Complete-Graph*, where each state $s_i \in \mathcal{S}$ is represented by a node in a fully-connected undirected-graph. Figure 5 illustrates the model.

$$P_{\text{stay}}^{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}, \; P_{\text{switch}}^{ij} = \begin{cases} 0, & \text{if } i = j \\ \frac{1}{n-1}, & \text{if } i \neq j \end{cases}, \quad \forall i, j \in [1, ..., n]. \tag{15}$$

The agent has the capability to shift between any of observation streams, similar to switching between channels of a television.

After every τ sample observations, the reward function estimate R along with the complete-graph transition model P are used to learn a new approximate value function

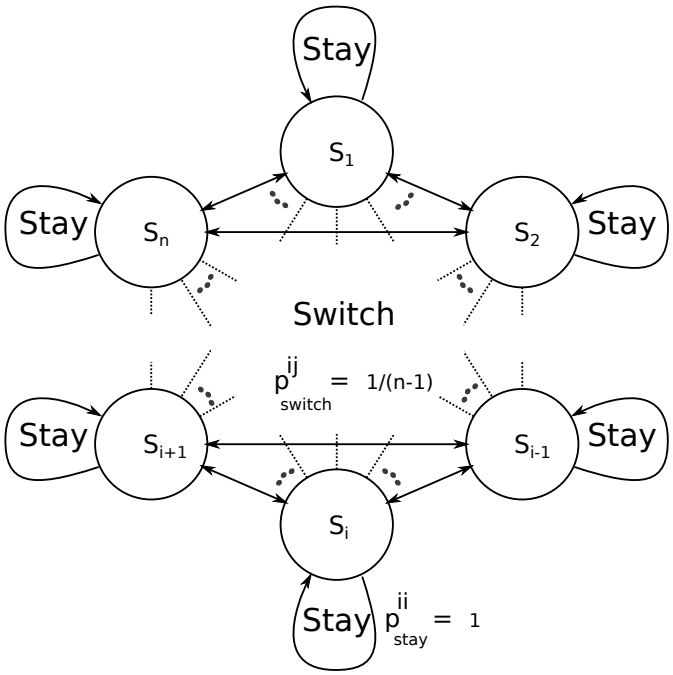

Figure 5: The state-action transition model of the agent's environment resembles that of a complete graph Markov chain.

Q and the stream selection policy π using Model-based Least Squares Policy Iteration (Model LSPI; Lagoudakis and Parr [2003]) RL algorithm. Next, we discuss how the algorithm with the updating abstraction and stream selection policy converges to the optimal solution discussed in Section 2.

4 Dynamical Analysis

In this section, we present a formal analysis of the dynamics of the Curious Dr. MISFA algorithm. The full dynamics of Curious Dr. MISFA is fairly complicated due to the use of the heuristic ϵ-greedy strategy that balances exploration and exploitation. We instead focus here only on two cases: (a) pure exploration ($\epsilon = 1$) and (b) pure exploitation ($\epsilon = 0$). We use a slowly decaying ϵ to smoothly transition from exploration to exploitation. Next, we show in Theorem 1 that for a given arbitrary deterministic abstraction estimator that ensures convergence to a fixed point, there exists a curiosity function Ω that indicates the speed of learning an abstraction.

Theorem 1. *Given an abstraction-estimator Θ that ensures convergence to a fixed-point, there exists a curiosity function $\Omega : \mathcal{X} \rightarrow [0, 1)$ corresponding to Θ that induces a total ordering on $\mathcal{X}$.*

Proof. Since Θ ensures convergence on a temporally coherent stationary input stream $\mathbf{x} \in \mathcal{X}$, there exists a minimal time $T_{\mathbf{x}} \in \mathbb{R}^+$ s.t. for

$$t > T_{\mathbf{x}}, \quad |\widehat{\phi}_t - \phi^*| < \delta, \tag{16}$$

where ϕ^* represents a fixed-point ($\in \mathbb{R}^{I \times J}$) and δ is a small non-negative scalar constant. $T_{\mathbf{x}}$ is called the convergence time for the stream $\mathbf{x}$. For non-stationary streams in $\mathcal{X}$, there is no fixed-point and therefore the above condition does not hold ($T_{\mathbf{x}} = \infty$). Let T denote the set of convergence times of all the streams $\mathbf{x} \in \mathcal{X}$. Therefore, there exists a function $\mathcal{T} : \mathcal{X} \rightarrow T$, s.t. $\mathcal{T}(\mathbf{x})$ denotes the convergence time of the input $\mathbf{x}$. It is straightforward to show that since T is a totally-ordered set, $\mathcal{T}$ induces a total ordering in $\mathcal{X}$. One can easily find an order-preserving transfer function $f : \mathcal{T} \rightarrow [0, 1)$, for example $1 - e^{-T}$, such that the composite function $\Omega = f \circ T$ induces a total ordering in $\mathcal{X}$. □

Theorem 1 ensures that the objective of the optimization problem discussed in Section 2 is well defined. Curious Dr. MISFA estimates the unknown Ω through curiosity rewards that are proportional to the learning progress of IncSFA. Next, we define the curiosity function of IncSFA and briefly discuss its dynamics.

4.1 Curiosity function of IncSFA

Here, we discuss briefly the dynamics of IncSFA as a function of Ω for a given temporally coherent observation stream $\mathbf{x}(t) \in \mathbb{R}^I$. To keep it simple, we assume that the CCIPCA (see Section 3.1) has converged, that is, the output of CCIPCA $\mathbf{z}_i(t) \in \mathbb{R}^I$ has unit variance (*whitened* output). We consider here only the first output component of IncSFA, but this analysis can trivially be extended for higher output components using *sequential-addition* technique [Kompella et al., 2012a].

Since $\mathbf{x}(t)$ is a temporally coherent stream, the correlation matrix $E[\dot{\mathbf{z}}_i \dot{\mathbf{z}}_i^T]$ 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 with corresponding eigenvalues ($\lambda_1 < \lambda_2 < ... < \lambda_I$). The eigenvectors $\{v_i | i = 1, 2, ..., I\}$ form an orthonormal basis spanning $\mathcal{R}^I$. The IncSFA weight vector $\widehat{\phi}^{\text{sfa}}(t)$ can then be represented as

$$\widehat{\phi}^{\text{sfa}}(t) = \sum_{i=1}^{I} a_i(t) v_i, \tag{17}$$

where $a_i(t)$ are non-negative constant coefficients. In our previous work [Luciw et al., 2013] we have shown that

$$a_i(t) = C_i a_I(t) \omega_i^t, a_I(t) = \frac{1}{\sqrt{1 + \sum_{j=1}^{I-1} C_j^2 \omega_j^{2t}}}, \omega_i = \left(1 - \frac{\eta^{\text{sfa}}(\lambda_i - \lambda_I)}{1 - \eta^{\text{sfa}} - \eta^{\text{sfa}} \lambda_I}\right), \tag{18}$$

$$\lim_{t \to \infty} a_{i \neq I} = 0, \lim_{t \to \infty} a_I = 1, \tag{19}$$

where $C_i = \frac{a_i(0)}{a_I(0)}$ and $0 < \omega_1 < ... < \omega_{I-1} < 1$. For the expected behavior over several random initializations, $\mathbb{E}[a_i(0)]$'s can be assumed to be the same, therefore $\mathbb{E}[C_i] = \mathbb{E}[a_i] / \mathbb{E}[a_I] = 1, \forall i$. It follows that if at time $t = T_x$, $a_{I-1} \leq \delta / I$, then $a_{i<I-1} < \delta / I$, $(1 - a_I) < \delta / I$ and $|\widehat{\phi}^{\text{sfa}}(t) - v_I| < \delta$ where δ is a small non-negative scalar constant. Therefore, on an average over random initializations, streams with higher ω_{I-1} will have higher convergence time. The curiosity function of IncSFA is defined as follows [Luciw et al., 2013]:

Definition 1. *The curiosity function of the IncSFA algorithm to extract the slowest feature from an observation stream $\mathbf{x}_i \in X$ is defined as*

$$\Omega(\mathbf{x}_i) = \omega_{I-1} = \left[1 - \frac{\eta^{sfa}(\lambda_{I-1} - \lambda_I)}{1 - \eta^{sfa} - \eta^{sfa} \lambda_I}\right], \tag{20}$$

where $\lambda_I \neq \lambda_{I-1}$ denote the smallest two eigenvalues of $E[\dot{\mathbf{z}}_i \dot{\mathbf{z}}_i^T]$, and $\mathbf{z}_i(t) \in \mathbb{R}^I$ is the whitened *output of $\mathbf{x}_i(t)$.*

Streams that have higher Ω values (close to 1) are more difficult to encode by IncSFA. From the above definition, it is also clear that streams with smaller λ_I (for example, streams that change slowly in time) or with very close λ_I and $\lambda_I - 1$ (for example, white noise) are difficult (or impossible) to encode by the IncSFA.

The curiosity function values of the observation streams are not known to the Curious Dr. MISFA agent. It estimates the values through curiosity rewards proportional to

the learning progress of IncSFA:

$$\xi^{\text{sfa}}(t) = \|d(\widehat{\phi}^{\text{sfa}}(t))/dt\| = \sqrt{\sum_{i=1}^{I} \dot{a}_i^2(t)}, \quad r^{\text{sfa}}(t) = -\dot{\xi}^{\text{sfa}}(t) \tag{21}$$

where $\dot{a}_i(t)$ can be found by differentiating $a_i(t)$ (Eq. (18)) w.r.t t and solving:

$$\dot{a}_i(t) = \frac{\omega_i^t}{\sqrt{1 + \sum_{j=1}^{I-1} \omega_j^{2t}}} \left[\ln(\omega_i) - \frac{\sum_{j=1}^{I-1} \omega_j^{2t} \ln(\omega_j)}{(1 + \sum_{j=1}^{I-1} \omega_j^{2t})} \right], \tag{22}$$

$$= \frac{\omega_i^t}{(1 + \sum_{j=1}^{I-1} \omega_j^{2t})^{\frac{3}{2}}} \left[\ln(\omega_i) + \sum_{j=1}^{I-1} \omega_j^{2t} \ln(\omega_i/\omega_j) \right]. \tag{23}$$

The range of ω is calculated as follows. Rearranging Eq. (18) we have,

$$\omega_i = \left(\frac{1 - \eta^{\text{sfa}} - \eta^{\text{sfa}} \lambda_i}{1 - \eta^{\text{sfa}} - \eta^{\text{sfa}} \lambda_I} \right), \tag{24}$$

where λ_i's also represent the *slowness measure* (Δ values) of the input components [Wiskott and Sejnowski, 2002]. Slowness measure indicates how slow a normalized signal (with unit variance) changes over time. For example, the Δ value of normalized white noise is 2. Signals that are encodable by IncSFA have $\Delta < 2$. Therefore,

$$\omega_{\text{min}} = \frac{1 - 3\eta^{\text{sfa}}}{1 - \eta^{\text{sfa}}} < \omega_1 < \omega < \Omega. \tag{25}$$

If ω's are uniformly distributed between ω_1 and Ω, due to the exponential distribution of $\dot{a}(t)$ w.r.t ω, the L^2 norm is close to the higher values of ω with the limit case of Ω when $b \to \infty$. To keep the analysis simple, we study the dynamics at the limit case $\omega = \Omega$. Additionaly, by selecting a small learning rate η^{sfa}, ω_{min} and ω_i's become close to Ω. Substituting $\omega_j = \omega_i = \Omega, \ \forall i, j \in [1, I]$ and $b = I - 1$ in Eq. (23), we get

$$\xi^{\text{sfa}}(t) \approx \frac{\Omega^t \ln(\Omega)}{\sqrt{1 + b\Omega^{2t}}} \sqrt{I}, \tag{26}$$

$$r^{\text{sfa}}(t) = -\dot{\xi}^{\text{sfa}}(t) = \frac{\Omega^t \ln^2(\Omega)}{(1 + b\Omega^{2t})^{\frac{5}{2}}} (1 - 2b\Omega^{2t}) \sqrt{I}. \tag{27}$$

Next, we find the relationship between the curiosity rewards for streams with different curiosity function values. The following lemma is useful for the analysis.

Lemma 1. *Let* $g(\omega, t) = \frac{\omega^t \ln^2(\omega)}{(1 + b\omega^{2t})^{\frac{5}{2}}} (1 - 2b\omega^{2t})$, $0 < \omega < 1$ *and* $b > 1$, *then*

1. $g(\omega, t)$ *decreases monotonously w.r.t.* ω, *if* $\frac{\ln(2b)}{2} < -t \ln(\omega) < \frac{\ln(\frac{4b}{5-\sqrt{21}})}{2}$.

2. $\operatorname{sgn}\left(g(\omega,t)\right) = \operatorname{sgn}\left(t + \dfrac{\ln(2b)}{2\ln(\omega)}\right)$.

Proof. 1. To prove the result, we show that $\frac{\partial g(\omega,t)}{\partial \omega} < 0$ when $\frac{\ln(2b)}{2} < -t\ln(\omega) < \frac{\ln(\frac{4b}{5-\sqrt{21}})}{2}$.

$$\begin{aligned}\frac{\partial g(\omega,t)}{\partial \omega} &= \frac{t\omega^{t-1}\ln^2(\omega)\left(1-2b\omega^{2t}\right)}{\left(1+b\omega^{2t}\right)^{\frac{5}{2}}} + \frac{2\omega^{t-1}\ln(\omega)\left(1-2b\omega^{2t}\right)}{\left(1+b\omega^{2t}\right)^{\frac{5}{2}}} \\ &\quad - \frac{4bt\omega^{3t-1}\ln^2(\omega)}{\left(1+b\omega^{2t}\right)^{\frac{5}{2}}} - \frac{5bt\omega^{3t-1}\ln^2(\omega)\left(1-2b\omega^{2t}\right)}{\left(1+b\omega^{2t}\right)^{\frac{7}{2}}} \\ &= \frac{\omega^{t-1}\ln(\omega)}{\left(1+b\omega^{2t}\right)^{\frac{7}{2}}}\left[\left(4b^2\omega^{4t} - 10b\omega^{2t} + 1\right)t\ln(\omega) - 4b^2\omega^{4t} - 2b\omega^{2t} + 2\right].\end{aligned}$$

Since $\ln(\omega) < 0$, $\frac{\partial g(\omega,t)}{\partial \omega} < 0$ if $\left[\left(4b^2\omega^{4t} - 10b\omega^{2t} + 1\right)t\ln(\omega) - 4b^2\omega^{4t} - 2b\omega^{2t} + 2\right] > 0$. Let $2b\omega^{2t} = x$. Substituting in the above equation, we have $\frac{\partial g(\omega,t)}{\partial \omega} < 0$

$$\begin{aligned}&\text{if,} \quad \left(x^2 - 5x + 1\right)t\ln(\omega) - x^2 - x + 2 > 0, \\ &\text{if,} \quad \left(x - \frac{5-\sqrt{21}}{2}\right)\left(x - \frac{5+\sqrt{21}}{2}\right)t\ln(\omega) + \left(x+2\right)\left(1-x\right) > 0, \\ &\text{if,} \quad \frac{5-\sqrt{21}}{2} < x < 1, \\ &\text{i.e. if,} \quad \frac{\ln(2b)}{2} < -t\ln(\omega) < \frac{\ln(\frac{4b}{5-\sqrt{21}})}{2} \approx \frac{\ln(9.58b)}{2}.\end{aligned}$$

2. The proof is straightforward. $g(\omega,t) > 0$ iff $(1-2b\omega^{2t}) > 0$, that is when $t > -\frac{\ln(2b)}{2\ln(\omega)}$ and vice versa. □

Lemma 2. *Let* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *be two streams s.t.* $\Omega_1 < \Omega_2$, $b > 1$ *and* $-\dfrac{\ln(2b)}{2\ln(\Omega_1)} < t < -\dfrac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_1)}$, *then* $r_1^{sfa}(t) > r_2^{sfa}(t)$.

Proof. For the cases when $-\frac{\ln(2b)}{2\ln(\Omega_1)} < t < -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_1)} < -\frac{\ln(2b)}{2\ln(\Omega_2)}$ and $-\frac{\ln(2b)}{2\ln(\Omega_1)} < t < -\frac{\ln(2b)}{2\ln(\Omega_2)} < -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_1)}$, from Lemma 1.2, $r_1^{\text{sfa}}(t) > 0 > r_2^{\text{sfa}}(t)$. Next, we consider the remaining case: $-\frac{\ln(2b)}{2\ln(\Omega_2)} < t < -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_1)}$. Since $\Omega_1 < \Omega_2$, $\frac{\ln(2b)}{2} < -t\ln(\Omega_2) < -t\ln(\Omega_1) < -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2}$. Therefore from Lemma 1.1, we have $r_1^{\text{sfa}}(t) > r_2^{\text{sfa}}(t)$. □

Curious Dr. MISFA uses the accumulation of the curiosity rewards over time to take optimal actions. Lemma 3 shows that the accumulated curiosity rewards generated through IncSFA are negative up to an initial time period.

Lemma 3. *Let* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *be two streams and* $\tau = -\dfrac{\ln(cb)}{2\ln(\Omega_1)}$

$$\begin{aligned} &\textit{s.t.} \qquad \Omega_1 < \Omega_2, \quad 2 < c < \frac{4}{5-\sqrt{21}} \approx 9.58, \quad b \geq 2 \\ &\textit{then,} \qquad \int_0^t r_1^{sfa}(t)dt < 0, \int_0^t r_2^{sfa}(t)dt < 0, \quad t \leq \tau. \end{aligned}$$

Proof. From Lemma 1.2, for $t < -\frac{ln(2b)}{2\ln(\Omega_1)}$, the result is straightforward. We consider the case for $t = \tau > -\frac{ln(2b)}{2\ln(\Omega_1)}$. Integrating $r^{\text{sfa}}(t)$ over time

$$\int_0^\tau r^{\text{sfa}}(t)dt = \frac{\Omega^t \ln(\Omega)\sqrt{I}}{(1+b\Omega^{2t})^{\frac{3}{2}}}\Big|_0^\tau, \tag{28}$$

$$= \left(\frac{1}{(1+b)^{\frac{3}{2}}} - \frac{\Omega^\tau}{(1+b\Omega^{2\tau})^{\frac{3}{2}}}\right)(-\ln(\Omega)\sqrt{I}). \tag{29}$$

Since $\tau = -\frac{\ln(cb)}{2\ln(\Omega_1)}$, $\Omega^{2\tau} = \frac{1}{cb}$. Substituting above we get

$$\int_0^\tau r^{\text{sfa}}(t)dt = \left(\frac{1}{(1+b)^{\frac{3}{2}}} - \frac{1}{\frac{\sqrt{b}}{c}(1+c)^{\frac{3}{2}}}\right)(-\ln(\Omega)\sqrt{I}). \tag{30}$$

Since $b \geq 2$, $(1+b)^{\frac{3}{2}}/\sqrt{b} > 3.67$ and $c < \frac{4}{5-\sqrt{21}}$, $(1+c)^{\frac{3}{2}}/c < 3.60$. Therefore,

$$\frac{(1+b)^{\frac{3}{2}}}{\sqrt{b}} > \frac{(1+c)^{\frac{3}{2}}}{c} \implies \frac{1}{(1+b)^{\frac{3}{2}}} < \frac{1}{\frac{\sqrt{b}}{c}(1+c)^{\frac{3}{2}}} \implies \int_0^\tau r^{\text{sfa}}(t)dt < 0. \tag{31}$$

Hence the result. □

Next, we find the relationship between the accumulated curiosity rewards between two streams.

Lemma 4. *Let* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *be two streams,* $b \geq 2$ *and* $\tau = -\dfrac{\ln(cb)}{2\ln(\Omega_1)}$

$$\begin{aligned} &\textit{s.t.} \qquad \Omega_1 < \Omega_2, \quad 2 \leq c < 6.75\left(\frac{\ln(\Omega_1)}{\ln(\Omega_2)}\right)^2 - 4.75, \\ &\textit{then,} \qquad \int_\tau^t r_1^{sfa}(t)dt > \int_\tau^t r_2^{sfa}(t)dt, \quad t > \tau. \end{aligned}$$

Proof. Case $\tau < t < -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_1)}$: The result is straightforward from Lemma 2.

Case $-\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_1)} < t < -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_2)}$: From Lemma 1.2 we have $r_1^{\text{sfa}}(t) > 0$, therefore $\int_\tau^t r_1^{\text{sfa}}(t)dt > 0$. From Lemma 3, we have $\int_\tau^t r_2^{\text{sfa}}(t)dt < 0$. The result follows.

Case $-\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_2)} < t$: We have $\int_\tau^t r_2^{\text{sfa}}(t)dt = \int_\tau^{-\frac{\ln(2b)}{2\ln(\Omega_2)}} r_2^{\text{sfa}}(t)dt + \int_{-\frac{\ln(2b)}{2\ln(\Omega_2)}}^t r_2^{\text{sfa}}(t)dt$. Since from Lemma 1.2, $\int_\tau^{-\frac{\ln(2b)}{2\ln(\Omega_2)}} r_2^{\text{sfa}}(t)dt < 0$, the result holds if

$$\int_\tau^t r_1^{\text{sfa}}(t)dt > \int_{-\frac{\ln(2b)}{2\ln(\Omega_2)}}^t r_2^{\text{sfa}}(t)dt \tag{32}$$

Integrating $\int_\tau^t r^{\text{sfa}}(t)dt$ and solving we get

$$\int_\tau^t r^{\text{sfa}}(t)dt = \frac{\Omega^t \ln(\Omega)\sqrt{I}}{(1+b\Omega^{2t})^{\frac{3}{2}}}\Big|_\tau^t = -\frac{\Omega^\tau \ln(\Omega)\sqrt{I}}{(1+b\Omega^{2\tau})^{\frac{3}{2}}} + \frac{\Omega^t \ln(\Omega)\sqrt{I}}{(1+b\Omega^{2t})^{\frac{3}{2}}}. \tag{33}$$

Let $f(\Omega,t) = -\frac{\Omega^t \ln(\Omega)\sqrt{I}}{(1+b\Omega^{2t})^{\frac{3}{2}}}$. First, we show that $f(\Omega_1,\tau) > f(\Omega_2, -\frac{\ln(2b)}{2\ln(\Omega_2)})$. Substituting for $\tau = -\frac{\ln(cb)}{2\ln(\Omega_2)}$ and solving we get,

$$f(\Omega_1,\tau) > f\left(\Omega_2, -\frac{\ln(2b)}{2\ln(\Omega_2)}\right) \tag{34}$$

$$\text{if}, \quad \frac{-\ln(\Omega_1)}{\frac{\sqrt{(b)}}{c}(1+c)^{\frac{3}{2}}} > \frac{-\ln(\Omega_2)}{\frac{\sqrt{(b)}}{2}(3)^{\frac{3}{2}}}, \tag{35}$$

$$\text{if}, \quad \frac{(1+c)^3}{c^2} < 6.75\left(\frac{\ln(\Omega_1)}{\ln(\Omega_2)}\right)^2, \tag{36}$$

$$\text{if}, \quad c+3+\frac{1+3c}{c^2} < c+4.75 < 6.75\left(\frac{\ln(\Omega_1)}{\ln(\Omega_2)}\right)^2 \qquad \because c>2, \tag{37}$$

which is the given condition. Next, we show that $f(\Omega_1,t) < f(\Omega_2,t)$. Differentiating $f(\Omega,t)$ w.r.t Ω, we get

$$\frac{\partial f(\Omega,t)}{\partial \Omega} = \frac{\Omega^{t-1}\left((2b\Omega^{2t}-1)t\ln(\Omega) - b\Omega^{2t} - 1\right)}{(1+b\Omega^{2t})^{\frac{5}{2}}}\sqrt{I}. \tag{38}$$

$$\text{sgn}\left(\frac{\partial f(\Omega,t)}{\partial \Omega}\right) = \text{sgn}\left((2b\Omega^{2t}-1)t\ln(\Omega) - b\Omega^{2t} - 1\right). \tag{39}$$

We check the condition when $f(\Omega,t)$ increases monotonously w.r.t Ω. Substituting $-t\ln(\Omega) = \frac{\ln(xb)}{2}$ and solving we get $\frac{\partial f(\Omega,t)}{\partial \Omega} > 0$,

$$\text{if}, \quad -0.5(\frac{2}{x}-1)\ln(xb) - \frac{1}{x} - 1 > 0, \tag{40}$$

$$\text{if}, \quad \frac{1}{x}\mathrm{e}^{\frac{2(1+x)}{(x-2)}} < b. \tag{41}$$

As x increases $\frac{1}{x}\mathrm{e}^{\frac{2(1+x)}{(x-2)}}$ decreases. Substituting for $x > \ln(\frac{4b}{5-\sqrt{21}})$, we get $\frac{1}{x}\mathrm{e}^{\frac{2(1+x)}{(x-2)}} < 1.71 < b$. Therefore at time t, $f(\Omega,t)$ increases monotonously w.r.t Ω's that satisfy $-\ln(\Omega) > \frac{\ln(\frac{4b}{5-\sqrt{21}})}{2t}$. Since $t > -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_2)} > -\frac{\ln(\frac{4b}{5-\sqrt{21}})}{2\ln(\Omega_1)}$ we have, $f(\Omega_1,t) < f(\Omega_2,t) \implies f(\Omega_1,\tau) - f(\Omega_1,t) > f(\Omega_2,\tau) - f(\Omega_2,t)$. Hence the result. □

Lemma 4 shows that after an initial time τ, the accumulated curiosity rewards corresponding to the easier-to-learn streams are greater than the difficult ones. The range of values of τ is more for streams with distant curiosity function values. Next, we show that the result of Lemma 4 follows when a similar relationship between certain exponential decay functions is held.

Lemma 5. *Let* $\widetilde{\xi}^{sfa}(t) = -\Omega^{qt}\ln(\Omega^q)\sqrt{I},\ q = \frac{4}{\ln(cb)}$. *Let* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *be two streams s.t.* $\int_\tau^t -\dot{\widetilde{\xi}}_1^{sfa}(t)dt > \int_\tau^t -\dot{\widetilde{\xi}}_2^{sfa}(t)dt,\ 2 \le c < 6.75\left(\frac{\ln(\Omega_1)}{\ln(\Omega_2)}\right)^2 - 4.75,,\ b \ge 2,\ \tau = -\frac{\ln(cb)}{2\ln(\Omega_1)}$,

$$\textit{then,} \qquad \int_\tau^t r_1^{sfa}(t)dt > \int_\tau^t r_2^{sfa}(t)dt, \quad t > \tau.$$

Proof. Let $\widetilde{r}^{\text{sfa}}$ denote $-\dot{\widetilde{\xi}}^{\text{sfa}}$. We find the condition $\frac{\partial \widetilde{r}^{\text{sfa}}(t)}{\partial \Omega} < 0$.

$$\frac{\partial \widetilde{r}^{\text{sfa}}(t)}{\partial \Omega} = q^2\Omega^{qt-1}\sqrt{I}\ln(\Omega)\big[qt\ln(\Omega) + 2\big]. \tag{42}$$

$$\frac{\partial \widetilde{r}^{\text{sfa}}(t)}{\partial \Omega} < 0, \quad \text{if } -t\ln(\Omega) < \frac{2}{q} = \frac{\ln(cb)}{2}. \tag{43}$$

Therefore, if $\widetilde{r}_1^{\text{sfa}}(t) > \widetilde{r}_2^{\text{sfa}}(t)$, then $\Omega_1 < \Omega_2$ for the values of $c < \frac{4}{5-\sqrt{21}}$. This implies Lemma 2 holds true. Following the proof similar to Lemma (4), we get if $\int_\tau^t \widetilde{r}_1^{\text{sfa}}(t)dt > \int_\tau^t \widetilde{r}_2^{\text{sfa}}(t)dt$ then $\Omega_1 < \Omega_2$, which implies $\int_\tau^t r_1^{\text{sfa}}(t)dt > \int_\tau^t r_2^{\text{sfa}}(t)dt$ from the result of Lemma (4) under the given conditions. □

Curious Dr. MISFA uses backward difference approximation to compute the curiosity rewards from $\xi^{\text{sfa}}(t)$. Lemma 5 gives a simpler expression to use for $\xi^{\text{sfa}}(t)$ (Eq. (26)) that preserves the relation between the accumulated curiosity rewards and the curiosity function values. Next, we discuss how Curious Dr. MISFA uses the accumulated curiosity rewards to find the stream with the least Ω.

4.2 Curious Dr. MISFA Dynamics

We present here the average dynamics of the Curious Dr. MISFA agent to extract the next easiest yet unknown abstraction. The outline of the analysis is as follows. First, we discuss the conditions that are assumed to be satisfied for the analysis. Then, we find the *optimal fixed points* for the adaptive abstraction $\widehat{\phi}$ and the observation stream selection policy π to solve the optimization problem (Theorems 2 & 3). In Theorems 4 & 5 we show that the RL framework within Curious Dr. MISFA that is detached from Ω,

converges to this optimal solution. The following definition is useful for the remaining of the section.

Definition 2. *At time t, let $\mathbf{x}_l$ denote the current easiest but not yet learned observation stream and s_l denote the corresponding state. Then, the index l is given by*

$$l = \underset{\forall i:\ \mathbf{x}_i \in X'}{\arg\min}\, \Omega(\mathbf{x}_i), \quad X' = X \setminus X^{\Phi_t}. \tag{44}$$

$X^{\Phi_t} \subseteq X$ denotes the set of encoded observation streams at time t (see Section 2) and $X \setminus X^{\Phi_t}$ denotes the set-theoretic difference equal to $\{\mathbf{x}_i \in X | \mathbf{x}_i \notin X^{\Phi_t}\}$. Next, we discuss the assumed conditions for the analysis.

Assumed Conditions: Curious Dr. MISFA has several online updating components. To guarantee optimal performance, several conditions are required to be held. We discuss here the set of necessary conditions and also the assumptions made to simplify the analysis. Later in the section, we discuss intuitively how the algorithm functions when a few of these assumptions are not held.

1. **Number of streams and abstraction dimensionality.** $n > 2$, $J = 1$. These assumptions imply that there are more than two observation streams and the slow-feature matrix $\widehat{\phi}^{\text{sfa}}$ is a column vector.

2. **Orthogonal fixed points.** Streams $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ are each encodable encodable by IncSFA and have orthogonal fixed points. That is, if $\{\phi_1^{\text{sfa}}, \phi_2^{\text{sfa}}, ..., \phi_n^{\text{sfa}}\}$ are abstractions learned by IncSFA for each observation stream, then $\phi_i^{\text{sfa}} \cdot \phi_j^{\text{sfa}} = 0$. This ensures that when IncSFA is making progress learning ϕ_i^{sfa} then, it does not make any learning progress towards other $\phi_{j \neq i}^{\text{sfa}}$.

3. **IncSFA convergence conditions.** Let $t' \in \mathbb{N}$ denote the time whenever a new adaptive abstraction $\widehat{\phi}$ is instantiated,

$$\eta^{\text{sfa}} \max(\lambda_1^1, ..., \lambda_1^n) < 0.5, \quad 0 < \eta^{\text{sfa}} \leq 0.5, \quad \|\widehat{\phi}^{\text{sfa}}(t')\|^2 = 1. \tag{45}$$

4. τ **condition.** Condition (46) determines the range of values for τ, $\forall \mathbf{x}_i \in X \setminus X^{\Phi_t}$,

$$\tau = -\frac{\ln(cb)}{2\ln(\Omega(\mathbf{x}_l))}, \quad 2 \leq c < 6.75 \left(\frac{\ln(\Omega(\mathbf{x}_l))}{\ln(\Omega(\mathbf{x}_i))}\right)^2 - 4.75. \tag{46}$$

5. **Curiosity function values.** Condition (47) determines how far apart the curiosity function values should be in order to be distinguished as two distinct streams.

$$\frac{-ln(\Omega(\mathbf{x}_i))}{-ln(\Omega(\mathbf{x}_l))} < \frac{1}{(1+(n-1)(1-\gamma)/\gamma)}, \quad \forall \mathbf{x}_i \in X \setminus X^{\Phi_t}. \tag{47}$$

γ is a constant discount-factor ($0 < \gamma < 1$) for RL.

6. **Other.** The reward function (Eq. 14) R generated by the algorithm has a Frobenius norm equal to one. It is assumed that R takes only non-negative values. This assumption is trivial since, a scalar positive constant can be added to R without having any effect on the policies learned by the Least Squares Policy Iteration (LSPI) reinforcement learning algorithm.

Let $\mathcal{X} = \{\mathbf{x} : \mathbf{x}(t) \in \mathbb{R}^I, I \in \mathbb{N}\}$ denote a set of of I-dimensional observation streams. Therefore, $X \subset \mathcal{X}$. Let Φ^* denote the space of all learnable abstractions by Θ for the input X satisfying Constraints (3)-(4) (see Section 2). Based on Ω, *optimal fixed-points* for the adaptive abstraction $\widehat{\phi}$ and the observation stream selection policy π are defined next.

Theorem 2. *At time t, the optimal fixed-point $\phi^* \in \Phi^*$ of the adaptive abstraction $\widehat{\phi}$ is equal to the J slow features of the observation stream $\mathbf{x}_l$ and the corresponding ROC clusters.*

Proof. The proof is straightforward for $J \geq 1$, however, for the rest of the analysis J is assumed to be 1. □

Theorem 3. *The optimal observation stream selection policy ($\pi^* : \Phi^* \times \mathcal{S} \rightarrow \mathcal{A}, \mathcal{A} = \{0\ (stay), 1\ (switch)\}$) to learn an abstraction $\phi_i \in \Phi^*$ is given by:*

$$\pi^*(\phi_i, s) = 1 - \mathbb{1}_{\{s_l\}}(s),\ \forall s \in \mathcal{S},$$

where $\mathbb{1}_{\{s_l\}}(s)$ is the Kronecker delta function: $\mathbb{1}_{\{s_l\}}(s = s_l) = 1, \mathbb{1}_{\{s_l\}}(s \neq s_l) = 0$.

Proof. The proof is straightforward and follows from Theorem 2. □

Theorem 3 determines the policy that optimizes the objective discussed in Section 2. It is such that the agent takes the action *stay* $(= 0)$ in the state s_l, which corresponds to the current easiest but not yet encoded observation stream $\mathbf{x}_l$, and takes the action

switch $(= 1)$ in the rest of the states. Next, we show that the RL's policy within Curious Dr. MISFA algorithm converges towards π^* based on the reward function defined in Eq. (14). The following lemmas are useful to show the result.

Lemma 6. *Let R denote the estimated reward function by the algorithm at any time t. Let π be any arbitrary deterministic observation stream selection policy and let k_0 and k_1 denote sets of states where the policy returns a zero (*stay*) and one (*switch*) respectively:*

$$k_0 = \{s \mid \pi(s) = 0, \forall s \in \mathcal{S}\}$$
$$k_1 = \{s \mid \pi(s) = 1, \forall s \in \mathcal{S}\}.$$

Then, the action values corresponding to each (s, a) tuple for the policy π are given by:

$$(a)\ Q^{\textit{stay}}_{s\in k_0} = \frac{R^{\textit{stay}}_{ss}}{1-\gamma} \tag{48}$$

$$(b)\ Q^{\textit{switch}}_{s\in k_1} = \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\textit{switch}}_{ss'} + \widehat{R}^{\textit{switch}}\right] + \widehat{R}^{\textit{stay}} \tag{49}$$

$$(c)\ Q^{\textit{switch}}_{s\in k_0} = \frac{1}{(n-1)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\textit{switch}}_{ss'} + \widehat{R}^{\textit{switch}} + (n-1+\gamma)\widehat{R}^{\textit{stay}} - \frac{\gamma R^{\textit{stay}}_{ss}}{1-\gamma}\right] \tag{50}$$

$$(d)\ Q^{\textit{stay}}_{s\in k_1} = R^{\textit{stay}}_{ss} + \frac{\gamma}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\textit{switch}}_{ss'} + \widehat{R}^{\textit{switch}}\right] + \widehat{R}^{\textit{stay}} \tag{51}$$

$$\textit{where,}\qquad \widehat{R}^{\textit{switch}} = \frac{\gamma}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\textit{switch}}_{s's''} \tag{52}$$

$$\widehat{R}^{\textit{stay}} = \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_0} R^{\textit{stay}}_{s''s''} \tag{53}$$

Proof. The value of a (s, a) tuple is the expected cumulative future reward that the agent can accumulate starting by executing the action a in the state s.

(a) $Q^{\text{stay}}_{s\in k_0} = \sum_{t=0}^{\infty}\gamma^t R^{\text{stay}}_{ss} P^{\text{stay}}_{ss} = \frac{R^{\text{stay}}_{ss}}{1-\gamma}$

(b) $Q^{\text{switch}}_{s\in k_1} = \sum_{s'\in k_0}\left[R^{\text{switch}}_{ss'} + \gamma Q^{\text{stay}}_{s'}\right] P^{\text{switch}}_{ss'} + \sum_{s'\in k_1\setminus s}\left[R^{\text{switch}}_{ss'} + \gamma Q^{\text{switch}}_{s'}\right] P^{\text{switch}}_{ss'}$

Substituting $P^{\text{switch}}_{ss'} = 1/(n-1)$ (see Section 3.2), $k_0 \cup k_1 = \mathcal{S}$ and the result from (a),

we get

$$= \frac{1}{n-1}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{1-\gamma}\sum_{s'\in k_0} R^{\text{stay}}_{s's'} + \gamma\sum_{s'\in k_1\setminus s} Q^{\text{switch}}_{s'}\right] \tag{54}$$

Taking a summation of Q^{switch}_{s} over all $s \in k_1$ and solving, we get,

$$\sum_{s''\in k_1} Q^{\text{switch}}_{s''} = \frac{1}{n-1}\left[\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''} + \frac{|k_1|\,\gamma}{1-\gamma}\sum_{s'\in k_0} R^{\text{stay}}_{s's'} + \gamma(|k_1|-1)\sum_{s'\in k_1} Q^{\text{switch}}_{s'}\right]$$

$$= \frac{\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''} + \frac{|k_1|\gamma}{1-\gamma}\sum_{s'\in k_0} R^{\text{stay}}_{s's'}}{(n-1-\gamma(|k_1|-1))} \tag{55}$$

Substituting Eq. (55) in Eq. (54) and solving we get,

$$Q^{\text{switch}}_{s\in k_1} = \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \widehat{R}^{\text{switch}}\right] + \widehat{R}^{\text{stay}}$$

(c) $$Q^{\text{switch}}_{s\in k_0} = \sum_{s'\in k_0\setminus s}\left[R^{\text{switch}}_{ss'} + \gamma Q^{\text{stay}}_{s'}\right]P^{\text{switch}}_{ss'} + \sum_{s'\in k_1}\left[R^{\text{switch}}_{ss'} + \gamma Q^{\text{switch}}_{s'}\right]P^{\text{switch}}_{ss'}$$

$$= \frac{1}{n-1}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{1-\gamma}\sum_{s'\in k_0\setminus s} R^{\text{stay}}_{s's'} + \gamma\sum_{s'\in k_1} Q^{\text{switch}}_{s'}\right] \tag{56}$$

Substituting Eq. (55) in Eq. (56) and solving we get,

$$Q^{\text{switch}}_{s\in k_0} = \frac{1}{(n-1)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \widehat{R}^{\text{switch}} + (n-1+\gamma)\widehat{R}^{\text{stay}} - \frac{\gamma}{1-\gamma}R^{\text{stay}}_{ss}\right]$$

(d) $$Q^{\text{stay}}_{s\in k_1} = R^{\text{stay}}_{ss} + \gamma Q^{\text{switch}}_{s\in k_1}$$

$$= R^{\text{stay}}_{ss} + \frac{\gamma}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \widehat{R}^{\text{switch}}\right] + \widehat{R}^{\text{stay}}$$

□

Lemma 7. *If* $R^{stay}_{s_l s_l} = \max(R)$ *and* $R^{stay}_{ss} < \dfrac{\gamma R^{stay}_{s_l s_l}}{(n-1)-\gamma(n-2)}$, $\forall s \in \mathcal{S}\setminus s_l$ *then,*

$$\arg\max_{\pi} Q^{\pi} = 1 - \mathbb{1}_{\{s_l\}}(s),\ \forall s \in \mathcal{S}$$

Proof. Let $\pi^{\text{opt}} = 1 - \mathbb{1}_{\{s_l\}}(s),\ \forall s \in \mathcal{S}$. The proof is straightforward if the following hold true:

1. (a) $Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_1} < Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_1}$ & (b) $Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_1}$
2. (a) $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_1} < Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_1}$ & (b) $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_1}$
3. (a) $Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_1} < Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_0}$ & (b) $Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_0}$
4. (a) $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_1} < Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_0}$ & (b) $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_0}$

Each of the above inequalities are proved in turn. We use the sets k_0 and k_1 that were defined in Lemma 6. For the policy π^{opt}, $k_0 = \{s_l\}$ and $k_1 = \mathcal{S} \setminus s_l$ ($(n-1)$ elements). Therefore, for any other policy $\pi \neq \pi^{\text{opt}}$, either $|k_1| = n$ or $|k_1| < (n-1)$. The result for $|k_1| = n$ (*switch* at all states) is straightforward to show. Here, the case $|k_1| < (n-1)$ is considered.

Proof for 1-(a): Using the condition $|k_1| < n-1$ in Eq. (49), we get,

$$Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_1} = \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_0} R^{\text{stay}}_{s''s''}$$

$$< \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))}\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_0} R^{\text{stay}}_{s''s''}$$

Using the condition $R^{\text{stay}}_{ss} < \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(n-1)-\gamma(n-2)},\ \forall s \in \mathcal{S}\setminus s_l$, we get

$$< \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))}\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\left[\frac{\gamma(|k_0|-1)}{(n-1)-\gamma(n-2)}+1\right] R^{\text{stay}}_{s_l s_l}$$

Substituting $|k_0| = n - |k_1|$ and solving, we get

$$= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))}\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma}{(1-\gamma)\,((n-1)-\gamma(n-2))} R^{\text{stay}}_{s_l s_l}$$

$$= Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_1}$$

Hence, $Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_1} < Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_1}$.

Proof for 1-(b): From Eq. (50) we have,

$$Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_0} = \frac{1}{(n-1)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma}{(1-\gamma)(n-1)}\left[\frac{(n-1+\gamma)}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_0} R^{\text{stay}}_{s''s''}\right] - \frac{\gamma}{(n-1)(1-\gamma)}R^{\text{stay}}_{ss}$$

$$= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{n-1}\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma(n-1+\gamma)\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1)\,(n-1-\gamma(|k_1|-1))}\right] + \frac{\gamma\left[(n-1+\gamma)\sum_{s''\in k_0} R^{\text{stay}}_{s''s''} - (n-1-\gamma(|k_1|-1))\,R^{\text{stay}}_{ss}\right]}{(1-\gamma)(n-1)\,(n-1-\gamma(|k_1|-1))}$$

Substituting the following in the first term of R.H.S.:

- $(n-1)-\gamma(n-2) < (n-1)$,
- since R has all non-negative entries $\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} < \sum_{s''\in k_0}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}$, and
- it can easily be shown that $\frac{n-1+\gamma}{n-1} < \frac{n-1-\gamma(|k_1|-1)}{n-1-\gamma(n-2)}$, we get,

$$Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma\sum_{s''\in k_0}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1)-\gamma(n-2)} + \frac{\gamma\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1-\gamma(n-2))}\right] + \frac{\gamma\left[(n-1+\gamma)\sum_{s''\in k_0} R^{\text{stay}}_{s''s''} - (n-1-\gamma(|k_1|-1))\,R^{\text{stay}}_{ss}\right]}{(1-\gamma)(n-1)\,(n-1-\gamma(|k_1|-1))}$$

$$= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s'} R^{\text{switch}}_{ss'} + \frac{\gamma\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1)-\gamma(n-2)}\right] + \frac{\gamma\left[(n-1+\gamma)\sum_{s''\in k_0} R^{\text{stay}}_{s''s''} - (n-1-\gamma(|k_1|-1))\,R^{\text{stay}}_{ss}\right]}{(1-\gamma)(n-1)\,(n-1-\gamma(|k_1|-1))}$$

Substituting $\sum_{s''\in k_0} R^{\text{stay}}_{s''s''} = \sum_{s''\in k_0\setminus s_l} R^{\text{stay}}_{s''s''} + R^{\text{stay}}_{s_l s_l}$ and the condition $R^{\text{stay}}_{ss} < \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(n-1)-\gamma(n-2)}$, $\forall s \in \mathcal{S}\setminus s_l$ in the second term of R.H.S., we get,

$$
\begin{aligned}
&< \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s'} R^{\text{switch}}_{ss'} + \frac{\gamma\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1)-\gamma(n-2)}\right] \\
&+ \frac{\gamma R^{\text{stay}}_{s_l s_l}\left[\frac{(n-1+\gamma)(|k_0|-1)\gamma}{(n-1)-\gamma(n-2)} + (n-1+\gamma) - \frac{(n-1-\gamma(|k_1|-1))\,\gamma}{(n-1)-\gamma(n-2)}\right]}{(1-\gamma)(n-1)\,(n-1-\gamma(|k_1|-1))} \\
&= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s'} R^{\text{switch}}_{ss'} + \frac{\gamma\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1)-\gamma(n-2)}\right] \\
&+ \frac{\gamma R^{\text{stay}}_{s_l s_l}\left[(n-1+\gamma)(n-|k_1|-1)\gamma + (n-1+\gamma)(n-1+2\gamma-n\gamma) - (n-1+\gamma-\gamma\,|k_1|)\gamma\right]}{(1-\gamma)(n-1)(n-1-\gamma(n-2))(n-1-\gamma(|k_1|-1))}
\end{aligned}
$$

Upon factoring we get,

$$
\begin{aligned}
&= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s'} R^{\text{switch}}_{ss'} + \frac{\gamma\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1)-\gamma(n-2)}\right] \\
&+ \frac{\gamma R^{\text{stay}}_{s_l s_l}\left[(n-1)(n-1-\gamma(|k_1|-1))\right]}{(1-\gamma)(n-1)(n-1-\gamma(n-2))(n-1-\gamma(|k_1|-1))} \\
&= Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_1}
\end{aligned}
$$

Hence, $Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < Q^{\text{switch},\pi^{\text{opt}}}_{s\in k_1}$.

Proof for 2-(a): $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_1} = R^{\text{stay}}_{ss} + \gamma Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s} < R^{\text{stay}}_{ss} + \gamma Q^{\text{switch},\pi^{\text{opt}}}_{s} = Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_1}$.
Hence, $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_1} < Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_1}$.

Proof for 2-(b): From Eq. (48) we have, $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_0} = \frac{R^{\text{stay}}_{ss}}{1-\gamma} = R^{\text{stay}}_{ss} + \frac{\gamma R^{\text{stay}}_{ss}}{1-\gamma}$. Using the condition $R^{\text{stay}}_{ss} < \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(n-1)-\gamma(n-2)}$, $\forall s \in \mathcal{S}\setminus s_l$, we get,

$$
Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < R^{\text{stay}}_{ss} + \gamma\left(\frac{\gamma R^{\text{stay}}_{s_l s_l}}{(1-\gamma)(n-1-\gamma(n-2))}\right) < R^{\text{stay}}_{ss} + \gamma Q^{\text{switch},\pi^{\text{opt}}}_{s} = Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_1}.
$$

Hence, $Q^{\text{stay},\pi\neq\pi^{\text{opt}}}_{s\in k_0} < Q^{\text{stay},\pi^{\text{opt}}}_{s\in k_1}$.

Proof for 3-(a): For the optimal policy π^{opt}, the set $k_0 = \{s_l\}$. Therefore, if $s \in k_0$, then $s = s_l$. Substituting this in the inequality 3-(a) that needs to proved, we get,

$$Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s_l\in k_1} < Q^{\text{switch},\pi^{\text{opt}}}_{s_l}. \tag{57}$$

For the policy π, since $s_l \in k_1$, this implies $s_l \notin k_0$. From Eq. (49) we have,

$$Q^{\text{switch},\pi\neq\pi^{\text{opt}}}_{s_l\in k_1} = \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_0} R^{\text{stay}}_{s''s''}$$

Substituting the following:

- $|k_1| < n-1$,
- since R has all non-negative entries $\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''} < \sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}$,
- using the condition $R^{\text{stay}}_{ss} < \frac{\gamma R^{\text{stay}}_{s_ls_l}}{(n-1)-\gamma(n-2)},\ \forall s \in \mathcal{S}\setminus s_l$, and
- since $s_l \notin k_0$, $\sum_{s''\in k_0} R^{\text{stay}}_{s''s''} < \frac{\gamma|k_0|R^{\text{stay}}_{s_ls_l}}{(n-1)-\gamma(n-2)}$, we get,

$$< \frac{1}{(n-1)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))}\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma(n-|k_1|)}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\frac{\gamma R^{\text{stay}}_{s_ls_l}}{(n-1-\gamma(n-2))}$$

Since $|k_1| \geq 1$ and $\gamma \leq 1$, it can be easily shown that $(n-|k_1|) \leq (n-1-\gamma(|k_1|-1))$. Using this result, we get,

$$\leq \frac{1}{(n-1)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))}\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma}{(1-\gamma)}\frac{\gamma R^{\text{stay}}_{s_ls_l}}{(n-1-\gamma(n-2))}$$

$$= \frac{1}{(n-1)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))}\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] + \frac{\gamma R^{\text{stay}}_{s_ls_l}}{(1-\gamma)(n-1)}\frac{(n-1)+\gamma-(n-1-\gamma(n-2))}{(n-1-\gamma(n-2))}$$

$$= \frac{1}{(n-1)} \left[\sum_{s' \in \mathcal{S} \setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))} \sum_{s'' \in \mathcal{S}} \sum_{s' \in \mathcal{S} \setminus s''} R^{\text{switch}}_{s's''} \right]$$
$$+ \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(1-\gamma)(n-1)} \frac{n-1+\gamma}{(n-1-\gamma(n-2))} - \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(1-\gamma)(n-1)} = Q^{\text{switch},\pi^{\text{opt}}}_{s \in k_0}$$

Hence, $Q^{\text{switch},\pi \neq \pi^{\text{opt}}}_{s \in k_1} < Q^{\text{switch},\pi^{\text{opt}}}_{s \in k_0}$.

Proof for 3-(b): As discussed in the Proof for 3-(a), $s = s_l$. Substituting the condition $|k_1| < n-1$ in Eq. (50), we get,

$$Q^{\text{switch},\pi \neq \pi^{\text{opt}}}_{s_l \in k_0} = \frac{1}{(n-1)} \left[\sum_{s' \in \mathcal{S} \setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(|k_1|-1))} \sum_{s'' \in k_1} \sum_{s' \in \mathcal{S} \setminus s''} R^{\text{switch}}_{s's''} \right]$$
$$+ \frac{\gamma}{(1-\gamma)(n-1)} \left[\frac{(n-1+\gamma)}{(n-1-\gamma(|k_1|-1))} \sum_{s'' \in k_0} R^{\text{stay}}_{s''s''} \right] - \frac{\gamma}{(n-1)(1-\gamma)} R^{\text{stay}}_{s_l s_l}$$

$$< \frac{1}{(n-1)} \left[\sum_{s' \in \mathcal{S} \setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))} \sum_{s'' \in k_1} \sum_{s' \in \mathcal{S} \setminus s''} R^{\text{switch}}_{s's''} \right]$$
$$+ \frac{\gamma}{(1-\gamma)(n-1)} \left[\frac{(n-1+\gamma)}{(n-1-\gamma(|k_1|-1))} \sum_{s'' \in k_0} R^{\text{stay}}_{s''s''} \right] - \frac{\gamma}{(n-1)(1-\gamma)} R^{\text{stay}}_{s_l s_l}$$

Substituting $\sum_{s'' \in k_0} R^{\text{stay}}_{s''s''} = \sum_{s'' \in k_0 \setminus s_l} R^{\text{stay}}_{s''s''} + R^{\text{stay}}_{s_l s_l}$ and the condition $R^{\text{stay}}_{ss} < \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(n-1)-\gamma(n-2)}$, $\forall s \in \mathcal{S} \setminus s_l$ in the second term of R.H.S. and solving, we get,

$$< \frac{1}{(n-1)} \left[\sum_{s' \in \mathcal{S} \setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(n-2))} \sum_{s'' \in k_1} \sum_{s' \in \mathcal{S} \setminus s''} R^{\text{switch}}_{s's''} \right]$$
$$+ \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(1-\gamma)(n-1)} \frac{n-1+\gamma}{(n-1-\gamma(n-2))} - \frac{\gamma R^{\text{stay}}_{s_l s_l}}{(1-\gamma)(n-1)} = Q^{\text{switch},\pi^{\text{opt}}}_{s_l \in k_0}$$

Hence, $Q^{\text{switch},\pi \neq \pi^{\text{opt}}}_{s \in k_0} < Q^{\text{switch},\pi^{\text{opt}}}_{s \in k_0}$.

Proof for 4-(a)&(b): This proof is straightforward since,

$$Q^{\text{stay},\pi \neq \pi^{\text{opt}}}_{s \in k_1 \text{or} k_0} < \max(Q^{\pi^{\text{opt}}}) = Q^{\text{stay},\pi^{\text{opt}}}_{s \in k_0}.$$

Hence, $Q^{\text{stay},\pi \neq \pi^{\text{opt}}}_{s \in k_1} < Q^{\text{stay},\pi^{\text{opt}}}_{s \in k_0}$ & $Q^{\text{stay},\pi \neq \pi^{\text{opt}}}_{s \in k_0} < Q^{\text{stay},\pi^{\text{opt}}}_{s \in k_0}$. □

The convergence of the algorithm's policy π and the adaptive abstraction $\widehat{\phi}$ to their respective optimal fixed-points is proved next.

Theorem 4. *Let $\{\pi_t\}_{t\in\mathbb{N}}$ denote the sequence of observation stream selection policies generated by the algorithm for $\epsilon = 1$. If $\sigma < |U|/(6N^{roc})$, $\beta < 1$ and Conditions (45)-(47) hold then,*

$$\lim_{t\to\infty} \pi_t(s) = \pi^*(\phi^*, s),\ \forall s \in \mathcal{S}$$

Proof. When $\epsilon = 1$, the algorithm performs a random walk over the states taking actions *stay* and *switch* with equal probability at each state. For each action, the agent receives a small set of τ samples from the observation stream corresponding to the transitioned state. Since the action *switch* shifts the agent's state uniformly randomly, the CCICPA weights (whitening vectors) of the adaptive abstraction $\widehat{\phi}^{\text{sfa}}$ converge quickly, while the CIMCA weights diffuse around randomly (due to the absence of a consistent temporal structure). Therefore, $\widehat{\phi}^{\text{sfa}}$ which is a product of CCIPCA and CIMCA weights, can be assumed to be a random variable. When the agent transitions to a new state (say at time t_0), the randomly initialized CIMCA weights are updated based on the temporally coherent samples from the corresponding observation stream, until the agent transitions out to a new state. We use the analysis discussed in Section 4.1 to find the relationship between the curiosity rewards accumulated in each state.

Let R^{cur} denote the steady state curiosity-reward function and R^{exp} denote the steady state expert-reward function ($R = R^{\text{cur}} + R^{\text{exp}}$). The proof has three parts:

1. We find the relationship between the steady state curiosity rewards for the stay action in each state.

2. We show that the curiosity rewards for the switch action are less than the stay action.

3. We show that the expert rewards are negligible compared to the dominant curiosity rewards.

We use these subparts to finally show the main result.

1. Here, we find the steady state curiosity-reward function term $R^{\text{cur}}(s_i, 0, s_i)$ for the *stay* action (a=0). To this end, we find the probability of a k-stay continuous action sequence (Figure 6) and find the corresponding average curiosity reward (see Eq. (14)). Starting from a random state $s \neq s_i$, the agent transitions to the state s_i by executing

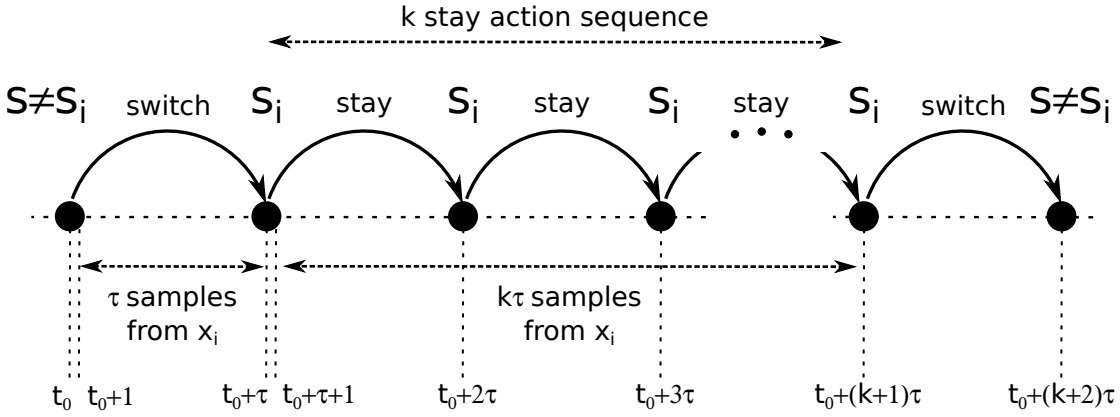

Figure 6: Transition diagram of a k-*stay* action sequence.

a *switch* action with a probability of $1/2(n-1)$ (see Eq. (15)). The agent receives τ samples from the stream $\mathbf{x}_i$ for this transition. The agent then executes its first *stay* action with a probability of $1/2$ and continues to observe τ samples from the same stream $\mathbf{x}_i$. The curiosity reward (computed through backward difference approx.) for the τ samples within the first stay action is equivalent to the curiosity rewards received for the time period $(\tau, 2\tau)$ of a randomly initialized $\widehat{\phi}^{\text{sfa}}$ at $t = t_0$.

$$r_0^{s_i,s_i} = \sum_{t'=\tau}^{2\tau} \left[\xi_i^{\text{sfa}}(t') - \xi_i^{\text{sfa}}(t'+1)\right] \tag{58}$$

$$= \xi_i^{\text{sfa}}(\tau) - \xi_i^{\text{sfa}}(2\tau). \tag{59}$$

From Lemma 5, $\xi_i^{\text{sfa}}(t)$ is an exponentially decaying function $\xi_i^{\text{sfa}}(2\tau) = \varsigma_i^{\tau}\xi_i^{\text{sfa}}(\tau)$, where $\varsigma_i = \Omega_i^q$.

$$r_0^{s_i,s_i} = \xi_i^{\text{sfa}}(\tau)(1-\varsigma_i^{\tau}) \tag{60}$$

The curiosity reward for the subsequent τ time steps = $\varsigma_i^{\tau} r_0^{s_i,s_i}$. The expected reward for all possible k-*stay* action sequences; $R^{\text{cur}}(s_i, 0, s_i) =$

$$\sum_{k=1}^{\infty} \Pr\left(s(t_0+(k+1)\tau) = s_i, ..., s(t_0+\tau) = s_i \middle| \{s(t_0), s(t_0+(k+2)\tau)\} \in S \setminus s_i\right) \times \frac{(1+...+\varsigma_i^{\tau(k-1)})r_0^{s_i,s_i}}{k}. \tag{61}$$

The probability term ($\Pr$) in the summation is the probability that the agent's state remains the same for the next k algorithm iterations after switching from another state.

Since the transition process between the states is Markovian, the probability term is equal to:

$$\Pr\left(s(t_0+(k+2)\tau)\in S\setminus s_i \middle| s(t_0+(k+1)\tau)=s_i\right)\times \prod_{p=1}^{k}\Pr\left(s(t_0+(p+1)\tau)=s_i \middle| s(t_0+p\tau)=s_i\right)\times \Pr\left(s(t_0+\tau)=s_i \middle| s(t_0)\in S\setminus s_i\right)=\frac{1}{2}\times\frac{1}{2^k}\times\frac{1}{2(n-1)}. \tag{62}$$

Substituting in Eq. (61), we get

$$R^{\text{cur}}(s_i,0,s_i)=\frac{1}{4(n-1)}\sum_{k=1}^{\infty}\frac{1}{2^k}\frac{(1-\varsigma_i^{\tau k})r_0^{s_i,s_i}}{k(1-\varsigma_i^{\tau})}, \tag{63}$$

$$=\frac{\xi_i^{\text{sfa}}(\tau)}{4(n-1)}\sum_{k=1}^{\infty}\frac{1}{k}\frac{1}{2^k}-\frac{1}{k}\left(\frac{\varsigma_i^{\tau}}{2}\right)^k. \tag{64}$$

Using Maclaurin series of $\log(1-\frac{1}{2})$ and $\log(1-\frac{\varsigma_i}{2})$ and solving, we get

$$R^{\text{cur}}(s_i,0,s_i)=\frac{\xi_i^{\text{sfa}}(\tau)}{4(n-1)}\ln(2-\varsigma_i^{\tau}). \tag{65}$$

If $\Omega(\mathbf{x}_i)<\Omega(\mathbf{x}_j)$, then $\xi_i^{\text{sfa}}(\tau)>\xi_j^{\text{sfa}}(\tau)$ and $\varsigma_i^{\tau}<\varsigma_j^{\tau}$ (see Lemma 5). Therefore, $R^{\text{cur}}(s_i,0,s_i)>R^{\text{cur}}(s_j,0,s_j)$. This implies, $R^{\text{cur}}(s_l,0,s_l)=\max_i R^{\text{cur}}(s_i,0,s_i)$.

2. Figure 7 shows the state transition diagram of a *switch* action from a state $s_{j\neq i}$ to the state s_i. From Lemma 3 and Condition (46), the accumulated curiosity rewards for the *switch* action for the period $(0,\tau)$ is less than zero. Therefore, $R^{\text{cur}}(s_i,1,s_j)<0$ and $R^{\text{cur}}(s_l,0,s_l)>R^{\text{cur}}(s_i,1,s_j)$.

3. Here, we discuss the contribution of the expert reward term (Eq. (11)). Slow feature outputs $\mathbf{y}(t)$ have unit variance, zero mean (see Section 3.1) and resemble half cosine functions [Franzius et al., 2007]. When $\epsilon=1$, the adaptive abstraction ($\widehat{\phi}^{\text{sfa}}$) outputs change rapidly resulting in high ROC errors. The average estimation error of $|U|$ ROC nodes, with N^{roc} max clusters initialized for each node, is $|U|/N^{\text{roc}}$. Therefore, for the values of $\sigma<|U|/(6N^{\text{roc}})$ (six-sigma of a Gaussian) and $\beta<1$, the Gaussian function output in Eq. (11) is of the order of magnitude less than 10^{-9}. Whereas, the dominant instantaneous curiosity rewards (Eqs. (60)-(26)) are of the order of minimum magnitude $>10^{-5}$ when $(\Omega<0.999), \tau>20$ and $I>3$. Therefore, the expert

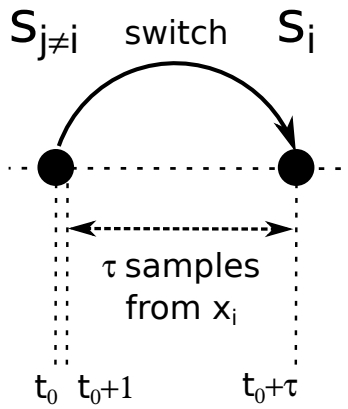

Figure 7: State transition diagram of the *switch* action.

rewards are negligible compared to the dominant curiosity rewards: $R \approx R^{\text{cur}}$. This implies, $R(s_l, 0, s_l) = \max R$.

From Eq. (65), we have

$$\frac{R(s_i, 0, s_i)}{R(s_l, 0, s_l)} = \frac{\xi_i^{\text{sfa}}(\tau)\ln(2-\varsigma_i^\tau)}{\xi_l^{\text{sfa}}(\tau)\ln(2-\varsigma_l^\tau)} < \frac{\ln(2-\varsigma_i^\tau)}{\ln(2-\varsigma_l^\tau)}.$$

Using Taylor series for $\ln(2-x)$ at x close to 1, we get $\ln(2-x) \approx -\ln(x)$. Therefore, we have

$$\frac{R(s_i, 0, s_i)}{R(s_l, 0, s_l)} < \frac{-\ln(\varsigma_i)}{-\ln(\varsigma_l)} = \frac{-\ln(\Omega_i)}{-\ln(\Omega_l)} < \frac{1}{(1+(n-1)(1-\gamma)/\gamma)}.$$

It follows from Lemma 7 and Theorem 3 that $\lim_{t\to\infty} \pi_t(s) = \pi^*(\phi^*, s),\ \forall s \in \mathcal{S}$. □

Theorem 4 shows that during pure exploration ($\epsilon = 1$), π_t converges to a policy with an action *stay* for the state (s_l) corresponding to the current *easiest but not yet encoded* observation stream ($\mathbf{x}_l$), and the action *switch* for rest of the states. Also, since the policies π_t and $\pi^*(\phi^*)$ are binary-vectors, it follows that $\exists t_c \in \mathbb{N}\ (t_0 < t_c < \infty),$ s.t. for $t = t_c,\ \pi_t = \pi^*(\phi^*)$.

Theorem 5. *Let $\{\widehat{\phi}_t\}_{t\in\mathbb{N}}$ denote the sequence of adaptive abstractions generated by the algorithm for $\epsilon = 0$. If $\pi_{t_c} = \pi^*(\phi^*)$, $R_{t_c}(s_l,\ \textit{stay}, s_l) = \max(R_{t_c})$, $\beta > \frac{0.367\sqrt{I}}{\tau}$ and Conditions (45),(46) and (47) hold, then*

$$\lim_{t\to\infty} \widehat{\phi}_t = \phi^*$$

Proof. When $\epsilon = 0$, the agent exploits the observation stream selection policy π_{t_c}. Therefore, the agent observes samples from $\mathbf{x}_l$ and IncSFA-ROC makes learning progress.

As a result, the IncSFA-ROC errors and the curiosity-rewards diminish exponentially (see the proof of the previous theorem). The expert-reward term in Eq. (11) now becomes dominant compared to the curiosity-reward term. If β is set such that the expert-reward term is greater than the maximum curiosity-reward, it ensures $R_t(s_l,\ \text{stay}, s_l) = \max(R_t)$. The maximum instantaneous curiosity rewards for IncSFA can be found by differentiating Eq. (27):

$$\frac{dr^{\text{sfa}}(t)}{dt} = \frac{\Omega^t \ln^3(\Omega)\sqrt{I}\,(4b^2\Omega^{4t} - 10b\Omega^{2t} + 1)}{(b\Omega^{2t}+1)^{\frac{7}{2}}}. \tag{66}$$

Solving for roots, we get the maximum when $-t\ln(\Omega) = \dfrac{\ln(\frac{4b}{5-\sqrt{21}})}{2} \approx 0.5\ln(9.58b)$. Substituting $t = \tau$ and simplifying by using the result $\max(\frac{\ln^2(x)}{\sqrt{x}}) = 16\text{e}^{-2}$, we get

$$\max r^{\text{sfa}}(t) = \frac{0.367\sqrt{I}}{\tau^2}. \tag{67}$$

For τ samples, $\max \int_\tau r^{\text{sfa}}(t) < \frac{0.367\sqrt{I}}{\tau}$. Setting $1 > \beta > \frac{0.367\sqrt{I}}{\tau}$ would make the weighted expert-reward term $\beta Z^{\delta,\sigma}(\langle\xi^{\text{roc}}\rangle_t^\tau)$ converge to a value greater than $\frac{0.367\sqrt{I}}{\tau}$ as $\langle\xi^{\text{roc}}\rangle_t^\tau \to \delta$, while also satisfying the constraint for Theorem 4. This ensures that the policy remains optimal $\pi_{t>t_c} = \pi^*(\phi^*)$ and from Theorem 2 the result follows. □

When $\langle\xi^{\text{roc}}\rangle_t^\tau < \delta$, the adaptive abstraction $\widehat{\phi}$ is frozen and saved to the abstraction set Φ_t ($\Phi_t \leftarrow \Phi_t \cup \widehat{\phi}$). Theorems 2-5 show that the saved abstraction satisfies Constraints (3)-(4) and the cardinality of the abstraction set increments by a value 1. The process repeats until all the abstractions have been learned.

Theorem 5 requires t_c, the time at which the policy π has converged. However, in practice t_c is not known *a priori* and is generally difficult to estimate. The ϵ-greedy strategy is a simple heuristic that we found to be useful for transitioning from $\epsilon = 1$ (pure exploration) to $\epsilon = 0$ (pure exploitation). By selecting a decay-constant close to 1 the algorithm has sufficient time for the policy to converge to the optimal. Later in Section 5 we discuss the overall hyper-parameters of the algorithm along with an intuition on how to tune them. Next, we discuss the performance of the algorithm when a few of the conditions that were assumed to be true for the above analysis are not held.

Case – Condition (47) is not held. Here, we discuss the performance of the algorithm when a few observation streams violate Condition 47. Let $r = \frac{\gamma}{(n-1-\gamma(n-2))}$.

Substituting r in Condition (47) we get

$$ln(\Omega(\mathbf{x}_i)) < r\ ln(\Omega(\mathbf{x}_j)), \forall \Omega(\mathbf{x}_i)) > \Omega(\mathbf{x}_j). \tag{68}$$

Definition 3. *A stream* $\mathbf{x}$ *is r-dominated by another stream* $\mathbf{x}'$ *if* $ln(\Omega(\mathbf{x})) < r\ ln(\Omega(\mathbf{x}'))$.

Using the above definition we show that the algorithm cannot find the easier stream to encode between two streams that are not r-dominated.

Theorem 6. *Let* $\{\pi_t\}_{t\in\mathbb{N}}$ *denote the sequence of observation stream selection policies generated by the algorithm for* $\epsilon = 1$. *Let* $\mathcal{S}_r$ *be the set of states whose corresponding observation streams are not r-dominated by* $\mathbf{x}_l$. *If Conditions (45) and (46) hold, then,* $\pi_t(s)$ *has two limits points equal to* $\left(1 - \mathbb{1}_{\{s_l\}}(s)\right)$ *or* $(1 - \mathbb{1}_{\mathcal{S}_r}(s))$, $\forall s \in \mathcal{S}$.

Proof. From Theorem 4, we get $\dfrac{R^{\text{stay}}_{ss}}{R^{\text{stay}}_{s_l s_l}} \geq \dfrac{\gamma}{(n-1-\gamma(n-2))},\ \forall s \in \mathcal{S}_r \setminus s_l$.

Let $R^{\text{stay}}_{ss} = \dfrac{\gamma R^{\text{stay}}_{s_l s_l}}{(n-1-\gamma(n-2))} + \epsilon_s$, where $\epsilon_s,\ \forall s \in \mathcal{S}_r \setminus s_l$ are non-negative constants. Let $\pi^* = 1 - \mathbb{1}_{\{s_l\}}(s)$ and $\widehat{\pi} = 1 - \mathbb{1}_{\mathcal{S}_r}(s),\ \forall s \in \mathcal{S}$. From Eq. (49) and substituting for R^{stay}_{ss} we have,

$$\begin{aligned}
Q^{\text{switch},\widehat{\pi}}_{s\in k_1} &= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] \\
&\quad + \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_0} R^{\text{stay}}_{s''s''} \\
&= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] \\
&\quad + \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\left(R^{\text{stay}}_{s_l s_l} + \sum_{s''\in k_0\setminus s_l} R^{\text{stay}}_{s''s''}\right)
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{(n-1+\gamma)}\left[\sum_{s'\in\mathcal{S}\setminus s} R^{\text{switch}}_{ss'} + \frac{\gamma}{(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}\right] \\
&\quad + \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\left[\left(\frac{\gamma(|k_0|-1)}{(n-1)-\gamma(n-2)}+1\right)R^{\text{stay}}_{s_l s_l} + \sum_{s''\in k_0\setminus s_l}\epsilon_{s''}\right]
\end{aligned}$$

$$= Q^{\text{switch},\pi^*}_{s\in k_1} - \frac{\gamma}{(n-1+\gamma)}\left[\frac{\sum_{s''\in\mathcal{S}}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1-\gamma(n-2))} - \frac{\sum_{s''\in k_1}\sum_{s'\in\mathcal{S}\setminus s''} R^{\text{switch}}_{s's''}}{(n-1-\gamma(|k_1|-1))}\right]$$
$$+ \frac{\gamma}{(1-\gamma)\,(n-1-\gamma(|k_1|-1))}\sum_{s''\in k_0\setminus s_l}\epsilon_{s''}$$
$$= Q^{\text{switch},\pi^*}_{s\in k_1} - A + B$$

Both A and B are non-zero. So, clearly when $B > A,\ Q^{\text{switch},\widehat{\pi}}_{s\in k_1} > Q^{\text{switch},\pi^*}_{s\in k_1}$. Therefore, $\arg\max_{\pi} Q^{\pi} \neq \pi^*$. Evaluating similarly for $Q^{\text{stay or switch},\widehat{\pi}}_{s\in k_1 \text{ or } k_0}$ we get, for the condition $B > A$, $\arg\max_{\pi} Q^{\pi} = \widehat{\pi}$. □

Theorem 6 shows that if a few observation streams violate the Condition (47), then π is not guaranteed to converge to the optimal policy (Theorem 3). It instead converges to a sub-optimal policy, which returns an action *stay* in all the states whose observation streams are not r-dominated with $\mathbf{x}_l$ and *switch* in all the remaining states. This differs from the optimal policy, where the action *stay* is returned for the state s_l and *switch* in all the other states. The suboptimal policy during exploitation makes the algorithm converge to an abstraction encoding any of the observation streams corresponding to the states in $\mathcal{S}_r$, with an uniform probability over multiple experiment trials. As an example, consider $X = \{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d\}$ with $\Omega(\mathbf{x}_a) < \Omega(\mathbf{x}_b) < \Omega(\mathbf{x}_c) < \Omega(\mathbf{x}_d)$. If $\mathbf{x}_a$ and $\mathbf{x}_b$ are not r-dominated, then the algorithm generates the following sequence of abstractions with equal probability over multiple trials:

1. $\Phi = \{\phi_1^{\mathbf{x}_a},\ \phi_2^{\mathbf{x}_b},\ \phi_3^{\mathbf{x}_c},\ \phi_4^{\mathbf{x}_d}\}$ (optimal),
2. $\Phi = \{\phi_1^{\mathbf{x}_b},\ \phi_2^{\mathbf{x}_a},\ \phi_3^{\mathbf{x}_c},\ \phi_4^{\mathbf{x}_d}\}$,

where $\phi_1^{\mathbf{x}_a}$ denotes the first abstraction learned corresponding to the stream $\mathbf{x}_a$. These two sequences differ only in the first two terms. Therefore, the sub-optimality is local and does not effect the order of the remaining streams. When the above experiment is executed over many trials, the algorithm converges to the optimal sequence (first sequence) for half the number of trials and another half to the sub-optimal sequence (second sequence). Intuitively, the Condition (47) denotes how far apart should a pair of observation streams be in terms of their curiosity function values, to be distinguished as two different streams by the algorithm. We find that the condition is easily met between

two difficult-to-learn streams than two very easy-to-learn streams. The reasons are as follows. Let b denote the difference of $b = \Omega(\mathbf{x}_i) - \Omega(\mathbf{x}_j)$. Substituting r and $b > 0$ in Condition (47) we get

$$b > b^r = \Omega(\mathbf{x}_j)^r - \Omega(\mathbf{x}_j). \tag{69}$$

The above condition denotes how far apart should be the learning difficulties of any two observation streams. It can be inferred that as $\Omega(\mathbf{x}_j) \to 1$, $b^r \to 0$. Therefore, for a given n and γ, the condition is most-likely met for observation streams with higher $\Omega(\mathbf{x}_j)$ (difficult-to-learn streams) than with lower $\Omega(\mathbf{x}_j)$. In the limiting case of $\gamma = 1$, the condition is always met. Therefore, setting γ close to 1 the condition can be met for most observation streams. The only drawback of selecting a γ very close to 1 is that it increases the convergence time of LSPI.

Case – $n \leq 2$**:** For $n = 2$, *i.e.* two states $\mathcal{S} = \{s_1, s_2\}$, the stochastic *switch* action is equivalent to a deterministic *switch* to the other state. Therefore, Curious Dr. MISFA can learn an abstraction by switching between the states (*i.e.*, only if both observation streams are individually encodable). If $\Omega(\mathbf{x}_{\text{mix}}) < \min(\Omega(\mathbf{x}_1), \Omega(\mathbf{x}_2))$, where $\mathbf{x}_{\text{mix}}$ denotes the mixture stream, then the algorithm will learn a policy that switches the agent's state to the other ($\pi = [1, 1]$). This results in an abstraction corresponding to the mixture stream. In this special case, the algorithm will learn a total of 3 abstractions. For $n = 1$, the solution is trivial, the algorithm learns an abstraction corresponding to the observation stream, irrespective of the observation stream selection policy.

Case – $J \geq 1$**:** The columns in the slow-feature matrix $\widehat{\phi}^{\text{sfa}}$ denote the slow-feature vectors ordered according to how fast or slow the corresponding output changes in time (see Section 3.1). IncSFA uses the *sequential addition* [Chen et al., 2001, Kompella et al., 2012a] technique to update all the slow-feature vectors simultaneously for each sample; the slowest feature is updated first, then the sequential addition technique shifts each observation into a space where the minor component of the current space will be the first PC, and all other PCs are reduced in order by one. Therefore, the curiosity function for the second slow feature is $\left[1 - \frac{\eta^{\text{sfa}}(\lambda_{K-2} - \lambda_{K-1})}{1 - \eta^{\text{sfa}} - \eta^{\text{sfa}}\lambda_{K-1}}\right]$, where K denotes the dimensionality of the *whitened* output of $\mathbf{x}_i(t)$. When $J > 1$, the overall learning difficulty of the input stream (Definition 1) is equal to the learning difficulty of the most

difficult slow feature component. The resultant curiosity function is as follows:

$$\Omega^{\text{res}}(\mathbf{x}_i) = \max_j \left\{ \left[1 - \frac{\eta^{\text{sfa}}(\lambda_{K-1} - \lambda_K)}{1 - \eta^{\text{sfa}} - \eta^{\text{sfa}}\lambda_K} \right], ..., \left[1 - \frac{\eta^{\text{sfa}}(\lambda_{K-j} - \lambda_{K-J+1})}{1 - \eta^{\text{sfa}} - \eta^{\text{sfa}}\lambda_{K-J+1}} \right] \right\},$$

where $\lambda_K \neq \lambda_{K-j}, j \in \{1, ..., J\}$ denote the J smallest eigenvalues of $E[\dot{\mathbf{z}}_i \dot{\mathbf{z}}_i^T]$, and $\mathbf{z}_i(t) \in \mathbb{R}^K$ is the *whitened* output of $\mathbf{x}_i(t)$.

In Curious Dr. MISFA, the Frobenius norm of the temporal difference of the slow-feature matrix is used to compute the curiosity rewards. In the case when $J > 1$, the weight change within the τ samples due to the easier-to-learn components may dominate the curiosity rewards (especially when J is small). These rewards may not truly estimate Ω^{res} of a stream. One approach to address this issue is to compute column-wise Frobenius norm and set the minimum as the reward:

$$\xi^{\text{sfa}}(t) = \left[\|\widehat{\phi}^{\text{sfa1}}(t) - \widehat{\phi}^{\text{sfa1}}(t-1)\|, ..., \widehat{\phi}^{\text{sfa}J}(t) - \widehat{\phi}^{\text{sfa}J}(t-1)\| \right], \tag{70}$$

$$r^{\text{sfa}}(t) = \min_j \left\{ -\int_\tau \dot{\xi}^{\text{sfa1}} dt, ..., -\int_\tau \dot{\xi}^{\text{sfa}J} dt \right\}. \tag{71}$$

where $\|.\|$ denotes the Frobenius norm, $\widehat{\phi}^{\text{sfai}}$ denotes the i^{th} column of $\widehat{\phi}^{\text{sfa}}$. This way, Curious Dr. MISFA learns the first abstraction corresponding to the stream with the least Ω^{res}.

Case – $\mathbf{x}_i$ is IncSFA encodable but not ROC encodable: If this assumption is not held and $\mathbf{x}_i$ is the current easiest and novel observation stream, then Theorem 5 will not hold true. The reasons are explained as follows. When $\epsilon = 1$, the agent's policy converges to $1 - \mathbb{1}_{\{s_i\}}(s),\ \forall s \in \mathcal{S}$. When the agent begins to exploit this policy, the curiosity-rewards diminish exponentially. If the slow-feature outputs are not correlated with the user-signal $\mathbf{u}$, ROC's estimation error $\xi^{\text{roc}}(t)$ will be high and therefore, the expert-reward term in Eq. (11) will be close to zero. As a result, the reward function term $R_t(s_i, \text{stay}, s_i)$ decreases exponential due to the absence of both curiosity and expert rewards. When the reward term falls below the one corresponding to the next easiest observation stream $R_t(s_i, \text{stay}, s_i) < R_t(s_j, \text{stay}, s_j)$, then $R_t(s_j, \text{stay}, s_j) = \max R_t$. The LSPI algorithm learns a new policy $1 - \mathbb{1}_{\{s_j\}}(s),\ \forall s \in \mathcal{S}$. If $\mathbf{x}_j$ is IncSFA-ROC encodable, then Theorem 5 holds true and the algorithm learns an abstraction corresponding to it. Otherwise, the above process repeats until the easiest IncSFA-ROC encodable observation stream is found. The resultant learned abstraction set will be optimally ordered, albeit, the time taken will be longer.

Case – Condition (46) is not held. From Lemma 3, we find that selecting a very small τ may result in the accumulation of negative rewards for the *stay* action. This may lead to an unstable result. One way to address this issue is to adapt τ such that the maximum of the reward function for the *stay* action is always positive, that is, $\max R(s_i, 0, s_i) > 0$, when $\epsilon = 1$.

5 Pseudocode and Hyper Parameters

The constant hyper parameters that need to be set for the working of the algorithm are as follows: (1) IncSFA learning rate η^{sfa}, (2) ROC amnesic rate η^{roc}, (3) ROC max clusters N^{roc}, (4) threshold δ, (5) ϵ decay multiplier, (6) τ, (7) σ and (8) β. IncSFA uses a constant learning rate η^{sfa} that is quite intuitive to set Kompella et al. [2012a]. The amnesic parameter η^{roc} is used to make ROC adaptive. Higher values will make ROC adapt faster to the new data, however at the cost of being less stable. Since, the learning rates are not adapted during the experiments, the effect of selecting different learning rates that ensure convergence of IncSFA-ROC, do not effect the outcome of the final result. The maximum number of clusters N^{roc} in ROC is set to encode multiple slow feature values for each $\mathbf{u}_i \in U$. Higher values can be used, however, very high values may lead to spurious clusters. δ is generally set to values close to zero < 1 depending on how well the expert modules need to encode the inputs. ϵ decay multiplier is set close to 1 for sufficient exploration and is reduced to a lower value when ϵ is low, to transition quickly to the pure exploitation mode. τ is usually set to a small number or can be adapted to keep the maximum of the reward function positive. The parameters σ and β correspond to the expert-reward term of the reward function 11. The effect of varying these parameters on the algorithm are discussed in detail through experiments in Section 6.2.

Algorithm 1 summarizes Curious Dr. MISFA. A *Python*-based implementation of the algorithm can be found here: `https://dl.dropboxusercontent.com/u/12734807/cdmisfa.zip`.

Algorithm 1: CURIOUS DR. MISFA

$\Phi_0 \leftarrow \{\}$, $\pi_0^b \leftarrow$ RANDOM (), $\widehat{\phi} \leftarrow 0$, $G \leftarrow$ true, $\widetilde{R} \leftarrow 0$

for *t ← 0 to* ∞ **do**

  $s_t \leftarrow$ current state, $a_t \leftarrow \pi_t^b(s_t)$ `//Sense`

  Take action a_t, observe next state $s_{t+1}(= P(s_t, a_t))$ and τ input samples $\left(\mathbf{x}(t;\tau) = \mathbf{x}_{s_{t+1}}(t;\tau), \mathbf{u}(t;\tau)\right)$

  for *each* ϕ *in* Φ_t **do**

    $-\int_\tau \dot{\xi}^{\text{sfa}} dt, \langle \xi^{\text{roc}} \rangle_t^\tau \leftarrow$ Compute-Error$((\mathbf{x}(t;\tau), \mathbf{u}(t;\tau)), \phi)$

    if $\langle \xi^{roc} \rangle_t^\tau < \delta$ **then**

      $G \leftarrow$ false `//Update gating flag`

    end

  end

  $r \leftarrow 0$, $\langle \xi^{\text{roc}} \rangle_t^\tau \leftarrow \infty$ `//Initialize default values`

  if G *is true* **then**

    `//Update adaptive IncSFA-ROC`

    $\widehat{\phi} \leftarrow \Theta\left((\mathbf{x}(t;\tau), \mathbf{u}(t;\tau)), \widehat{\phi}\right)$

    $-\int_\tau \dot{\xi}^{\text{sfa}} dt, \langle \xi^{\text{roc}} \rangle_t^\tau \leftarrow$ Compute-Error$\left((\mathbf{x}(t;\tau), \mathbf{u}(t;\tau)), \widehat{\phi}\right)$

    $r \leftarrow -\int_\tau \dot{\xi}^{\text{sfa}} dt + \beta Z^{\delta,\sigma}(\langle \xi^{\text{roc}} \rangle_t^\tau)$

  end

  `//Update reward function`

  $\widetilde{R}_{a_t}^{s_t s_{t+1}} \leftarrow \alpha r + (1-\alpha)\widetilde{R}_{a_t}^{s_t s_{t+1}}$, $R_t \leftarrow \widetilde{R}/\|\widetilde{R}\|$

  `//Update observation stream selection policy`

  $\pi_{t+1} \leftarrow$ Model-LSPI (S, A, P, R_t)

  `//Update behavior policy`

  $\pi_{t+1}^b \leftarrow \epsilon$-greedy$(\pi_{t+1})$

  if $\langle \xi^{roc} \rangle_t^\tau < \delta$ **then**

    $\Phi_{t+1} \leftarrow \Phi_t \cup \widehat{\phi}$ `//Save Module`

    $\pi_{t+1}^b \leftarrow$ RANDOM (), $\widehat{\phi} \leftarrow 0$, $G \leftarrow$ true, $\widetilde{R} \leftarrow 0$ `//Reset`

  end

end

6 Experimental Results

We present here experimental results conducted on oscillatory test streams to support the theoretical analysis presented in Section 4. More studies on the types of representations learned by the IncSFA algorithm, and applications of Curious Dr. MISFA on humanoid platforms with high-dimensional video inputs can be found in our previous work [Kompella et al., 2012a, Luciw et al., 2013, Kompella et al., 2014, 2015].

6.1 Proof of Concept

In this experiment, the convergence of the algorithm is illustrated for an input that consists of three 2D nonlinear oscillatory audio streams $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$, each encodable by IncSFA:

$$\mathbf{x}_1 \quad : \quad \begin{cases} x_1(t) = \sin(4\,\theta_t - \pi/4.) - \cos(44\,\theta_t)^2 \\ x_2(t) = \cos(44\,\theta_t) \end{cases}, \tag{72}$$

$$\mathbf{x}_2 \quad : \quad \begin{cases} x_1(t) = \sin(3\,\theta_t) + \cos(27\,\theta_t)^2 \\ x_2(t) = \cos(27\,\theta_t) \end{cases}, \text{ and} \tag{73}$$

$$\mathbf{x}_3 \quad : \quad \begin{cases} x_1(t) = \cos(12\,\theta_t) \\ x_2(t) = \cos(2\,\theta_t) + \cos(12\,\theta_t)^2 \end{cases}, \tag{74}$$

where $\theta_t = 2\pi(t\%500)/500$, $\%$ denotes the modulo operator. The environment has three states $\mathcal{S} = \{s_1, s_2, s_3\}$ associated with the observation streams. Since a user-signal is unavailable, the algorithm assumes $\mathbf{u}(t)$ to indicate a time-index of a period = 500, $U = [0, 499]$ and $\mathbf{u}(t) = t\%(500)$. Figure 8(a) illustrates the environment. The slowest feature in the stream $\mathbf{x}_1$ is $\mathbf{y}_1(t) = x_1(t) + x_2(t)^2 = \sin(4\,\theta_t - \pi/4.)$. For the streams $\mathbf{x}_2$ and $\mathbf{x}_3$, the features are $\mathbf{y}_2(t) = x_1(t) - x_2(t)^2 = \sin(3\,\theta_t)$ and $\mathbf{y}_3(t) = -x_1(t)^2 + x_2(t)^2 = \cos(2\,\theta_t)$ respectively. To extract these slow features, each observation stream is expanded via a polynomial expansion of degree 2 to a 5 dimensional stream (see Section 3.1). The expanded streams have the following curiosity function values: $\Omega_1 = 0.98979$, $\Omega_2 = 0.99611$, and $\Omega_3 = 0.99924$. Therefore, observation stream $\mathbf{x}_1$ is the easiest stream to encode followed by $\mathbf{x}_2$ and then $\mathbf{x}_3$.

Experiment parameters: We use a fixed parameter setting for the entire experiment. $\eta^{\text{sfa}} = 0.05, J = 1$. There are a total of $p = 500$ ROC clustering nodes (see

Section 3.1). $N^{\text{roc}} = 2$. $\delta = 0.3, \eta^{\text{roc}} = 0.2$. We initialized ϵ to 1.1, so that the agent explores long enough. However, when used as a probability, any value of $\epsilon > 1$ is considered as 1. ϵ decays after every algorithm iteration with a multiplier equal to 0.998 and is set to 0.99 when $\epsilon < 0.9$. $\gamma = 0.99, \tau = 100, \alpha = 0.0198 (= 2/N + 1)$ (a moving average of period 100 algorithm iterations). σ and β are set to 50 and 0.01 respectively based on the conditions from Theorems 4 and 5 (we discuss the effect of selecting different values in Section 6.2). To avoid the influence of any large initial noisy IncSFA-ROC weight changes, we clip the curiosity-rewards between $(-0.5, 0.5)$.

The dynamics of the algorithm can be observed by studying the time varying reward function R_t, action value function Q and the ROC estimation error $\xi^{\text{roc}}(t)$. Figures 8(b) and (c) show the reward function and the normalized value function for a single run of the experiment. Both figures share a common legend. Solid lines represent the reward in Figure 8(b) and the value in Figure 8(c) for the action *stay* in each state s_i. The dotted lines in Figure 8(c) represent the value for the action *switch* in each state s_i and in Figure 8(b) they represent the *marginalized* reward for the action *switch* at each state s_i, ($\frac{1}{2}\sum_j R(s_i, \textit{switch}, s_j)$).

For the sake of explanation, the learning process can be thought of as passing through three phases, where each phase corresponds to learning a single abstraction module.

Phase 1: At the beginning of Phase 1, the agent starts exploring by executing either *stay* or *switch* at each state. After a few hundred algorithm iterations, the reward function begins to stabilize and is such that $R(s_1, \textit{stay}) > R(s_2, \textit{stay}) > R(s_3, \textit{stay}) > 0$, ordered according to the learning difficulty of the observation streams. However, the reward components for the *switch* action are either close to zero or negative. The resultant value function learned is such that the *stay* action is state s_1 has the highest value. Therefore, the policy π converges to the optimal policy (*i.e.*, to *stay* at the state corresponding to the easiest observation stream $\mathbf{x}_1$ and *switch* at every other state). As ϵ decays, the agent begins to exploit the learned policy, and the adaptive IncSFA-ROC abstraction $\widehat{\phi}$ converges to ϕ^* (slow feature corresponding to the observation stream $\mathbf{x}_1$). The ROC estimation error (Figure 8(d)) decreases and falls below the threshold δ, at which point, the abstraction is added to the abstraction set Φ. The increase in the reward value of $R(s_1, \textit{stay})$ near the end of the phase is caused by the expert-reward

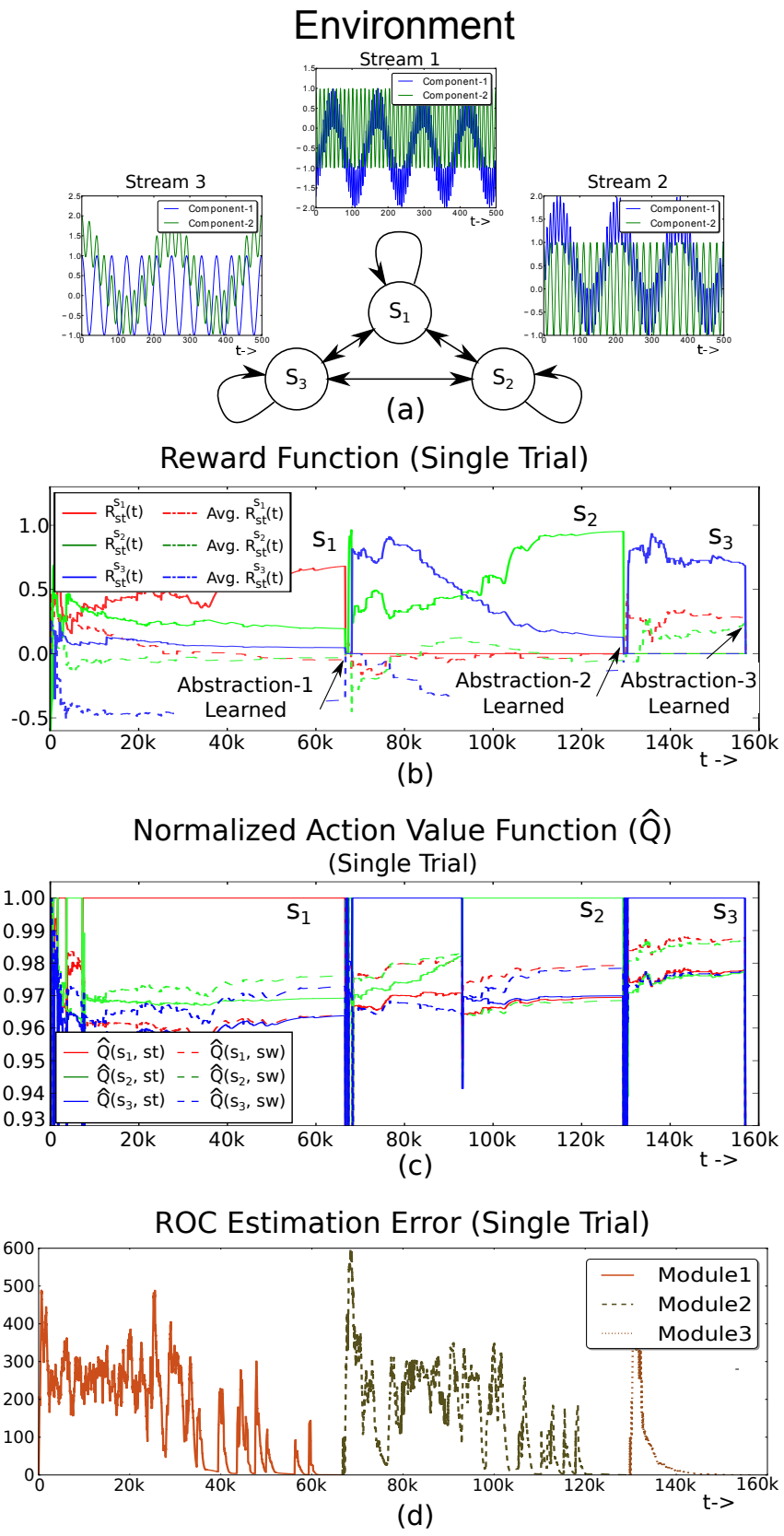

Figure 8: **Synthetic Streams**: See text for details. (Figures are best viewed in color)

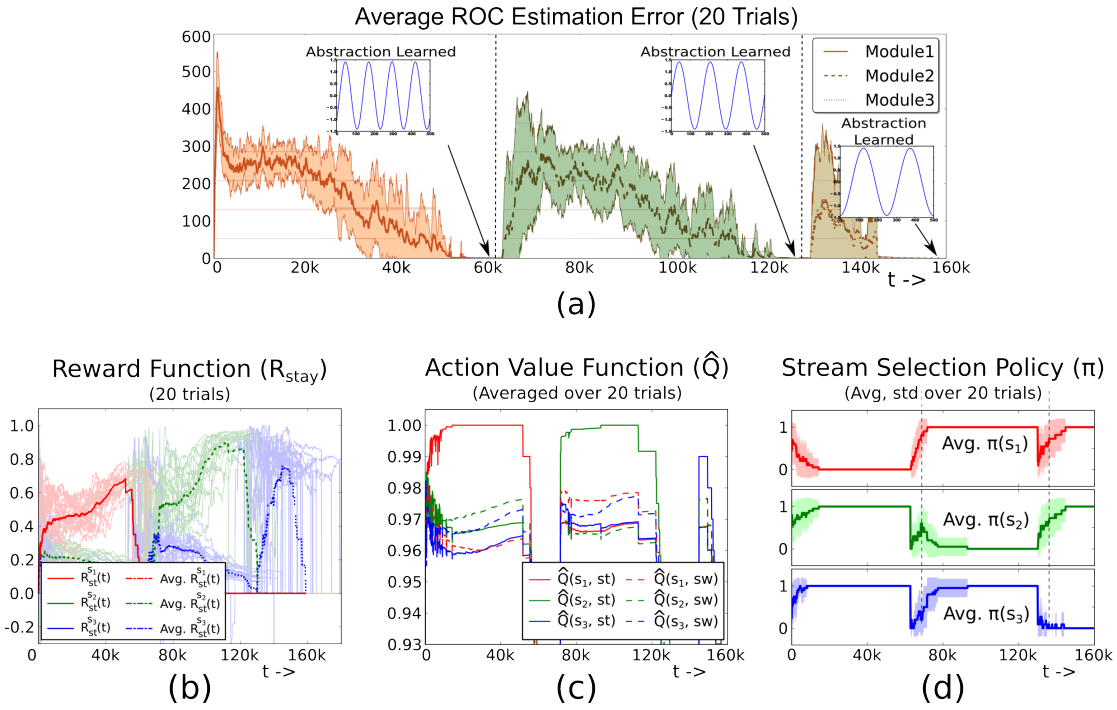

Figure 9: Results of the synthetic streams experiment conducted over 20 Trials. See text for details. (Figures are best viewed in color)

term in Eq. (11). Both ϵ and R are reset and the algorithm enters Phase 2 at ($t \approx 70k$).

Phase 2: The agent begins to explore again, however, it does not receive any reward for the (s_1, *stay*) tuple as the observations are filtered by the gating system. After a few hundred algorithm iterations, $R(s_2, \textit{stay}) > R(s_3, \textit{stay}) > R(s_1, \textit{stay}) = 0$, the adaptive abstraction converges, but to the slow feature corresponding to the observation stream $\mathbf{x}_2$.

Phase 3: The process continues again until the third abstraction is learned.

Figure 9 shows results of the experiment conducted for 20 trials with different random initializations. Figure 9(a) shows the average ROC estimation error plot, where the shaded region represents the standard deviation. Figure 9(b) shows the reward function of only *stay* action for all the 20 trials. The bold lines represent their average over the 20 trials. It is clear that for all the 20 trials the algorithm learns the abstraction corresponding to the easiest stream $\mathbf{x}_1$ as its first abstraction module, followed by a module for $\mathbf{x}_2$ and $\mathbf{x}_3$. Figures 9(c) and (d) show plots of the average normalized value function and the averaged policy with standard deviation (shaded region) over 20 trials. This experiment result shows that the algorithm learns abstractions for the observation streams

in the order of the increasing learning difficulty supporting the theoretical analysis of the problem.

6.2 The Significance of Setting β and σ

Theorems 4 and 5 discussed the broad range of values for setting the parameters σ and β for pure exploration and exploitation phases, such that, the algorithm can learn the optimal solution. However, when using the heuristic decaying ϵ-greedy strategy to smoothly transition from pure exploration to exploitation, we find that these parameters require further tuning to achieve optimal performance. We present here quantitative experimental results that show the effect of selecting different tuples of (σ, β) on the algorithm. To this end, we use the environment and the rest of the parameters discussed in the previous section to conduct the experiments. We focus here only on the policy learned for module 1.

Using the conditions provided in Theorems 4 and 5, we select values for σ around $|U|/(6N^{\text{roc}}) = 42$ and for β around $\frac{0.367\sqrt{I=5}}{\tau=100} = 0.008$; $\sigma \in \{0.01, 20, 50, 70, 150, 300\}$ and $\beta \in \{\ 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. For each tuple of (σ, β), results are collected with 20 different random initializations. Each experiment terminates either when the module 1 is learned (*i.e.*, when $\xi^{\text{roc}} < \delta$) or the number of iterations cross a value greater than 650, whichever happens earlier. The learned observation stream selection policy is averaged over the 20 experiment trials ($\widehat{\pi}$) for each tuple (σ, β). The plots of these results are shown in Figure 10(a). The optimal policy for learning module 1 is $\pi^* = [0, 1, 1]$. For a clearer visualization, the average policy values are plotted against $(\sigma, \log(\beta))$. From the plots, it is clear that the optimal performance is achieved for a large range of tuples $(20, 0.005$ to $0.5)$, $(50$ to $70, 0.005$ to $0.01)$, $(150$ to $300, 0.005)$ and a near-optimal performance for quite a few other values. This result is also evident in Figures 10(b)-(c) that show the norm of $(\widehat{\pi} - \pi^*)$ averaged (marginalized) over all the values of σ and β respectively. The minimum error for the values of β independent to σ is at $\beta = 0.005$ and the same for the values of σ independent to β is at $\sigma = 20 - 50$. It can also be inferred from the figures that the algorithm is sensitive to the parameter β as compared to σ.

This experiment shows that the the conditions in Theorems 4 and 5 are a good

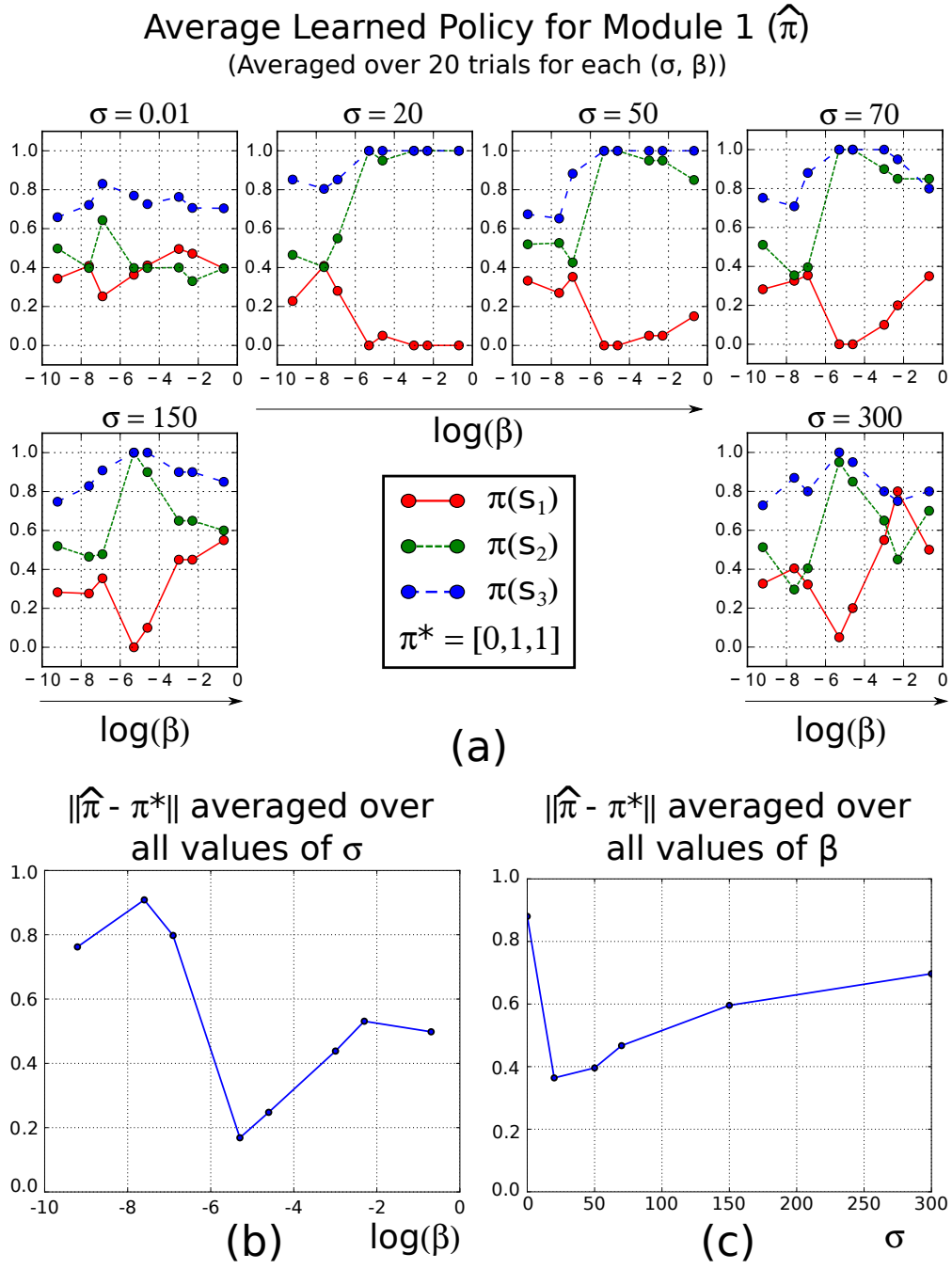

Figure 10: Experiments with different (σ, β) values. (a) Averaged observation stream selection policy $\widehat{\pi}$ learned for module 1. The policy $\widehat{\pi}$ is averaged over 20 trials of random initializations. The optimal policy for module 1 is $\pi^* = [0, 1, 1]$. The algorithm with values of σ around 20 to 70 and $\log \beta$ around -6 to -3 learns the optimal. This is also evident from the norm of $(\widehat{\pi} - \pi^*)$ averaged (marginalized) over all values of (b) σ and (c) β. See text for more details.

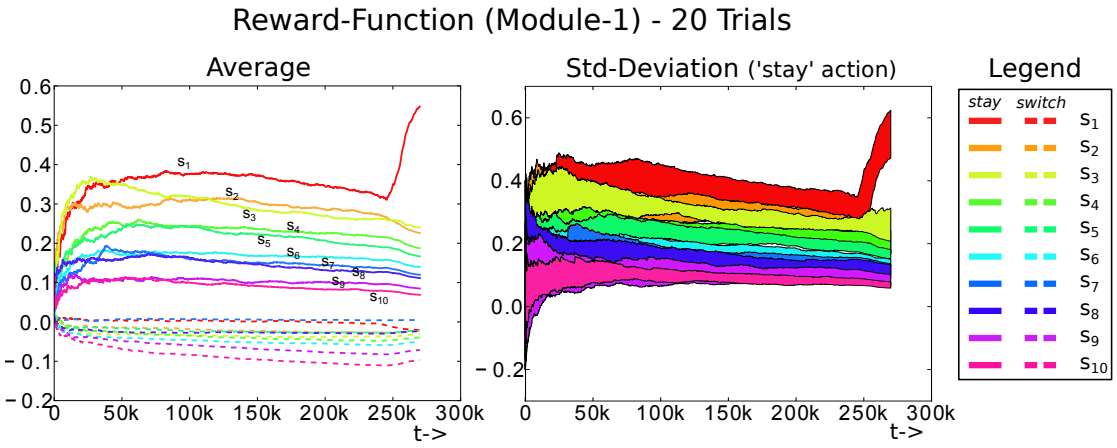

Figure 11: **10 Different Observation Streams**: See text for details. (Figures are best viewed in color)

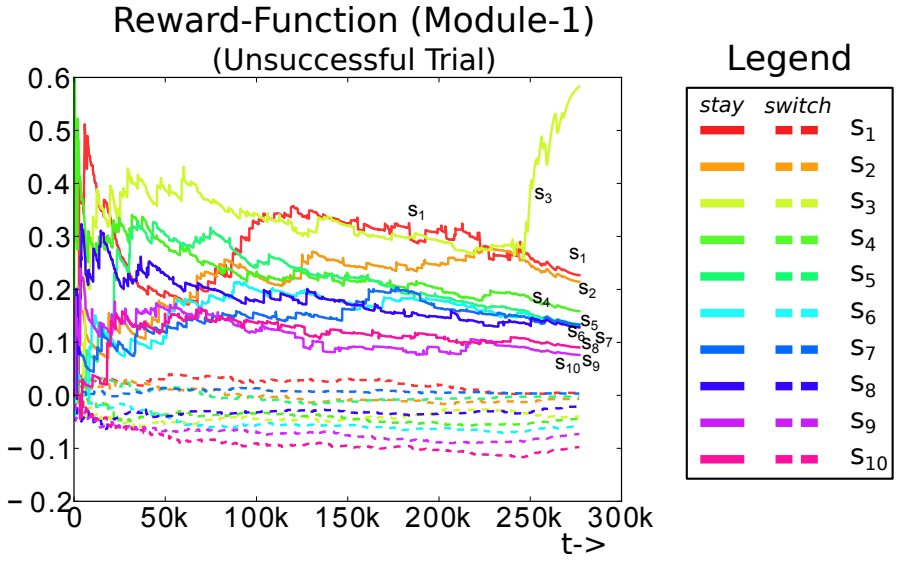

Figure 12: Reward function of the unsuccessful trial. (Figure is best viewed in color)

starting values to get a near-optimal performance of the algorithm. Next, we discuss an experiment conducted over a larger number of observation streams.

6.3 10 Different Observation Streams

The next experiment demonstrates that the algorithm scales well to a larger number (10) of different observation streams (similar to the ones in Experiment 1). These streams have the following increasing curiosity-function values ($\Omega_1 - \Omega_{10}$): (0.981140, 0.984279, 0.987169, 0.989791, 0.991922, 0.993411, 0.995511, 0.996260, 0.997685,

0.998256). Observation stream $\mathbf{x}_1$ is the easiest to learn compared to the other streams. We use parameters similar to the previous experiment, except for the decay multiplier which is set to 0.9999 for an increased exploration, and is reset to a value 0.99 when $\epsilon < 0.9$.

The experiment is conducted for 20 trails with different random seed initialization. In 19 out of the 20 trials, the algorithm successfully converged to the optimal solution. Figure 11 shows the average and standard-deviation of the reward function of module-1 for 20 trials. Clearly, the expected reward function stabilizes to a state such that: $R(s_1,$ *stay*$) > R(s_2,$ *stay*$) > R(s_3,$ *stay*$) > R(s_4,$ *stay*$) > R(s_5,$ *stay*$) > R(s_6,$ *stay*$) > R(s_7,$ *stay*$) > R(s_8,$ *stay*$) > R(s_9,$ *stay*$) > R(s_{10},$ *stay*$) > 0$.

However, for the one unsuccessful trial an abstraction corresponding to the observation stream $\mathbf{x}_3$ was learned as the first abstraction. Figure 12 shows the reward function for the unsuccessful trial. During exploration (higher values of ϵ, the reward function did not yet stabilize in the order of Ω values of the observation streams. Therefore, as the ϵ decreased to zero, the result converged to a suboptimal solution. The result can be improved by using a larger decay multiplier. In this experiment, we showed the result for only the first module since other modules follow a similar trend (if not better).

These experimental results demonstrate that the algorithm with the ϵ-greedy strategy learns the optimal solution discussed in Section 2. The method as presented above can be used in several online learning applications and is especially suited for acquisition of abstractions and skills on humanoid platforms [Luciw et al., 2013, Kompella et al., 2014]. We discuss next a few design modifications of the algorithm to extend its application to mobile robots in maze environments.

7 Extensions to Maze Environments

In this section we explore the application of Curious Dr. MISFA to environments such as a room-maze where each room has a time-varying audio or a video source (Figure 13). Therefore, each room represents a state s_i of Curious Dr. MISFA. In such cases, the environment's transition dynamics (see Section 3.2) are not similar to that of a complete-graph model, *i.e.*, the agent cannot *switch* between all the rooms without passing through the other rooms. We present here design modifications to the algorithm

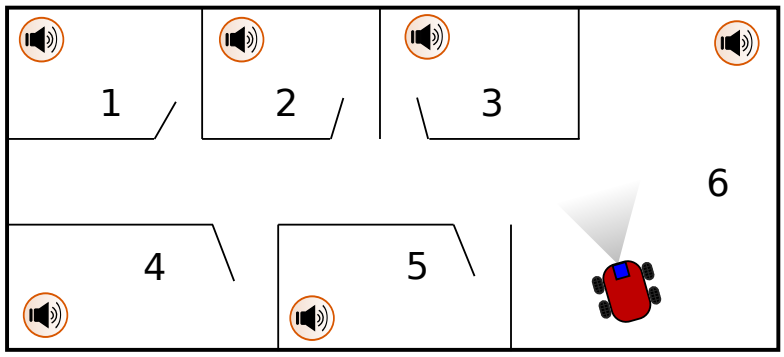

Figure 13: An example room maze scenario.

that enable the agent to take deterministic actions (unlike the stochastic *switch* action) to move to the state with the easiest encodable observation stream to learn an abstraction. It is hard to provide theoretical guarantees to these modifications because they depend on the unknown transition model of the maze. Instead, we present experimental results later in the section to demonstrate the algorithm's performance in such domains. The following are the design modifications of the algorithm for maze environments:

Environment's Transition Model: Curious Dr. MISFA uses a model-based LSPI algorithm to learn the observation stream selection policy (see Section 3.2). Therefore, the transition model for general maze environments needs to be learned *a priori*. It can be learned either by using lookup tables in deterministic environments or using Bayesian inference in stochastic environments. However, we provide the transition model *a priori* to the algorithm in the experiments discussed later in this section.

Observation Stream Selection Policy: A drawback of using an ϵ-greedy strategy over the *stay-switch* policy is that for a large number of observation streams in maze environments, it takes a considerable amount of time to get to a desired state. This can be improved by simultaneously learning another deterministic policy π^d defined over the same state space $\mathcal{S}$ but with a deterministic action-space ($\mathcal{A}^d = \{a_1^d, ..., a_n^d\}$). Let P^d denote the transition model of the internal environment for the action-space $\mathcal{A}^d$. When the agent shifts to a state s_i, $\forall i \in \{1, ..., n\}$, this implies that the agent took an action a_i^d, and vice-versa. The *switch* action stochastically selects an observation stream, while the action $a_i^d \in \mathcal{A}^d$ deterministically selects the observation stream $\mathbf{x}_i$. The agent therefore maintains a pair of value functions, one for the *stay-switch* action space $\mathcal{A}$ and the other for the deterministic action space $\mathcal{A}^d$. The agent chooses between

the two policies π and π^d probabilistically based on a decaying parameter ν (similar to the ϵ-greedy strategy).

Reward Function: For maze environments, the desired state may not be reachable in few time-steps. In such cases the discount-factor γ may lead to suboptimal behavior [Mahadevan, 1996]. To minimize this, average-reward reinforcement learning approaches [Mahadevan, 1996] can be used. We instead use a simple trick by modifying Eq. (14) as follows:

$$R_t(s) = \mathbb{1}_{\{s_l\}}(s),\ s_l = \arg\max_{s_i} \left(\widetilde{R}_t(s_i, 0, s_i)\right),\ s \in \mathcal{S} \tag{75}$$

Therefore, at any time t, the observation stream selection policy learned using this reward function is a shortest path to get to the state corresponding to the maximum reward.

Basis Functions: We use *linear function approximation* methods to find approximate value functions for large discrete maze environments. The value function is represented as a linear combination of basis functions. The selection of basis functions plays an important role in solving the problem. *Krylov Basis Functions* [KBFs; Petrik, 2007] are *reward-sensitive* basis-vectors $\mathcal{K}$, which are constructed by taking the product of reward function R with geometric powers of transition matrix P of a policy:

$$\mathcal{K} = \{R, PR, P^2R, ...\}. \tag{76}$$

Proto-Value Basis Functions [PVFs; Mahadevan and Maggioni, 2007] are however *reward-insensitive* basis-vectors, which are constructed by finding the eigenvectors of the symmetric graph Laplacian matrix based on the neighborhood relationships among the states. PVFs capture invariant subspaces (*bottlenecks*) of the model transition matrix. However, they lead to poor approximations when the reward function is spiky, because the basis vectors are smooth. While KBFs tend to work well for spiky reward functions, they require costly re-computations of the basis functions whenever the reward function changes. *Augmented Krylov Basis Functions* [AKBFs; Petrik, 2007] combines the methods to take advantage of both their approximation properties. This basis is constructed by augmenting a finite number of Krylov-basis and proto-value basis vectors, followed by an iterative orthogonalization using *Arnoldi iteration technique* [Arnoldi, 1951]. We use AKBFs for evaluating the *stay-switch* observation stream selection policy π_t and the deterministic observation stream selection policy π_t^d.

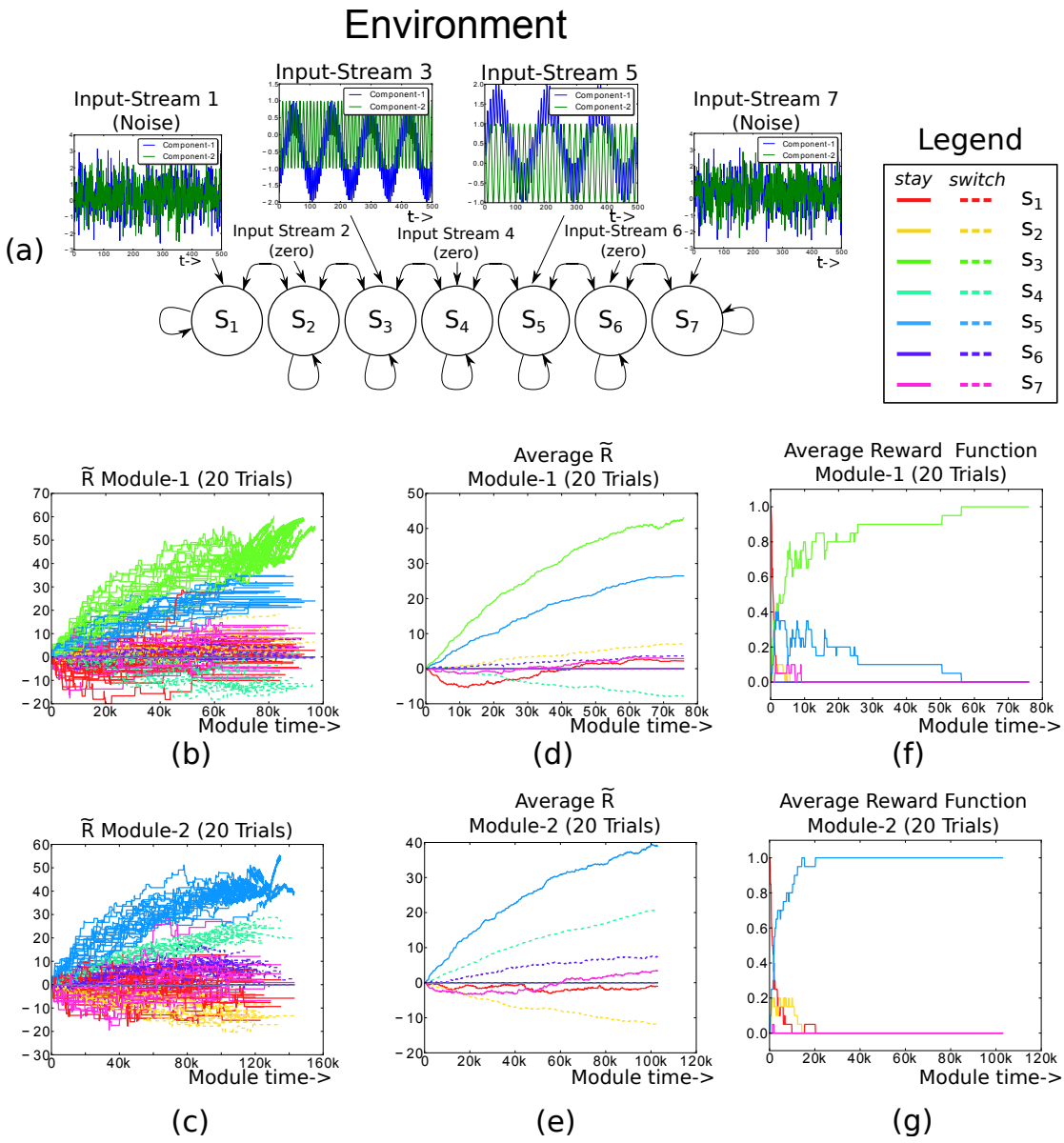

Figure 14: **Maze Environment with Noisy Streams**: See text for details. (Figures are best viewed in color)

These design modifications together enable Curious Dr. MISFA to be applied to maze environments with time-varying observation streams. Next, we present results of experiments conducted in such environments.

7.1 Maze Environment with Noisy Streams

Here, we test the design modifications discussed above on a bounded 1D-chain maze environment and in the presence of incompressible noisy streams as shown in Figure

14(a). Each state corresponds to a room with an audio source (see Figure 13). Each state has only two neighbors, except for the boundary states. Action $switch$ shifts the agent's state to one of its 2 neighboring states uniformly randomly. States s_1 and s_7 are associated with a white noise stream, and s_3 and s_5 are associated with two streams, shown in Eq. (72) and Eq. (73), respectively. The rest of the states have no observation streams (zero value). The 2D observation streams are expanded to 5 dimensions to handle non-linearity in the input. Based on the Ω values, observation stream $\mathbf{x}_1$ is the easiest stream to encode followed by $\mathbf{x}_2$ and then $\mathbf{x}_3$.

Experiment parameters: The values for τ, β, γ, α and the IncSFA-ROC parameters are set to the same values as in Experiment 6.1. σ is set to 5. Action-space $\mathcal{A}^d$ is equal to $\{a_1^d = -1(\text{Left}), a_2^d = 0(\text{Home}), a_3^d = 1(\text{Right})\}$. *Left* and *Right* actions shifts the agent's state (with probability 1) from s_i to s_{i-1} and s_{i+1} respectively, while *Home* action makes the agent to remain in the same state. The initial ϵ and ν values are set to $\epsilon = 1.1$ and $\nu = 1.2$, with a 0.999 decay multiplier for both. However, when $\epsilon < 0.9$, the decay multiplier is set to 0.992 to speed up the experiment. γ_d is set to 0.9.

The experiment is conducted for 20 trails with different random seed initializations. Figures 14(b)-(c) show plots of $\widetilde{R}_t$ (see Eq. 75) over time for each trial and Figures 14(d)-(e) show the average. The average $\widetilde{R}_t$ values corresponding to the noise are close to zero. Figures 14(f)-(g) show the average thresholded reward function over time (see Eq. 75). The algorithm successfully converges to the optimal solution in all the 20 trials avoiding the noisy streams. The two abstractions corresponding to the observation streams $\mathbf{x}_3$ and $\mathbf{x}_5$ are learned sequentially.

7.2 Large Maze Environment with Duplicated Streams

Here, we evaluate the algorithm on a larger maze environment as shown in the Figure 15(a). The environment has 100 grid points. Each grid point topologically represents a room (see Figure 13) with an arbitrarily associated audio stream such that, there are in total 10 grid points each of $\mathbf{x}_1$ (Eq. (72)), $\mathbf{x}_2$ (Eq. (73)) and a random stream. The remaining grid points are associated with an empty (zero) stream. The agent is unaware of the audio stream distribution and can traverse along the grid points to observe samples from the associated time-varying audio streams. The objective here is to learn an

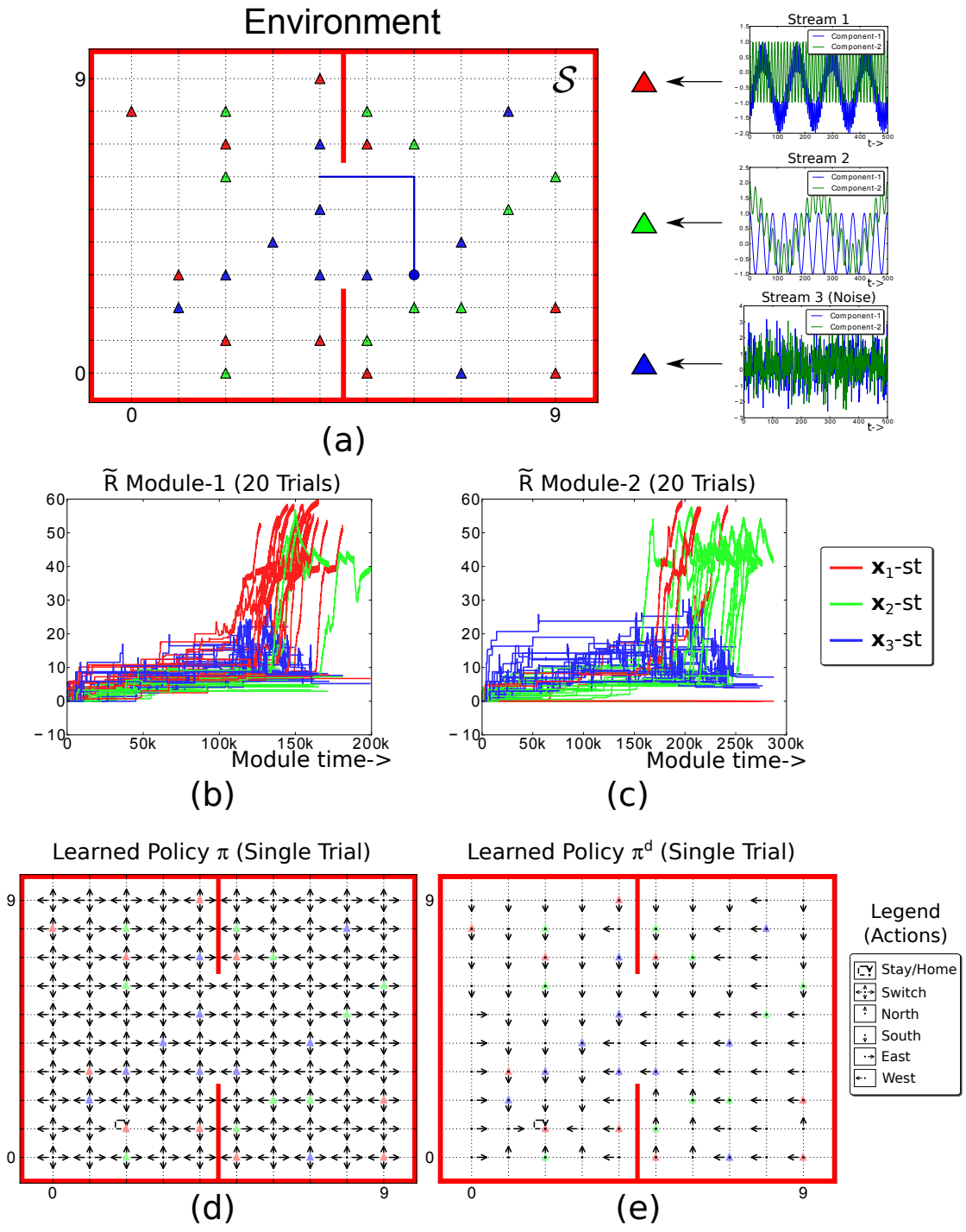

Figure 15: **Large Maze with Duplicated Streams**: See text for details. (Figures are best viewed in color)

abstraction corresponding to $\mathbf{x}_1$ first by moving into any of the grid points containing $\mathbf{x}_1$, followed by an abstraction for $\mathbf{x}_2$. Since there are in total 100 observation streams, Curious Dr. MISFA's environment has 100 states ($\mathcal{S} = \{s_1, ..., s_{100}\}$). Each state has

only four neighbors (except for the boundary states). Action $switch$ shifts the agent's state to one of its 4 neighboring states uniformly randomly.

Experiment parameters: Action-space $\mathcal{A}^d$ is $\{a_1^d = -2(\text{North}), a_2^d = -1(\text{South}), a_3^d = 0(\text{Home}), a_4^d = 1(\text{East}), a_5^d = 2(\text{West})\}$. IncSFA-ROC parameters are set to the same values as in Experiment 6.1. We use *Augmented Krylov* basis functions (see Section 7), with 10 Krylov bases, 30 LEM bases for computing π and 0 Krylov bases, 40 LEM bases for computing π^d. γ_d is set to 0.85. The values of the remaining algorithm parameters are set to the same values in the previous experiment (Section 7.1).

The agent begins by exploring the grid world using the policy described in Algorithm 2. When ϵ is close to 1, it executes the *stay-switch* actions uniformly randomly. The reward function is updated and the policies π and π^d begin to converge. The optimal stay-switch policy for learning module-1 returns the action *stay* in one or more of the states (grid-points) that contain the oscillatory signal $\mathbf{x}_1$ and *switch* in all the other states. As ϵ and ν decay, the agent begins to exploit the policy π initially and π^d later on. The converged policy π^d enables the agent to get to the optimal state quickly, which speeds up learning the corresponding abstraction. When $\epsilon \approx 0$, the agent stays in one of the grid points (that contains $\mathbf{x}_1$) and a corresponding abstraction is learned. The process repeats, the gating system prevents learning an abstraction corresponding to $\mathbf{x}_1$ again. Therefore, the states corresponding to $\mathbf{x}_2$ are most rewarding now and the agent learns a second abstraction corresponding to it.

The experiment is conducted for 20 trails with different random initializations. Figures 14(b)-(c) show the plot of $\widetilde{R}_t$ for each trial. Each red curve represents the maximum reward of all the 10 states associated with the audio stream $\mathbf{x}_1$ for each trial. While the green and the blue curves represent the same but for streams $\mathbf{x}_2$ and random stream respectively. In 16 out of the 20 trials, an abstraction corresponding to the audio stream $\mathbf{x}_1$ (Eq. (72)) is learned first followed by audio stream $\mathbf{x}_2$ (Eq. (73)). Figure 14(d)-(e) show the learned policies π and π^d for a single trial. The arrows in the figure indicate the actions $\mathcal{A} = \{$*stay*, *switch*$\}$ and the 5 deterministic actions $\mathcal{A}^d = \{a_1^d = -2(\text{North}), a_2^d = -1(\text{South}), a_3^d = 0(\text{Home}), a_4^d = 1(\text{East}), a_5^d = 2(\text{West})\}$.

This experiment demonstrates that Curious Dr. MISFA can successfully be applied to maze environments. The algorithm learns an abstraction while simultaneously developing a policy to get to the grid point with the easiest learnable observation stream.

8 Discussion and Conclusion

This section discusses the related research carried out by others, current limitations of the method and the future work that might address these limitations. To the best of our knowledge, the learning problem introduced in this paper has not been tackled with any of the current existing, practically implementable, intrinsically-motivated reinforcement learning methods, or anything else. We provided a theoretical analysis and an empirical evaluation to justify that the method achieves the desired optimal performance under the constraints mentioned in the paper. However, the following section compares structurally how our method differs from some of the relevant prior research work.

8.1 Related Work

Curious Dr. MISFA learns multiple *feature abstractions* from action sequences that are specific (but not limited) to a few localized parts of the environment. This is closely related to learning abstractions for *options*. The *options* framework (Sutton et al., 1999) formalizes planning over temporally extended courses of actions (*temporal abstractions*) via the semi-Markov Decision Process (MDP). Each option is applicable over a part of the environment, has its own subgoal(s), and has its own policy. Each option has a set of initiation states (from which the option can be started), a policy for action selection, and a termination probability upon each state. Konidaris et al. [2009, 2010] show how each option might be assigned with an abstraction from a library of many sensorimotor abstractions. The abstractions have typically been hand-designed and learning was assisted by human-demonstration. Without any external guidance, Curious Dr. MISFA autonomously builds a compact library of abstractions that can be used for options.

Mugan and Kuipers [2012] Qualitative Learner of Action and Perception system is designed to learn simplified predictable knowledge, potentially useful for learning behaviors from autonomous and/or curiosity-driven exploration (Mugan and Kuipers, 2012). It discretizes low-level sensorimotor experience through defining landmarks in the variables and observing contingencies between landmarks. It builds predictive models on this low-level experience, which it later uses to generate plans of action. It

either selects its actions randomly (early) or such that it expects to make fast progress in the performance of the predictive models (artificial curiosity). The sensory channels are preprocessed so that the input variables, for example, track the positions of the objects in the scene. A major difference between this system and ours is that Curious Dr. MISFA can potentially operate upon the raw pixels directly [Kompella et al., 2015], instead of assuming the existence of a low-level sensory model that can e.g., track the positions of the objects in the scene. Through IncSFA, features emerge from raw visual processing, and this feature development is tightly coupled with the curiosity-driven learning.

PowerPlay (Schmidhuber, 2013, Srivastava et al., 2013) 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. PowerPlay, unlike most online-learning algorithms has no problems with forgetting. Similar to PowerPlay, in Curious Dr. MISFA when a new representation is learned well enough to be internally predictable (low feature output estimator error), it is frozen and added to a long-term memory storage, and therefore already learned representations are not lost. However, neither PowerPlay (nor other intrinsically motivated reinforcement learning methods) have been applied to high-dimensional video data.

8.2 Limitations and Future Work

In the following, we will briefly list the current limitations of the Curious Dr. MISFA framework and insights for future work:

- **Raw information processing.** Curious Dr. MISFA is based on the IncSFA algorithm that updates slow feature abstractions online directly from raw-inputs (see Section 3.1), including high-dimensional image inputs [Kompella et al., 2015]. Slow features learned through IncSFA are linear. To extract higher non-linearities in the inputs, hierarchical extensions of IncSFA (H-IncSFA) over an *expanded input* in quadratic space [Luciw et al., 2012, Wiskott and Sejnowski, 2002] or the recently proposed Deeply-Learned SFA [DL-SFA; Sun et al., 2014] may be used. DL-SFA adopts the notion of 3D convolution and max-pooling to capture abstract, structural and translational invariant features. As future work, we plan to

combine such non-linear hierarchical structures to improve the quality of the slow feature abstractions learned.

- **Continuous model.** Curious Dr. MISFA uses a ROC clustering algorithm that learns a discrete model mapping the adaptive slow feature outputs with respect to the user-signal observations $\mathbf{u}(t)$ (see Section 3.1). We plan to use a continuous state predictor to avoid discretization of the slow feature outputs. This continuous predictive model can help a subsequent reinforcement learner to quickly learn continuous policies.

- **Sensor fusion.** And finally, we have only applied Curious Dr. MISFA on either oscillatory streams or high-dimensional visual inputs from the onboard cameras of an iCub humanoid robot [Luciw et al., 2013, Kompella et al., 2014, 2015]. As our future work, we plan to build slow features abstractions by using different sensory modalities such as tactile and audio in addition to the visual inputs. This should be straightforward addition to Curious Dr. MISFA, since IncSFA is agnostic to the modality of the sensory information. The raw inputs of different modalities can be concatenated as a single input and fed to the IncSFA algorithm, without causing too much computational overhead (since IncSFA has a linear update complexity [Kompella et al., 2012a]). Related work on combining sensory modalities using SFA methods have shown to achieve good results [Höfer et al., 2012].

8.3 Conclusion

We have presented an autonomous curiosity-driven modular incremental slow feature learning algorithm that learns invariant slow feature abstractions from multiple time-varying input observation streams, sequentially, in the order of increasing learning difficulty. The method continually estimates the initially unknown learning difficulty through intrinsic rewards generated by exploring the observation streams using a *stay-switch* action selection mechanism. The architecture of the method includes (a) a reinforcement-learner that generates policies to select an input stream based on the intrinsic rewards, (b) an adaptive IncSFA-ROC module that updates an abstraction based on the incoming observations, and (c) a gating-system that prevents encoding inputs

that have been previously encoded. We formalized the learning problem as an optimization problem and presented a formal analysis to prove that the Curious Dr. MISFA algorithm converges to the optimal-solution under a few mild conditions. Experimental results show that the method successfully learns abstractions in the order of increasing learning difficulty, for a variety of experimental settings.

With the growing success of the Slow Feature Analysis (SFA) among many problems and scenarios, this modular incremental version of SFA contributes to the field of artificial intelligence by enabling skill development in *curiosity*-driven agents. In other works (Kompella et al., 2012b, Luciw et al., 2013, Kompella et al., 2015) we have shown results on how Curious Dr. MISFA enables an iCub robot to carry out real-time intrinsically motivated interactions with the environment to uncover slow features from the raw video inputs. In future work, Curious Dr. MISFA could be extended to enable the iCub to learn increasingly complex slow-feature representations in more general environments.

Acknowledgments

We acknowledge Sohrob Kazerounian, Faustino Gomez and Alan Lockett's assistance in revising this paper. This work was funded through SNF grant #138219 (Theory and Practice of Reinforcement Learning II) and through the 7th framework program of the EU under grant #270247 (NeuralDynamics project).

References

H. Abut, editor. *Vector Quantization*. IEEE Press, Piscataway, NJ, 1990.

W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Q. Appl. Math*, 9(17):17–29, 1951.

H. Barlow. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12(3):241–253, 2001.

T. Chen, S. I. Amari, and N. Murata. Sequential extraction of minor components. *Neural Processing Letters*, 13(3):195–201, 2001. ISSN 1370-4621.

P. Comon. Independent component analysis, A new concept? *Signal Processing*, 36: 287–314, 1994.

P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

M. Franzius, H. Sprekeler, and L. Wiskott. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology*, 3(8):e166, 2007.

Z. Gábor, Z. Kalmár, and C. Szepesvári. Multi-criteria reinforcement learning. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 197–205. Morgan Kaufmann Publishers Inc., 1998.

L. Gisslén, M. Luciw, V. Graziano, and J. Schmidhuber. Sequential constant size compressors for reinforcement learning. In *Artificial General Intelligence*, pages 31–40. Springer, 2011.

I. D. Guedalia, M. London, and M. Werman. An on-line agglomerative clustering method for nonstationary data. *Neural Computation*, 11(2):521–540, 1999.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, August 2002.

S. Höfer, M. Spranger, and M. Hild. Posture recognition based on slow feature analysis. In *Language Grounding in Robots*, pages 111–130. Springer, 2012.

O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to isomap nonlinear dimension reduction. In *Proceedings of the twenty-first international conference on Machine learning*, page 56. ACM, 2004.

I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

M. Klapper-Rybicka, N. N. Schraudolph, and J. Schmidhuber. Unsupervised learning in lstm recurrent neural networks. In *Lecture Notes on Comp. Sci. 2130, Proc. Intl. Conf. on Artificial Neural Networks (ICANN-2001)*, pages 684–691. Springer: Berlin, Heidelberg, 2001.

T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 3rd edition, 2001.

V. R. Kompella, M. Luciw, and J. Schmidhuber. Incremental slow feature analysis. In *Proc. 20th International Joint Conference of Artificial Intelligence (IJCAI)*, pages 1354–1359, 2011a.

V. R. Kompella, L. Pape, J. Masci, M. Frank, and J. Schmidhuber. Autoincsfa and vision-based developmental learning for humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, pages 622–629, Bled, Slovenia, 2011b.

V. R. Kompella, M. Luciw, and J. Schmidhuber. Incremental slow feature analysis: Adaptive low-complexity slow feature updating from high-dimensional input streams. *Neural Computation*, 24(11):2994–3024, 2012a.

V. R. Kompella, M. Luciw, M. Stollenga, L. Pape, and J. Schmidhuber. Autonomous learning of abstractions using curiosity-driven modular incremental slow feature analysis. In *Proc. of the Joint Conference on Development and Learning and Epigenetic Robotics (ICDL-EPIROB)*, pages 1–8, San Diego, 2012b. IEEE.

V. R. Kompella, M. Stollenga, M. Luciw, and J. Schmidhuber. Explore to see, learn to perceive, get the actions for free: Skillability. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2705–2712. IEEE, 2014.

V. R. Kompella, M. Stollenga, M. Luciw, and J. Schmidhuber. Continual curiosity-driven skill acquisition from high-dimensional video inputs for humanoid robots. *Artificial Intelligence*, 2015.

G. Konidaris and A. G. Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems*, pages 1015–1023, 2009.

G. Konidaris, S. Kuindersma, A. G. Barto, and R. Grupen. Constructing skill trees for reinforcement learning agents from demonstration trajectories. In *Advances in Neural Information Processing Systems*, pages 1162–1170, 2010.

M. G. Lagoudakis and R. Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

H. Lee, Y. Largman, P. Pham, and A.Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104.

R. Legenstein, N. Wilbert, and L. Wiskott. Reinforcement learning on slow features of high-dimensional input streams. *PLoS Computational Biology*, 6(8), 2010. ISSN 1553-734X.

S. Lindstädt. Comparison of two unsupervised neural network models for redundancy reduction. In M. C. Mozer, P. Smolensky, D. S. Touretzky, J. L. Elman, and A. S. Weigend, editors, *Proc. of the 1993 Connectionist Models Summer School*, pages 308–315. Hillsdale, NJ: Erlbaum Associates, 1993.

M. Luciw, V. R. Kompella, and J. Schmidhuber. Hierarchical incremental slow feature analysis. In *Workshop on Deep Hierarchies in Vision*, Vienna, 2012.

M. Luciw, V. R. Kompella, S. Kazerounian, and J. Schmidhuber. An intrinsic value system for developing multiple invariant representations with incremental slowness learning. *Frontiers in Neurorobotics*, 7, 2013.

S. Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, 22(1-3):159–195, 1996.

S. Mahadevan and M. Maggioni. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *Journal of Machine Learning Research*, 8(2169-2231):16, 2007.

G. Mitchison. Removing time variation with the anti-hebbian differential synapse. *Neural Computation*, 3(3):312–320, 1991.

J. Mugan and B. Kuipers. Autonomous learning of high-level states and actions in continuous environments. *IEEE Transactions on Autonomous Mental Development*, 4(1):70–86, 2012.

E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5(6):927–935, 1992. ISSN 0893-6080.

L. Pape, C. M. Oddo, M. Controzzi, C. Cipriani, A. Förster, M. C. Carrozza, and J. Schmidhuber. Learning tactile skills through curious exploration. *Frontiers in neurorobotics*, 6, 2012.

D. Peng and Z. Yi. A new algorithm for sequential minor component analysis. *International Journal of Computational Intelligence Research*, 2(2):207–215, 2006.

D. Peng, Z. Yi, and W. Luo. Convergence analysis of a simple minor component analysis algorithm. *Neural Networks*, 20(7):842–850, 2007. ISSN 0893-6080.

M. Petrik. An analysis of laplacian methods for value function approximation in mdps. In *IJCAI*, pages 2574–2579, 2007.

M. B. Ring. *Continual Learning in Reinforcement Environments*. PhD thesis, University of Texas at Austin, 1994.

J. Schmidhuber. Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE press, 1991.

J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992a.

J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992b.

J. Schmidhuber. Learning unambiguous reduced sequence descriptions. In J. E. Moody, S. J. Hanson, and R. P. Lippman, editors, *Advances in Neural Information Processing Systems 4 (NIPS 4)*, pages 291–298. Morgan Kaufmann, 1992c.

J. Schmidhuber. Artificial curiosity based on discovering novel algorithmic predictability through coevolution. In *Congress on Evolutionary Computation (CEC)*, pages 1612–1618. IEEE Press, 1999a.

J. Schmidhuber. Neural predictors for detecting and removing redundant information. In H. Cruse, J. Dean, and H. Ritter, editors, *Adaptive Behavior and Learning*. Kluwer, 1999b.

J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006a.

J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006b.

J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230 –247, 2010a. ISSN 1943-0604. doi: 10.1109/TAMD.2010.2056368.

J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010b.

J. Schmidhuber. Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in psychology*, 4, 2013.

H. Sprekeler. On the relation of slow feature analysis and laplacian eigenmaps. *Neural Computation*, 23(12):3287–3302, 2011.

H. Sprekeler, T. Zito, and L. Wiskott. An extension of slow feature analysis for nonlinear blind source separation. *Journal of Machine Learning Research*, 15:921–947, 2014.

R. K. Srivastava, B. R. Steunebrink, and J. Schmidhuber. First Experiments with POWERPLAY. *Neural Networks*, 2013. ISSN 0893-6080. doi: 10.1016/j.neunet.2013.01.022.

J. Storck, S. Hochreiter, and J. Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks, Paris*, volume 2, pages 159–164. EC2 & Cie, 1995.

L. Sun, K. Jia, T. H. Chan, Y. Fang, G. Wang, and S. Yan. Dl-sfa: Deeply-learned slow feature analysis for action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2625–2632. IEEE, 2014.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. Cambridge, MA, MIT Press, 1998.

R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1): 181–211, 1999.

C. Theriault, N. Thome, and M. Cord. Dynamic scene classification: Learning motion descriptors with slow features analysis. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2603–2610. IEEE, 2013.

P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning*, 84 (1):51–80, 2011.

G. Wallis and E.T. Rolls. Invariant face and object recognition in the visual system. *Progress in Neurobiology*, 51(2):167–194, 1997.

J. Weng, Y. Zhang, and W. Hwang. Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):1034–1040, 2003.

L. Wiskott. Estimating driving forces of nonstationary time series with slow feature analysis. *arXiv preprint cond-mat/0312317*, 2003.

L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

D. Zhang, D. Zhang, S. Chen, K. Tan, and K. Tan. Improving the robustness of online agglomerative clustering method based on kernel-induce distance measures. *Neural processing letters*, 21(1):45–51, 2005.

Y. Zhang and J. Weng. Convergence analysis of complementary candid incremental principal component analysis. *Michigan State University*, 2001.

Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):436–450, 2012.