Slow Feature Analysis For Curiosity-Driven Agents

Varun Raj Kompella

(www.idsia.ch/~kompella)

Co-organizers: Matthew Luciw and Juergen Schmidhuber

The Swiss AI Lab IDSIA, Switzerland

Lugano, Switzerland

http://www.atlanticolugano.ch/images/lugano05.jpg

Co-Organizers
Dr. Matthew D. Luciw

- Webpage: www.idsia.ch/~luciw
- Currently working at Neurala, Inc, Boston US.
- Offshoot of Boston University neuromorphic research
- Working with NASA, USAF and private customers to design deep learning autonomy algorithms for ground/air robots.
- www.neurala.com

neurala
brains for bots

NASA

Co-Organizers
Prof. Juergen Schmidhuber

- Director of the Swiss AI Lab IDSIA, Lugano, Switzerland.
- Professor of AI at the Faculty of Computer Science at the University of Lugano (USI) and SUPSI.
- He is known for his work on machine learning, artificial intelligence, artificial neural networks, digital physics and low-complexity art.
- Webpage: www.idsia.ch/~juergen

USI/SUPSI
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale
IDSIA

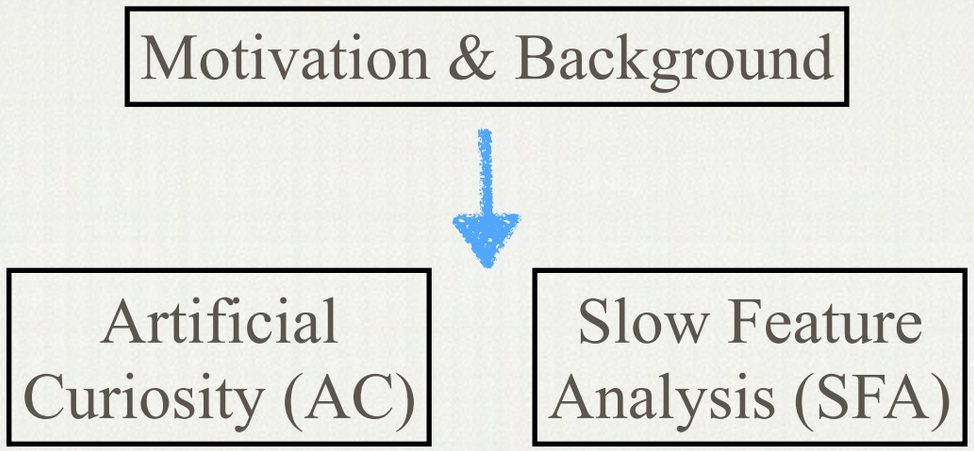
Tutorial Webpage

- http://www.idsia.ch/~kompella/mywork/WCCI_Tutorial.html
- Related material: Slides, Papers, Videos, Codes, Online-browser implementations.
- Will be constantly updated for new material.

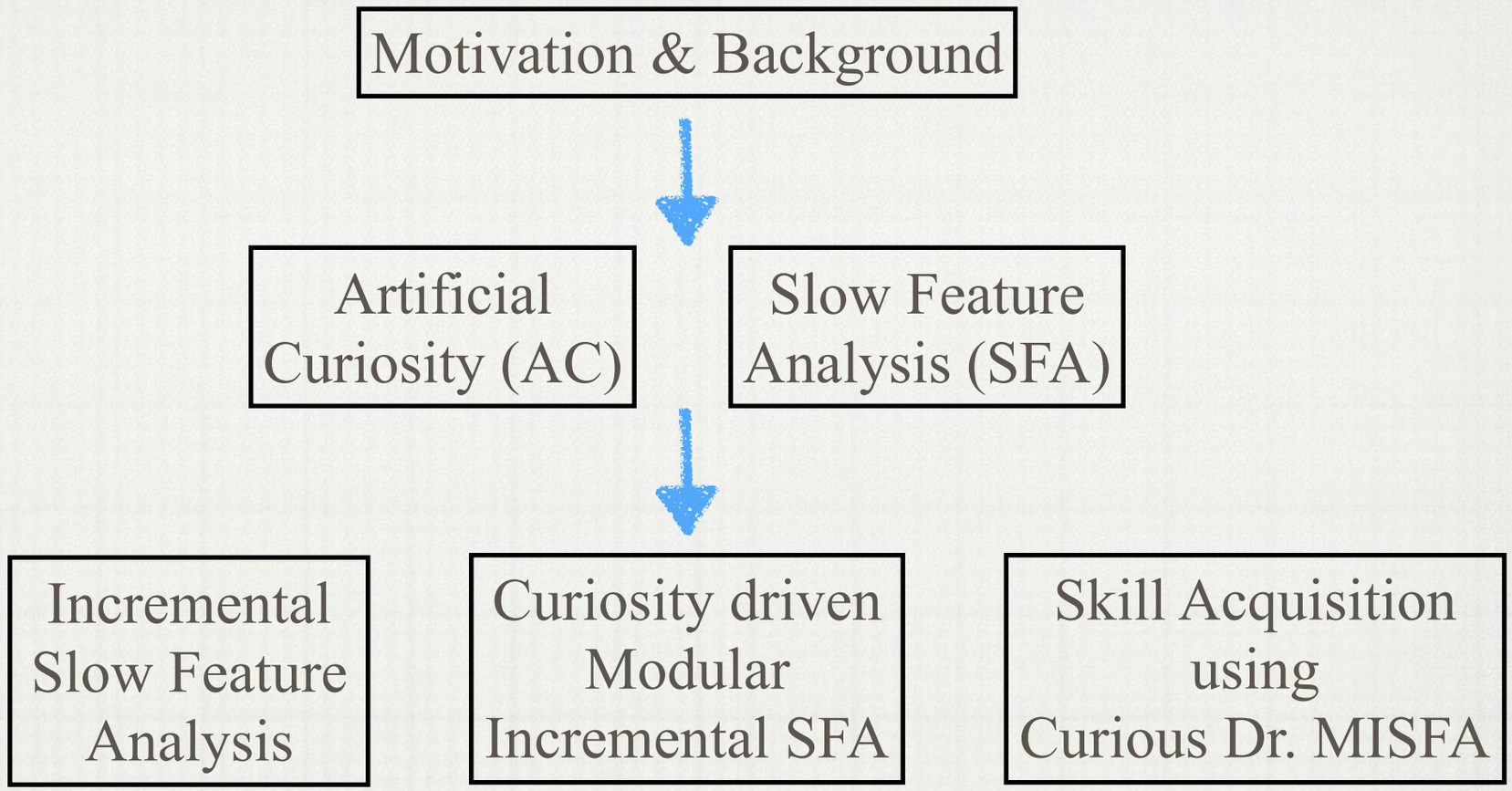
Tutorial Outline

Motivation & Background

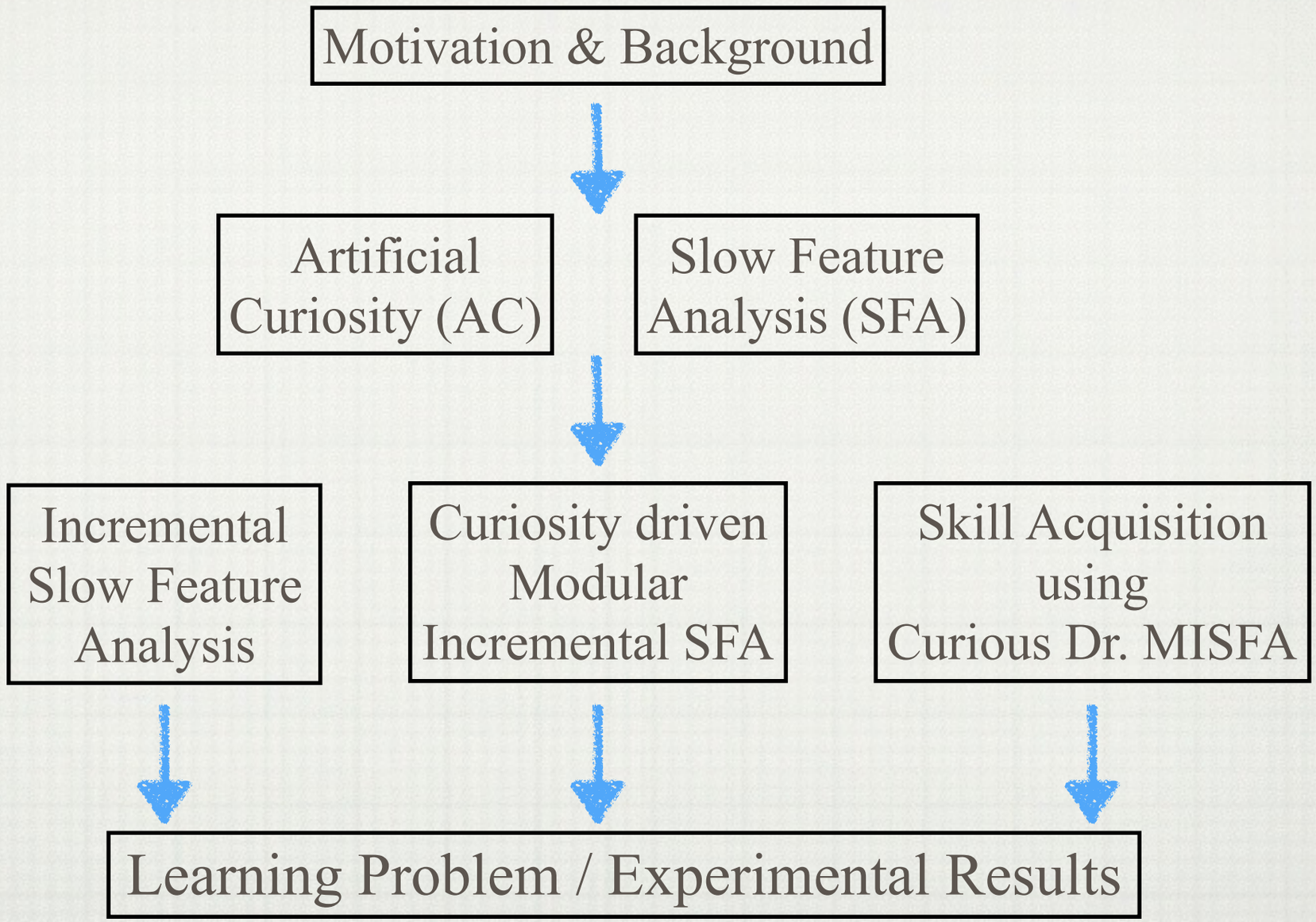
Tutorial Outline



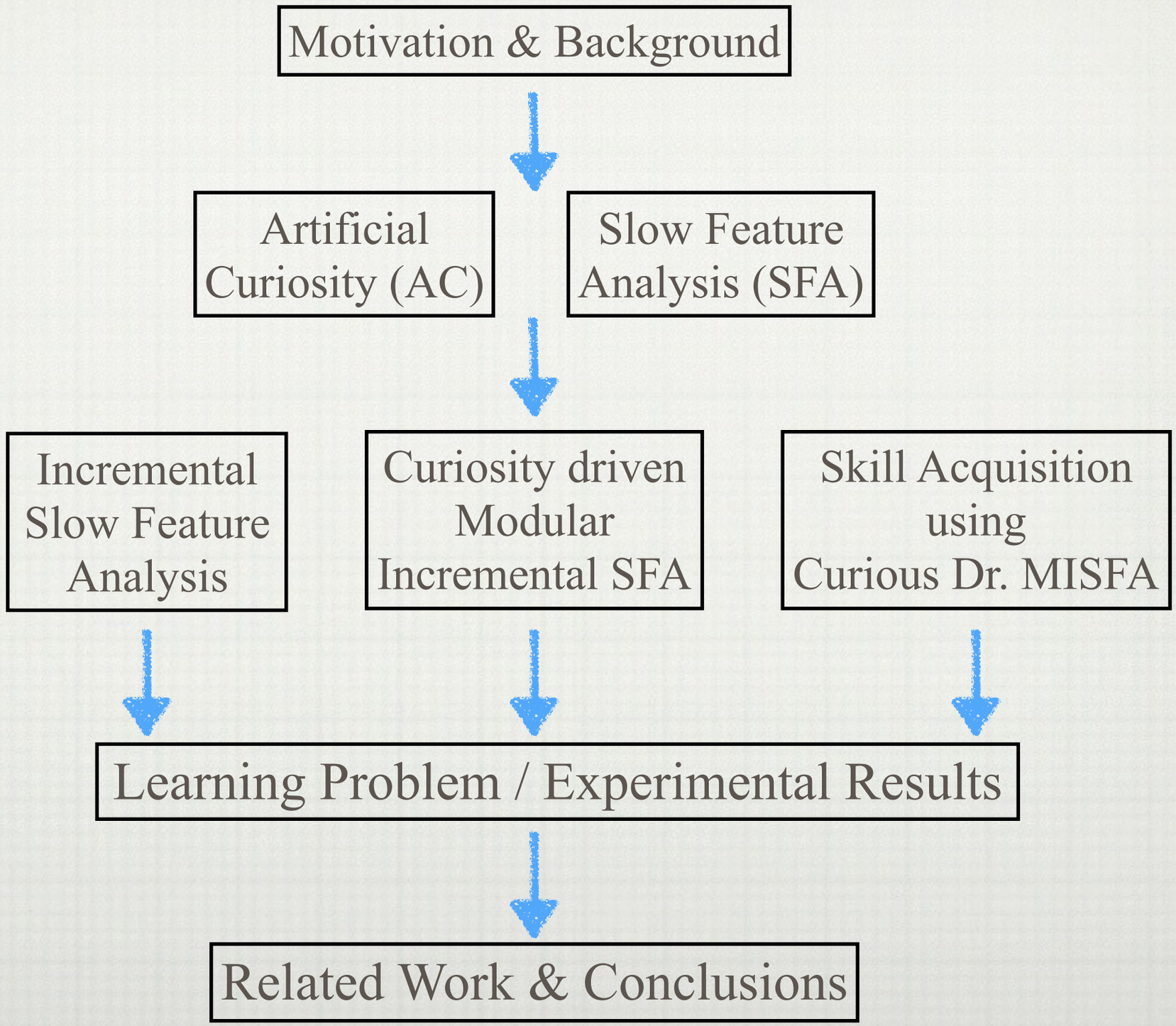
Tutorial Outline



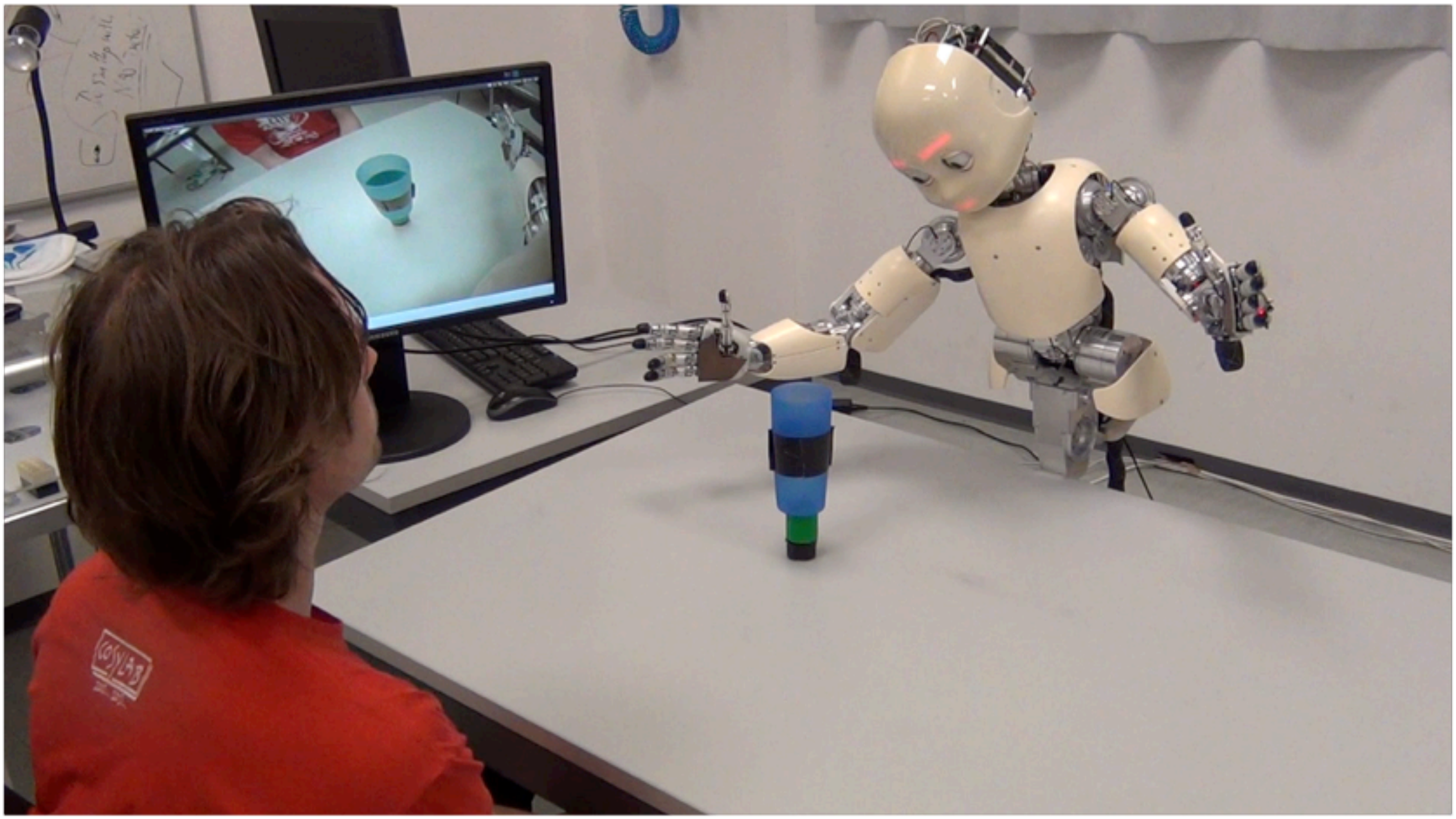
Tutorial Outline



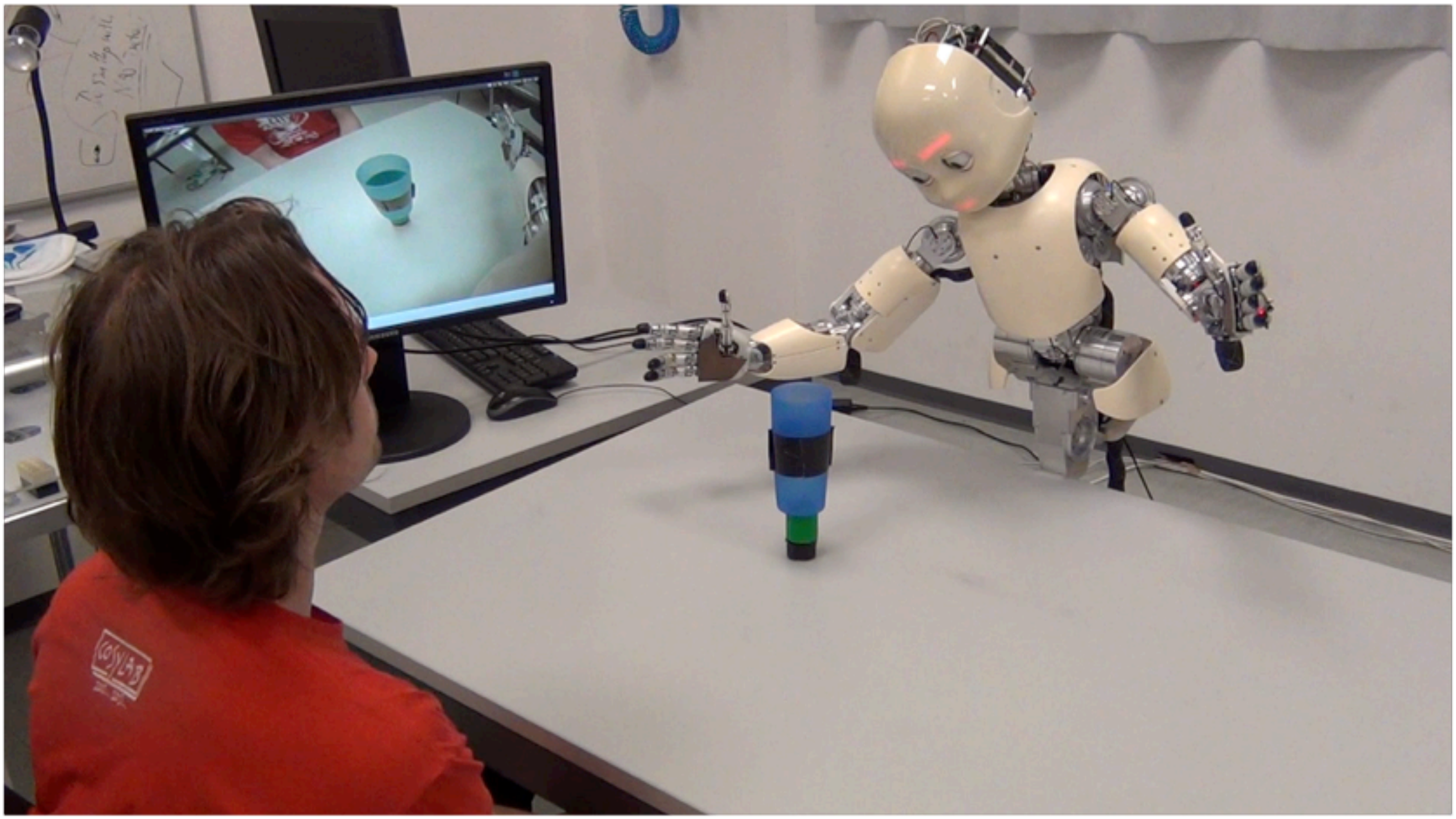
Tutorial Outline



Sneak Preview (What to Expect)

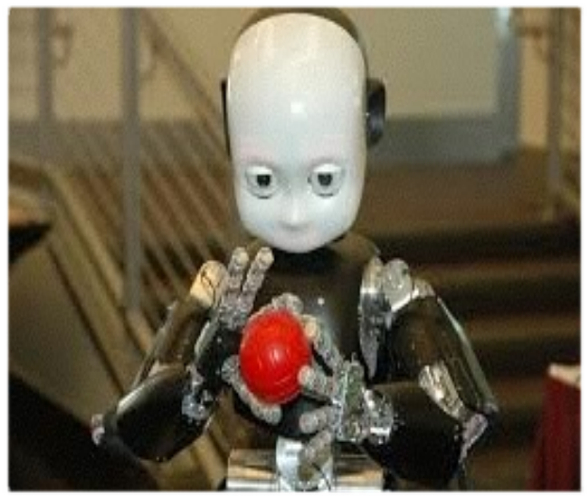


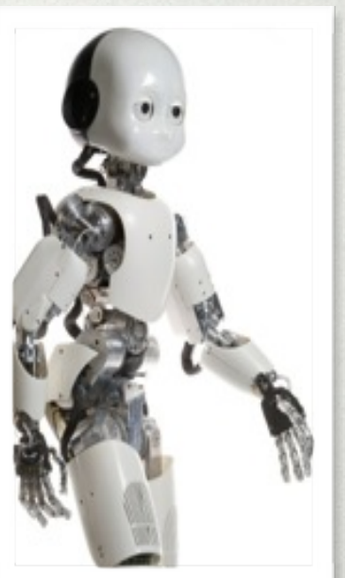
Sneak Preview (What to Expect)



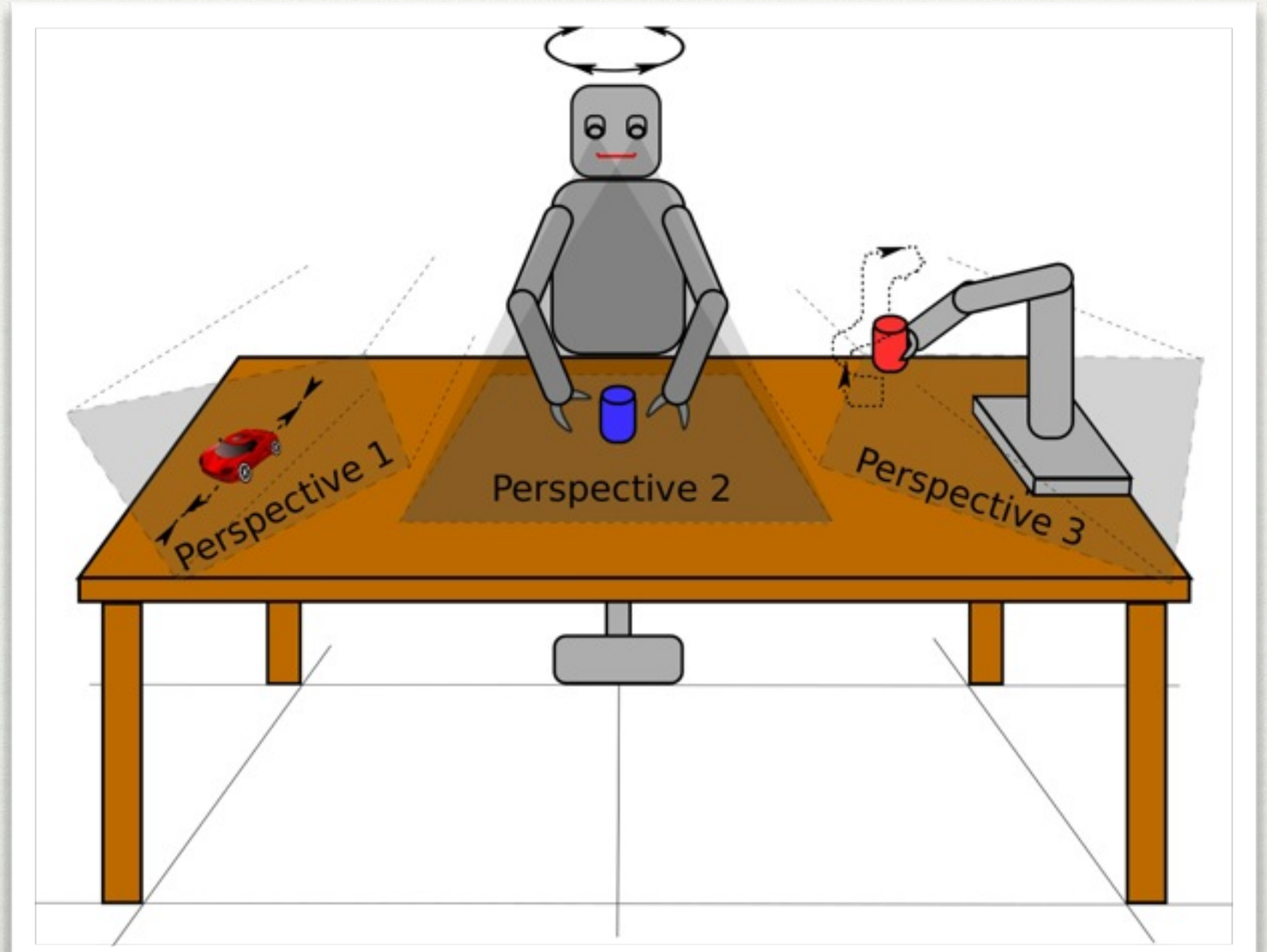
Anthropomorphic Robots

(ASIMO, PETMAN, iCUB)



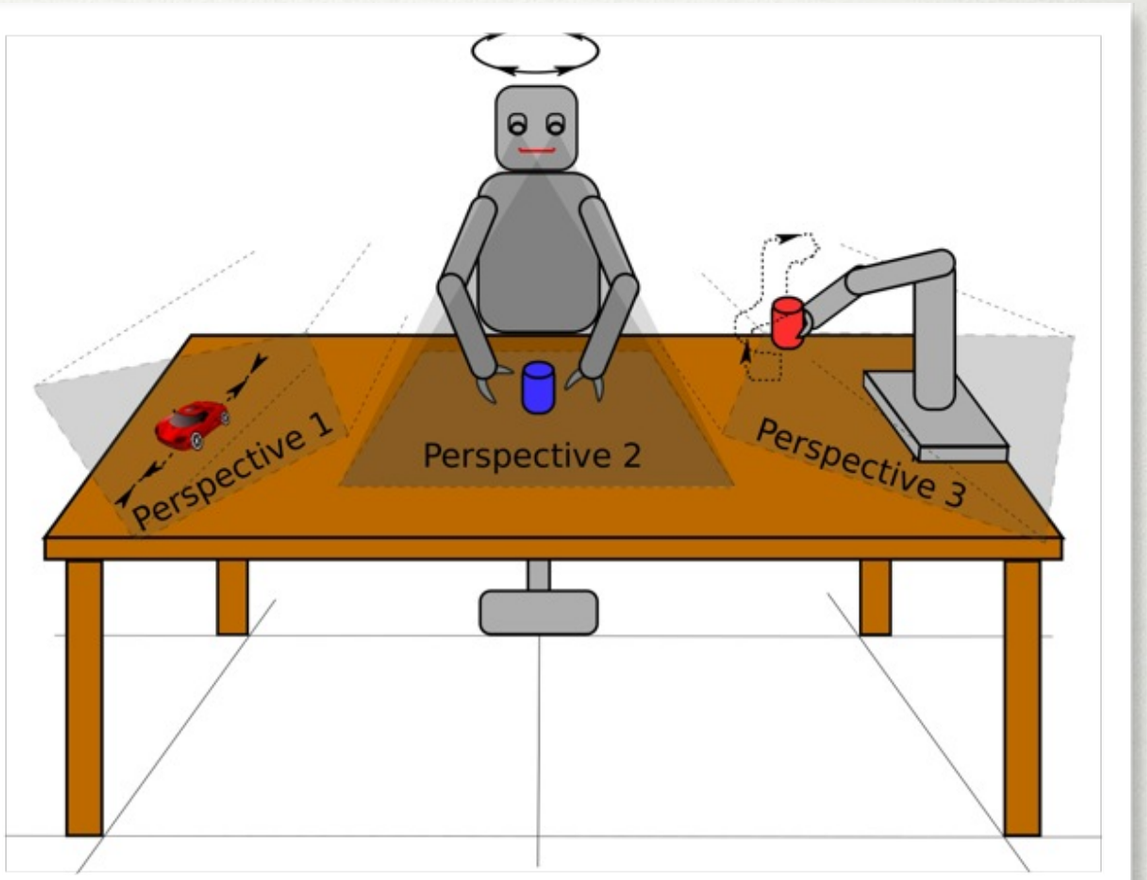


Learning Behaviors from High-Dimensional Sensory Inputs?



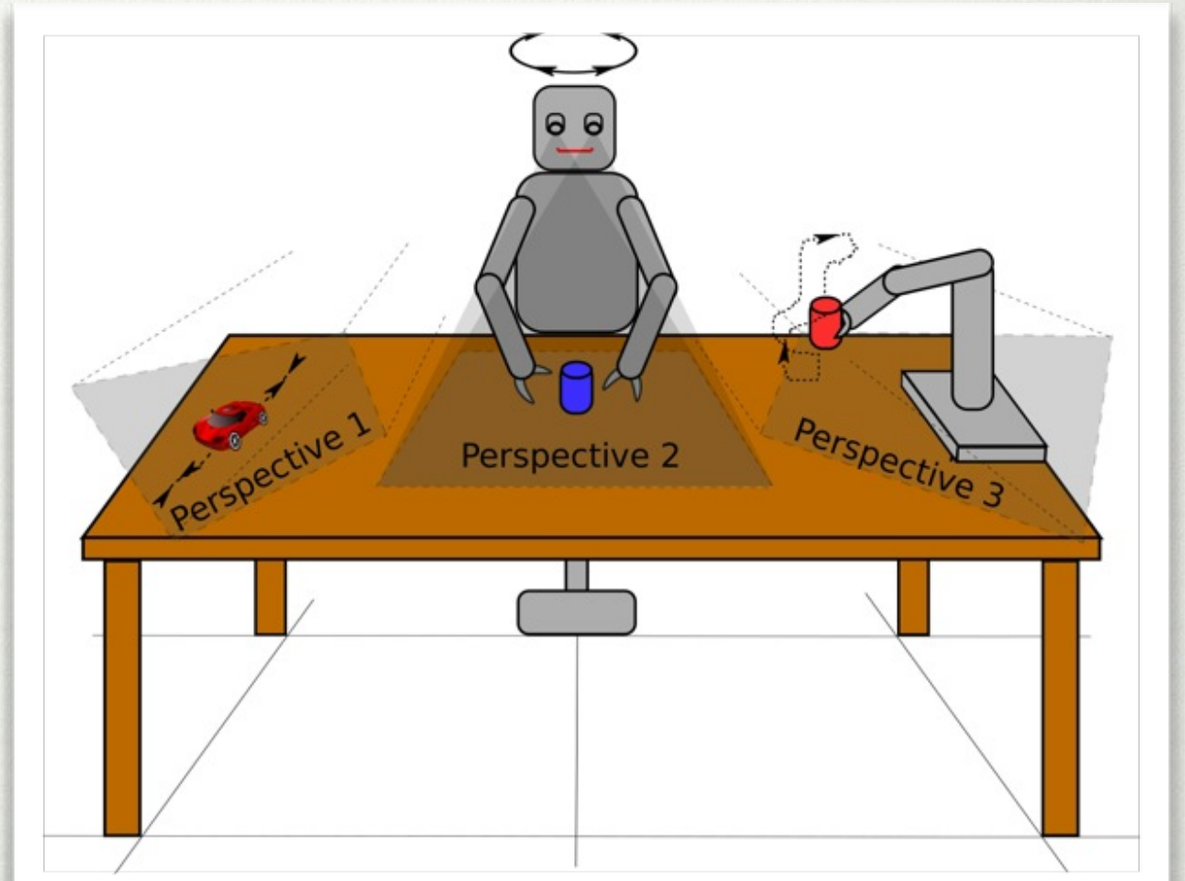
Learning Behaviors from High-Dimensional Sensory Inputs?

- Reinforcement Learning - Task Specific Behaviors via trial-error interactions



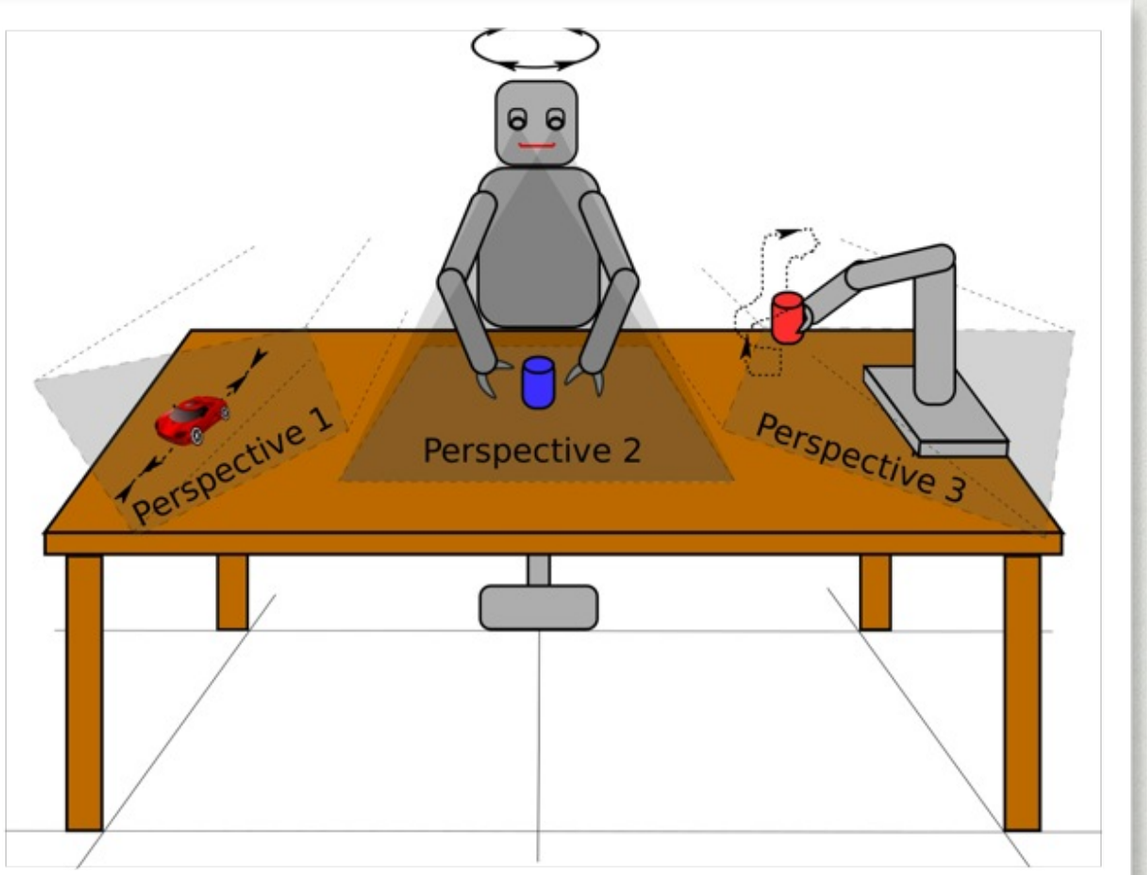
Learning Behaviors from High-Dimensional Sensory Inputs?

- Reinforcement Learning - Task Specific Behaviors via trial-error interactions
- Difficulty: Pure random-exploration in large sensory and joint state spaces.
  - eg: (320 x 240) x (~50) continuous state variables!

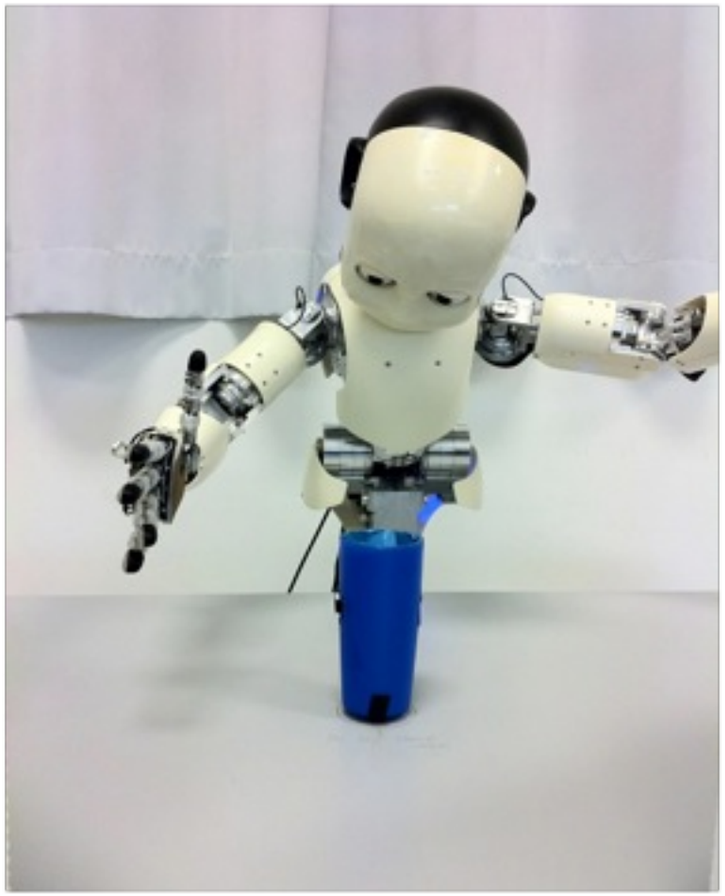


Learning Behaviors from High-Dimensional Sensory Inputs?

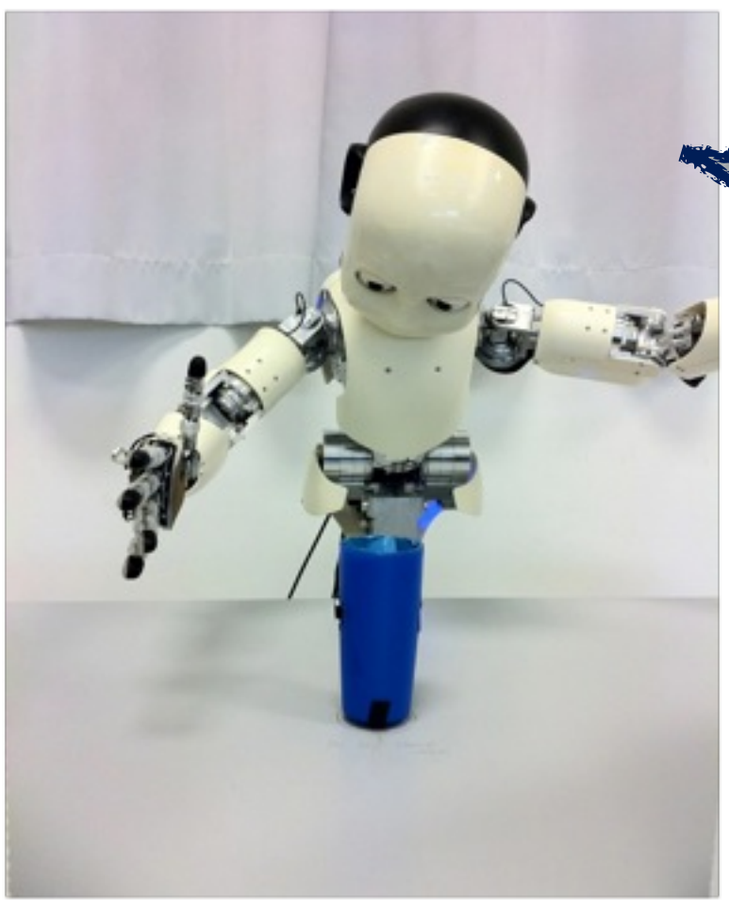
- Reinforcement Learning - Task Specific Behaviors via trial-error interactions
- Difficulty: Pure random-exploration in large sensory and joint state spaces.
  - eg: (320 x 240) x (~50) continuous state variables!
- Self supervised learning



Challenge in AI?

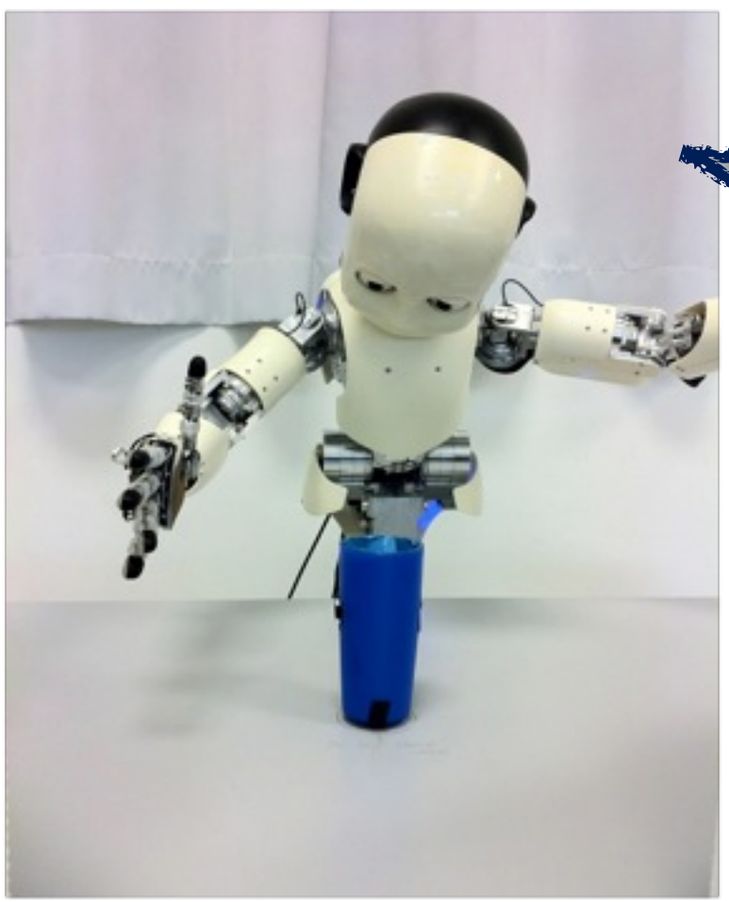


Challenge in AI?



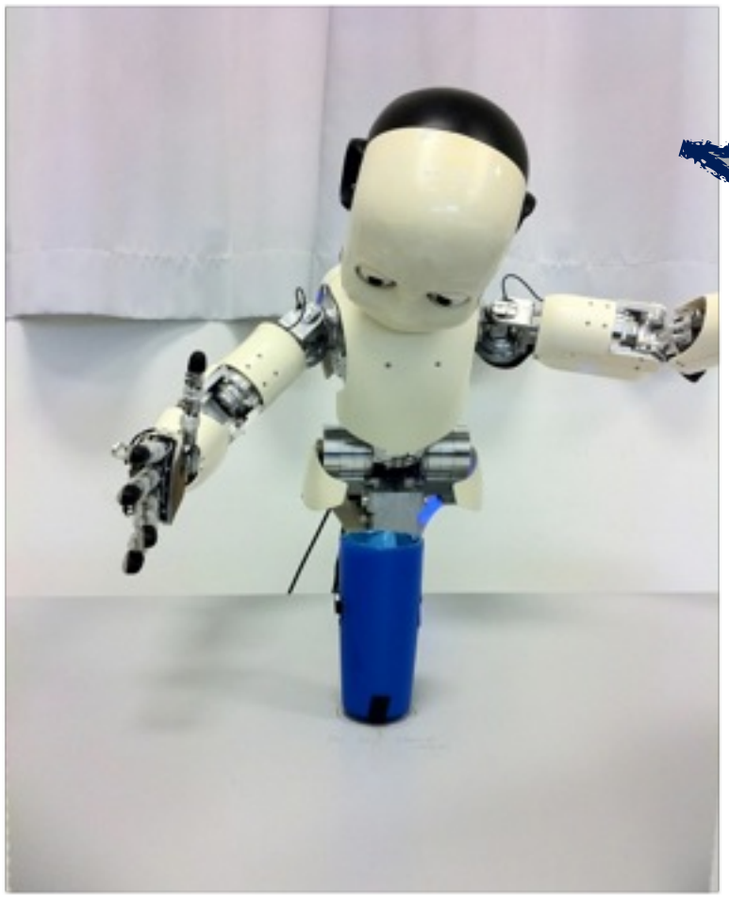
Wait…. Is this Interesting ??

Challenge in AI?



= What is this ??

Challenge in AI?



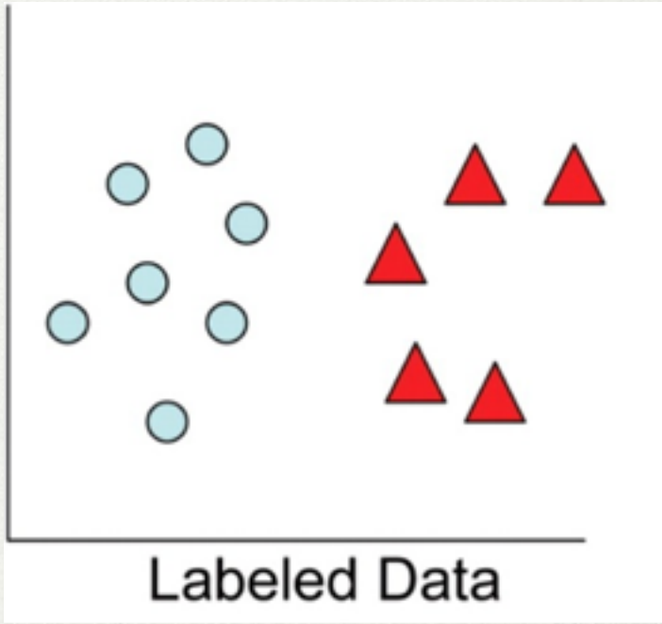
How can I push the cup?
grasp the cup?
•
•
•

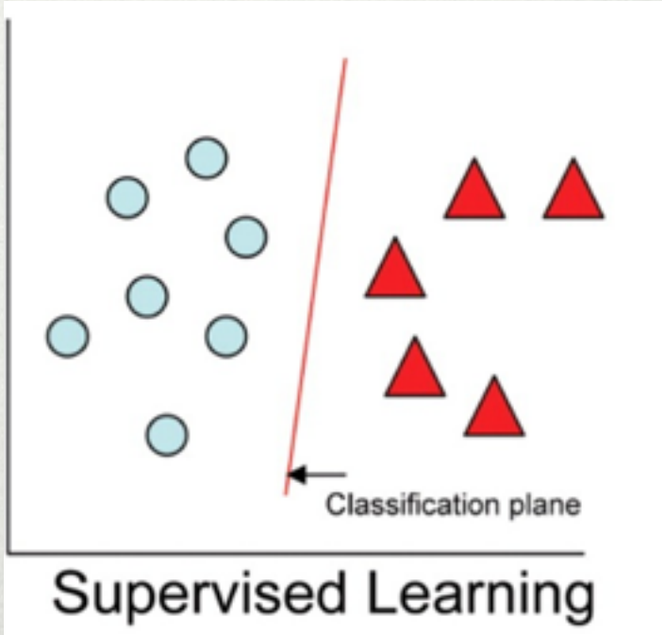
Supervised Learning (SL)

Training examples: (x_i, y_i), $x_i \sim D(x)$, $y_i = f(x_i)$. x-samples, y-labels.

Learning Objective: Improve a hypothesis h(x) by minimising a loss function L(y, h(x)).

Examples: Artificial Neural Networks, Support Vector Machine



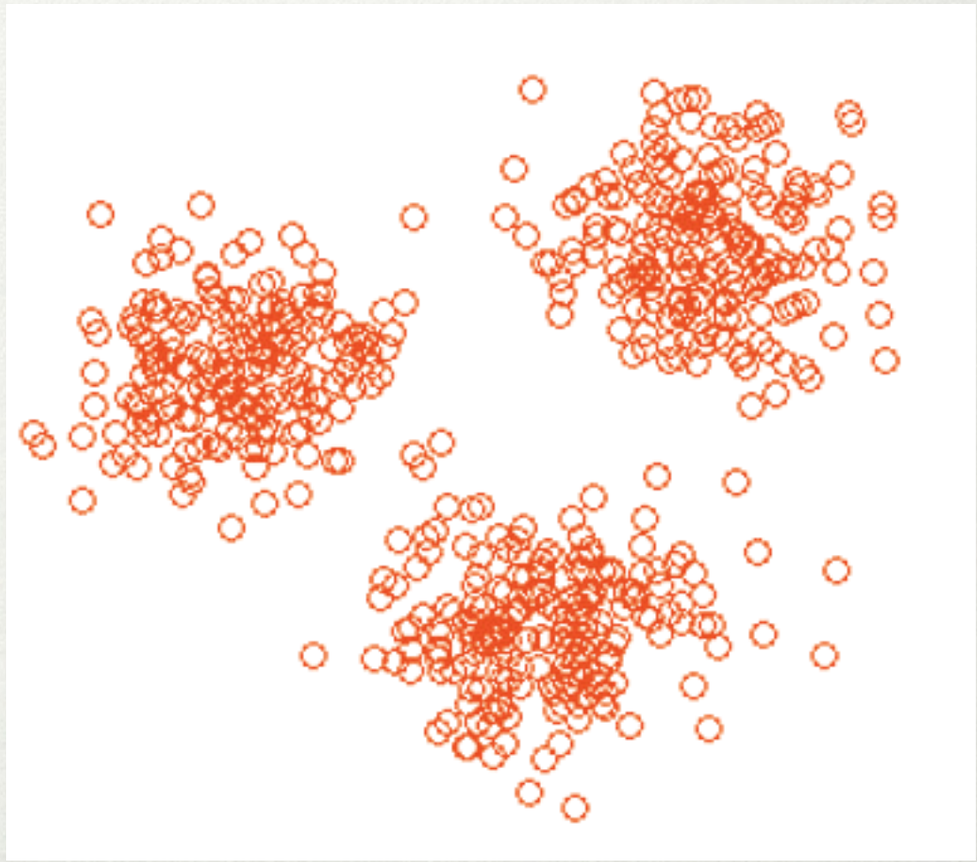


Unsupervised Learning (UL)

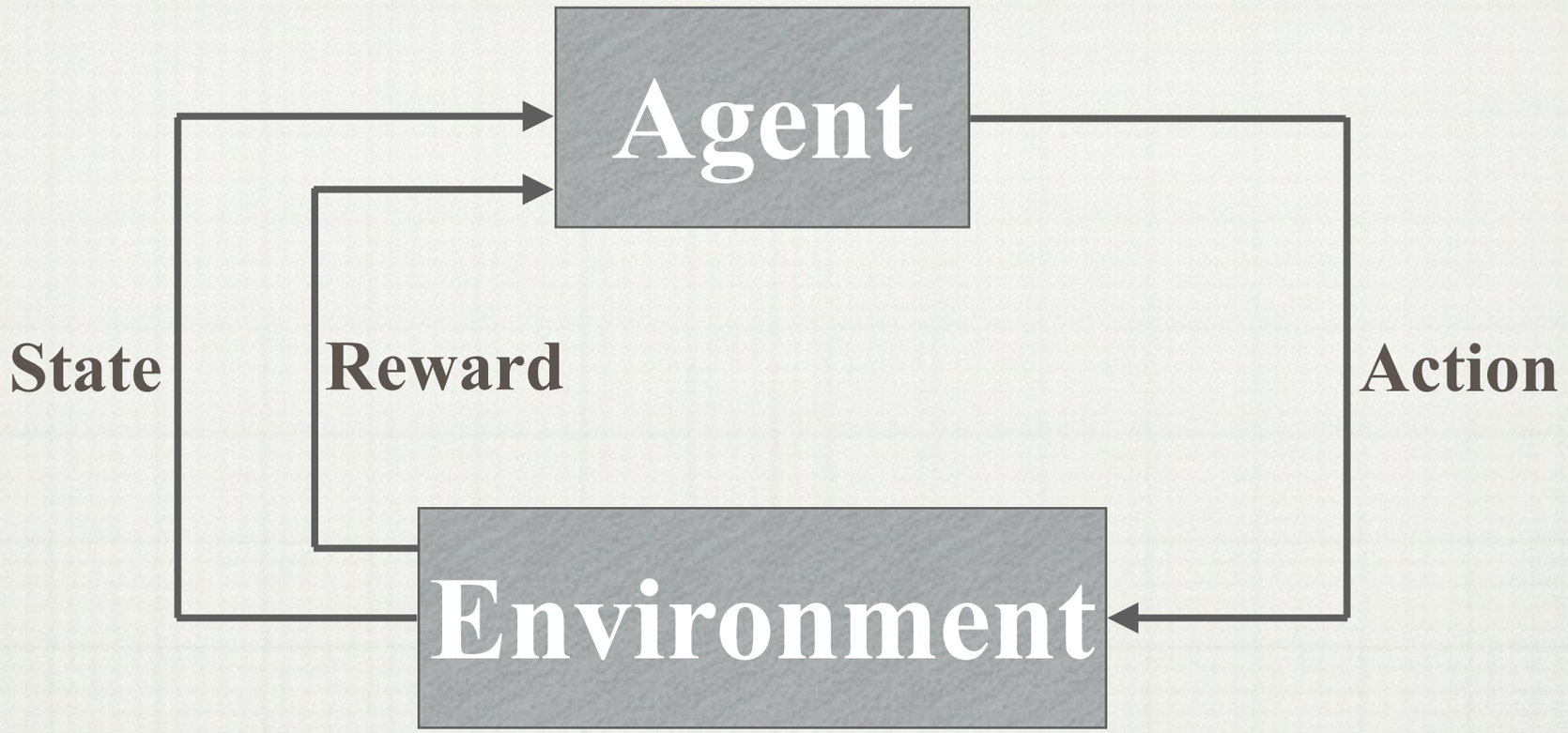
Training examples: (x_i), $x_i \sim D(x)$. x-samples.

Learning Objective: Minimise a loss function that reflects some statistical structure of the input.

Examples: Principal Component Analysis, Hidden Markov Models



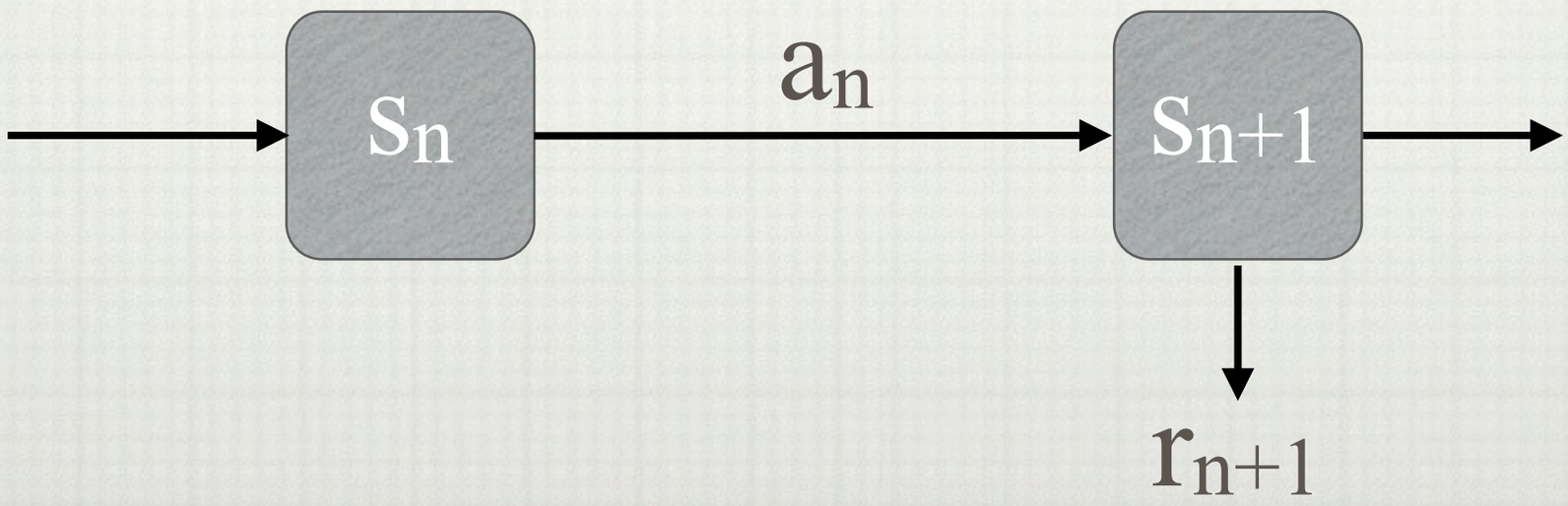
Reinforcement Learning (RL)



- **Learning Objective**: Take actions to maximise expected cumulative rewards.
- **Value Function**: A number that denotes how valuable a state-action is in solving the objective.
- **Policy**: Mapping between states and actions.
- **Types of RL**: Batch, Open-ended.

Markov Decision Process (MDP)

Defined as a **4-tuple** (S, A, P, R)
S - Fully observable state space; **A** - Space of actions
P - Stationary state transition fn. ; **R** - Stationary reward fn.
Possesses the **Markov Property**



Intrinsic Motivation

Artificial Curiosity (Schmidhuber, 1991)

Artificial Curiosity (Schmidhuber, 1991)

- Provides a mathematical formalism for describing curiosity and creativity.

Artificial Curiosity (Schmidhuber, 1991)

- Provides a mathematical formalism for describing curiosity and creativity.
- **Interesting**: Learnable but as-yet-unknown aspects of the environment.

Artificial Curiosity (Schmidhuber, 1991)

- Provides a mathematical formalism for describing curiosity and creativity.
- **Interesting**: Learnable but as-yet-unknown aspects of the environment.
- **Un-interesting**: Predictable or inherently unlearnable (noise)

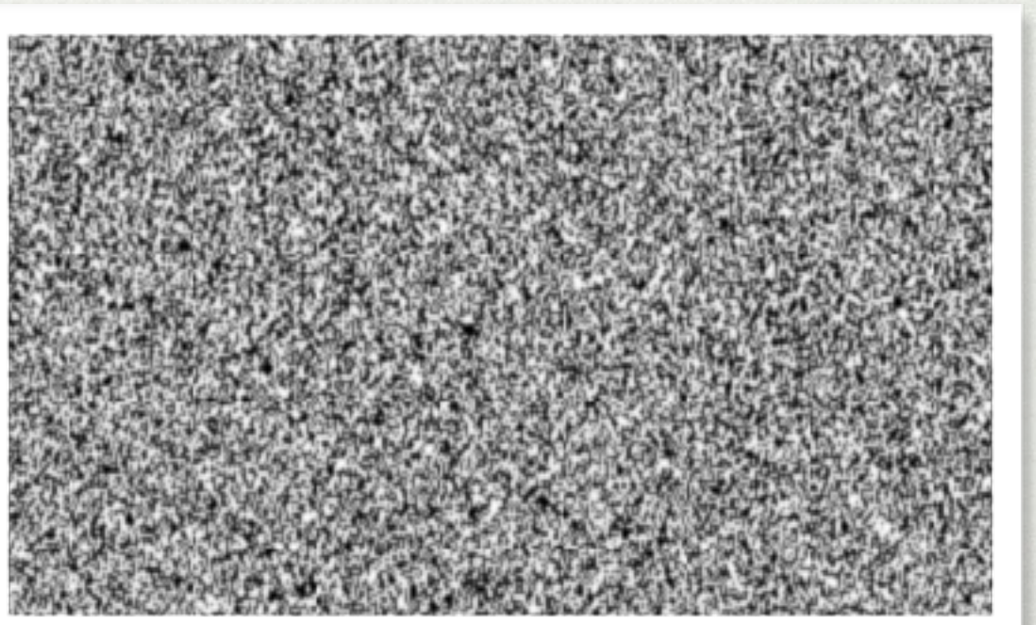
Artificial Curiosity (Schmidhuber, 1991)

- Provides a mathematical formalism for describing curiosity and creativity.
- **Interesting**: Learnable but as-yet-unknown aspects of the environment.
- **Un-interesting**: Predictable or inherently unlearnable (noise)
- **Curiosity Rewards**: Proportional to the improvement of an internal model/predictor of the environment.

Artificial Curiosity

Hi ... I am Homer

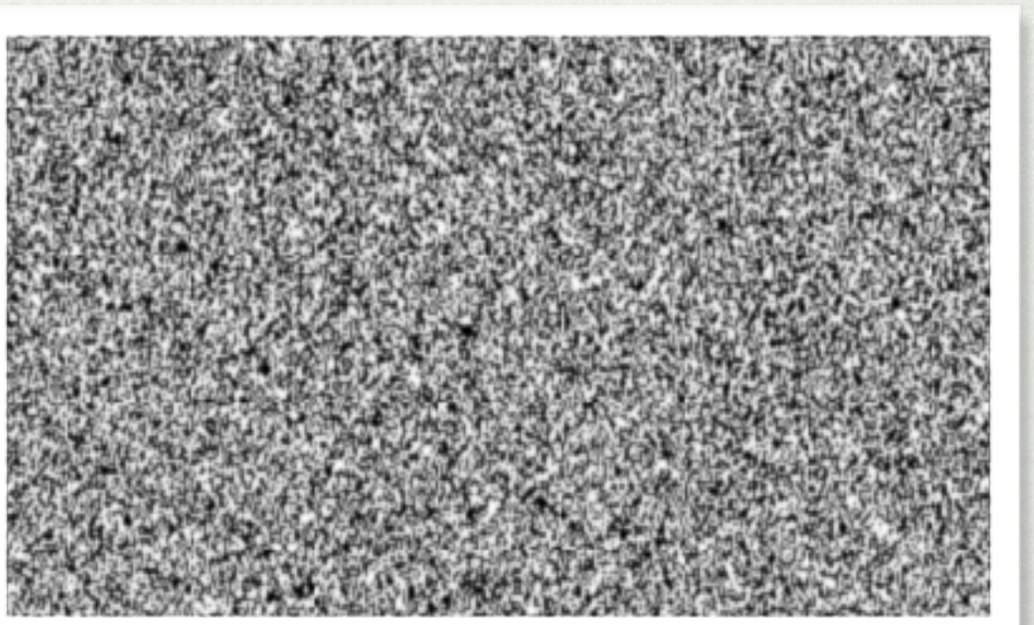
Artificial Curiosity



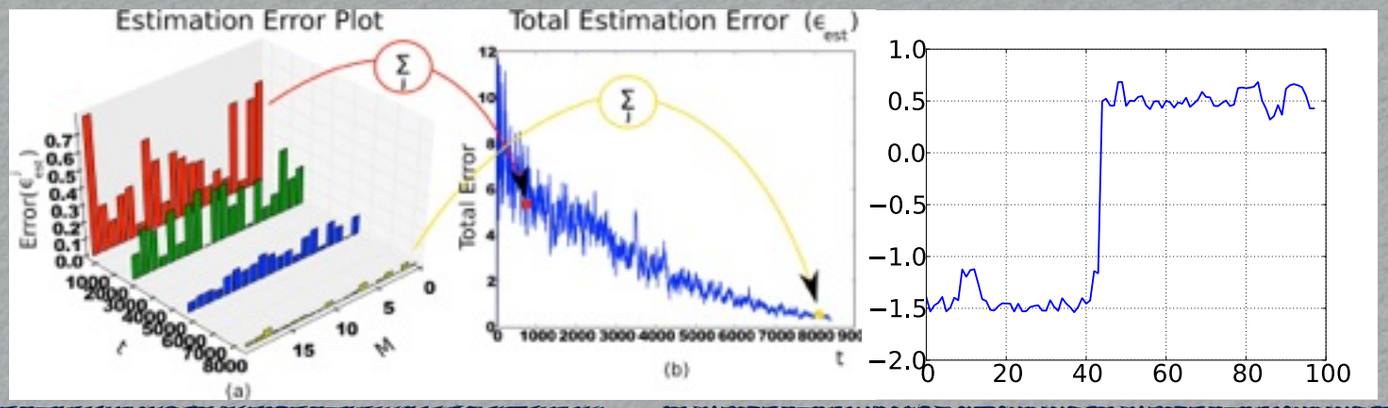
Artificial Curiosity

?????@@@@@!!!!!!

Boring!



Artificial Curiosity



Interesting!

Artificial Curiosity

Not interesting anymore!

Artificial Curiosity

Now this slide is boring! Move on!

Curiosity-Driven RL (CDRL)

- CDRL objective : Maximise accumulation of expected cumulative curiosity rewards.

Curiosity-Driven RL (CDRL)

- CDRL objective : Maximise accumulation of expected cumulative curiosity rewards.

- Agent is motivated to explore wherever it is makes maximum learning progress.

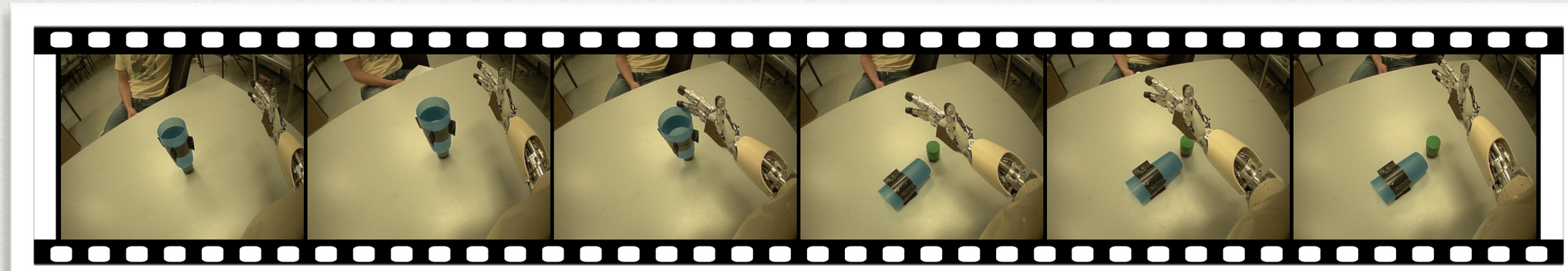
Curiosity-Driven RL (CDRL)

- CDRL objective : Maximise accumulation of expected cumulative curiosity rewards.
  - Agent is motivated to explore wherever it is makes maximum learning progress.
- **Task-independent** intrinsic reward function

Curiosity-Driven RL (CDRL)

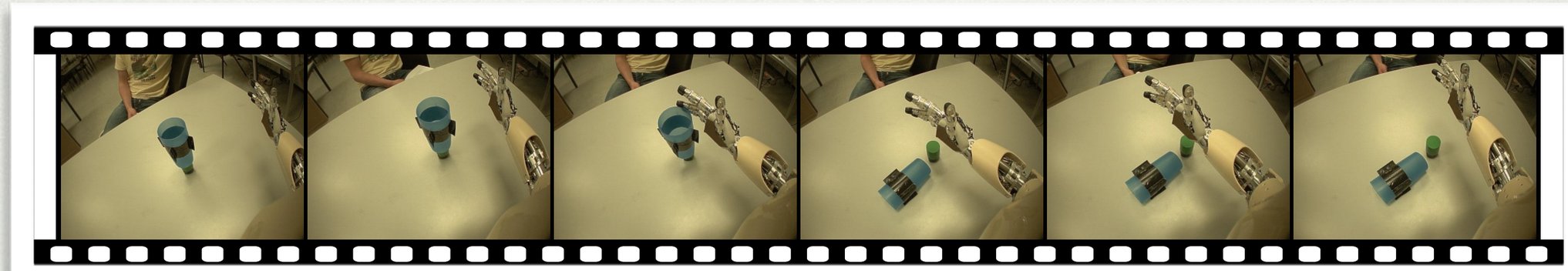
- CDRL objective : Maximise accumulation of expected cumulative curiosity rewards.
  - Agent is motivated to explore wherever it is makes maximum learning progress.
- **Task-independent** intrinsic reward function
- Challenges in CDRL: High dimensionality of the state spaces, non-stationary rewards

High-Dimensionality of State-Space



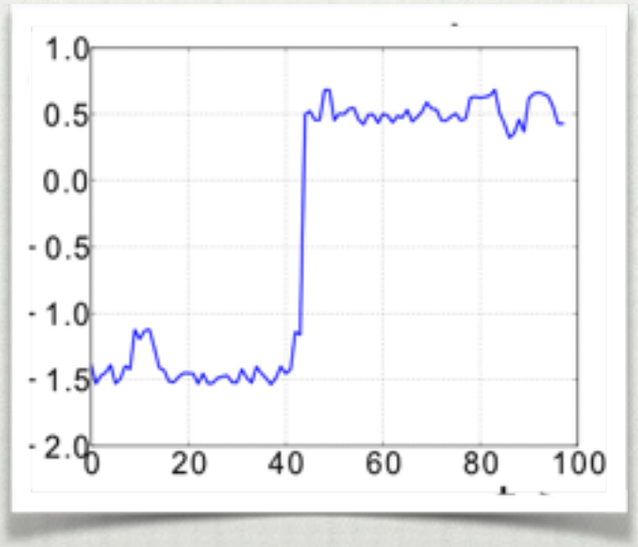
640 x 480 Dim

High-Dimensionality of State-Space



640 x 480 Dim

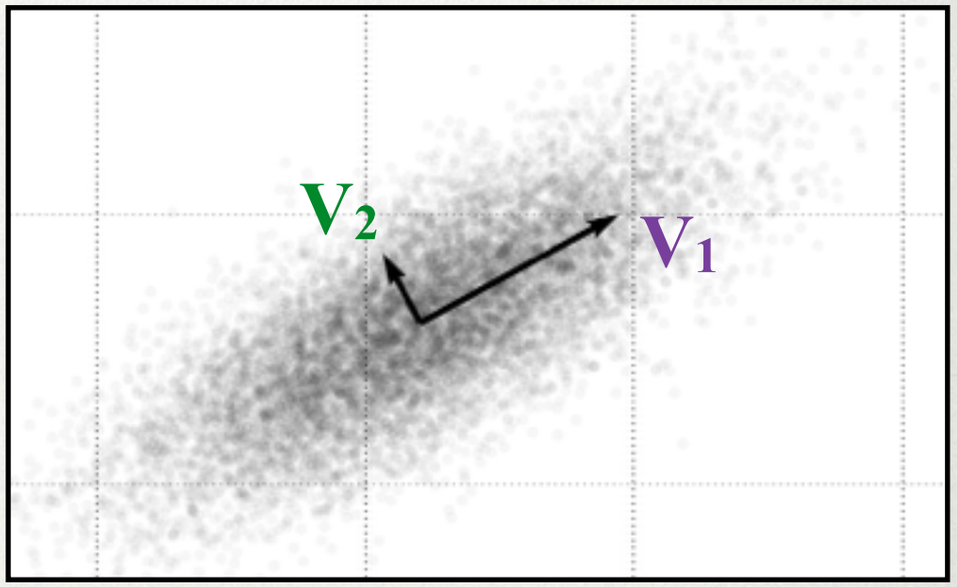
1 Dim

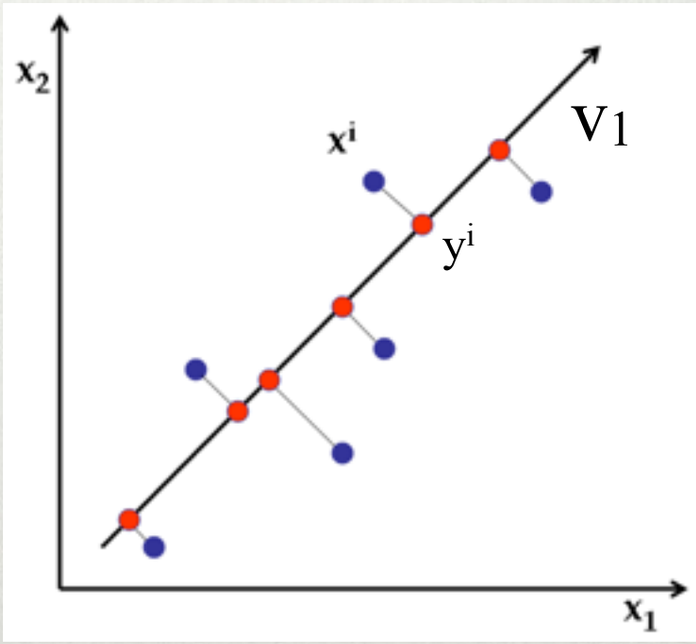


Principal Component Analysis (PCA)

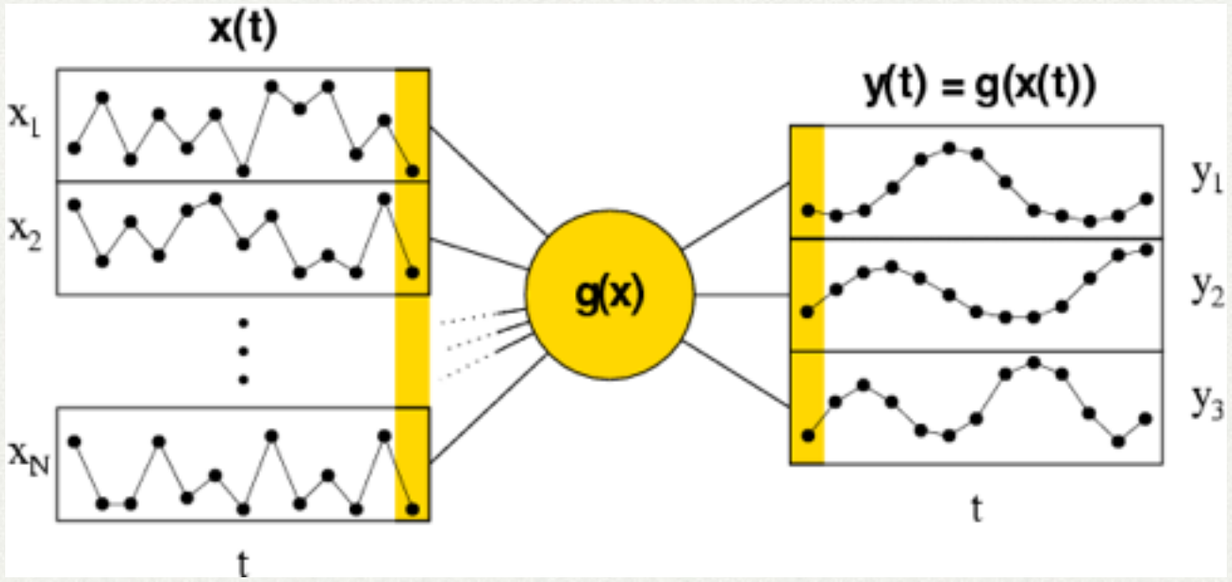
- Unsupervised Learning method
- Extracts linear uncorrelated feature vectors, which represent directions of maximal variance in the input data.
- Eigen-Decomposition
- Dimensionality reduction: Project input data on fewer PCA vectors (m < n):

$$\underset{(m \times 1)}{y} = \underset{(m \times n)}{V_{pca}^{T}} \bullet \underset{(n \times 1)}{x}$$



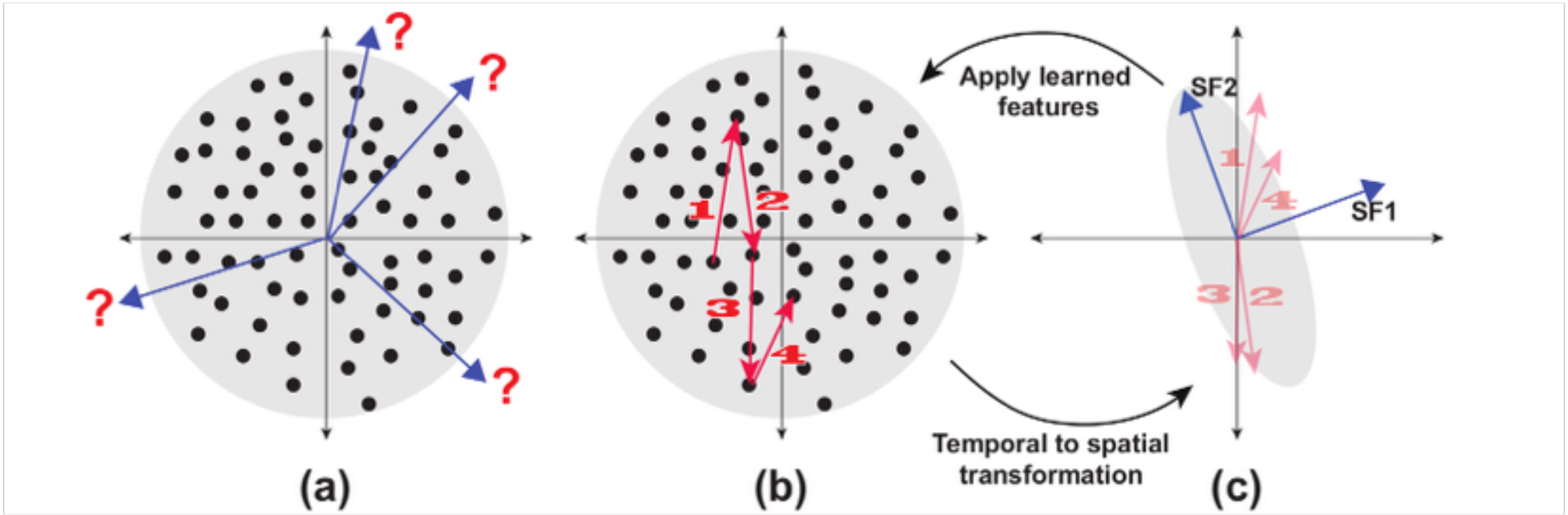


Slow Feature Analysis (Wiskott and Sejnowski, 2002)



- Unsupervised learning method
- Uses temporal coherence in the input data
- Extracts slowly varying components from rapidly changing raw sensory inputs (**Slowness Learning**)

PCA vs SFA


?
?
?
?
1
2
3
4
Apply learned features
SF2
SF1
1
4
3
2
Temporal to spatial transformation
(a)
(b)
(c)

SFA Formalised

Given an I-dimensional input signal $\mathbf{x}(t)$, find a set of J instantaneous real-valued functions $\mathbf{g}(x)$, which generate a J-dimensional output signal $\mathbf{y}(t)$, such that for each $j \in \{1, ..., J\}$

Definition

$\Delta_j := \Delta(y_j) := \langle \dot{y}_j^2 \rangle$ is minimal, with the constraints

$$\langle y_j \rangle = 0 \quad \text{(zero mean)}, \tag{1}$$

$$\langle y_j^2 \rangle = 1 \quad \text{(unit variance)}, \tag{2}$$

$$\forall i < j : \langle y_i y_j \rangle = 0 \quad \text{(decorrelation and order)}, \tag{3}$$

SFA Implementation via Eigen-Vector based approach (Linear)

SFA Implementation via Eigen-Vector based approach (Linear)

- **Whitening**: Input is normalised to have zero mean and unit variance via PCA-1 (V_1)

SFA Implementation via Eigen-Vector based approach (Linear)

- **Whitening**: Input is normalised to have zero mean and unit variance via PCA-1 (V_1)
- **Derivative**: A fast approximation of the whitened input is computed using backward-difference.

SFA Implementation via Eigen-Vector based approach (Linear)

- **Whitening**: Input is normalised to have zero mean and unit variance via PCA-1 (V_1)
- **Derivative**: A fast approximation of the whitened input is computed using backward-difference.
- **Minor Components**: Extract components with least Eigen-values of the derivative signal using PCA-2 (V_2)

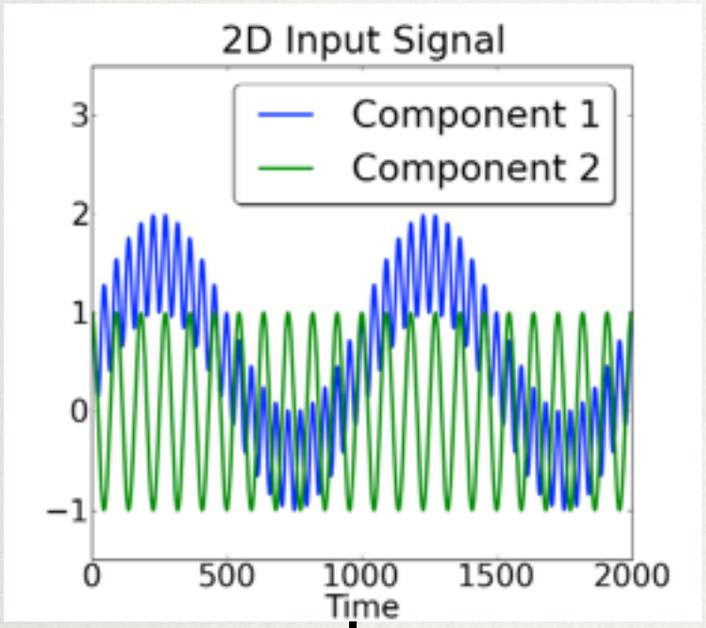
SFA Implementation via Eigen-Vector based approach (Linear)

- **Whitening**: Input is normalised to have zero mean and unit variance via PCA-1 (V_1)
- **Derivative**: A fast approximation of the whitened input is computed using backward-difference.
- **Minor Components**: Extract components with least Eigen-values of the derivative signal using PCA-2 (V_2)
- **Slow Features**: $V_{sf} = V_1 \bullet V_2$

SFA Implementation via Eigen-Vector based approach (Linear)

- **Whitening**: Input is normalised to have zero mean and unit variance via PCA-1 (V_1)
- **Derivative**: A fast approximation of the whitened input is computed using backward-difference.
- **Minor Components**: Extract components with least Eigen-values of the derivative signal using PCA-2 (V_2)
- **Slow Features**: $V_{sf} = V_1 \bullet V_2$
- **SFA Outputs**: $y(t) = V_{sf}^{T} \bullet x(t)$

SFA Examples

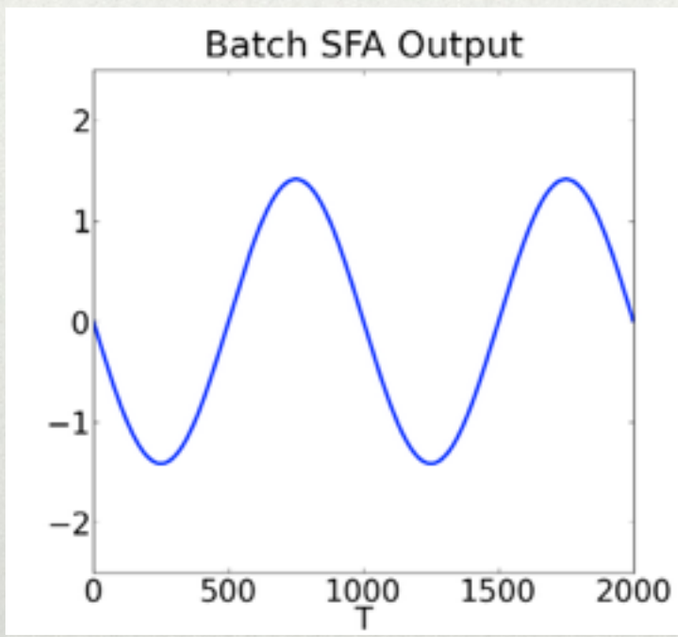


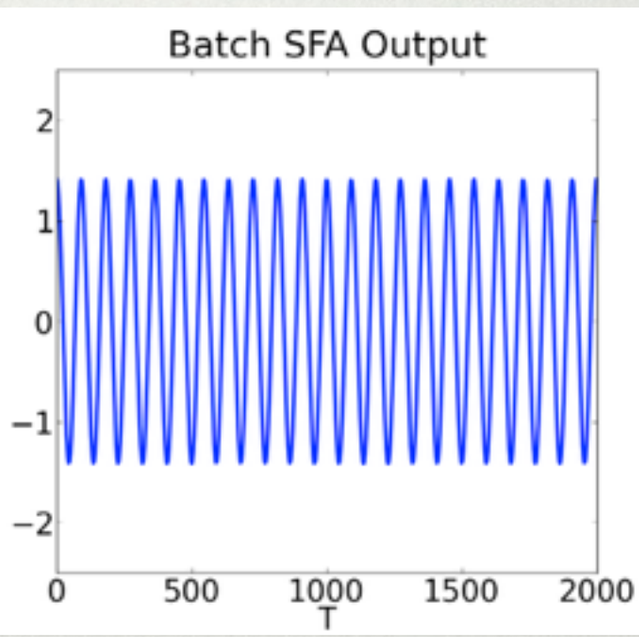
$$\breve{x}_1(t) = \sin(t) + \cos(11\,t)^2,$$

$$\breve{x}_2(t) = \cos(11\,t),\ t \in [0, 2\pi],$$

QUADRATIC EXPANSION

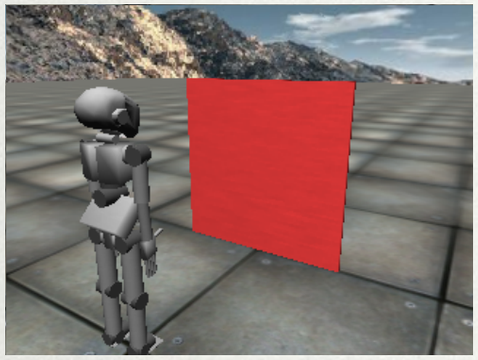
SFA



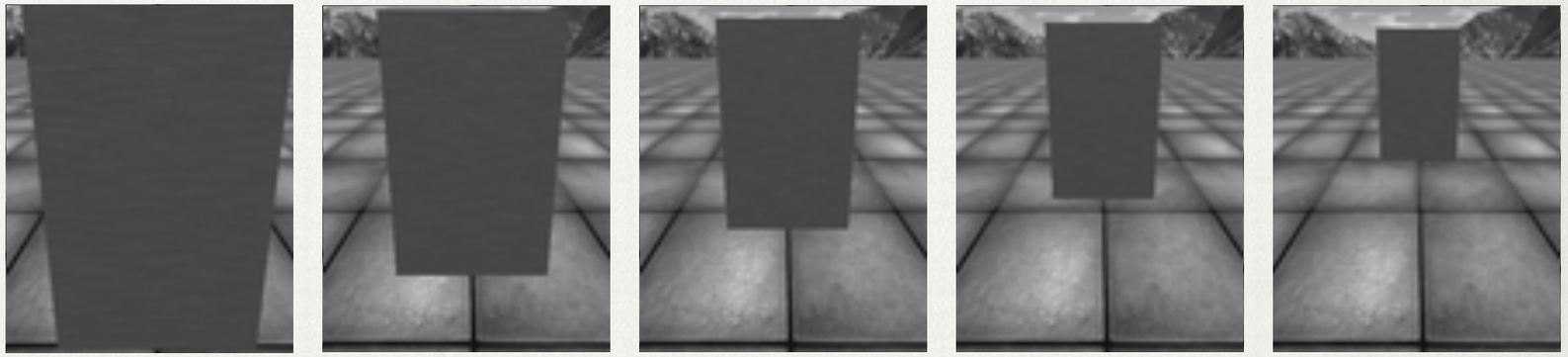


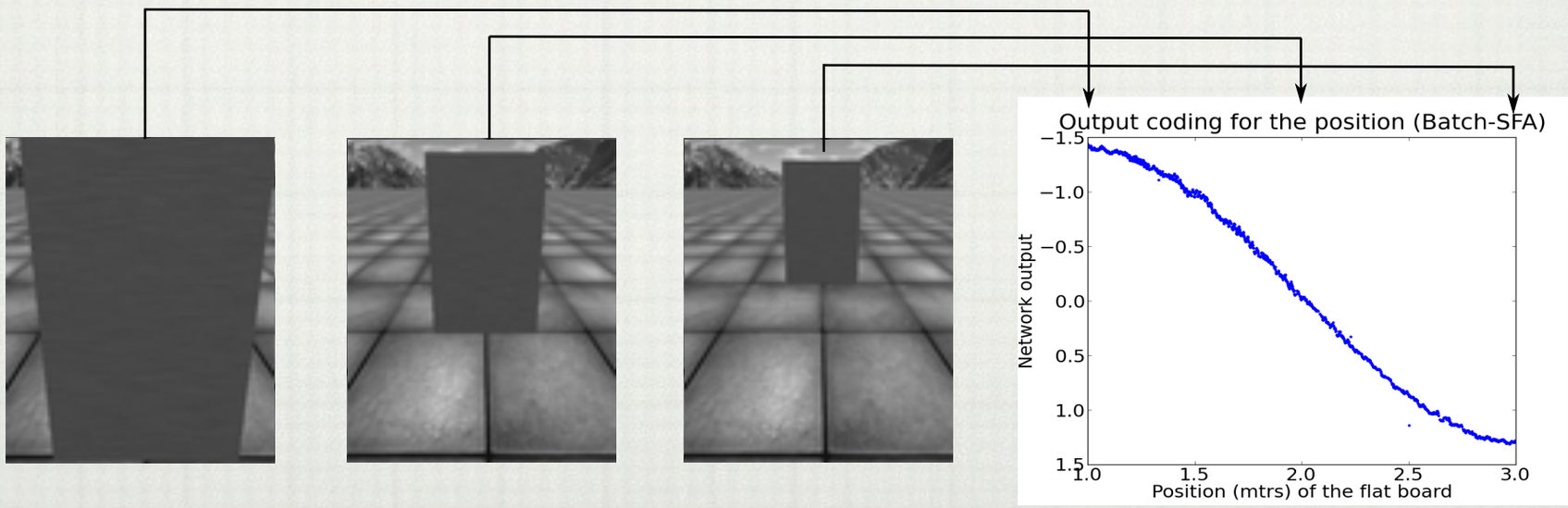
SFA Examples

Environment



Input Image Stream

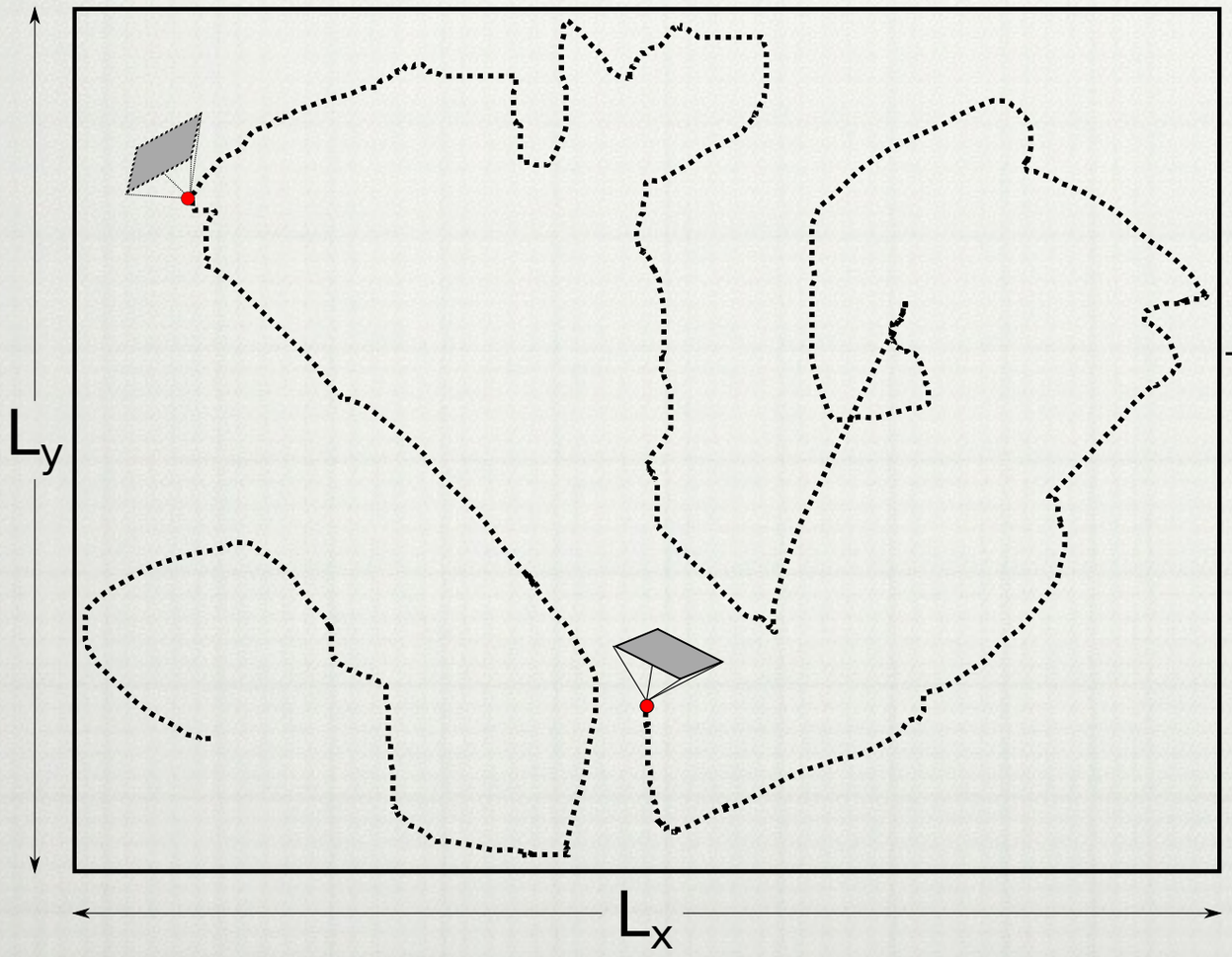




First Slow Feature Output

SFA Examples

3D Room Environment



Input Image Stream

SFA

$(\pm\cos(\pi x/L_x), \pm\cos(\pi y/L_y), \ldots)$

Invariant Slow-Feature Outputs

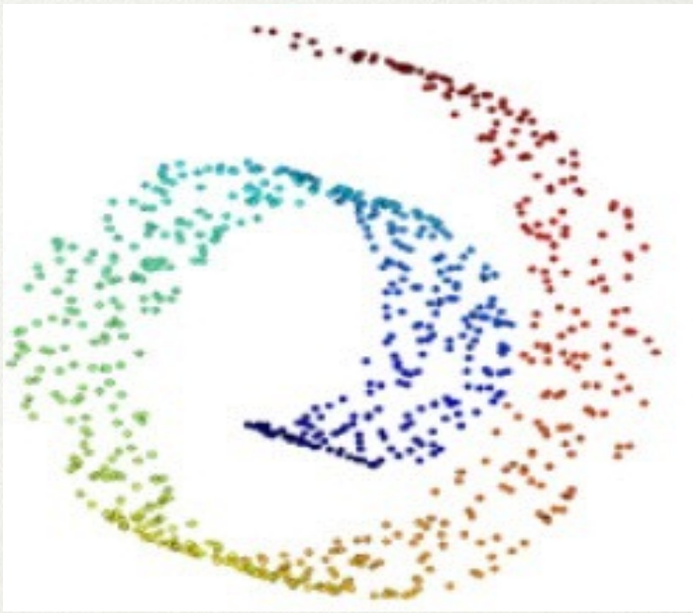
SFA for RL?

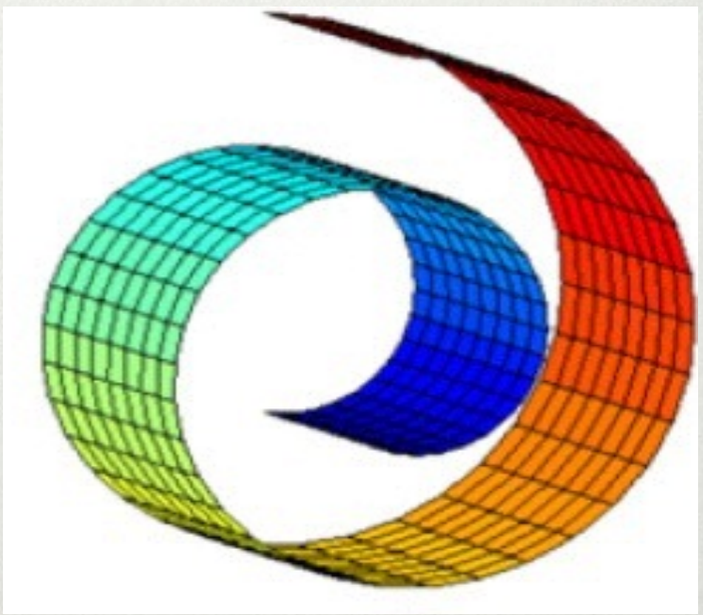
SFA for RL?

- Laplacian Eigen Maps (LEMs):

SFA for RL?

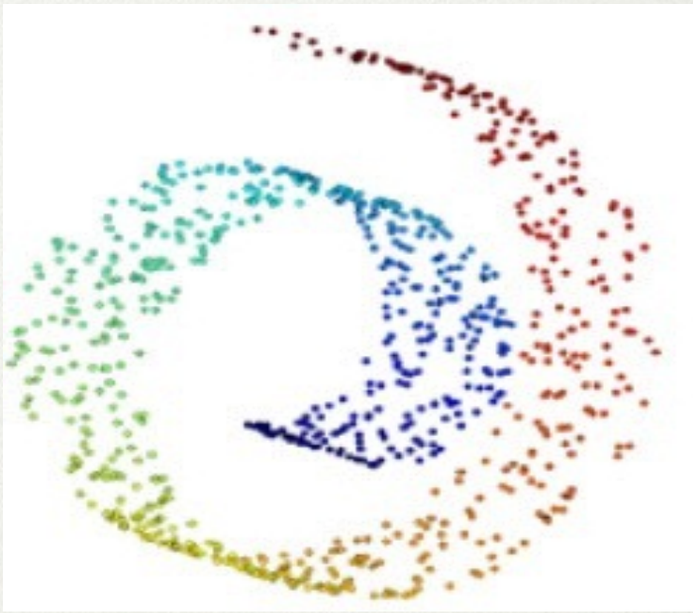
- Laplacian Eigen Maps (LEMs):
  - (a) A **low-dimensional embedding** of the state-space, which captures the local-connectedness between the states (Adjacency).

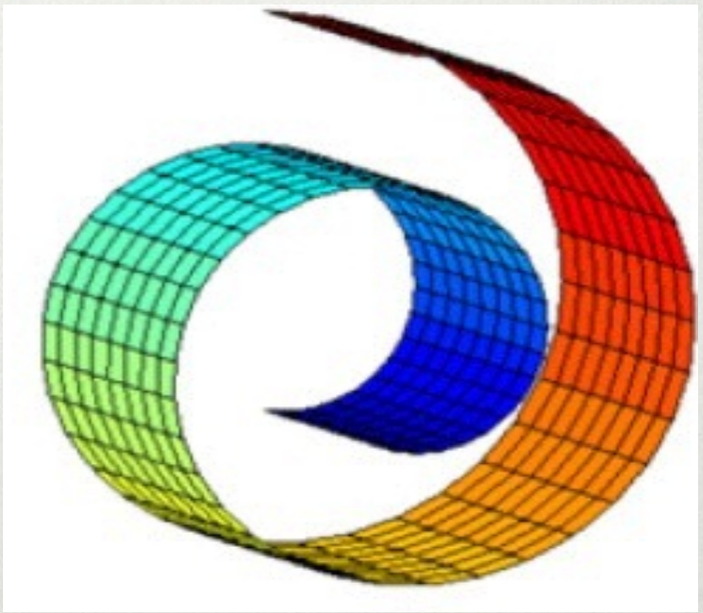




SFA for RL?

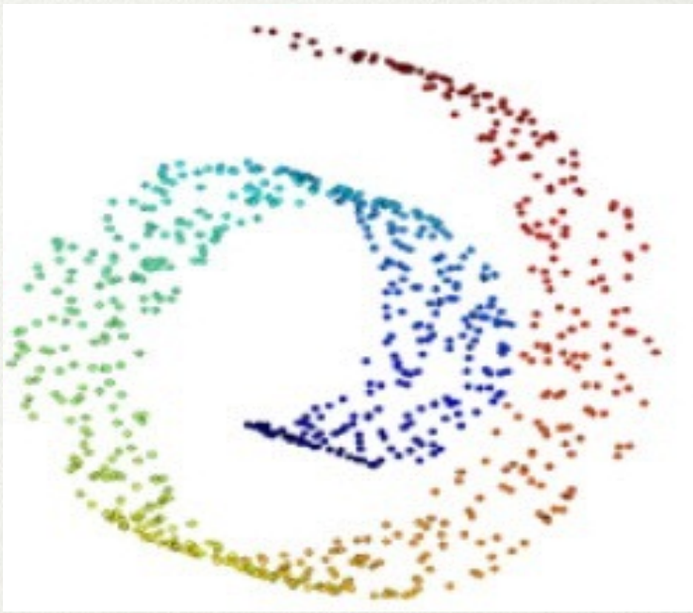
- Laplacian Eigen Maps (LEMs):
  - (a) A **low-dimensional embedding** of the state-space, which captures the local-connectedness between the states (Adjacency).
  - (b) Useful for representing value-functions for RL (Proto-value functions).

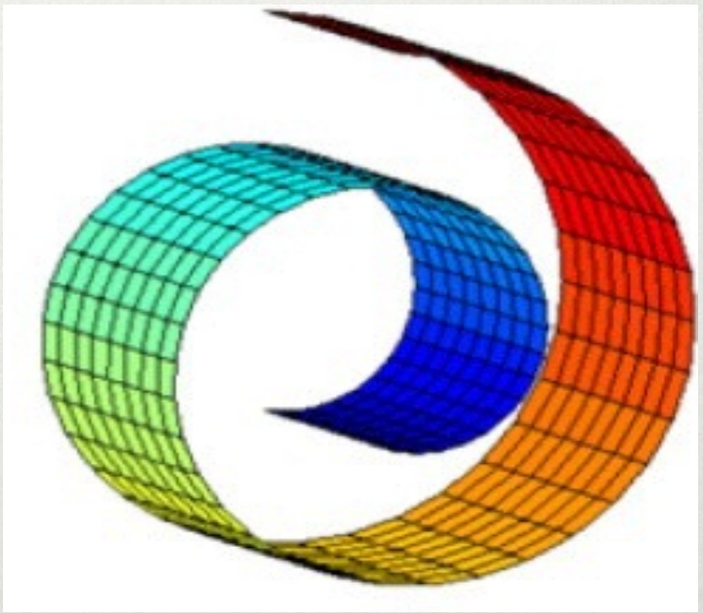




SFA for RL?

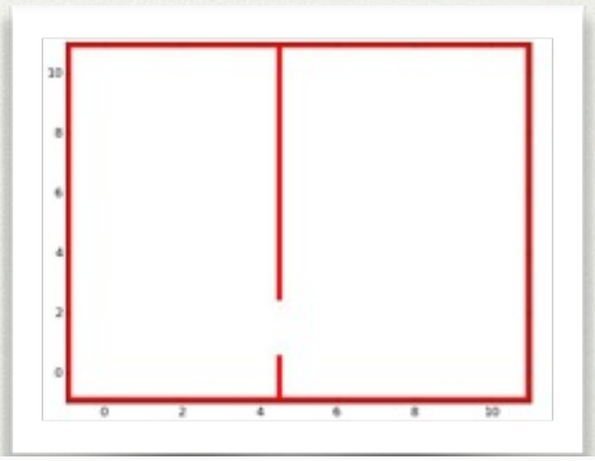
- Laplacian Eigen Maps (LEMs):
  - (a) A **low-dimensional embedding** of the state-space, which captures the local-connectedness between the states (Adjacency).
  - (b) Useful for representing value-functions for RL (Proto-value functions).
  - (c) LEMs are non-parametric - dimensionality scales with the number of data points.

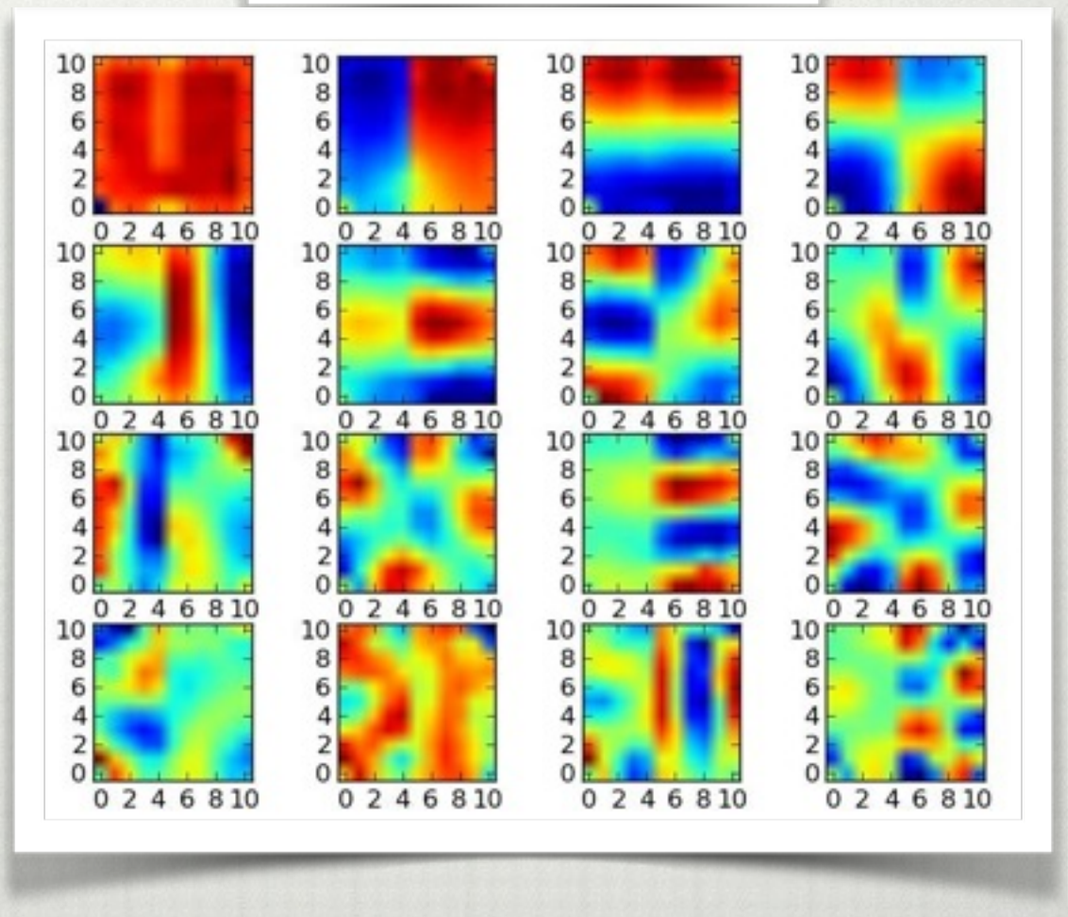




SFA for RL?

- SFA is a low-complex linear **function** approximation of LEMs

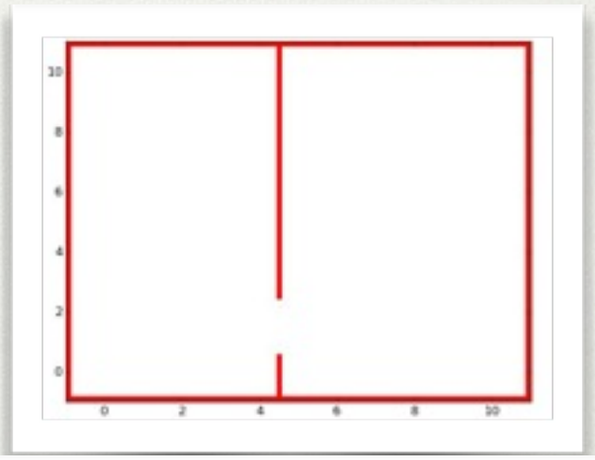


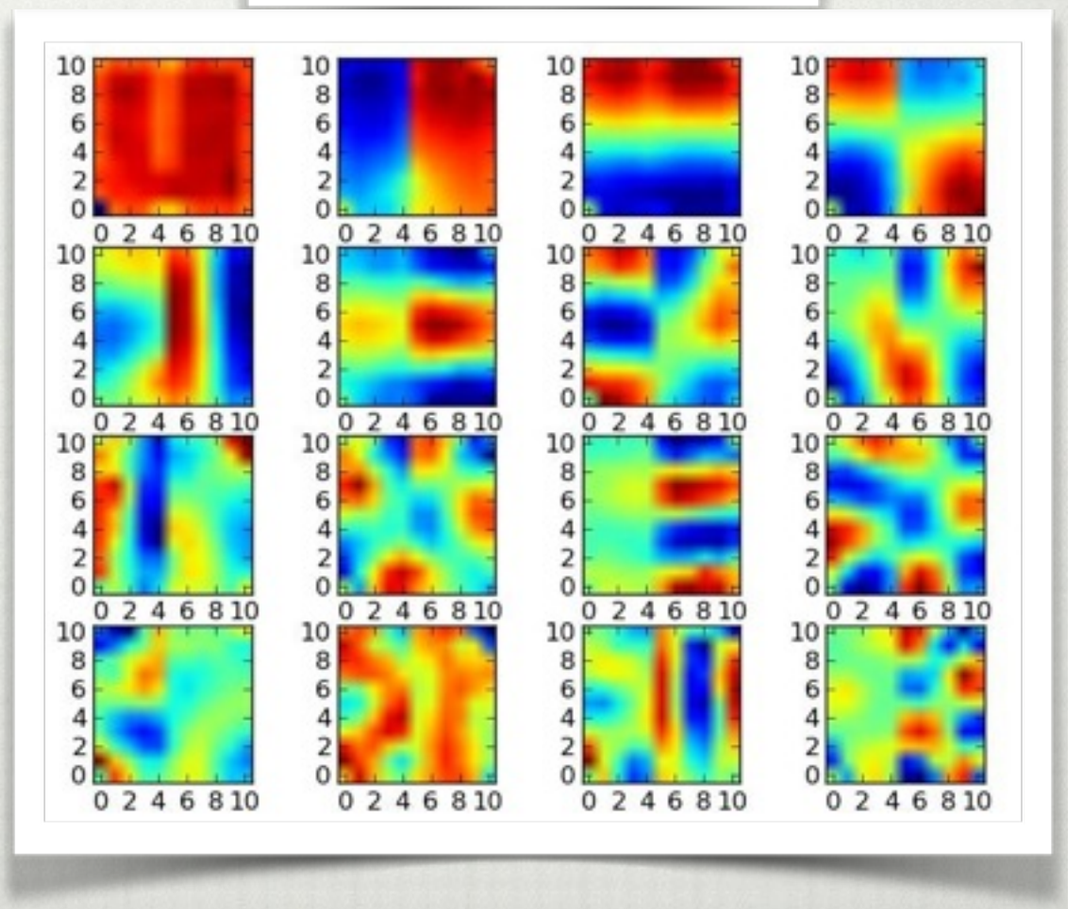


SFA for RL?

- SFA is a low-complex linear **function** approximation of LEMs

- SFA is a parametric method that generalises to unseen data.



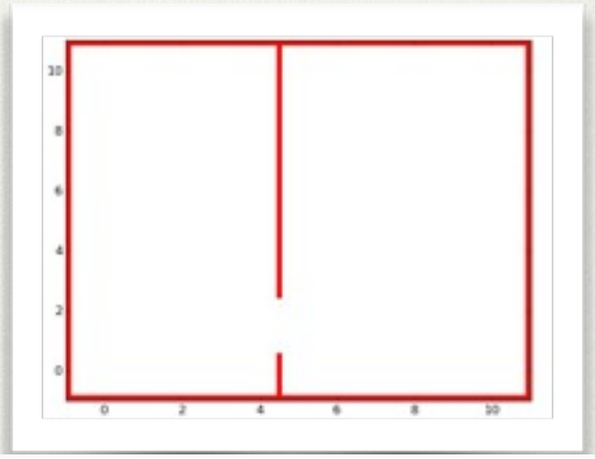


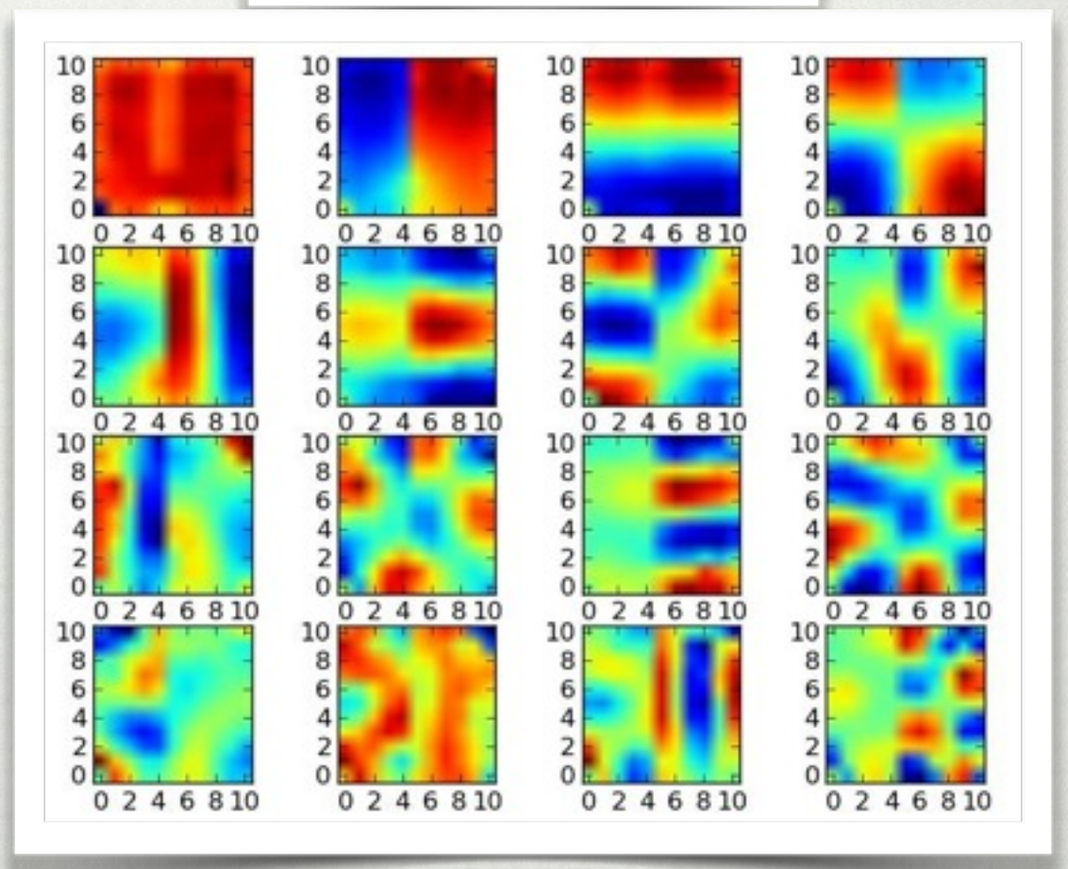
SFA for RL?

- SFA is a low-complex linear **function** approximation of LEMs

- SFA is a parametric method that generalises to unseen data.

- Can therefore be a good preprocessor for RL (Legenstein et al. 2010, Kompella et al. 2011, Luciw and Schmidhuber, 2012).





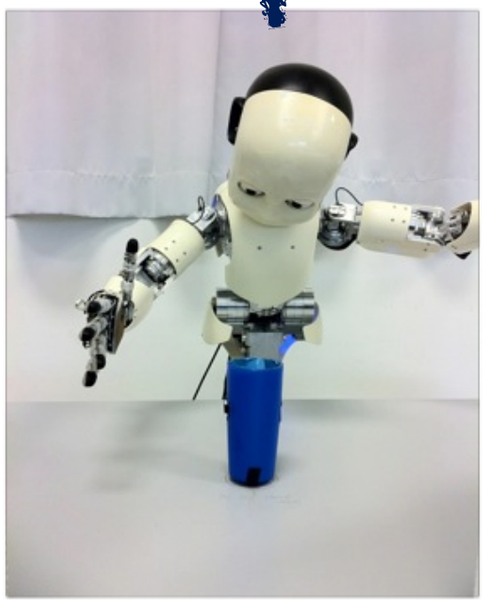
SFA for Open-Ended Online RL?

SFA for Open-Ended Online RL?

- Limitations of SFA for open-ended online learning RL agents:
  - Estimates covariance matrices via batch processing.
  - Cubic computational complexity.

Incremental Slow Feature Analysis (IncSFA)

Slow, but one step at a time!



Incremental Slow Feature Analysis
(Kompella et al., 2011)

Incremental Slow Feature Analysis
(Kompella et al., 2011)

- Incremental SFA (IncSFA) is an online implementation of SFA

Incremental Slow Feature Analysis
(Kompella et al., 2011)

- Incremental SFA (IncSFA) is an online implementation of SFA
  - Adaptive to changing input statistics

Incremental Slow Feature Analysis
(Kompella et al., 2011)

- Incremental SFA (IncSFA) is an online implementation of SFA
  - Adaptive to changing input statistics
  - Linear computational complexity

Incremental Slow Feature Analysis
(Kompella et al., 2011)

- Incremental SFA (IncSFA) is an online implementation of SFA
  - Adaptive to changing input statistics
  - Linear computational complexity
  - Reduced sensitivity to outliers

Incremental Slow Feature Analysis
(Kompella et al., 2011)

- Incremental SFA (IncSFA) is an online implementation of SFA
  - Adaptive to changing input statistics
  - Linear computational complexity
  - Reduced sensitivity to outliers
  - Adds to the biological plausibility of batch SFA

Batch SFA Re-Visited

- **Whitening**: Input is normalised to have zero mean and unit variance via PCA-1 (V_1)

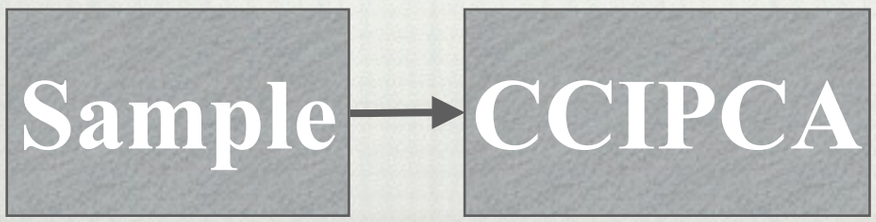
- **Minor Components**: Extract components with least Eigen-values of the derivative signal using PCA-2 (V_2)

- **Slow Features**: $V_{sf} = V_1 \bullet V_2$

IncSFA Simplified

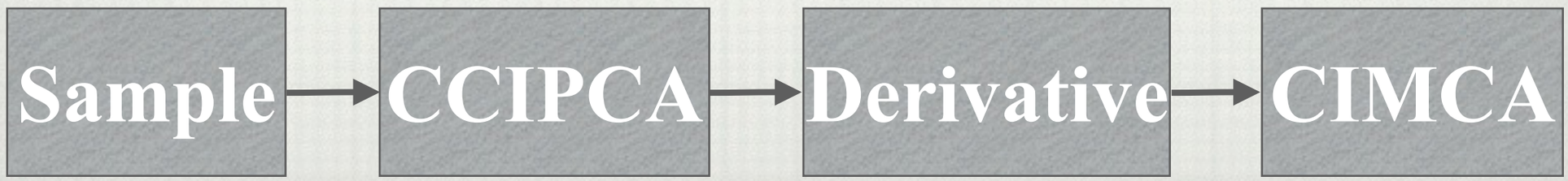
IncSFA Simplified

- Incremental **whitening** using Candid Covariance-free Incremental PCA (CCIPCA).



IncSFA Simplified

- Incremental **whitening** using Candid Covariance-free Incremental PCA (CCIPCA).
- Incremental extraction of **minor components** using Covariance-free Incremental MCA (CIMCA).



Candid Covariance-free Incremental PCA (CCIPCA; Weng et al.)

- Updates eigenvalues and eigenvectors from each centred observation.

Algorithm 2: CCIPCA-Update($\mathbf{V}, K, \mathbf{u}, \eta$)

```
  //Candid Covariance-Free Incremental PCA
```

$\mathbf{u}_1 \leftarrow \mathbf{u}$

for $i \leftarrow 1$ *to* K **do**

```
    //Principal component update
```

$$\mathbf{v}_i \leftarrow (1-\eta)\,\mathbf{v}_i + \eta \left[\frac{\mathbf{u}_i \cdot \mathbf{v}_i}{\|\mathbf{v}_i\|}\,\mathbf{u}_i\right]$$

```
    //Residual
```

$$\mathbf{u}_{i+1} = \mathbf{u}_i - \left(\mathbf{u}_i^T \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}\right)\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$$

end

return V

Candid Covariance-free Incremental PCA (CCIPCA; Weng et al.)

- Updates eigenvalues and eigenvectors from each centred observation.

- Hebbian Update ($\Delta w \propto \eta x_i y_i$) + Residual

Algorithm 2: CCIPCA-Update($\mathbf{V}, K, \mathbf{u}, \eta$)

```
//Candid Covariance-Free Incremental PCA
```

$\mathbf{u}_1 \leftarrow \mathbf{u}$

for $i \leftarrow 1$ *to* K **do**

```
//Principal component update
```

$$\mathbf{v}_i \leftarrow (1-\eta)\,\mathbf{v}_i + \eta\left[\frac{\mathbf{u}_i \cdot \mathbf{v}_i}{\|\mathbf{v}_i\|}\,\mathbf{u}_i\right]$$

```
//Residual
```

$$\mathbf{u}_{i+1} = \mathbf{u}_i - \left(\mathbf{u}_i^T \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}\right)\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$$

end

return V

Candid Covariance-free Incremental PCA (CCIPCA; Weng et al.)

- Updates eigenvalues and eigenvectors from each centred observation.
- Hebbian Update ($\Delta w \propto \eta x_i y_i$) + Residual
- Intuitive to tune learning rate

Algorithm 2: CCIPCA-Update($\mathbf{V}, K, \mathbf{u}, \eta$)

```
  //Candid Covariance-Free Incremental PCA
```

$\mathbf{u}_1 \leftarrow \mathbf{u}$

for $i \leftarrow 1$ *to* K **do**

```
    //Principal component update
```

$$\mathbf{v}_i \leftarrow (1-\eta)\,\mathbf{v}_i + \eta \left[\frac{\mathbf{u}_i \cdot \mathbf{v}_i}{\|\mathbf{v}_i\|}\,\mathbf{u}_i\right]$$

```
    //Residual
```

$$\mathbf{u}_{i+1} = \mathbf{u}_i - \left(\mathbf{u}_i^T \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}\right)\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|}$$

end

return V

Covariance-free Minor Component Analysis (CIMCA; Peng et al.)

- Based on Peng's MCA modifications:
  - (a) Covariance free
  - (b) Normalisation step

Algorithm 3: CIMCA-Update($\hat{\phi}, J, \dot{\mathbf{z}}, \gamma, \eta$)

```
//Covariance-Free Incremental MCA
//φ̂_i : i^th column of φ̂
```

$\mathbf{l}_1 \leftarrow 0$

for $i \leftarrow 1$ *to* J **do**

```
//Minor component update
```

$\hat{\phi}_i \leftarrow (1-\eta)\hat{\phi}_i - \eta\left[(\dot{\mathbf{z}} \cdot \hat{\phi}_i)\,\dot{\mathbf{z}} + \mathbf{l}_i\right].$

```
//Normalize
```

$\hat{\phi}_i \leftarrow \hat{\phi}_i / \|\hat{\phi}_i\|.$

```
//Lateral competition from "lower" components
```

$\mathbf{l}_{i+1} \leftarrow \gamma \sum_j^i (\hat{\phi}_j \cdot \hat{\phi}_i)\hat{\phi}_j$

end

return W

Covariance-free Minor Component Analysis (CIMCA; Peng et al.)

- Based on Peng's MCA modifications:
  - (a) Covariance free
  - (b) Normalisation step

- Anti-Hebbian + Sequential addition.

Algorithm 3: CIMCA-Update($\hat{\phi}, J, \dot{\mathbf{z}}, \gamma, \eta$)

```
//Covariance-Free Incremental MCA
//φ̂_i :  i^th column of φ̂
```
$\mathbf{l}_1 \leftarrow 0$

for $i \leftarrow 1$ *to* J **do**

```
//Minor component update
```
$\hat{\phi}_i \leftarrow (1-\eta)\hat{\phi}_i - \eta\left[(\dot{\mathbf{z}}\cdot\hat{\phi}_i)\,\dot{\mathbf{z}} + \mathbf{l}_i\right].$

```
//Normalize
```
$\hat{\phi}_i \leftarrow \hat{\phi}_i / \|\hat{\phi}_i\|.$

```
//Lateral competition from "lower" components
```
$\mathbf{l}_{i+1} \leftarrow \gamma \sum_j^i (\hat{\phi}_j \cdot \hat{\phi}_i)\hat{\phi}_j$

end

return W

Covariance-free Minor Component Analysis (CIMCA; Peng et al.)

- Based on Peng's MCA modifications:
  - (a) Covariance free
  - (b) Normalisation step
- Anti-Hebbian + Sequential addition.
- Constant learning rate - no need for dynamic rate scheduling.

Algorithm 3: CIMCA-Update($\hat{\phi}, J, \dot{\mathbf{z}}, \gamma, \eta$)

```
//Covariance-Free Incremental MCA
//φ̂_i :  i^th column of φ̂
```
$\mathbf{l}_1 \leftarrow 0$

for $i \leftarrow 1$ *to* J **do**

```
    //Minor component update
```
$\hat{\phi}_i \leftarrow (1-\eta)\hat{\phi}_i - \eta\left[(\dot{\mathbf{z}}\cdot\hat{\phi}_i)\,\dot{\mathbf{z}} + \mathbf{l}_i\right].$

```
    //Normalize
```
$\hat{\phi}_i \leftarrow \hat{\phi}_i / \|\hat{\phi}_i\|.$

```
    //Lateral competition from "lower" components
```
$\mathbf{l}_{i+1} \leftarrow \gamma \sum_j^i (\hat{\phi}_j \cdot \hat{\phi}_i)\hat{\phi}_j$

end

return W

Algorithm 1: IncSFA(J, K, θ)

//Incremental update of J slow features from samples $\mathbf{x} \in \mathcal{R}^I$
//$\mathbf{V}$: K columns: PCs of $\mathbf{x}$
//$\hat{\phi}$: J columns: SFs
//$\mathbf{v}^Y$: First PC in $\dot{\mathbf{z}}$-space
//$\bar{\mathbf{x}}$: Mean of $\mathbf{x}$

1 $\{\mathbf{V}, \hat{\phi}, \mathbf{v}^Y, \bar{\mathbf{x}}\} \leftarrow$ Initialize ()

Algorithm 1: IncSFA(J, K, θ)

```
//Incremental update of J slow features from samples x∈ℜ^I
//V  :  K columns:  PCs of x
//φ̂  :  J columns:  SFs
//v^Y :  First PC in ż-space
//x̄  :  Mean of x
```

1 $\{\mathbf{V}, \hat{\phi}, \mathbf{v}^Y, \bar{\mathbf{x}}\} \leftarrow$ Initialize ()

2 **for** $t \leftarrow 1$ *to* ∞ **do**

3 $\quad \mathbf{x} \leftarrow$ Sense($worldstate$)

4 $\quad \{\eta_t^{PCA}, \eta_t^{MCA}\} \leftarrow$ LrnRateSchedule (θ, t)

Sense

Algorithm 1: IncSFA(J, K, θ)

```
//Incremental update of J slow features from samples x∈ℛ^I
//V :  K columns:  PCs of x
//φ̂ :  J columns:  SFs
//v^Y :  First PC in ż-space
//x̄ :  Mean of x
```

1 $\{\mathbf{V}, \hat{\phi}, \mathbf{v}^Y, \bar{\mathbf{x}}\} \leftarrow$ Initialize ()

2 **for** $t \leftarrow 1$ *to* ∞ **do**

3 $\quad \mathbf{x} \leftarrow$ Sense(*worldstate*)

4 $\quad \{\eta_t^{PCA}, \eta_t^{MCA}\} \leftarrow$ LrnRateSchedule (θ, t)

5 $\quad \bar{\mathbf{x}} \leftarrow (1 - \eta_t^{PCA})\,\bar{\mathbf{x}} + \eta_t^{PCA}\,\mathbf{x}$ //Update mean

6 $\quad \mathbf{u} \leftarrow (\mathbf{x} - \bar{\mathbf{x}})$ //Centering

Sense

Update mean &
de-mean the input

Algorithm 1: IncSFA(J, K, θ)

```
//Incremental update of J slow features from samples x∈ℛ^I
//V  :  K columns:  PCs of x
//φ̂  :  J columns:  SFs
//v^Y :  First PC in ż-space
//x̄  :  Mean of x
```

1 $\{\mathbf{V}, \hat{\phi}, \mathbf{v}^Y, \bar{\mathbf{x}}\} \leftarrow$ Initialize ()

2 **for** $t \leftarrow 1$ *to* ∞ **do**

3 $\quad \mathbf{x} \leftarrow$ Sense($worldstate$)

4 $\quad \{\eta_t^{PCA}, \eta_t^{MCA}\} \leftarrow$ LrnRateSchedule (θ, t)

5 $\quad \bar{\mathbf{x}} \leftarrow (1 - \eta_t^{PCA})\,\bar{\mathbf{x}} + \eta_t^{PCA}\,\mathbf{x}$ //Update mean

6 $\quad \mathbf{u} \leftarrow (\mathbf{x} - \bar{\mathbf{x}})$ //Centering

```
  //Candid Covariance-Free Incremental PCA
```

7 $\quad \mathbf{V} \leftarrow$ CCIPCA-Update $(\mathbf{V}, K, \mathbf{u}, \eta_t^{PCA})$

8 $\quad \mathbf{S} \leftarrow$ ConstructWhiteningMatrix $(\mathbf{V})$

9 $\quad$ **If** $t > 1$ **then** $(\mathbf{z}_{prev} \leftarrow \mathbf{z}_{curr})$ //Store prev.

```
  //Whitening and dim.  reduction
```

10 $\quad \mathbf{z}_{curr} \leftarrow \mathbf{S}^T \mathbf{u}$

]

]

]

]

]

]

]

Sense

Update mean & de-mean the input

Update whitening matrix & normalize the input

Algorithm 1: IncSFA(J, K, θ)

```
//Incremental update of J slow features from samples x∈ℛ^I
//V  :  K columns:  PCs of x
//φ̂  :  J columns:  SFs
//v^Y :  First PC in ż-space
//x̄  :  Mean of x
```

1 $\{\mathbf{V}, \hat{\phi}, \mathbf{v}^Y, \bar{\mathbf{x}}\} \leftarrow$ Initialize ()

2 **for** $t \leftarrow 1$ *to* ∞ **do**

3 $\quad \mathbf{x} \leftarrow$ Sense($worldstate$)

4 $\quad \{\eta_t^{PCA}, \eta_t^{MCA}\} \leftarrow$ LrnRateSchedule (θ, t)

5 $\quad \bar{\mathbf{x}} \leftarrow (1 - \eta_t^{PCA})\,\bar{\mathbf{x}} + \eta_t^{PCA}\,\mathbf{x}$ //Update mean

6 $\quad \mathbf{u} \leftarrow (\mathbf{x} - \bar{\mathbf{x}})$ //Centering

```
  //Candid Covariance-Free Incremental PCA
```

7 $\quad \mathbf{V} \leftarrow$ CCIPCA-Update $(\mathbf{V}, K, \mathbf{u}, \eta_t^{PCA})$

8 $\quad \mathbf{S} \leftarrow$ ConstructWhiteningMatrix $(\mathbf{V})$

9 $\quad$ **If** $t > 1$ **then** $(\mathbf{z}_{prev} \leftarrow \mathbf{z}_{curr})$ //Store prev.

```
  //Whitening and dim.  reduction
```

10 $\quad \mathbf{z}_{curr} \leftarrow \mathbf{S}^T \mathbf{u}$

11 $\quad$ **if** $t > 1$ **then**

12 $\quad\quad \mathbf{z} \leftarrow (\mathbf{z}_{curr} - \mathbf{z}_{prev})$ //Approx. derivative

Sense

Update mean & de-mean the input

Update whitening matrix & normalize the input

Compute derivative of whitened input

Algorithm 1: IncSFA(J, K, θ)

```
//Incremental update of J slow features from samples x∈ℛ^I
//V  :  K columns:  PCs of x
//φ̂  :  J columns:  SFs
//v^γ :  First PC in ż-space
//x̄  :  Mean of x
```

1 $\{\mathbf{V}, \hat{\phi}, \mathbf{v}^{\gamma}, \bar{\mathbf{x}}\} \leftarrow$ Initialize ()

2 **for** $t \leftarrow 1$ *to* ∞ **do**

3 $\quad \mathbf{x} \leftarrow$ Sense($worldstate$)

4 $\quad \{\eta_t^{PCA}, \eta_t^{MCA}\} \leftarrow$ LrnRateSchedule (θ, t)

5 $\quad \bar{\mathbf{x}} \leftarrow (1 - \eta_t^{PCA})\,\bar{\mathbf{x}} + \eta_t^{PCA}\,\mathbf{x}$ //Update mean

6 $\quad \mathbf{u} \leftarrow (\mathbf{x} - \bar{\mathbf{x}})$ //Centering

$\quad$ //Candid Covariance-Free Incremental PCA

7 $\quad \mathbf{V} \leftarrow$ CCIPCA-Update $(\mathbf{V}, K, \mathbf{u}, \eta_t^{PCA})$

8 $\quad \mathbf{S} \leftarrow$ ConstructWhiteningMatrix $(\mathbf{V})$

9 $\quad$ **If** $t > 1$ **then** $(\mathbf{z}_{prev} \leftarrow \mathbf{z}_{curr})$ //Store prev.

$\quad$ //Whitening and dim. reduction

10 $\quad \mathbf{z}_{curr} \leftarrow \mathbf{S}^T\mathbf{u}$

11 $\quad$ **if** $t > 1$ **then**

12 $\quad\quad \mathbf{z} \leftarrow \left(\mathbf{z}_{curr} - \mathbf{z}_{prev}\right)$ //Approx. derivative

$\quad\quad$ //For seq. addition (γ)

13 $\quad\quad \mathbf{v}^{\gamma} \leftarrow$ CCIPCA-Update $(\mathbf{v}^{\gamma}, 1, \mathbf{z}, \eta_t^{PCA})$

14 $\quad\quad \gamma \leftarrow \mathbf{v}^{\gamma}/\|\mathbf{v}^{\gamma}\|$

$\quad\quad$ //Covariance-free Incremental MCA

15 $\quad\quad \hat{\phi} \leftarrow$ CIMCA-Update $(\hat{\phi}, J, \dot{\mathbf{z}}, \gamma, \eta_t^{MCA})$

16 $\quad$ **end**

17 $\quad \mathbf{y} \leftarrow \mathbf{z}_{curr}^T\hat{\phi}$ //Slow feature output

18 **end**

Sense

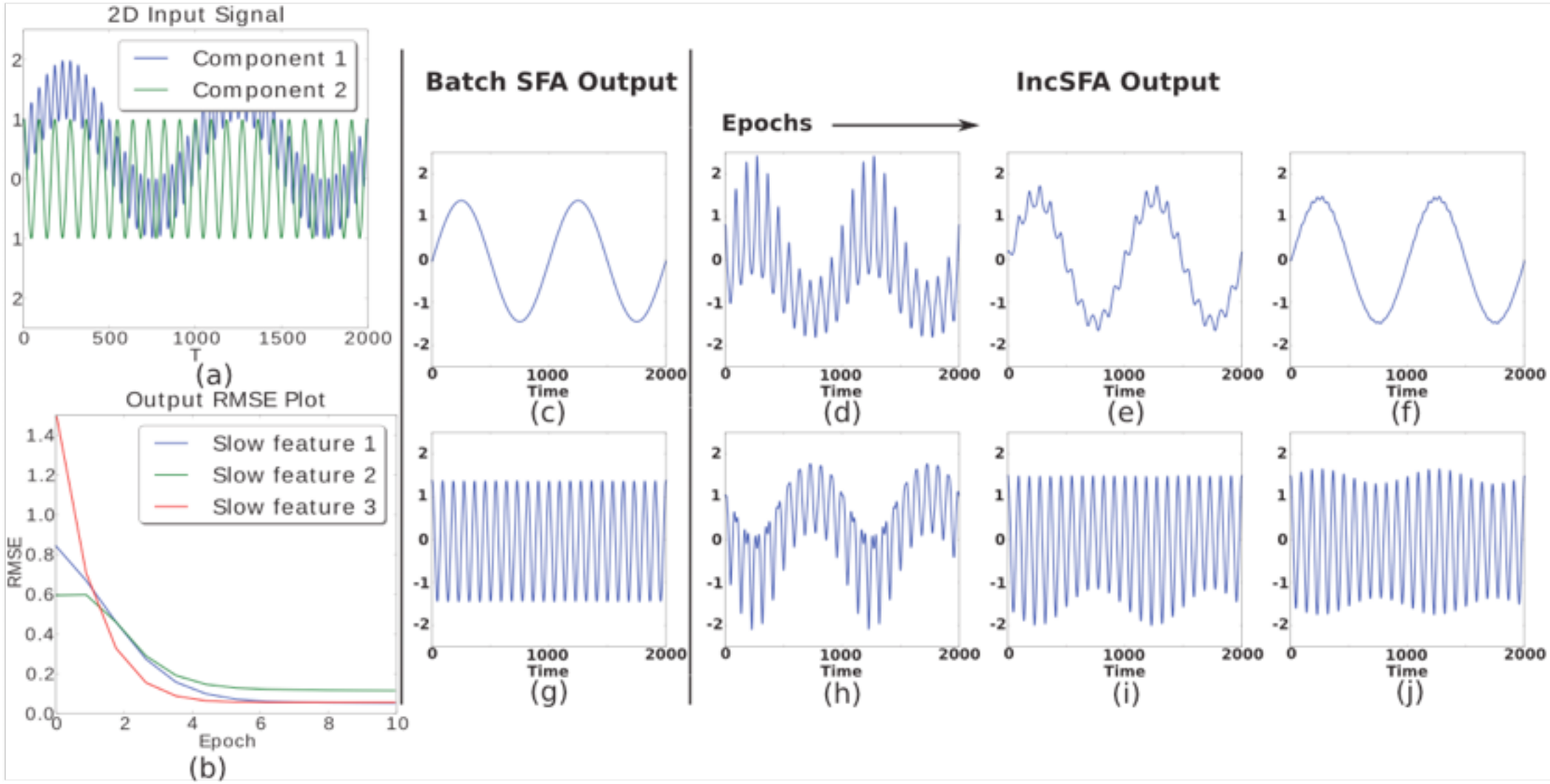
Update mean & de-mean the input

Update whitening matrix & normalize the input

Compute derivative of whitened input

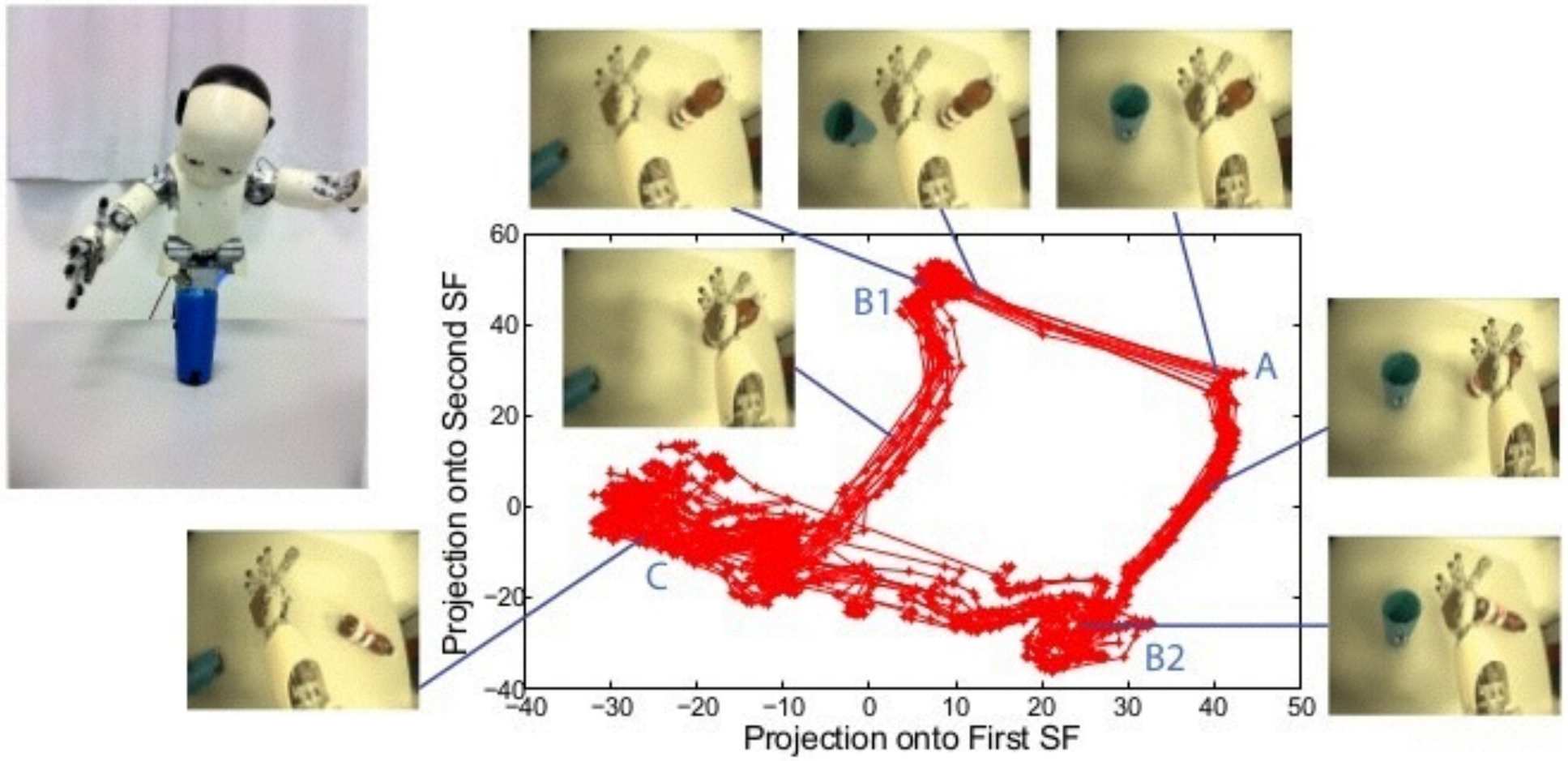
Update minor component vectors

Experimental Results



- http://www.idsia.ch/~kompella/mywork/WCCI_Tutorial.html

Experimental Results



- http://www.idsia.ch/~kompella/mywork/WCCI_Tutorial.html

Take-Home Message & Questions?

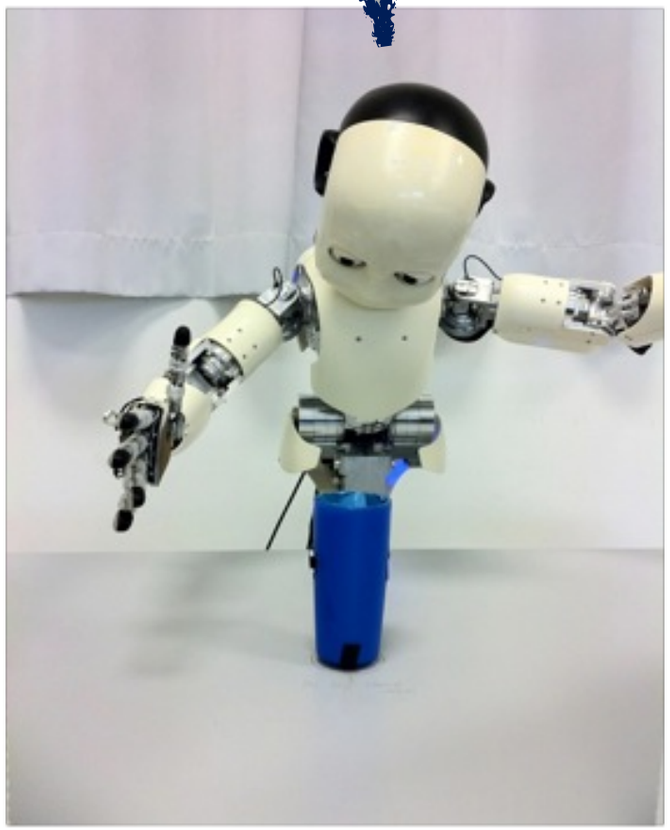
Take-Home Message & Questions?

- ☐ IncSFA is a low-complex incremental slow feature analysis algorithm.

Take-Home Message & Questions?

- IncSFA is a low-complex incremental slow feature analysis algorithm.
- FAQs:
  - Is IncSFA truly incremental?
  - How easy is it to set the learning rates?
  - Do IncSFA features always converge to their batch counterparts?
  - Is IncSFA code available for public?
    - **Python**: http://www.idsia.ch/~kompella/codes/IncSFA.zip
    - **Matlab**: http://www.idsia.ch/~luciw/incsfa.html

5 minutes break!



Recap

Recap

- **Task**: An *online* learning framework that is
  — curiosity-driven and
  — enables skills acquisition from
  — high-dimensional video inputs for humanoid robots.

Recap

- **Task**: An *online* learning framework that is
  — curiosity-driven and
  — enables skills acquisition from
  — high-dimensional video inputs for humanoid robots.

- How to make the robot curious?

  - The theory of Artificial Curiosity

Recap

- **Task**: An *online* learning framework that is
  — curiosity-driven and
  — enables skills acquisition from
  — high-dimensional video inputs for humanoid robots.

- How to make the robot curious?

  - The theory of Artificial Curiosity

- How to handle high dimensional video inputs online?

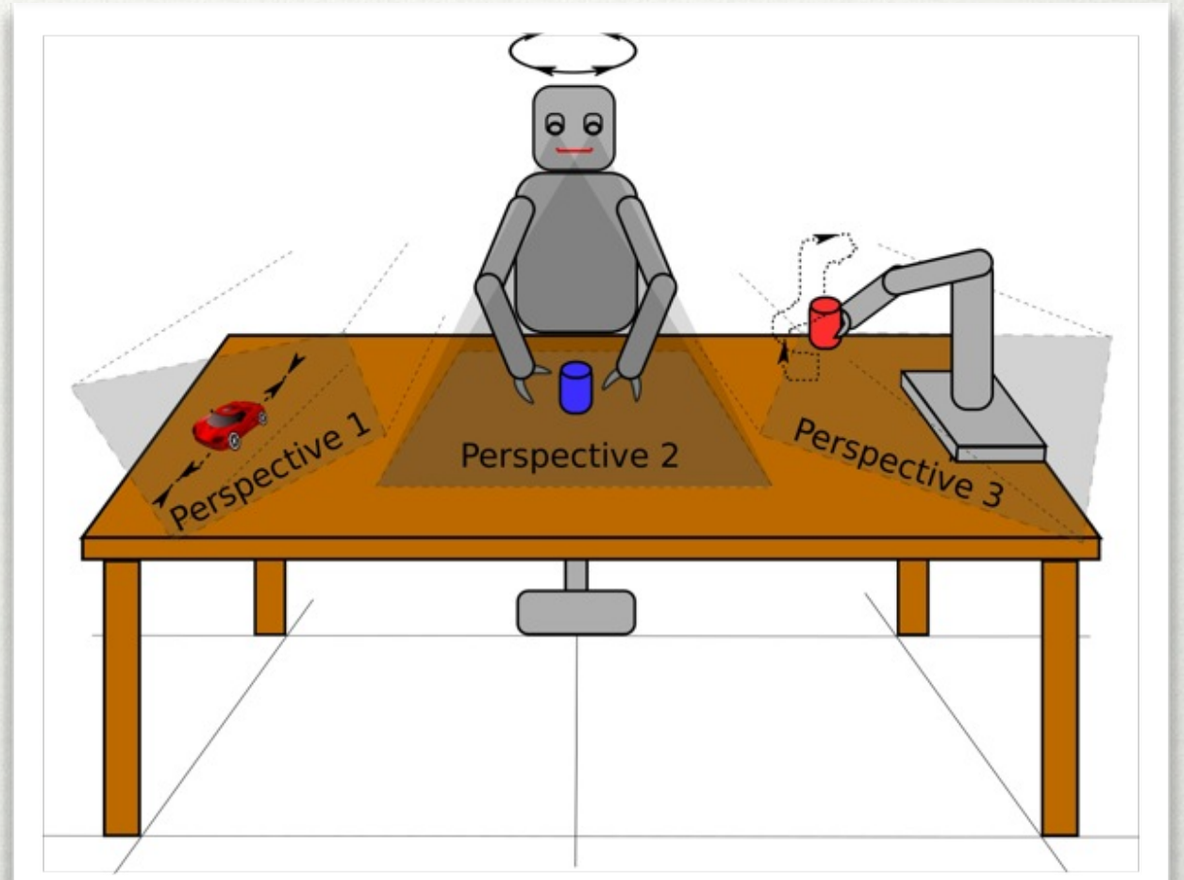
  - Incremental Slow Feature Analysis (IncSFA)

IncSFA for Open-ended RL?

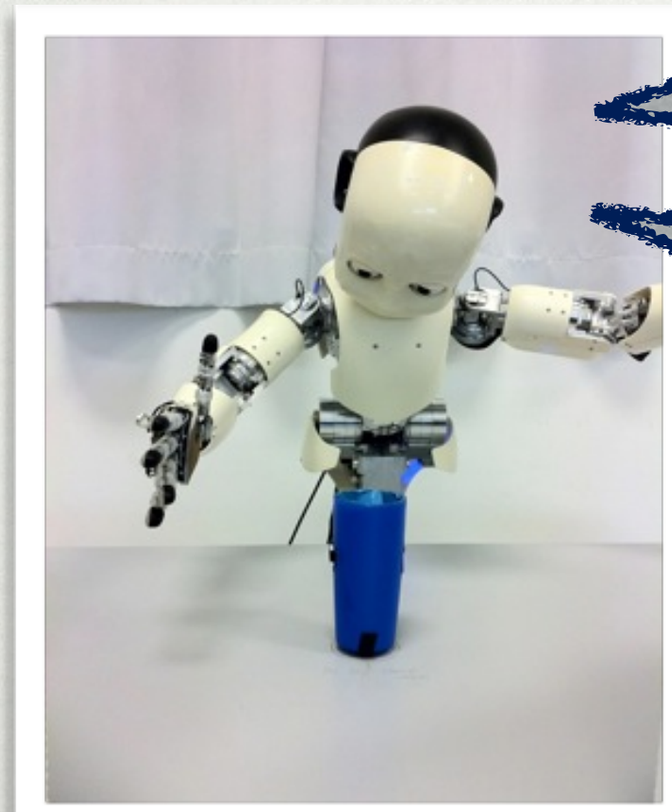
- IncSFA gradually forgets previously learned representations.

IncSFA for Open-ended RL?

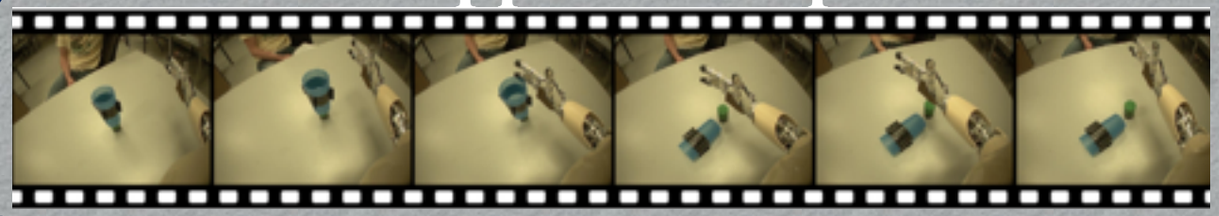
- IncSFA gradually forgets previously learned representations.



Curiosity Driven
Modular Incremental Slow Feature Analysis

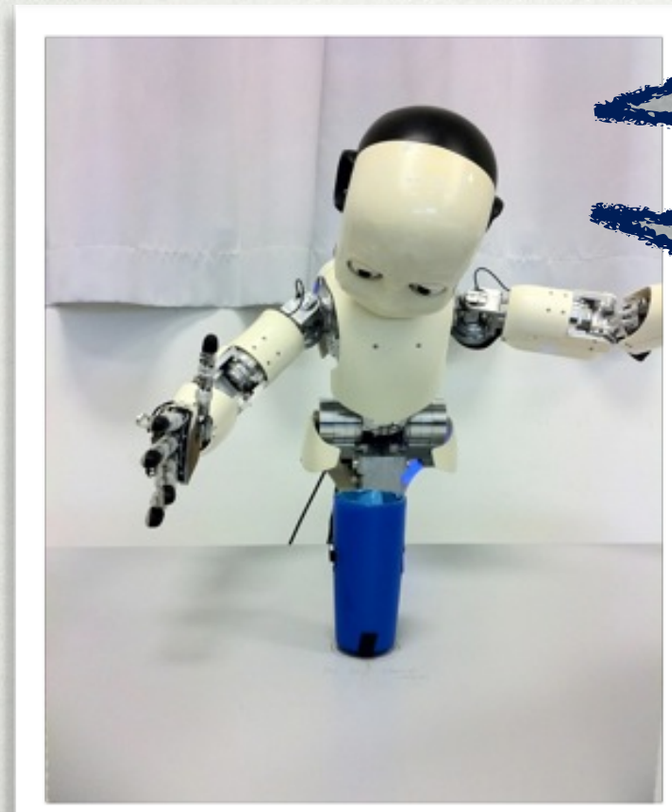


Topple the cup

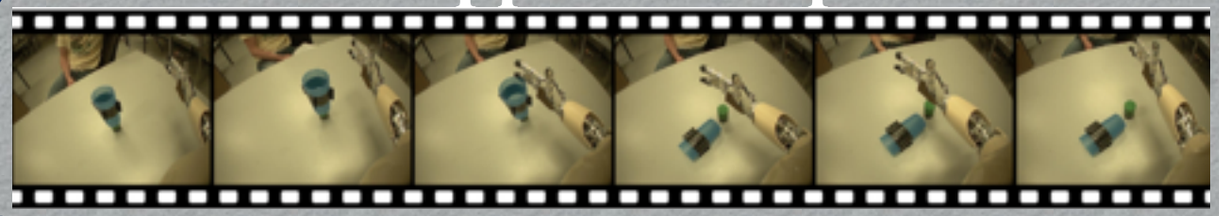


Grasp the cup

Curiosity Driven
Modular Incremental Slow Feature Analysis

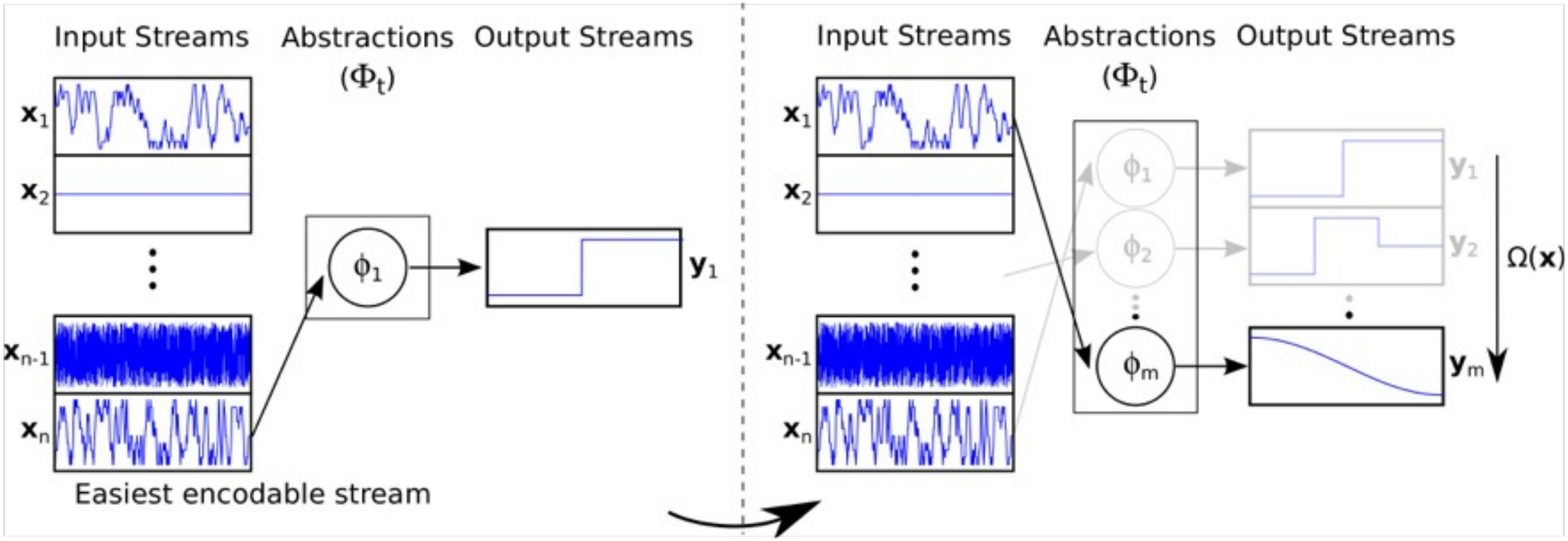


Topple the cup

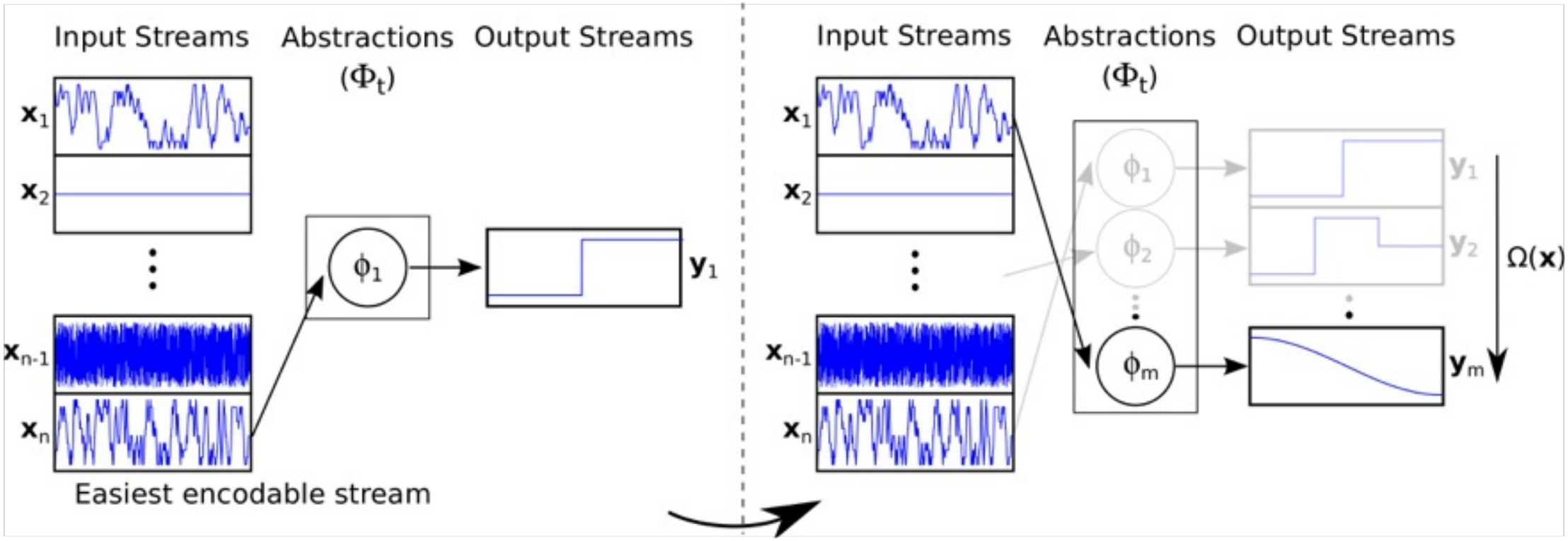


Grasp the cup

Curious Dr. MISFA
(Kompella et al. 2012)

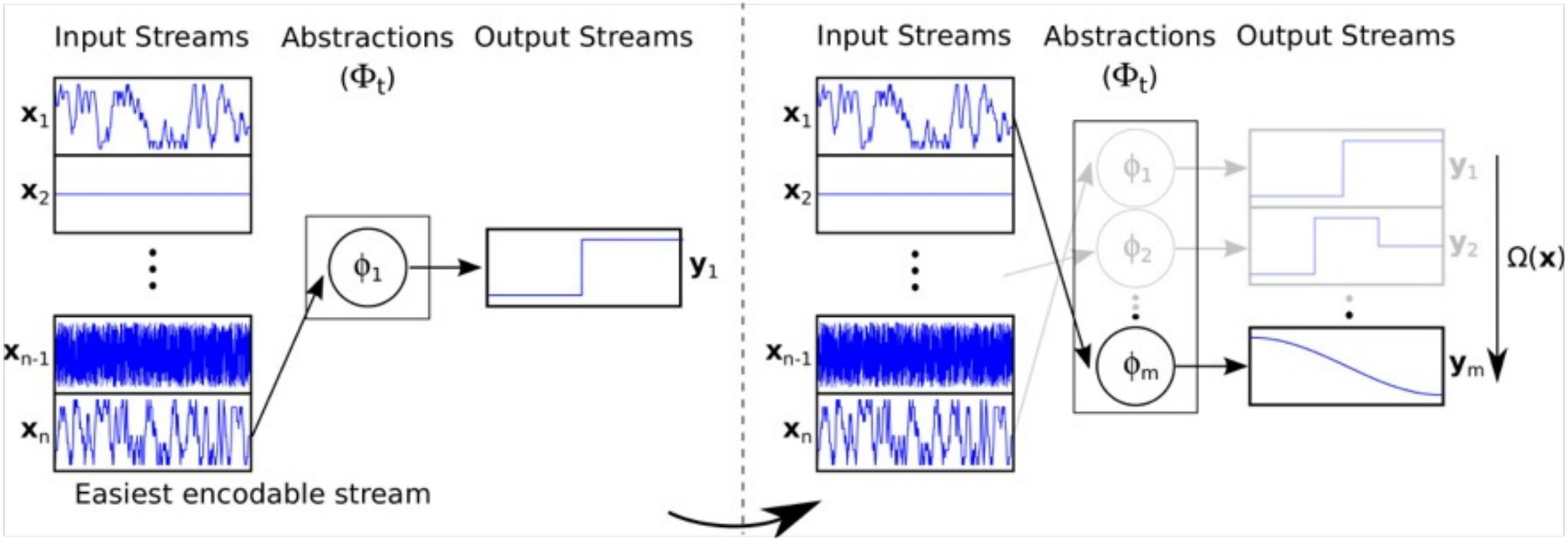


Curious Dr. MISFA
(Kompella et al. 2012)



Updating Abstraction

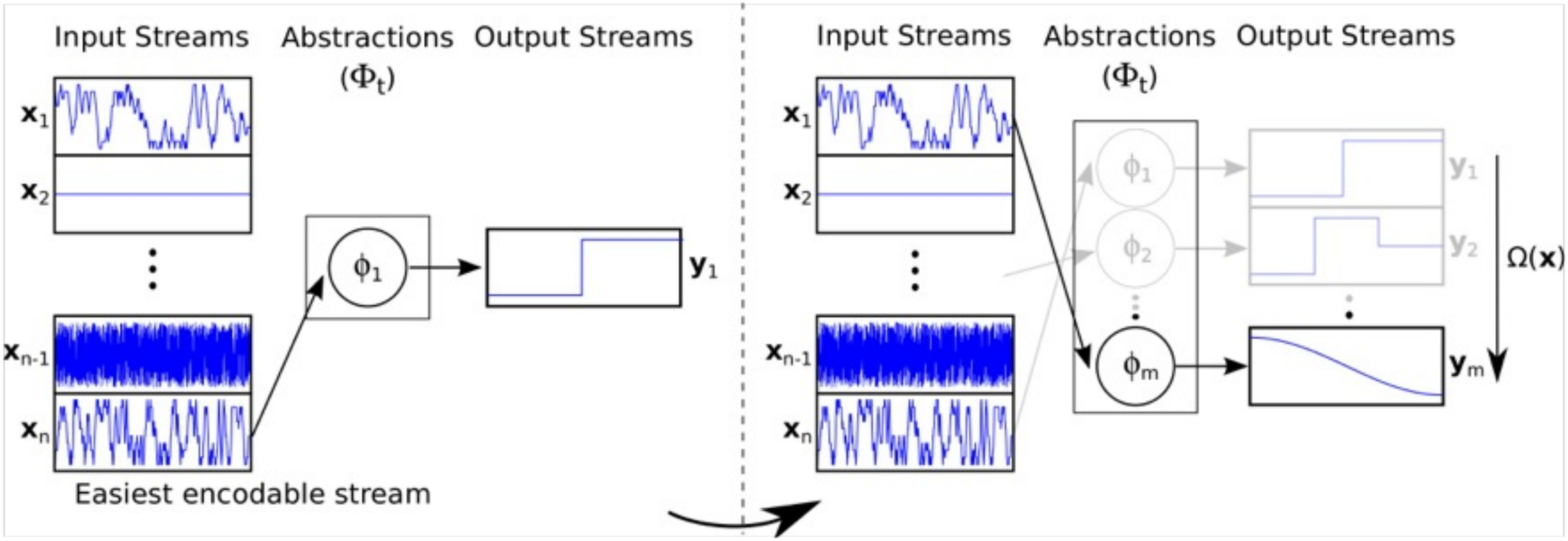
Curious Dr. MISFA
(Kompella et al. 2012)



Updating Abstraction

Updating Input-Stream Selection Policy

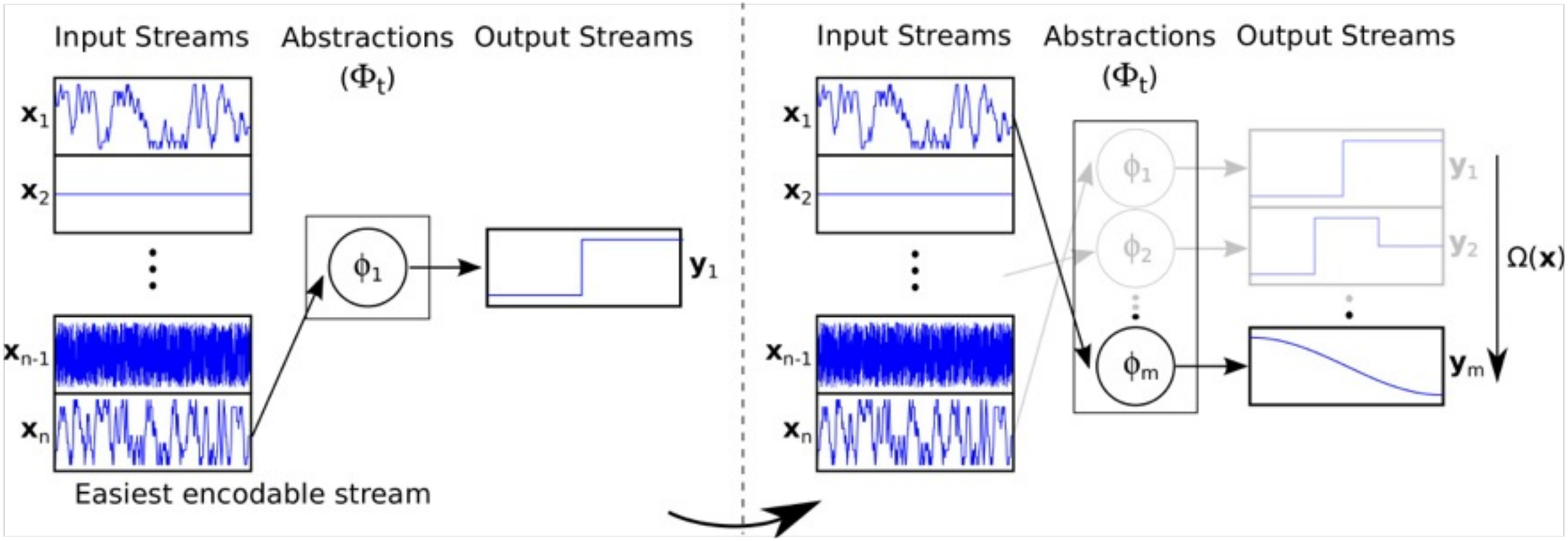
Curious Dr. MISFA
(Kompella et al. 2012)



Updating Abstraction

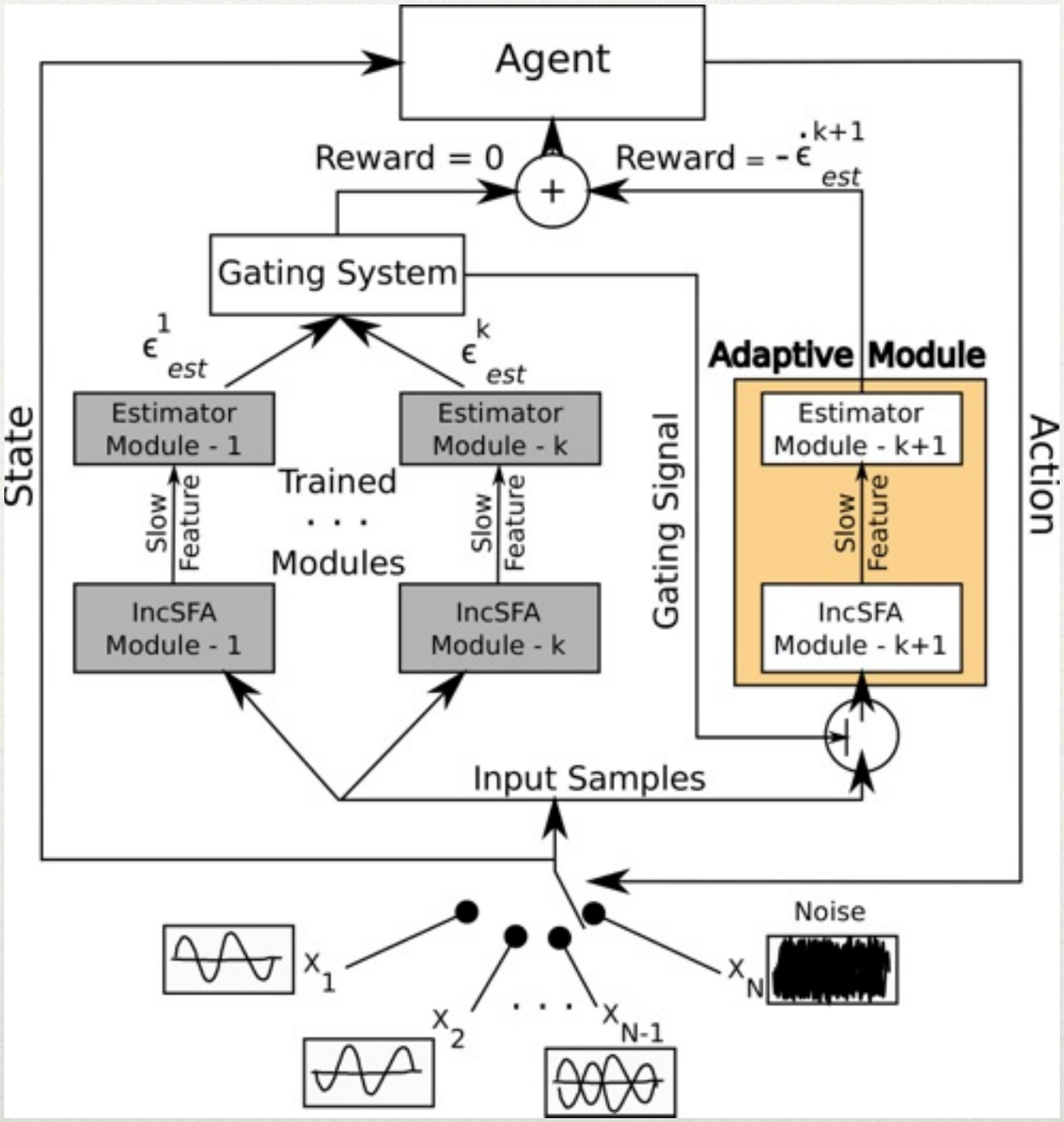
Updating Input-Stream Selection Policy

Curious Dr. MISFA
(Kompella et al. 2012)



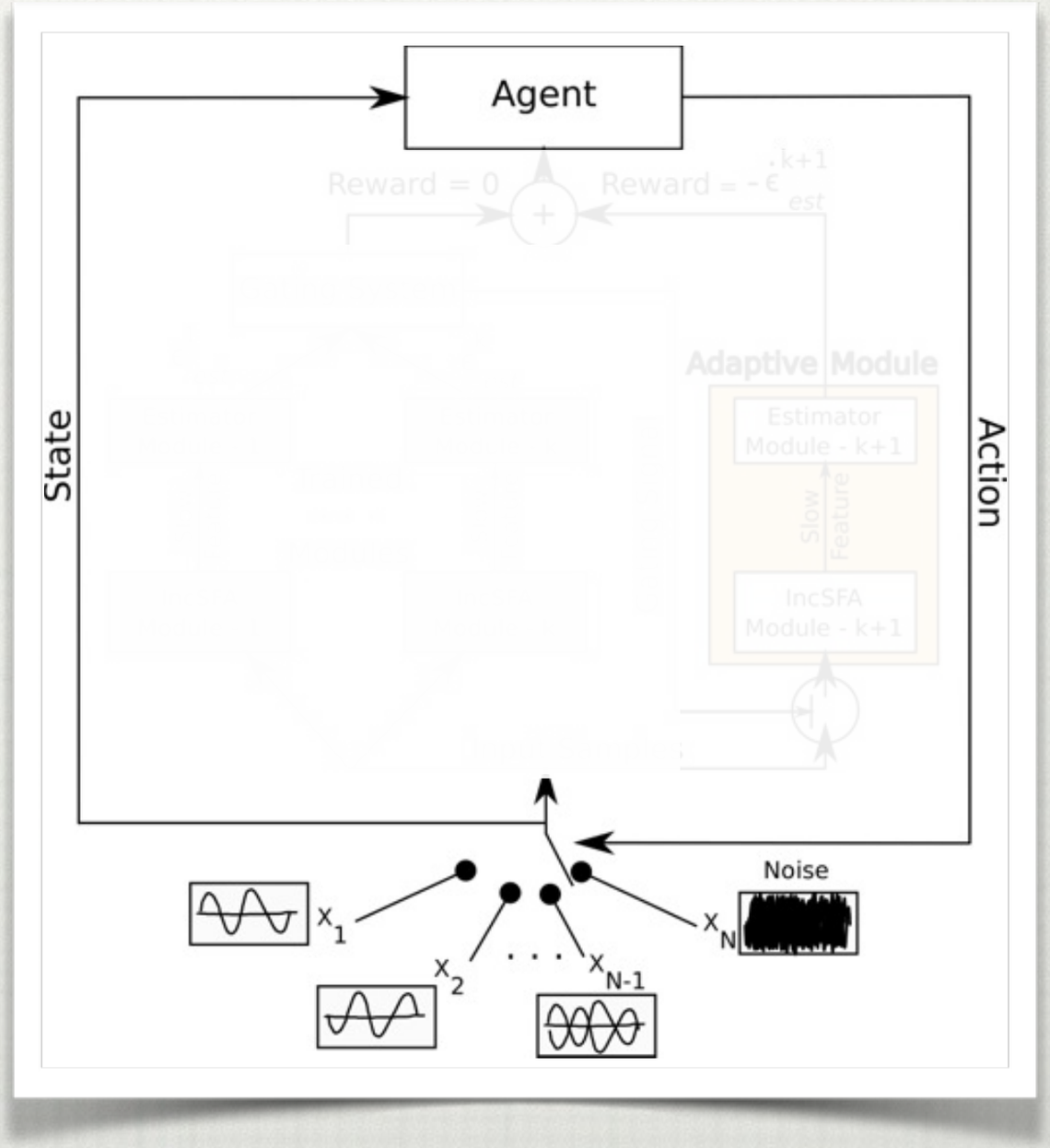
Updating Abstraction ~ **Solved** Updating Input-Stream Selection Policy

Curious Dr. MISFA Architecture



Curious Dr. MISFA Architecture

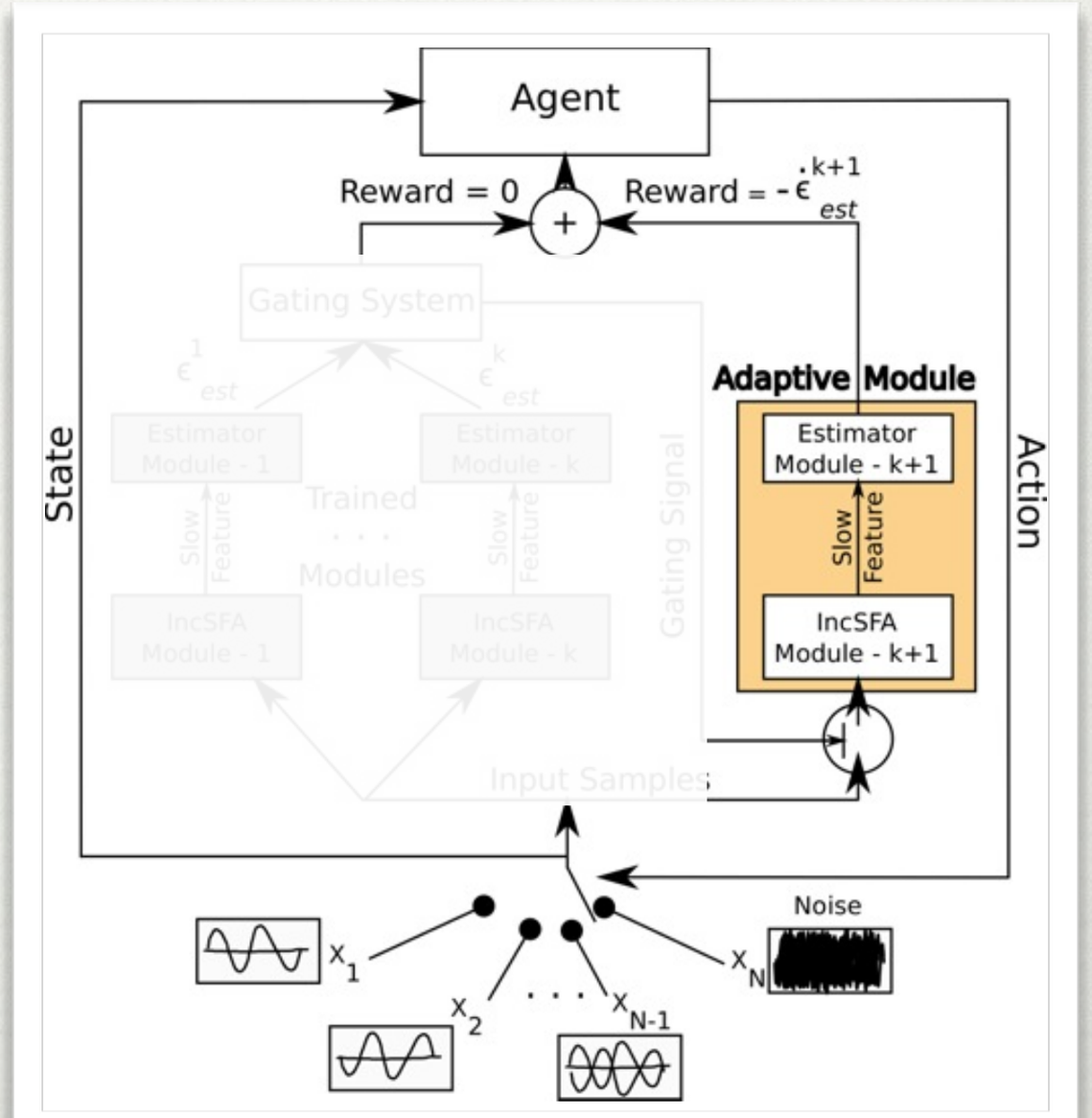
1. An internal RL agent



Curious Dr. MISFA Architecture

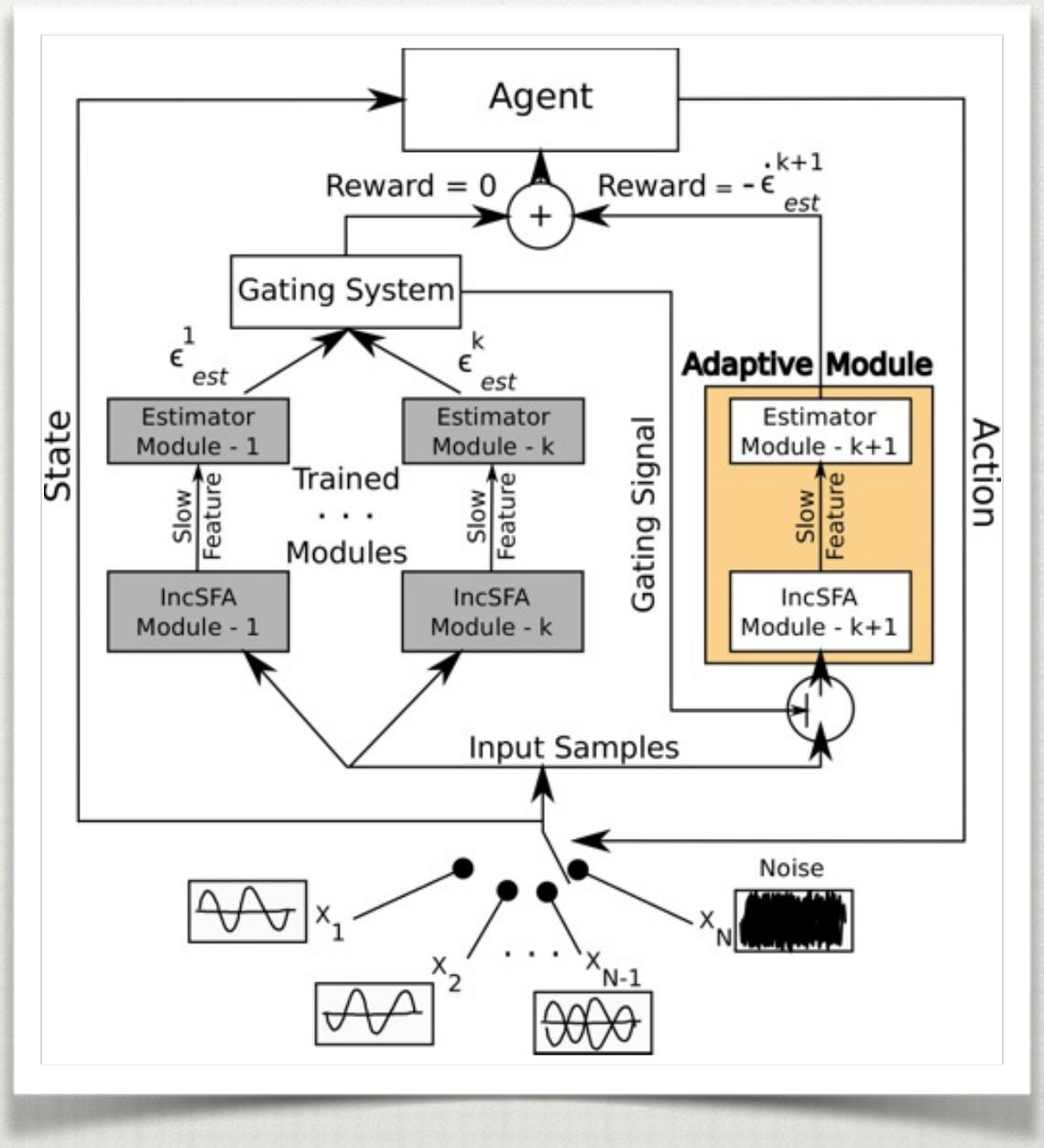
1. An internal RL agent

2. An adaptive IncSFA-ROC abstraction module

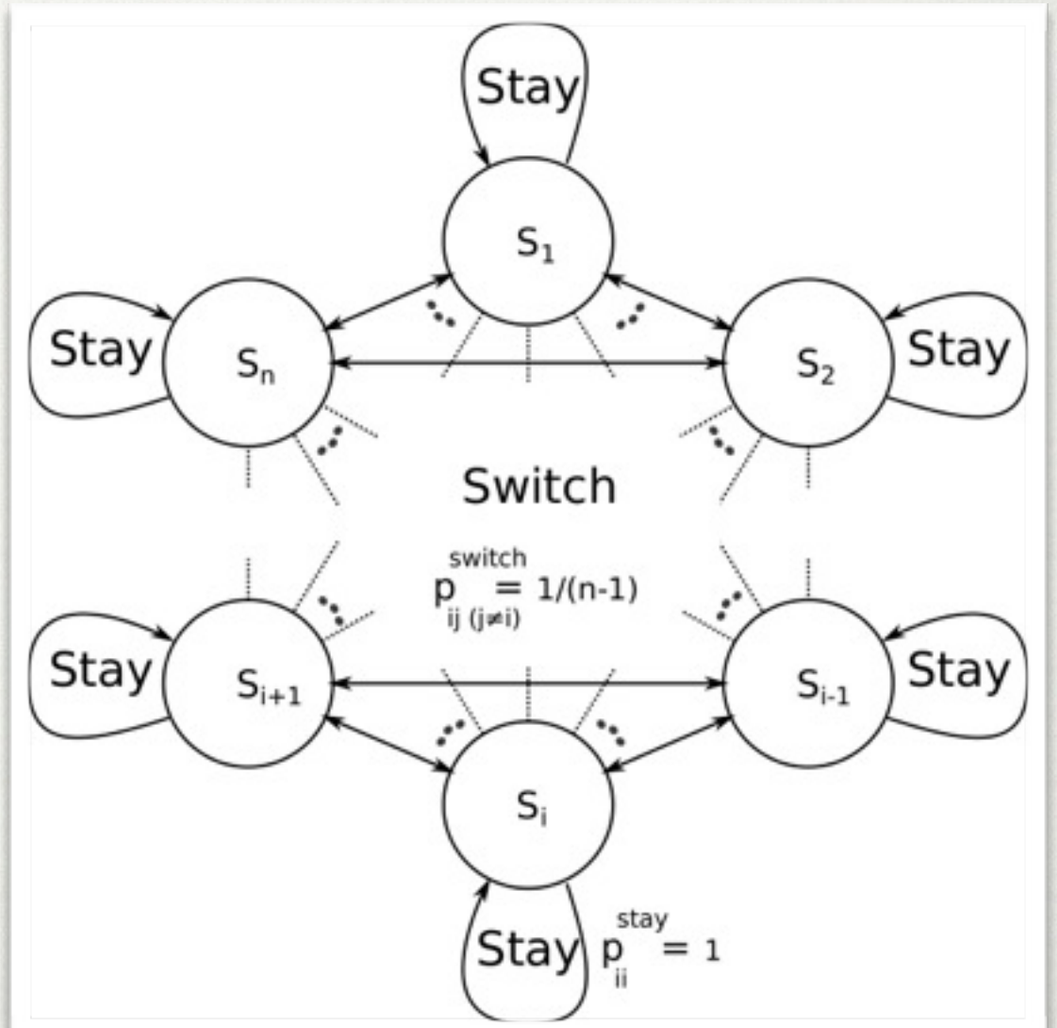


Curious Dr. MISFA Architecture

1. An internal RL agent
2. An adaptive IncSFA-ROC abstraction module
3. A gating system

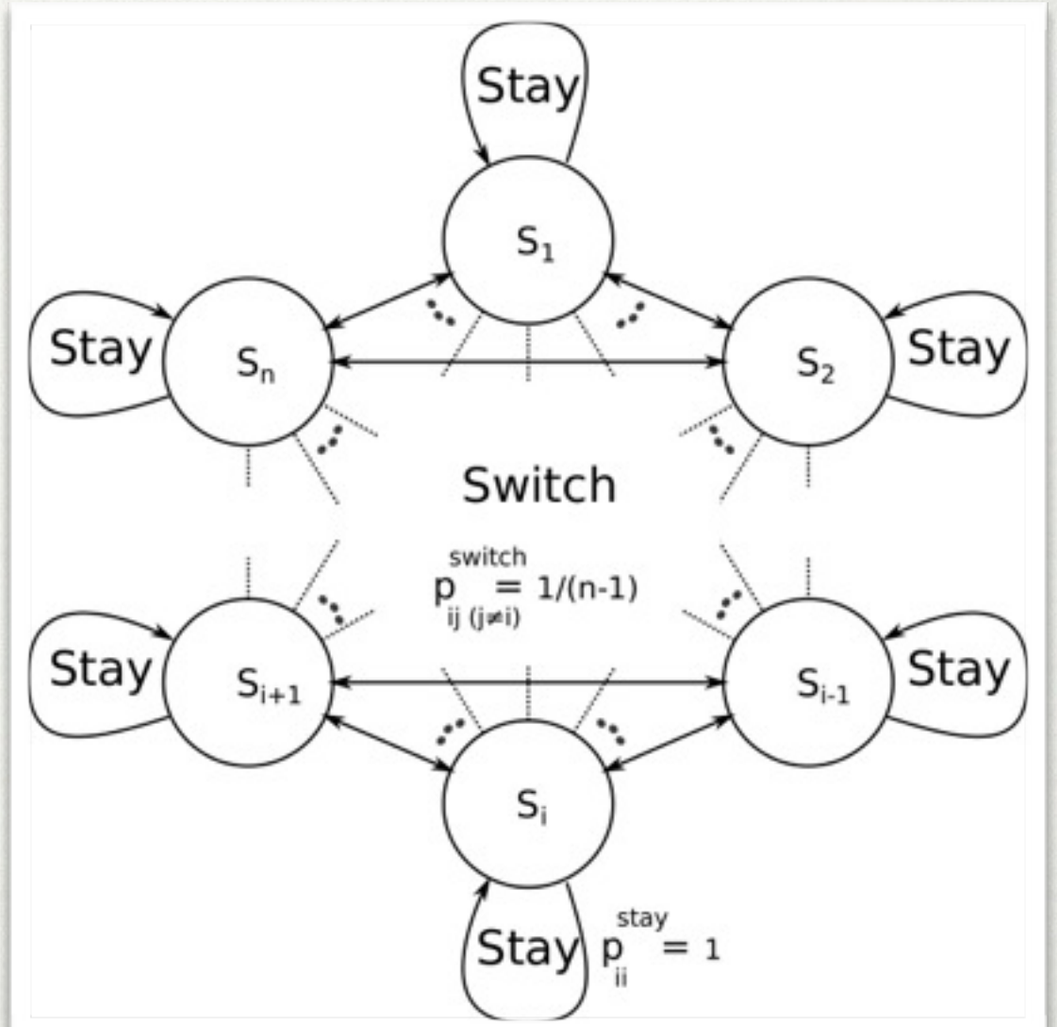


(1) Curious Dr. MISFA's RL Agent



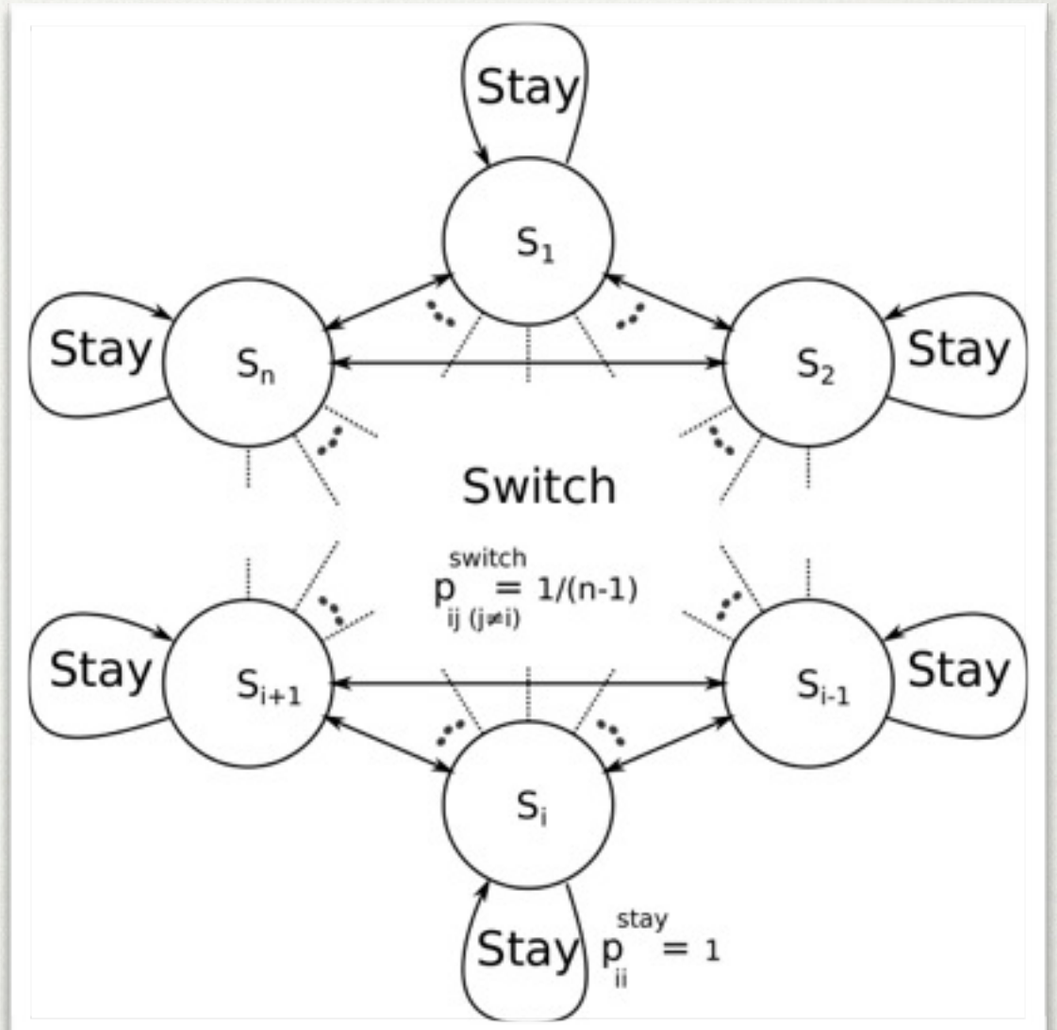
(1) Curious Dr. MISFA's RL Agent

- Internal Environment



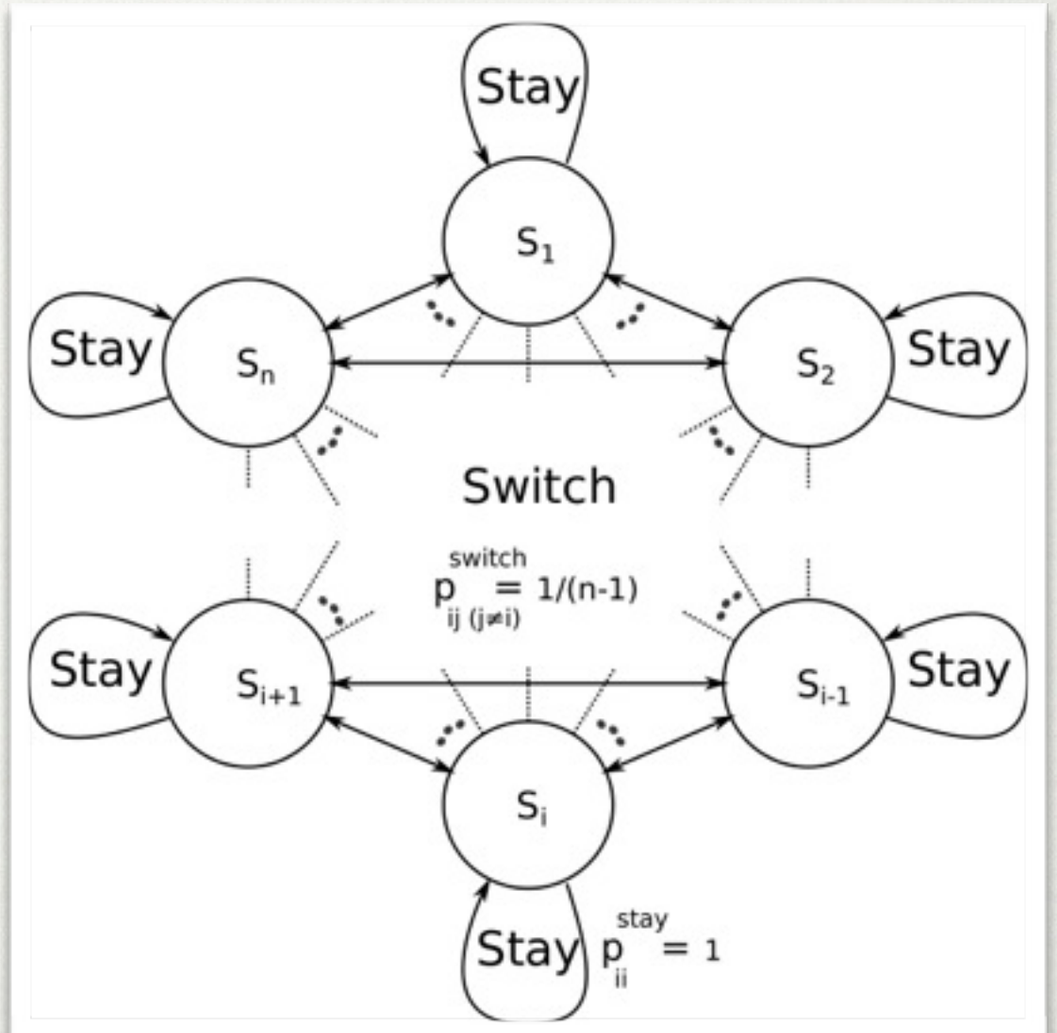
(1) Curious Dr. MISFA's RL Agent

- Internal Environment
- Discrete states (S^{int}): $\{s_1^{int}, \ldots, s_n^{int}\}$ equal to the number of input streams



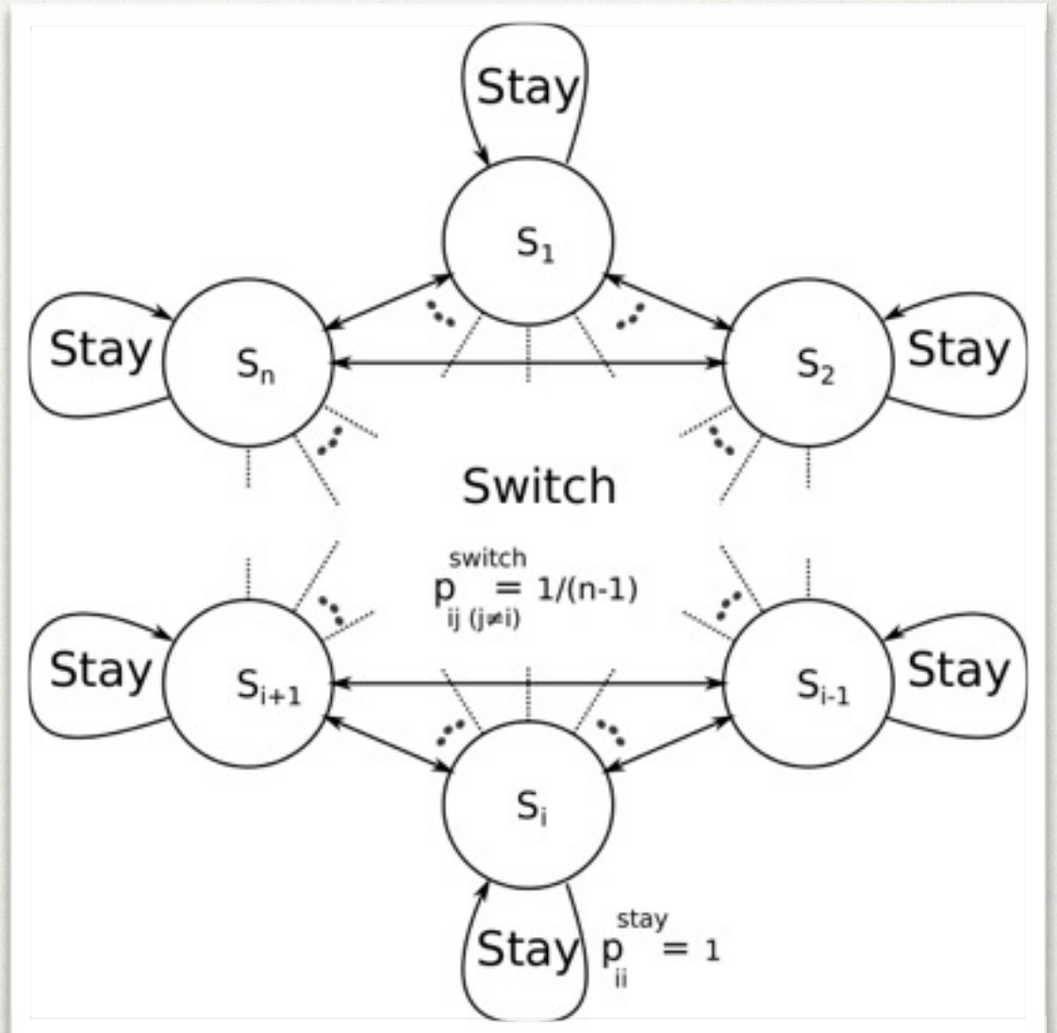
(1) Curious Dr. MISFA's RL Agent

- Internal Environment
- Discrete states (S^{int}): $\{s_1^{int}, \ldots, s_n^{int}\}$ equal to the number of input streams
- At each state s_i^{int} the agent observes τ-samples from x_i.



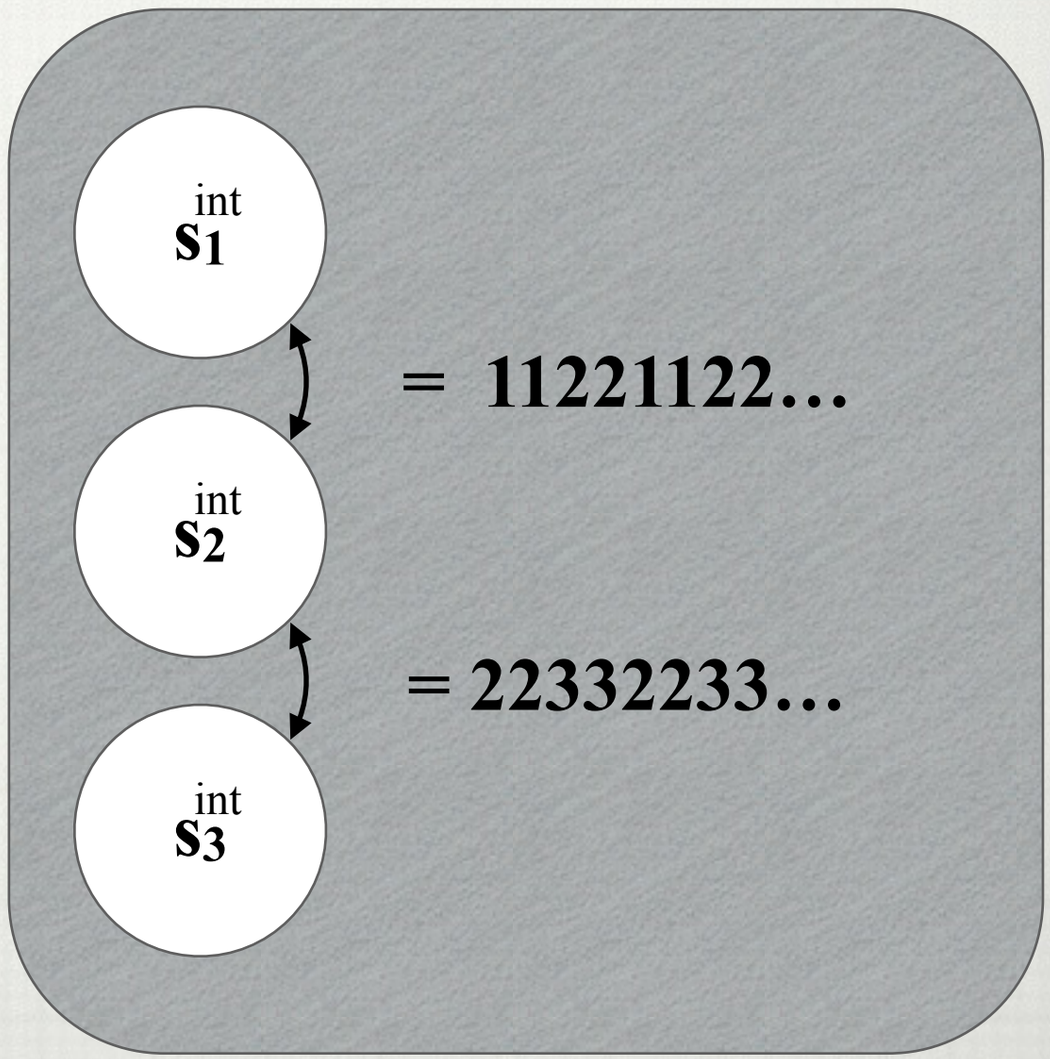
(1) Curious Dr. MISFA's RL Agent

- Internal Environment
- Discrete states (S^{int}): $\{s_1^{int}, \ldots, s_n^{int}\}$ equal to the number of input streams
- At each state s_i^{int} the agent observes τ-samples from x_i.
- Actions (A^{int}): {stay, switch}



(1) Curious Dr. MISFA's RL Agent

- Internal Environment
- Discrete states (S^{int}): $\{s_1^{int}, \ldots, s_n^{int}\}$ equal to the number of input streams
- At each state s_i^{int} the agent observes τ-samples from x_i.
- Actions (A^{int}): {stay, switch}



(2) Curious Dr. MISFA's Abstraction Estimator

- Coupled IncSFA-ROC algorithm

(2) Curious Dr. MISFA's Abstraction Estimator

- Coupled IncSFA-ROC algorithm
- IncSFA learns slow feature abstractions from input.

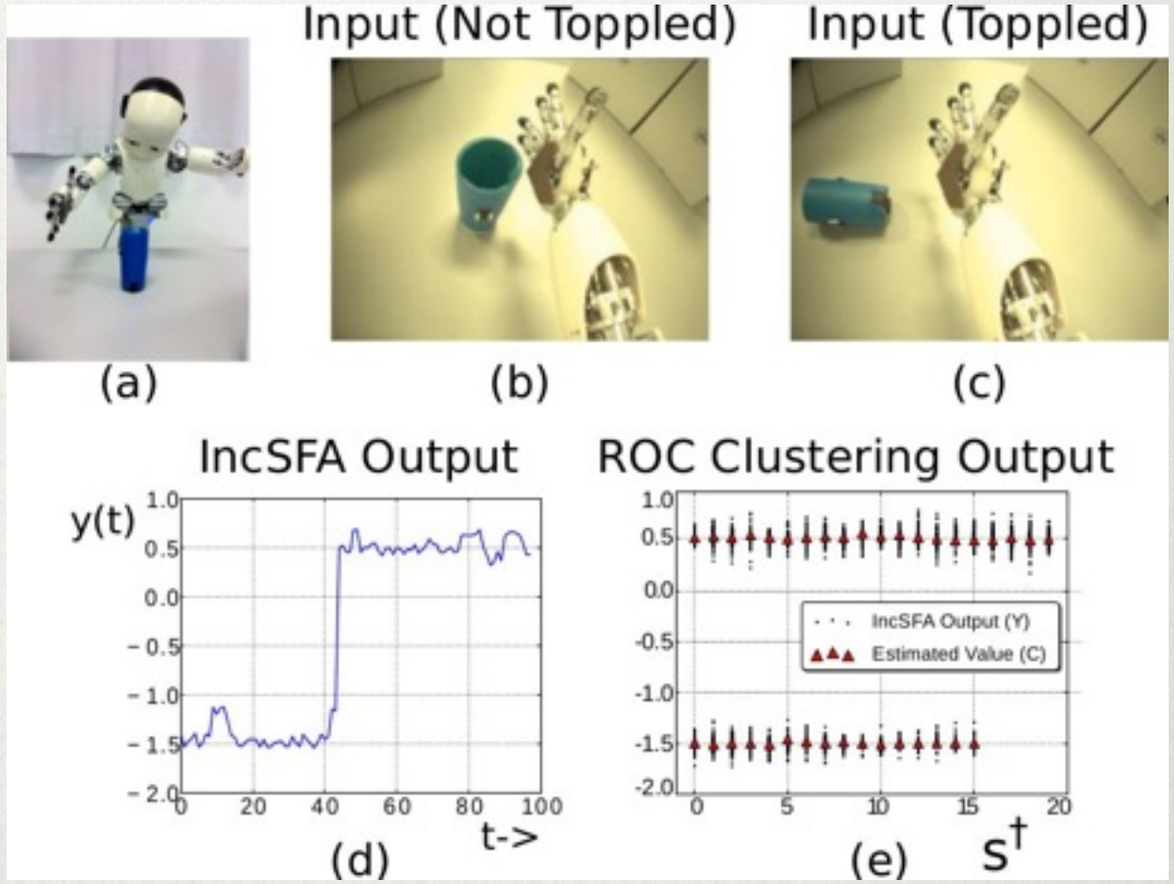
(2) Curious Dr. MISFA's Abstraction Estimator

- Coupled IncSFA-ROC algorithm
- IncSFA learns slow feature abstractions from input.
- Robust Online Clustering (ROC; Guedalia et al.) is a clustering algorithm that learns a discrete mapping between the slow feature outputs w.r.t. to the *subjective* state space ($S^{\dagger}$).

  — Online agglomerative clustering algorithm
  — Cluster centres merge if they are similar to each other

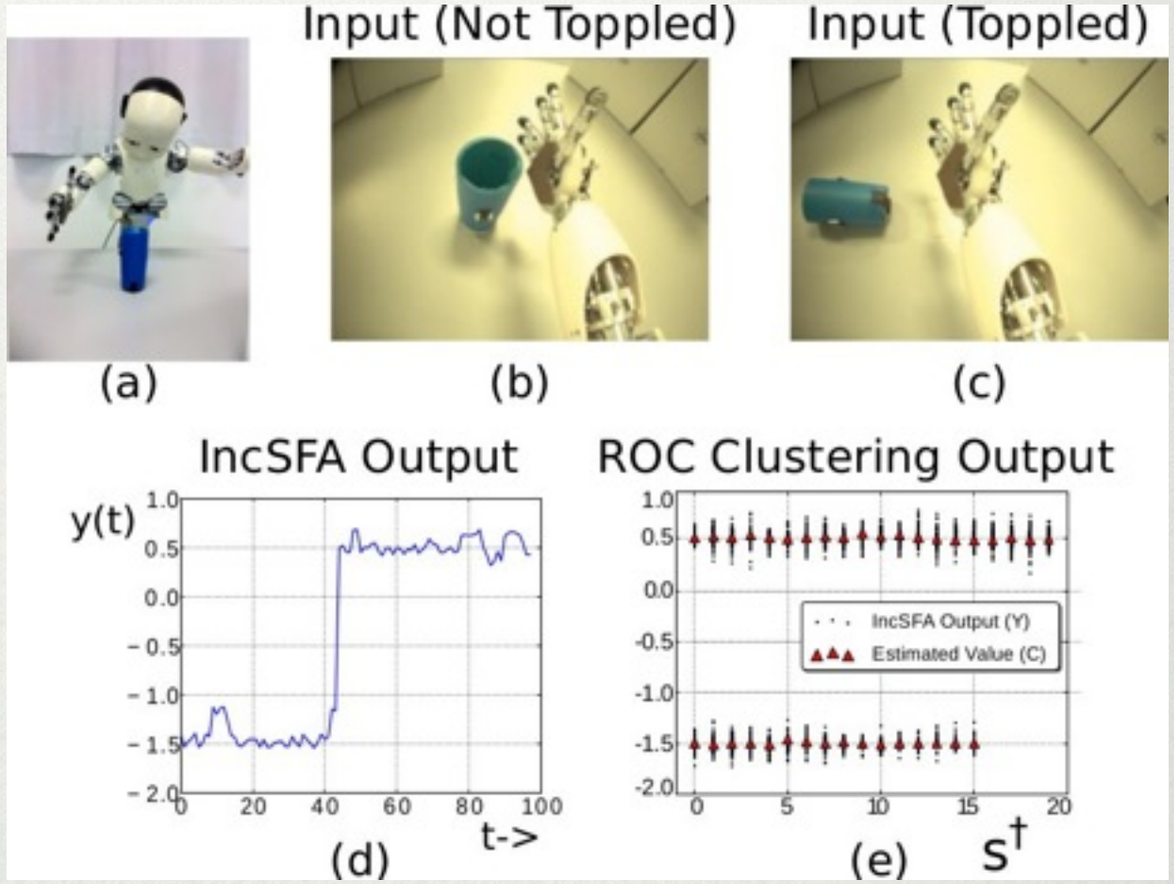
Subjective State Space ($S^\dagger$)

- Proprioceptive states and discrete values of previously encoded abstraction outputs.



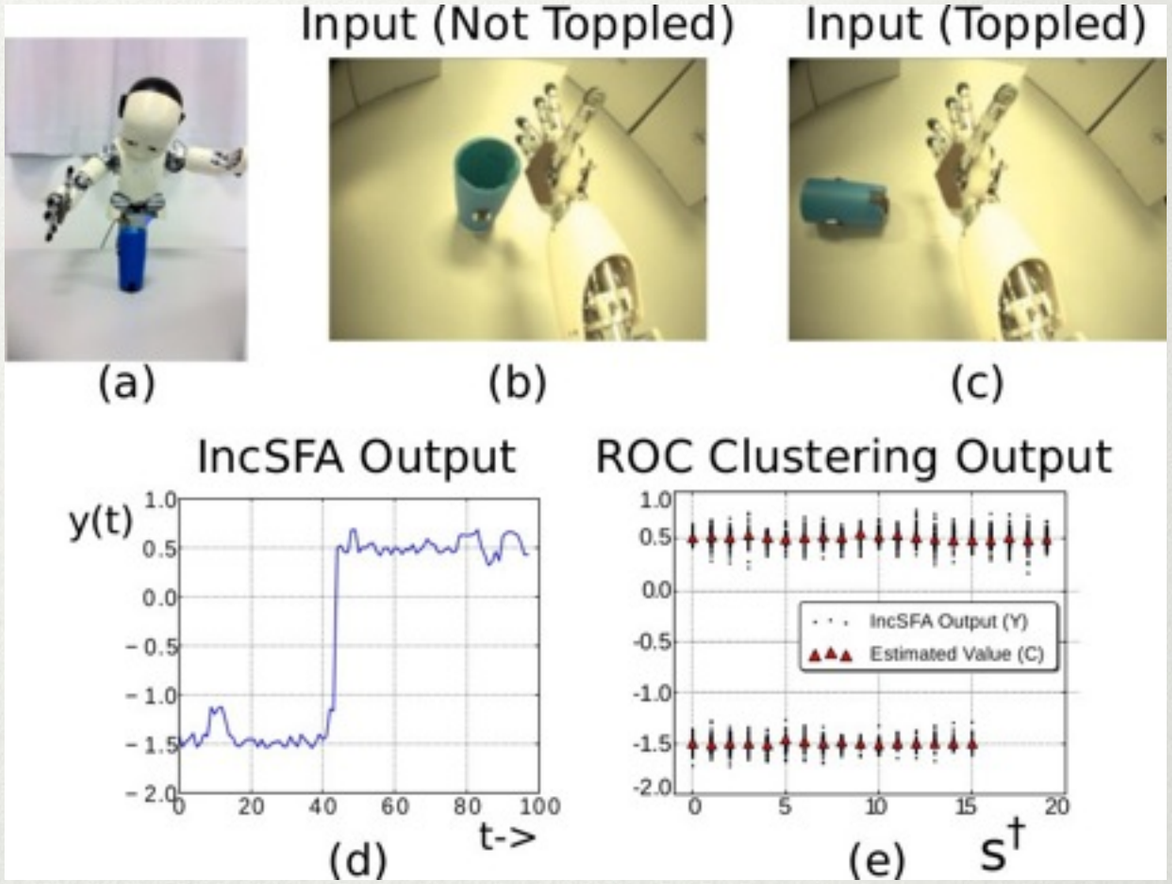
Subjective State Space ($S^\dagger$)

- Proprioceptive states and discrete values of previously encoded abstraction outputs.
- Multiple nodes of ROC algorithm are used to map slow-feature outputs to the subjective states.

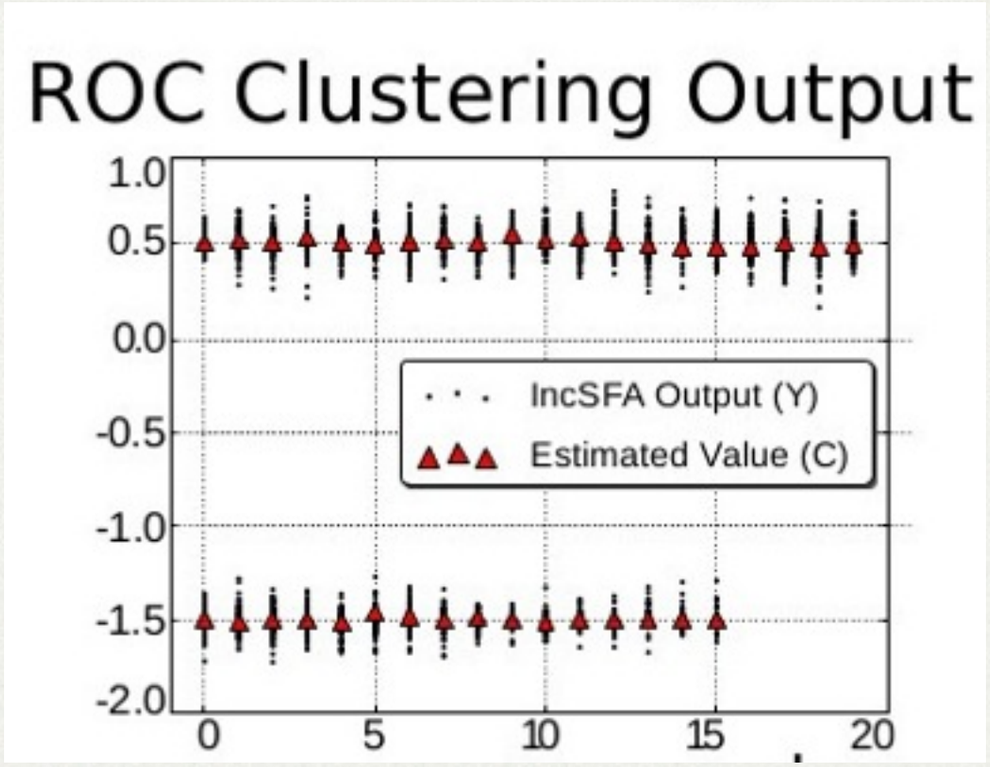


Subjective State Space ($S^{\dagger}$)

- Proprioceptive states and discrete values of previously encoded abstraction outputs.
- Multiple nodes of ROC algorithm are used to map slow-feature outputs to the subjective states.
- Dimensionality of the subjective-state space increases as the robot acquires more abstractions.

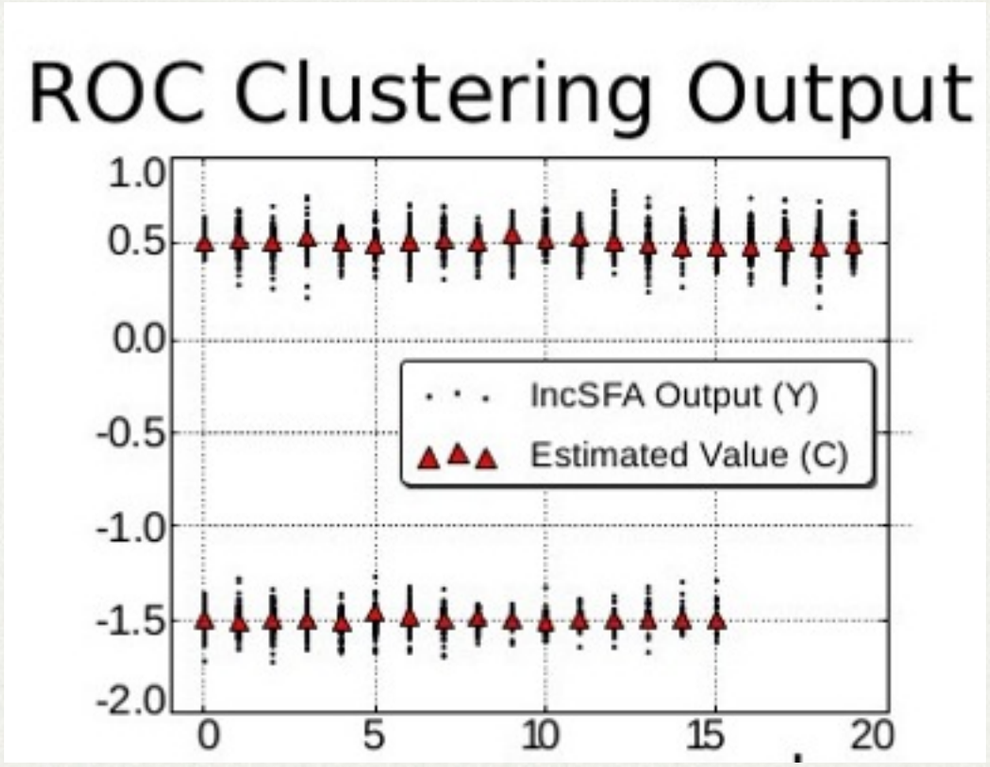


Curiosity Reward



- For each slow-feature output, the closest ROC node is activated and the corresponding mean-squared estimation error is computed.

Curiosity Reward



- For each slow-feature output, the closest ROC node is activated and the corresponding mean-squared estimation error is computed.
- A total estimation error is computed as a sum of stored errors corresponding to all the nodes.

Curiosity Reward

- Derivative of the total estimation error (ξ) is calculated via backward difference approximation:

$$\dot{\xi}(t) = \xi(t) - \xi(t\text{-}1)$$

Curiosity Reward

- Derivative of the total estimation error (ξ) is calculated via backward difference approximation:

$$\dot{\xi}(t) = \xi(t) - \xi(t\text{-}1)$$

- The agent's reward function is updated at every iteration as follows:

Curiosity Reward

- Derivative of the total estimation error (ξ) is calculated via backward difference approximation:

$$\dot{\xi}(t) = \xi(t) - \xi(t\text{-}1)$$

- The agent's reward function is updated at every iteration as follows:

$$R^{\text{int}}(s^{\text{int}}, s^{\text{int}}_{-}, a^{\text{int}}) = (1-\eta)\, R^{\text{int}}(s^{\text{int}}, s^{\text{int}}_{-}, a^{\text{int}}) + \eta \sum_{t}^{t+\tau} -\dot{\xi}(t)$$

where $0 < \eta < 1$ is a discount factor, τ is the number of observed samples (x), $(s^{\text{int}}, s^{\text{int}}_{-}) \in \mathcal{S}^{\text{int}}$ and $a^{\text{int}} \in \{\text{stay, switch}\}$.

Input Stream Selection Policy (π^{int})

Input Stream Selection Policy (π^{int})

- $\pi^{int} : S^{int} \rightarrow \{stay, switch\}$

Input Stream Selection Policy (π^{int})

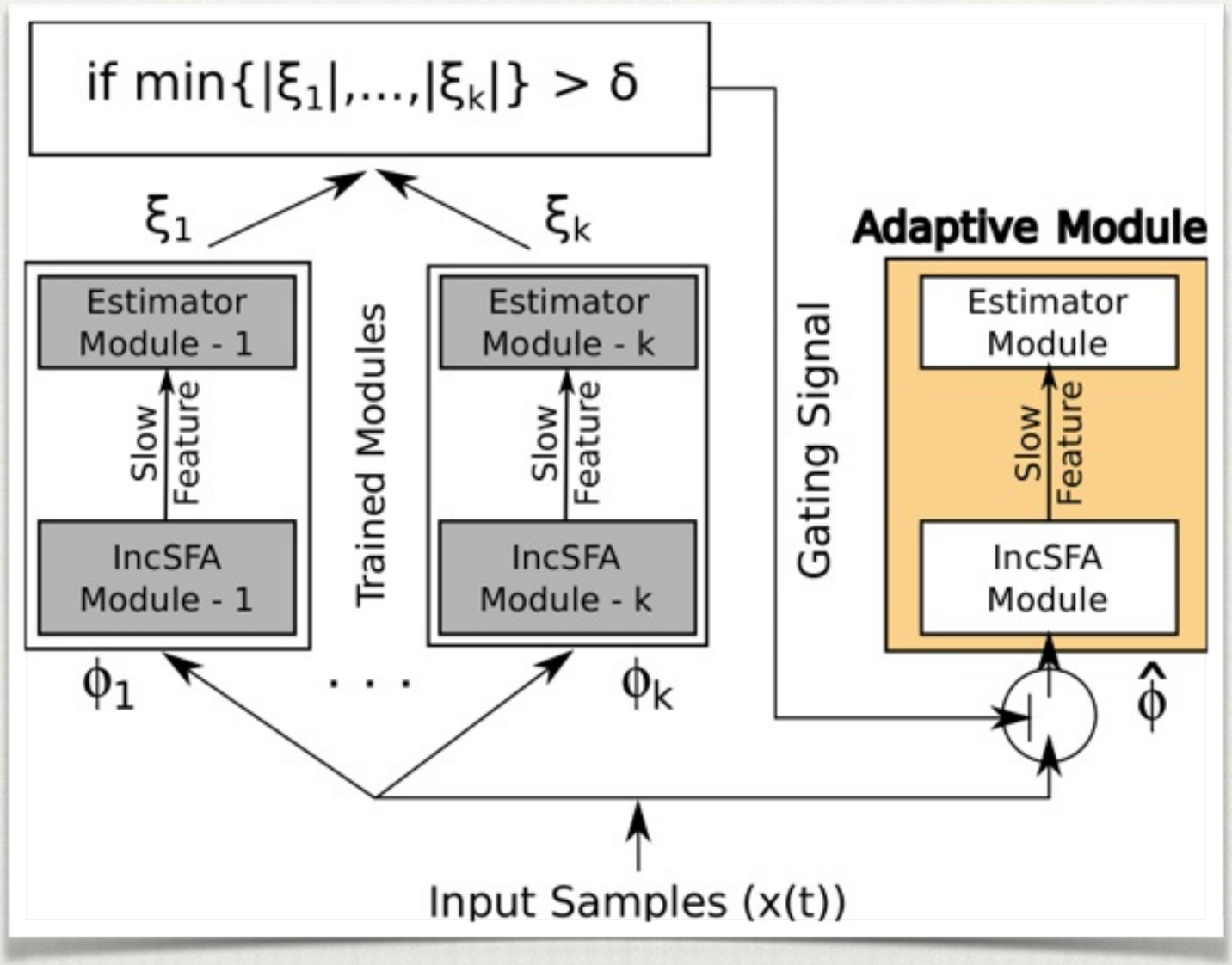
- $\pi^{int} : S^{int} \rightarrow \{stay, switch\}$

- Learned through Least-Squares Temporal Difference (LSTD) approach using the current estimate of the reward function.

Input Stream Selection Policy (π^{int})

- $\pi^{int} : S^{int} \rightarrow \{stay, switch\}$
- Learned through Least-Squares Temporal Difference (LSTD) approach using the current estimate of the reward function.
- Agent explores using a decaying epsilon-greedy strategy over the input-stream selection policy.

(c) Gating System



Experimental Validation

- Curiosity Function: Quantifies learning difficulty of the abstraction estimator ($\Omega : X \rightarrow [0,1]$)

Experimental Validation

- Curiosity Function: Quantifies learning difficulty of the abstraction estimator ($\Omega : X \rightarrow [0,1]$)
- Ω induces a total ordering among the input streams.

Experimental Validation

- Curiosity Function: Quantifies learning difficulty of the abstraction estimator ($\Omega : X \rightarrow [0,1]$)
- Ω induces a total ordering among the input streams.
- Easier to encode input streams have lower Ω values.

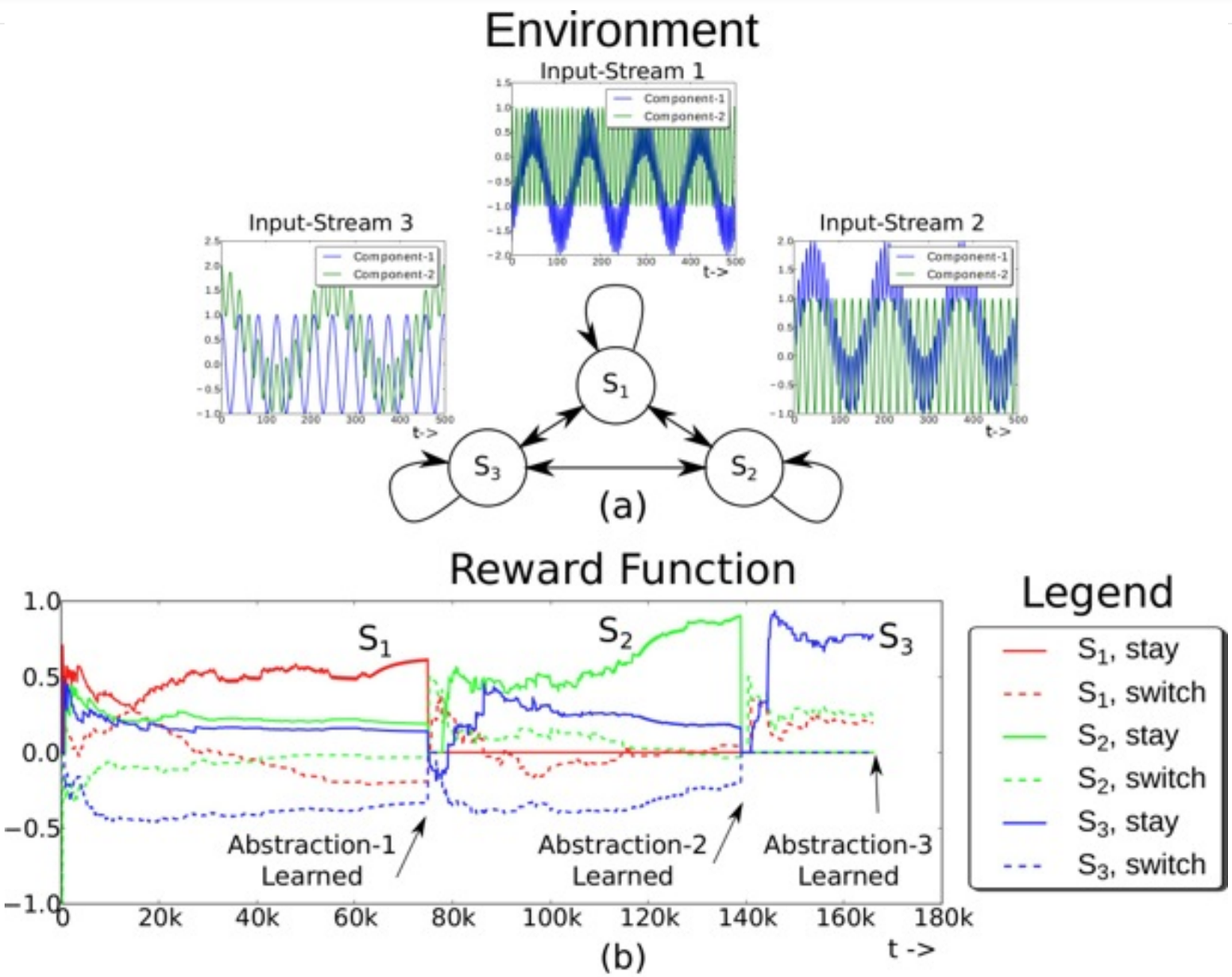
Experimental Validation

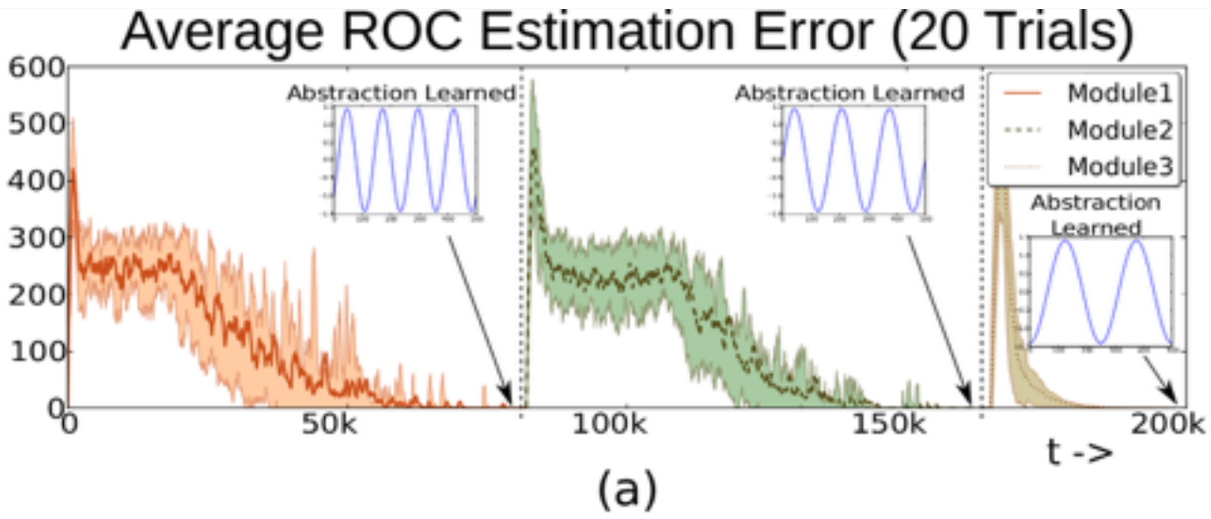
- Curiosity Function: Quantifies learning difficulty of the abstraction estimator ($\Omega : X \rightarrow [0,1]$)
- Ω induces a total ordering among the input streams.
- Easier to encode input streams have lower Ω values.

Definition 1 *We define the curiosity-function of the IncSFA algorithm for an input stream* $\mathbf{x}(t) \in \mathbb{R}^I$ *as*

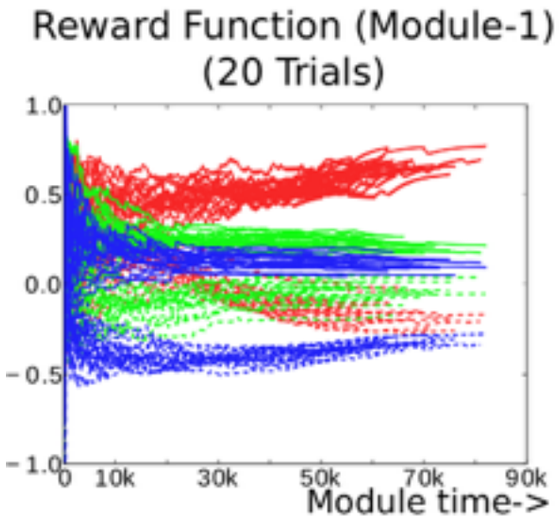
$$\Omega(\mathbf{x}) = \left[1 - \frac{\eta^{mca}(\lambda_{K-1} - \lambda_K)}{1 - \eta^{mca} - \eta^{mca}\lambda_K}\right] \tag{23}$$

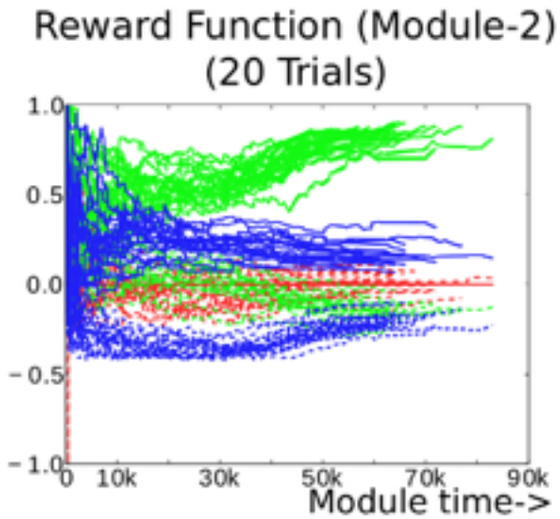
where λ_K *and* λ_{K-1} *denote the smallest two eigenvalues of* $E[\dot{\mathbf{z}}(t)\dot{\mathbf{z}}(t)^T]$, $\mathbf{z}(t) \in \mathbb{R}^K$ *is the* whitened *output of* $\mathbf{x}(t)$.

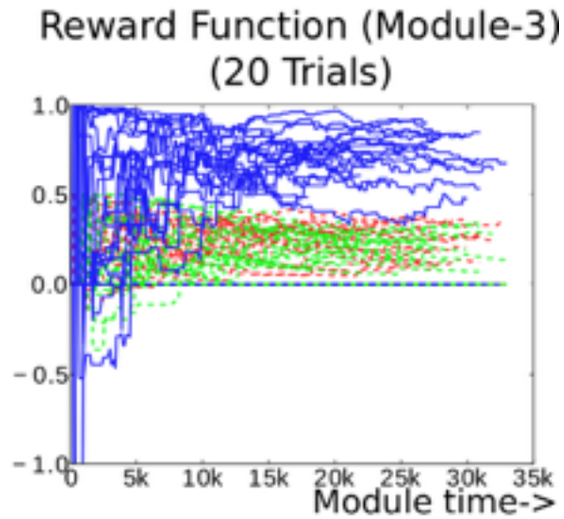

Environment
Input-Stream 1
Component-1
Component-2
Input-Stream 3
Input-Stream 2
t->
S_1
S_2
S_3
(a)
Reward Function
S_1
S_2
S_3
Abstraction-1 Learned
Abstraction-2 Learned
Abstraction-3 Learned
t ->
(b)
Legend
S_1, stay
S_1, switch
S_2, stay
S_2, switch
S_3, stay
S_3, switch
ROC Estimation Error
Module1
Module2
Module3
t->
(c)

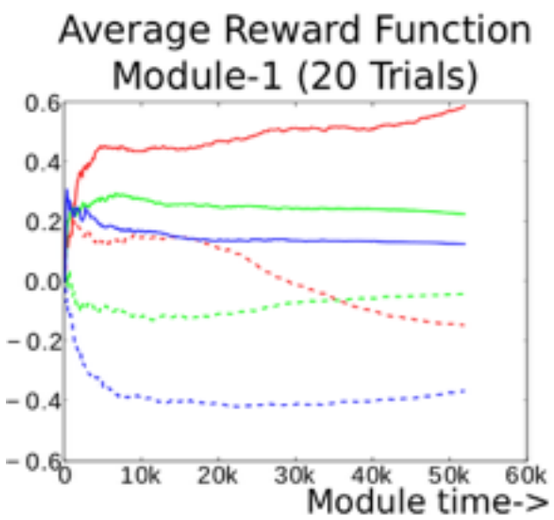

Average ROC Estimation Error (20 Trials)
Abstraction Learned
Abstraction Learned
Abstraction Learned
Module1
Module2
Module3
600
500
400
300
200
100
0
0
50k
100k
150k
200k
t ->
(a)

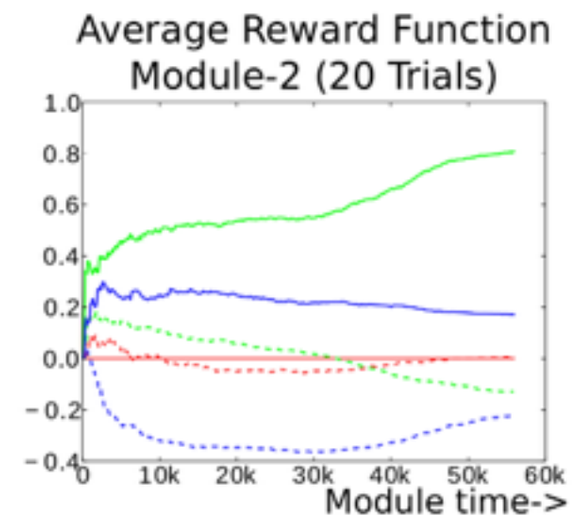

Legend
S_1, stay
S_1, switch
S_2, stay
S_2, switch
S_3, stay
S_3, switch

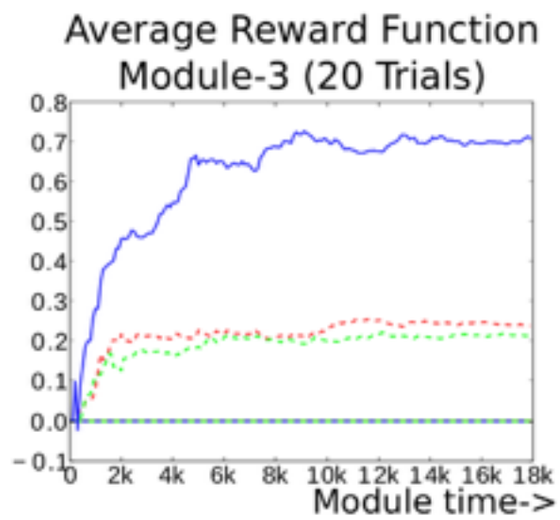

Reward Function (Module-1)
(20 Trials)
1.0
0.5
0.0
−0.5
−1.0
0
10k
30k
50k
70k
90k
Module time->
(b)


Reward Function (Module-2)
(20 Trials)
1.0
0.5
0.0
−0.5
−1.0
0
10k
30k
50k
70k
90k
Module time->
(c)


Reward Function (Module-3)
(20 Trials)
1.0
0.5
0.0
−0.5
−1.0
0
5k
10k
15k
20k
25k
30k
35k
Module time->
(d)


Average Reward Function
Module-1 (20 Trials)
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
10k
20k
30k
40k
50k
60k
Module time->
(e)

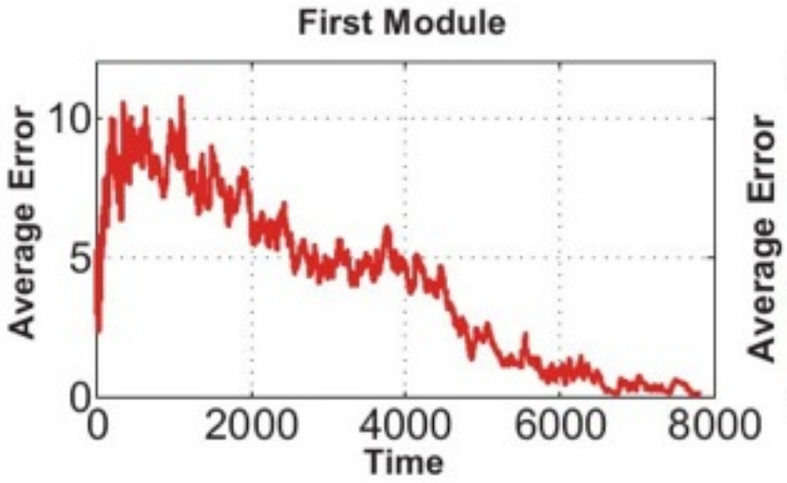

Average Reward Function
Module-2 (20 Trials)
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
10k
20k
30k
40k
50k
60k
Module time->
(f)


Average Reward Function
Module-3 (20 Trials)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
0
2k
4k
6k
8k
10k
12k
14k
16k
18k
Module time->
(g)

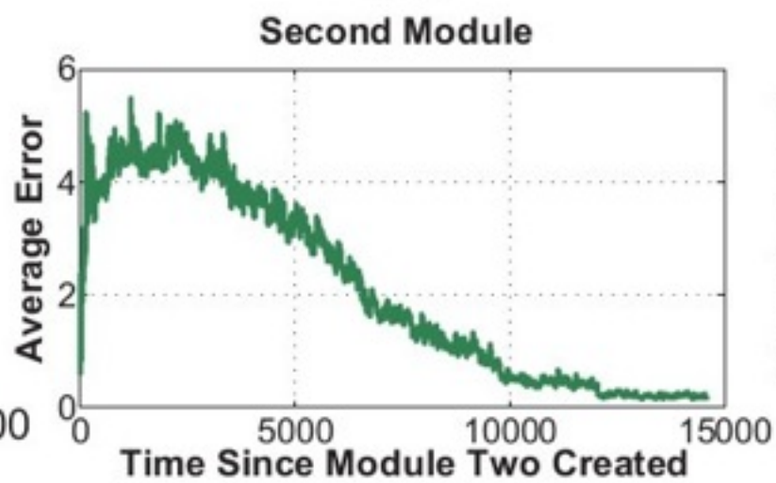
x_1

x_2

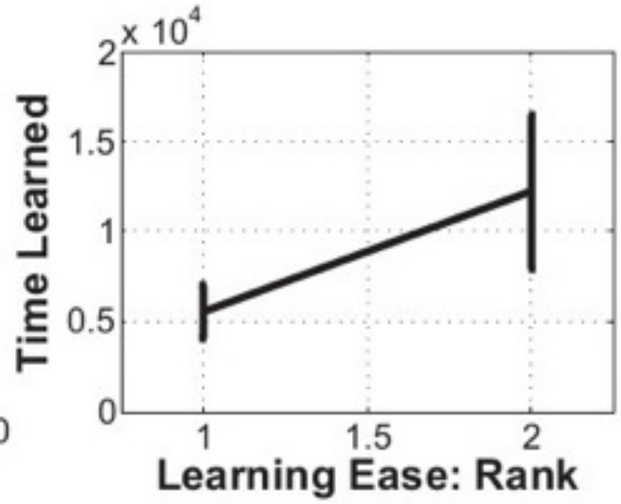
(a)


First Module
Average Error
Time
0
5
10
0
2000
4000
6000
8000

(b)

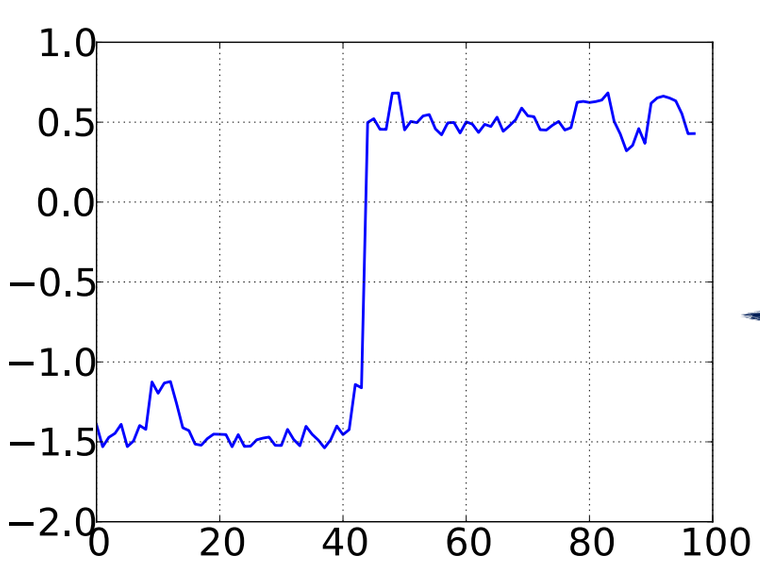

Second Module
Average Error
Time Since Module Two Created
0
2
4
6
0
5000
10000
15000

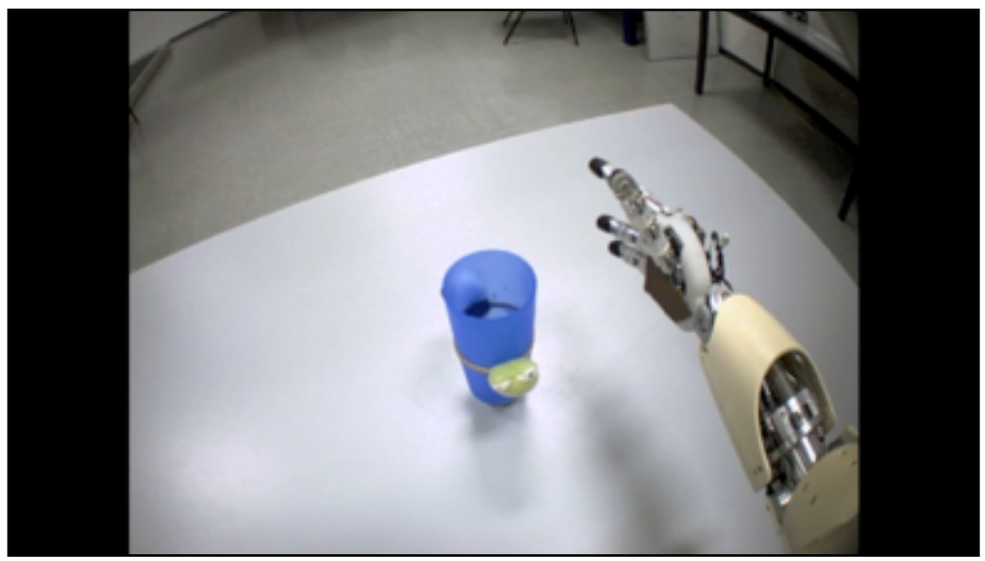
(c)


x 10^4
Time Learned
Learning Ease: Rank
0
0.5
1
1.5
2
1
1.5
2

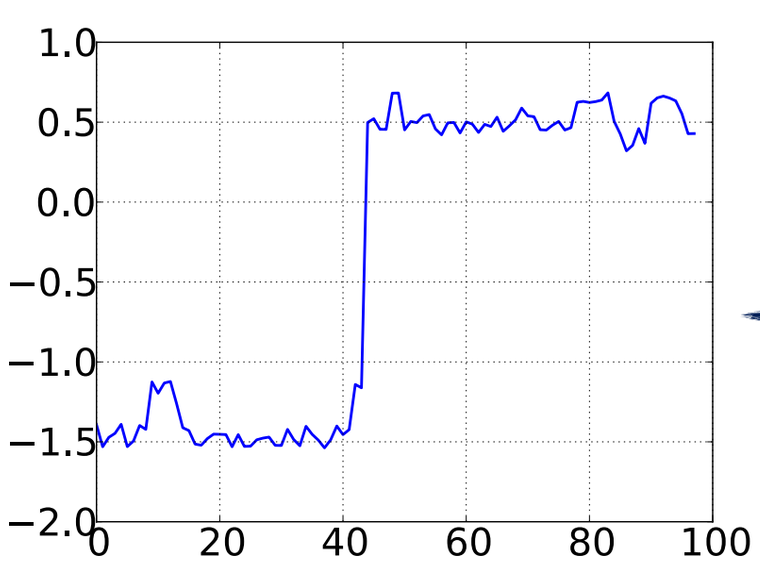
(d)

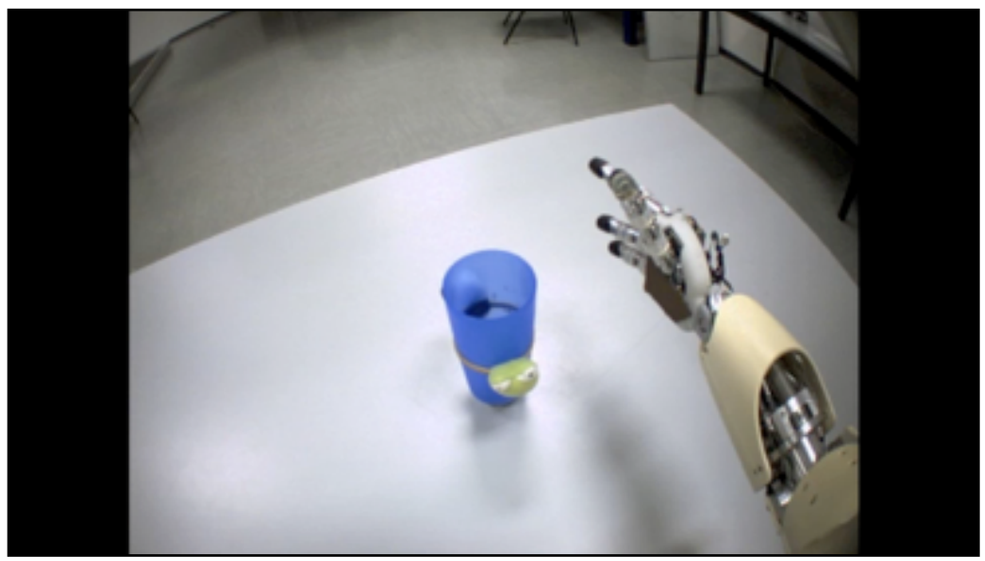
Abstraction Learning to Skill Acquisition





Abstraction Learning to Skill Acquisition





Skill Acquisition using Curious Dr. MISFA (Kompella et al. 2014)

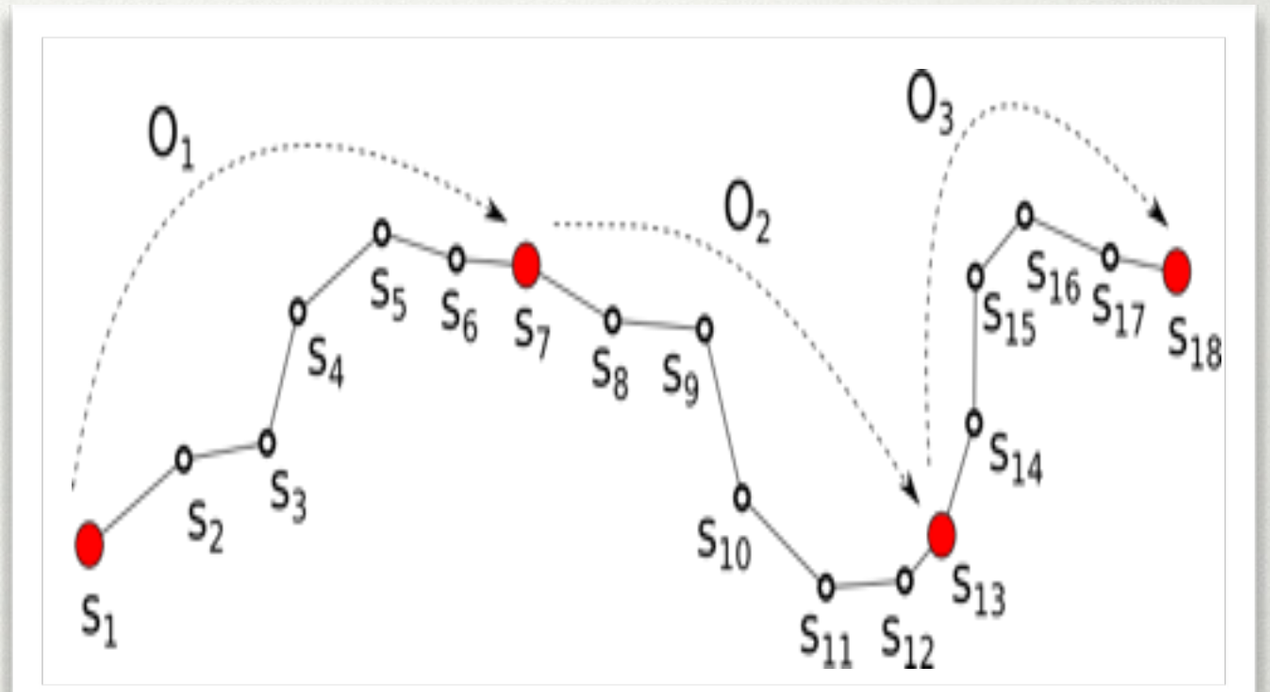
- Task independent curiosity-driven skill acquisition algorithm

Skill Acquisition using Curious Dr. MISFA (Kompella et al. 2014)

- Task independent curiosity-driven skill acquisition algorithm
- Combines Curious Dr. MISFA with the options Framework
  - Curious Dr. MISFA - Extracts slow feature abstractions as quickly as possible
  - Options - Builds the abstractions into skills simultaneously

Options Framework

- Option: Temporally extended courses of actions.



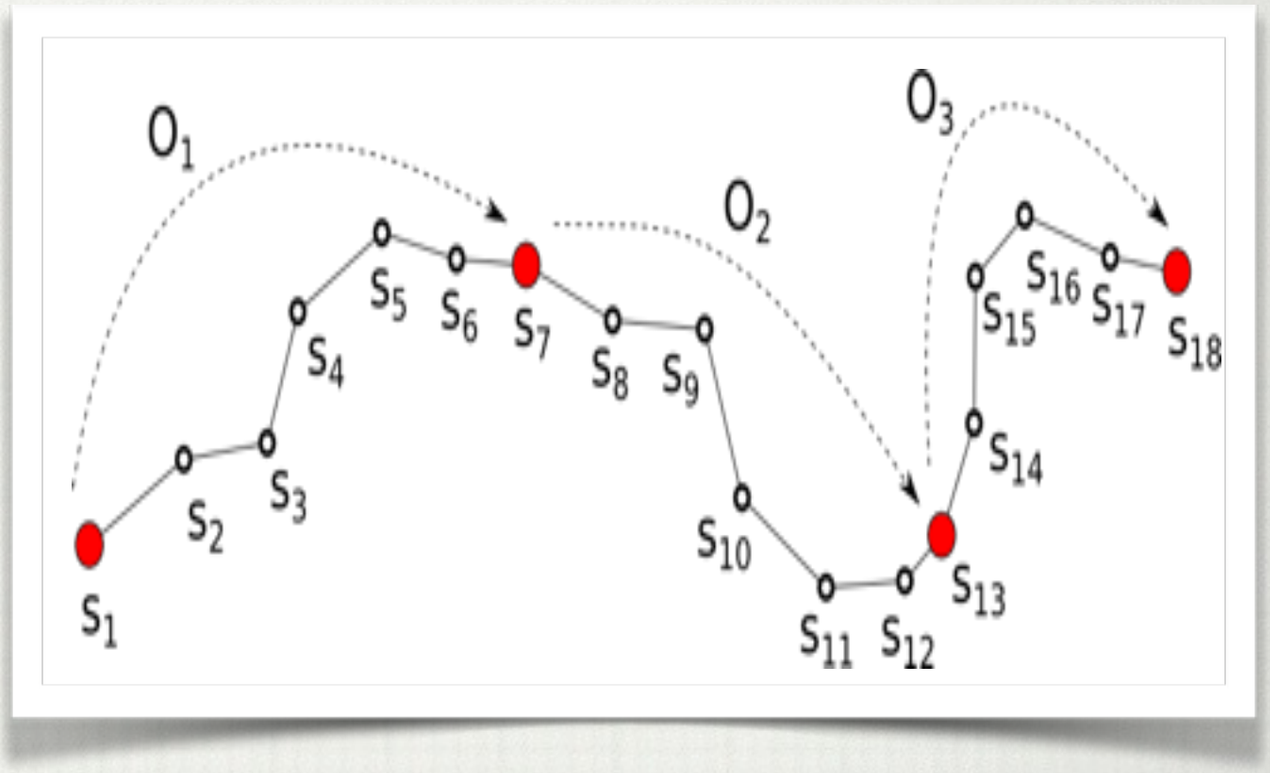
Options Framework

- Option: Temporally extended courses of actions.
- Formally defined as a tuple $O = \langle I, \beta, \pi \rangle$:

$I \subseteq S$ (Initiation set),

$\beta : S \rightarrow [0,1]$ (Termination cond.),

$\pi : I \times A \rightarrow [0,1]$ (Option policy)



Options Framework

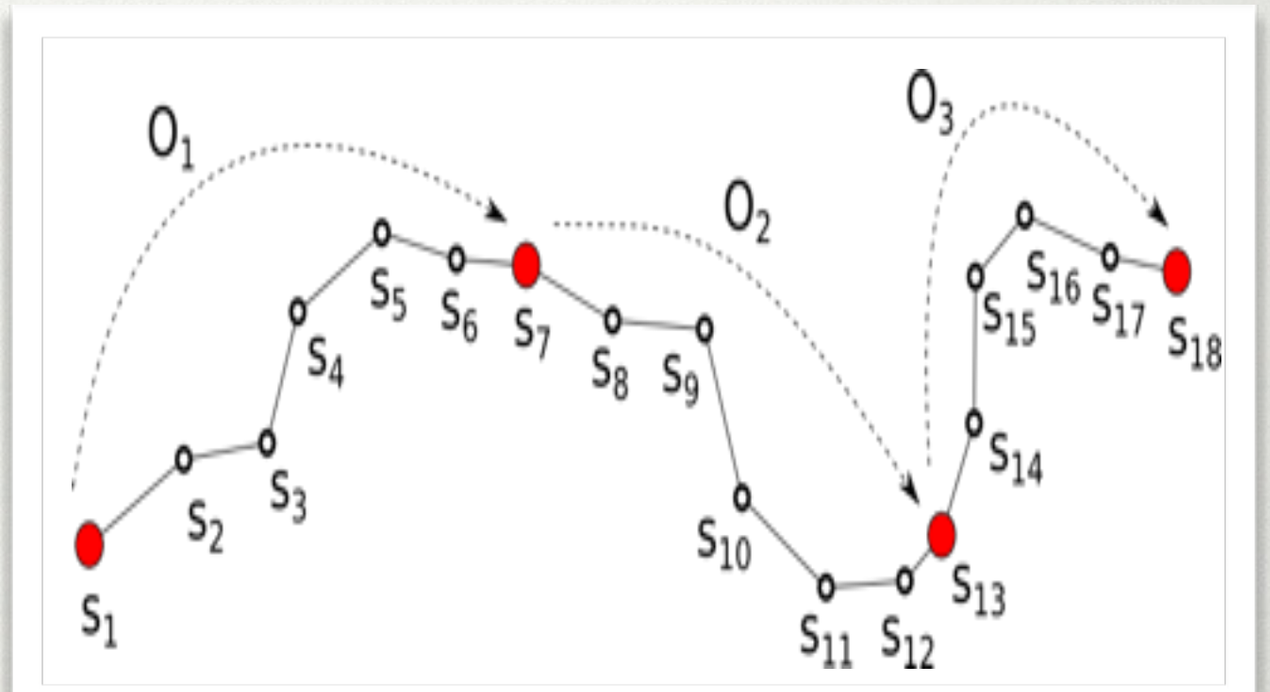
- Option: Temporally extended courses of actions.
- Formally defined as a tuple $O = \langle I, \beta, \pi \rangle$:

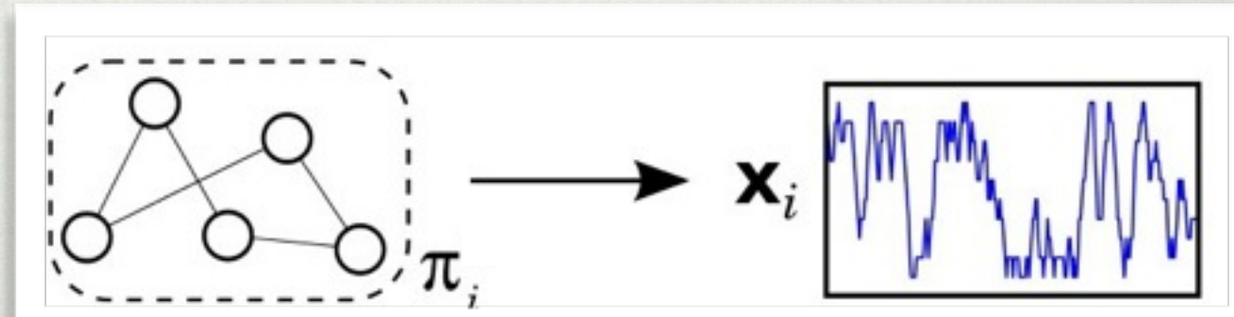
  $I \subseteq S$ (Initiation set),

  $\beta : S \rightarrow [0,1]$ (Termination cond.),

  $\pi : I \times A \rightarrow [0,1]$ (Option policy)

- Executing option → High-dimensional observations



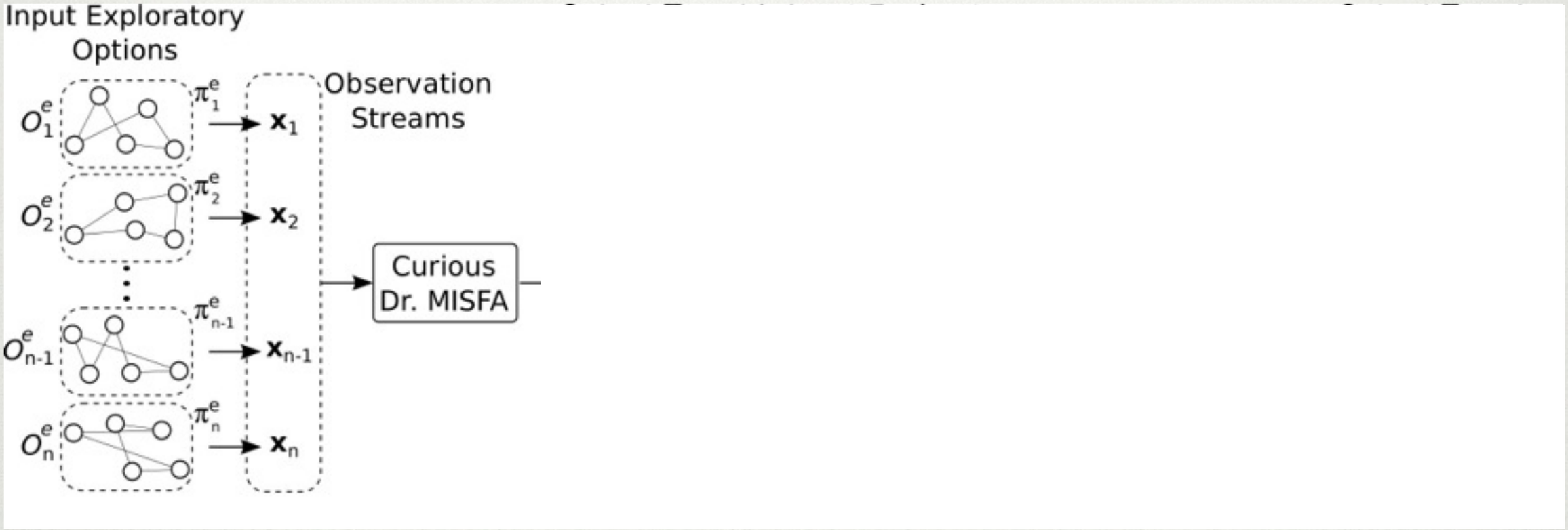


Learning Process

- Random exploratory behaviors → Deterministic and increasingly complex target behaviors

(Exploratory Options) (Target Options)

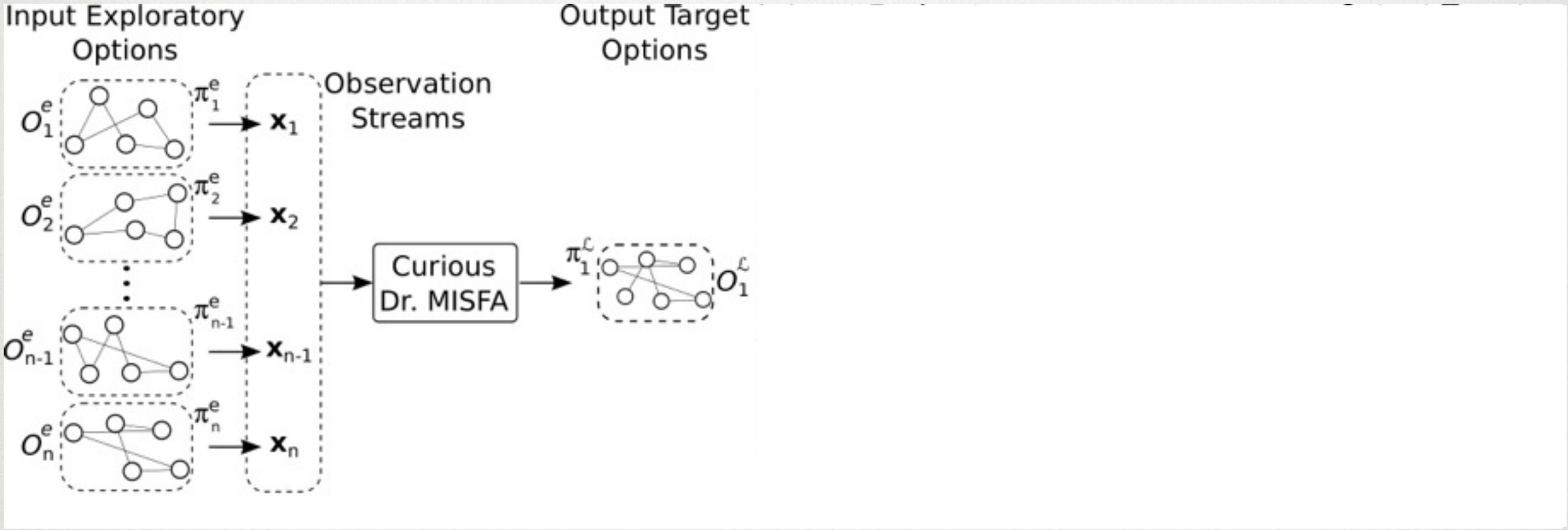
Learning Process



- Random exploratory behaviors → Deterministic and increasingly complex target behaviors

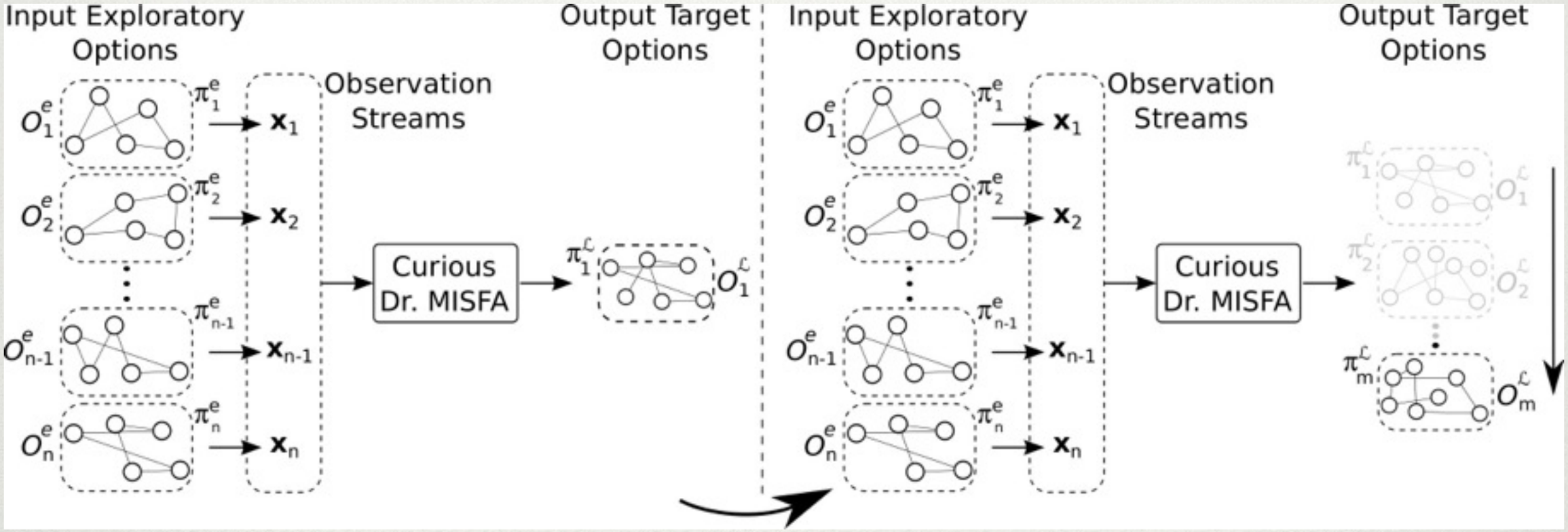
(Exploratory Options) (Target Options)

Learning Process



- Random exploratory behaviors (Exploratory Options) → Deterministic and increasingly complex target behaviors (Target Options)

Learning Process

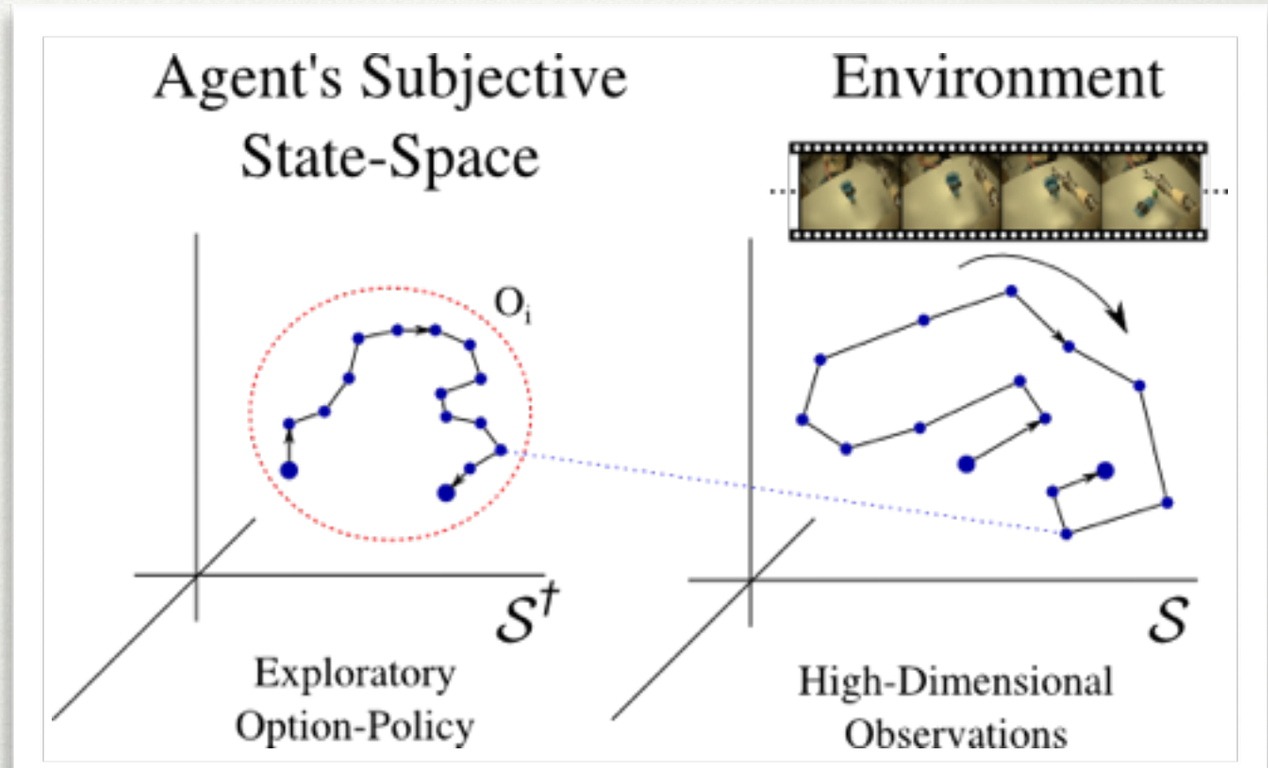


- Random exploratory behaviors → Deterministic and increasingly complex target behaviors

(Exploratory Options) (Target Options)

Exploratory Options (O^e)

- $O^e = \langle I^e, \beta^e, \pi^e \rangle$
- $I^e \subseteq S^\dagger$
- $\boldsymbol{\pi}^e$: Exploratory-option's stochastic policy is a random walk in option's state-space
- β^e : The option terminates after $\boldsymbol{\tau}$ time-steps since its execution.



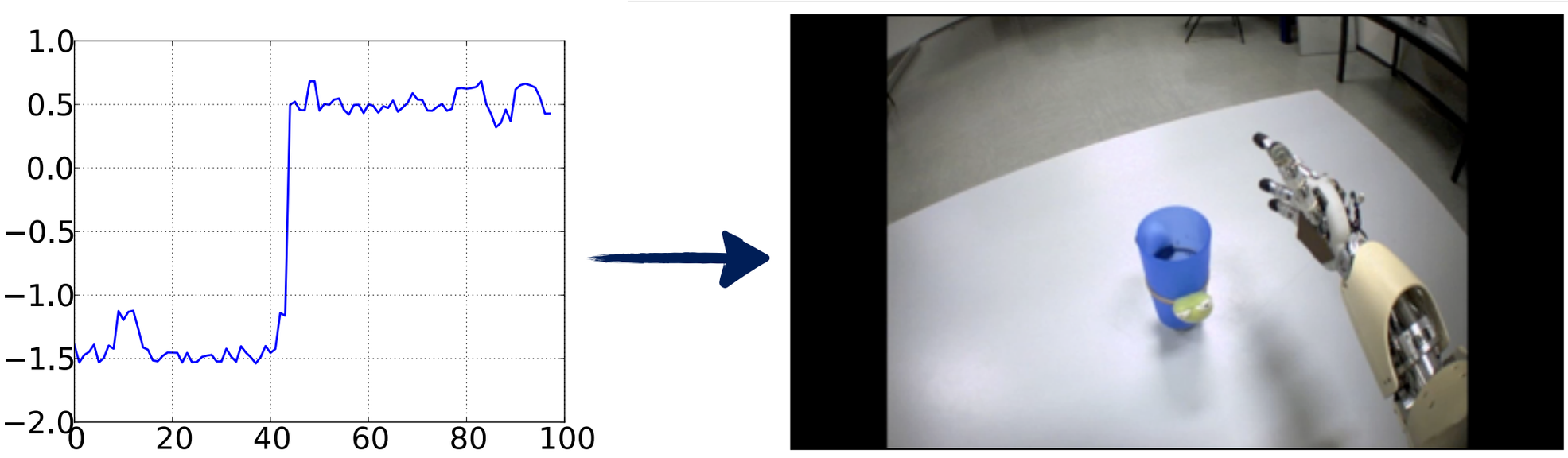
Target Options (Acquired Skills)

Target Options (Acquired Skills)

- $O^{\mathcal{L}} = \langle I^{\mathcal{L}}, \beta^{\mathcal{L}}, \phi, \pi^{\mathcal{L}} \rangle$

Target Options (Acquired Skills)

- $O^{\mathcal{L}} = \langle I^{\mathcal{L}}, \beta^{\mathcal{L}}, \phi, \pi^{\mathcal{L}} \rangle$
- Target option's policy maximises the variation of observed feature-outputs.



Learning a Target Option ($\boldsymbol{O}^{\mathcal{L}}$)

- Say Curious Dr. MISFA learns an abstraction ϕ_i corresponding to x_j

Learning a Target Option ($\boldsymbol{O}^{\mathcal{L}}$)

- Say Curious Dr. MISFA learns an abstraction ϕ_i corresponding to x_j
- Initiation set: $I_i^{\mathcal{L}} = (I_j^e \times S^{\phi_i})$

Learning a Target Option ($\boldsymbol{O}^{\mathcal{L}}$)

- Say Curious Dr. MISFA learns an abstraction ϕ_i corresponding to x_j
- Initiation set: $I_i^{\mathcal{L}} = (I_j^e \times S^{\phi_i})$
- Target option policy ($\pi_i^{\mathcal{L}} : I_i^{\mathcal{L}} \rightarrow A$) : Developed through LSPI using:
  - Estimated option's transition model from samples generated by $\boldsymbol{\pi}_j^e$.
  - Estimated options' reward model using diff. of subseq. abstraction activations:

Learning a Target Option ($O^{\mathcal{L}}$)

- Say Curious Dr. MISFA learns an abstraction ϕ_i corresponding to x_j
- Initiation set: $I_i^{\mathcal{L}} = (I_j^e \times S^{\phi_i})$
- Target option policy ($\pi_i^{\mathcal{L}} : I_i^{\mathcal{L}} \rightarrow A$) : Developed through LSPI using:

  - Estimated option's transition model from samples generated by π_j^e.
  - Estimated options' reward model using diff. of subseq. abstraction activations:

$$R^{O_i^{\mathcal{L}}}(s^\dagger, a, s'^\dagger_-) = (1-\alpha) R^{O_i^{\mathcal{L}}}(s^\dagger, a, s'^\dagger_-) + \alpha \|\mathbf{y_i}(t) - \mathbf{y_i}(t-1)\|$$

where $\mathbf{y_i}(t) = \phi_i \left(\mathcal{U}(\mathcal{P}(s_-, \pi_j^e(s'^\dagger_-))) \right)$ and $\mathbf{y_i}(t-1) = \phi_i \left(\mathcal{U}(\mathcal{P}(s, \pi_j^e(s^\dagger))) \right)$

Learning a Target Option ($O^{\mathcal{L}}$)

- Say Curious Dr. MISFA learns an abstraction ϕ_i corresponding to x_j
- Initiation set: $I_i^{\mathcal{L}} = (I_j^e \times S^{\phi_i})$
- Target option policy ($\pi_i^{\mathcal{L}} : I_i^{\mathcal{L}} \rightarrow A$) : Developed through LSPI using:

  - Estimated option's transition model from samples generated by π_j^e.
  - Estimated options' reward model using diff. of subseq. abstraction activations:

$$R^{O_i^{\mathcal{L}}}(s^\dagger, a, s'^\dagger_-) = (1-\alpha) R^{O_i^{\mathcal{L}}}(s^\dagger, a, s'^\dagger_-) + \alpha \|\mathbf{y_i}(t) - \mathbf{y_i}(t-1)\|$$

$$\text{where} \quad \mathbf{y_i}(t) = \phi_i\left(\mathcal{U}(\mathcal{P}(s_-, \pi_j^e(s'^\dagger_-)))\right) \text{ and } \mathbf{y_i}(t-1) = \phi_i\left(\mathcal{U}(\mathcal{P}(s, \pi_j^e(s^\dagger)))\right)$$

- Termination Condition: Option terminates whenever the agent observes the maximum of estimated reward model.

Summary of Policies Involved

- π^{int} : Internal Policy that is learned to determine which exploratory option to execute.

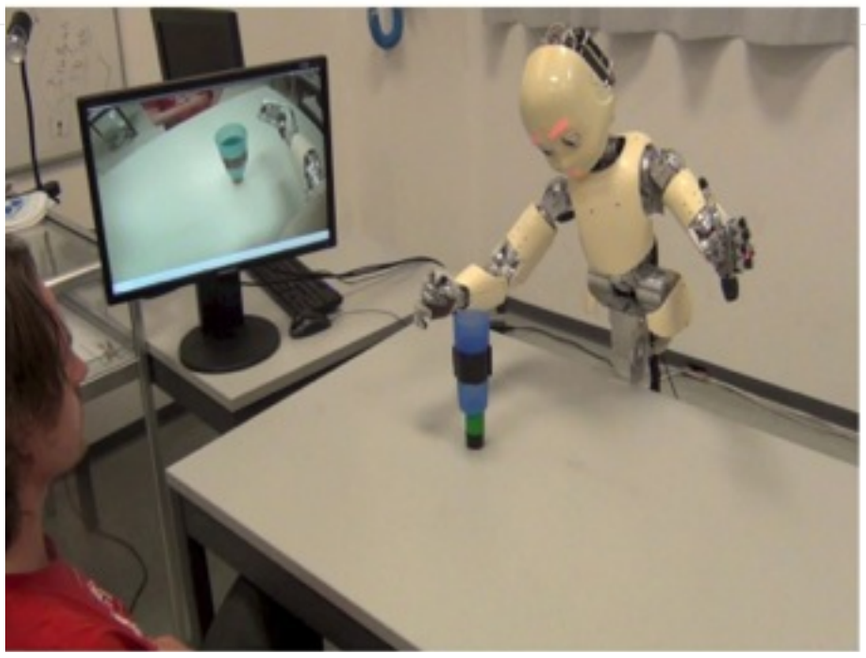
Summary of Policies Involved

- π^{int} : Internal Policy that is learned to determine which exploratory option to execute.
- π^{e} : Exploratory-option's stochastic policy that is used to observe high-dimensional observations to update slow feature abstraction.

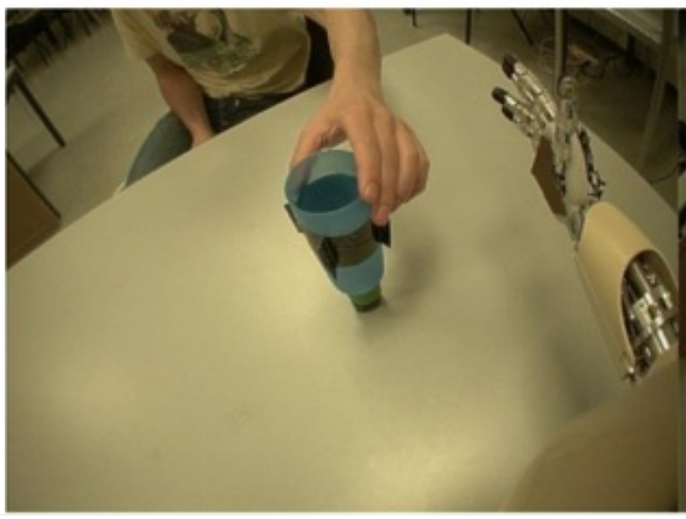
Summary of Policies Involved

- π^{int} : Internal Policy that is learned to determine which exploratory option to execute.
- π^{e} : Exploratory-option's stochastic policy that is used to observe high-dimensional observations to update slow feature abstraction.
- $\pi^{\mathcal{L}}$: Target-option's deterministic policy that is learned to maximize variation in the slow-feature abstraction output.

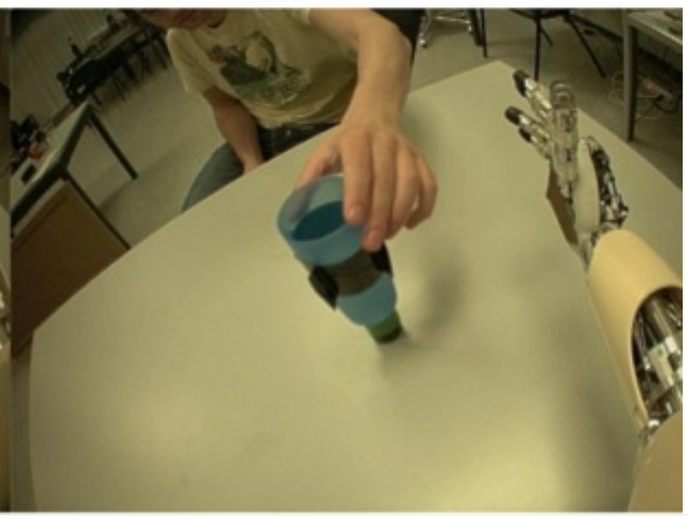
Experimental Results

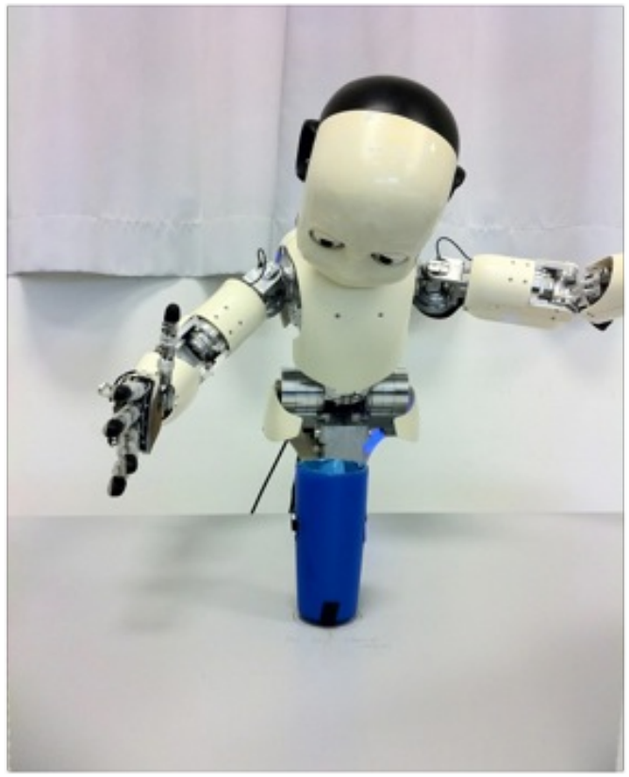


iCub's Left-Camera Image



iCub's Right-Camera Image

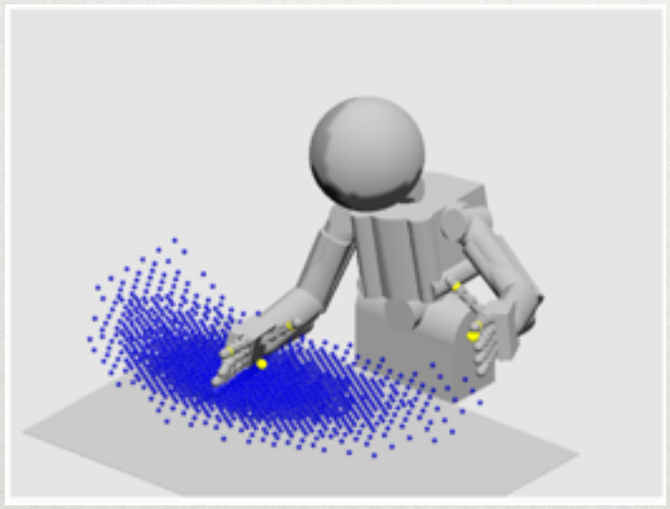




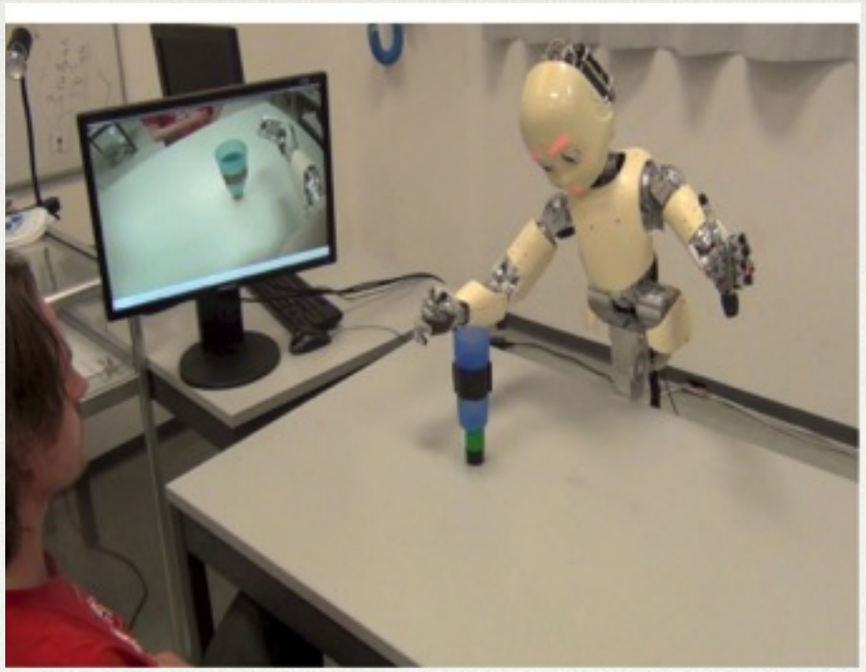
iCub's state space dimensionality

Vision: 640 x 480 pixels
Joints: 51 motors
Total = 640 x 480 x 51
= (15,667,200) states variables!

Our Focus:



Low-dimensional joint state space embedding learned apriori



Experiment 1

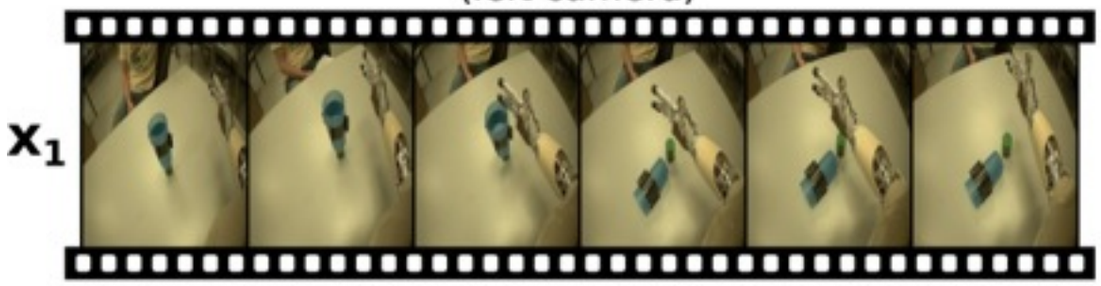
iCub Learns to Topple a Cup

https://www.youtube.com/watch?v=OTqdXbTEZpE

https://www.youtube.com/watch?v=OTqdXbTEZpE

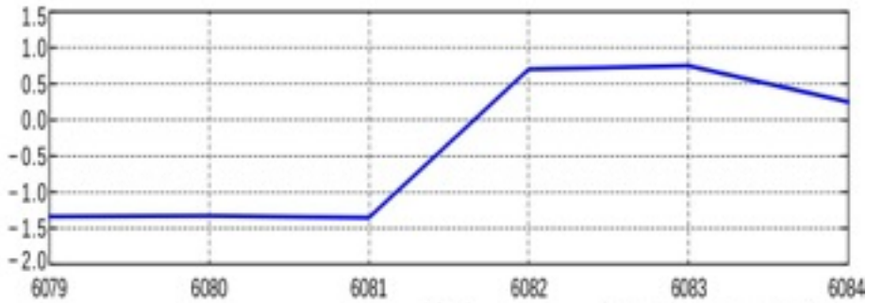
Sample Input Observation Stream

(left camera)



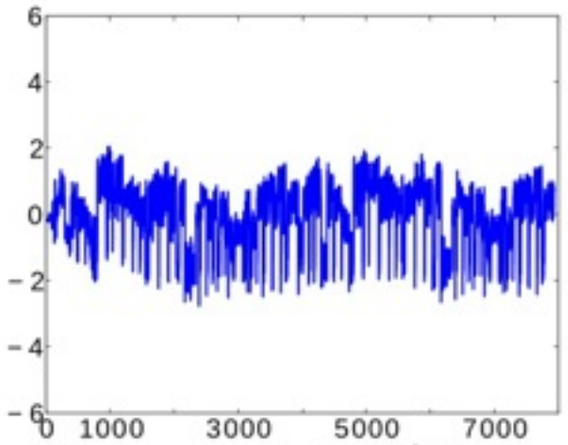
(a)

Sample IncSFA Output (**y**)



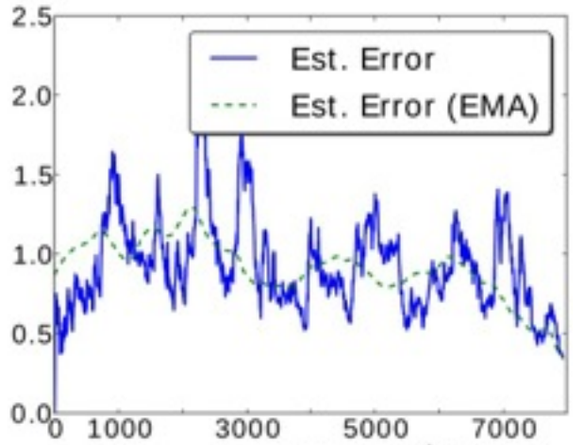
(b)

IncSFA Output (**y**)



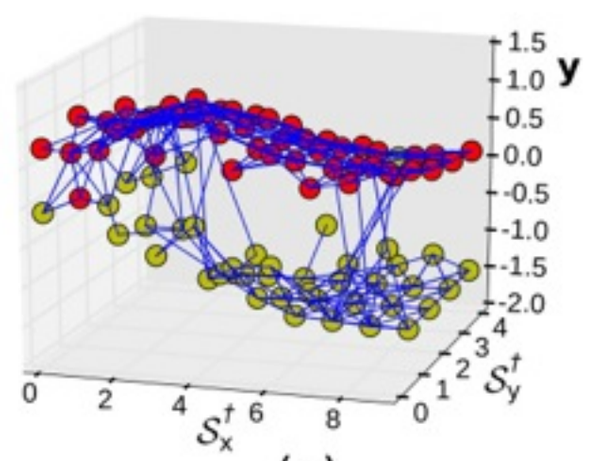
(c)

ROC Est. Error



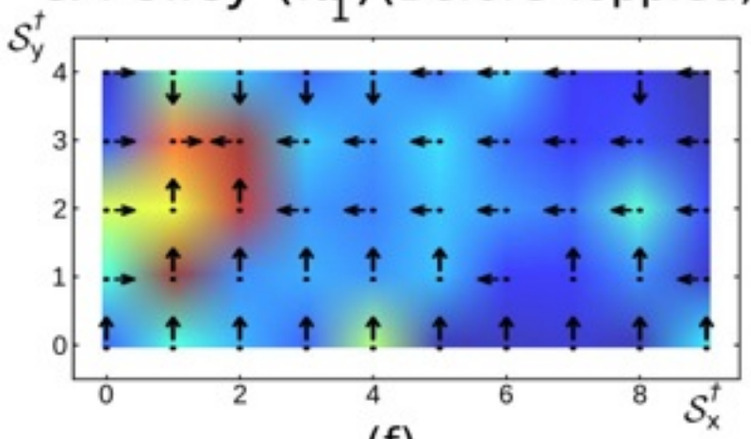
(d)

ROC Cluster Centers



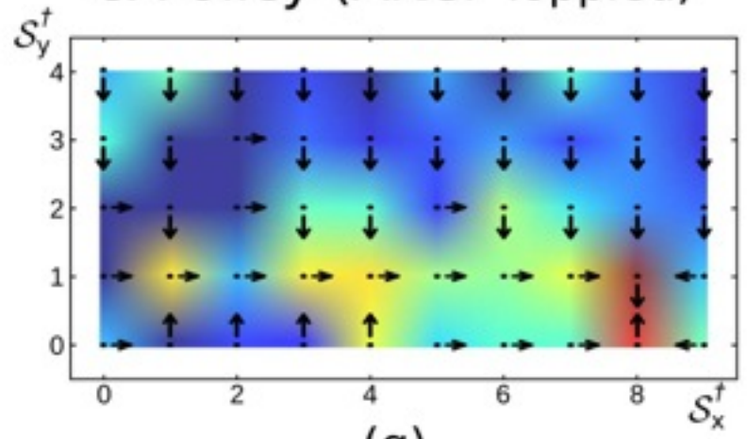
(e)

State-Reward Function ($R^{O_1^{\mathcal{L}}}$) & Policy ($\pi_1^{\mathcal{L}}$)(Before Toppled)



(f)

State-Reward Function ($R^{O_1^{\mathcal{L}}}$) & Policy (After Toppled)



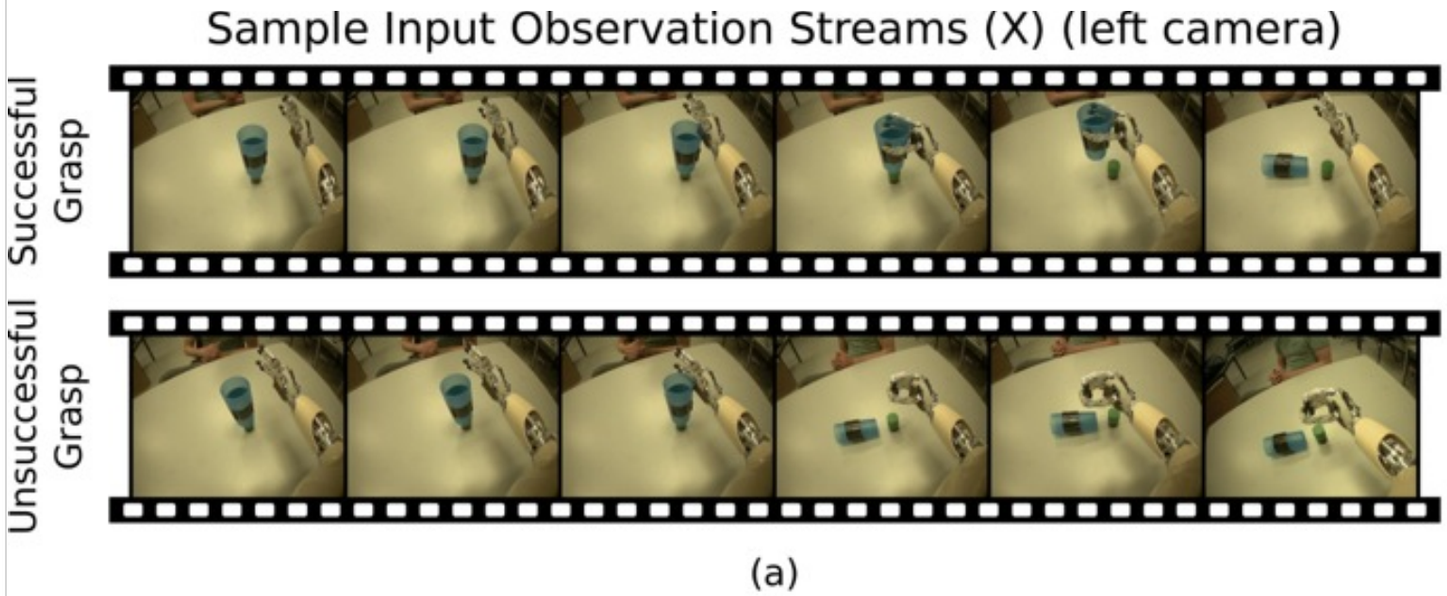
(g)

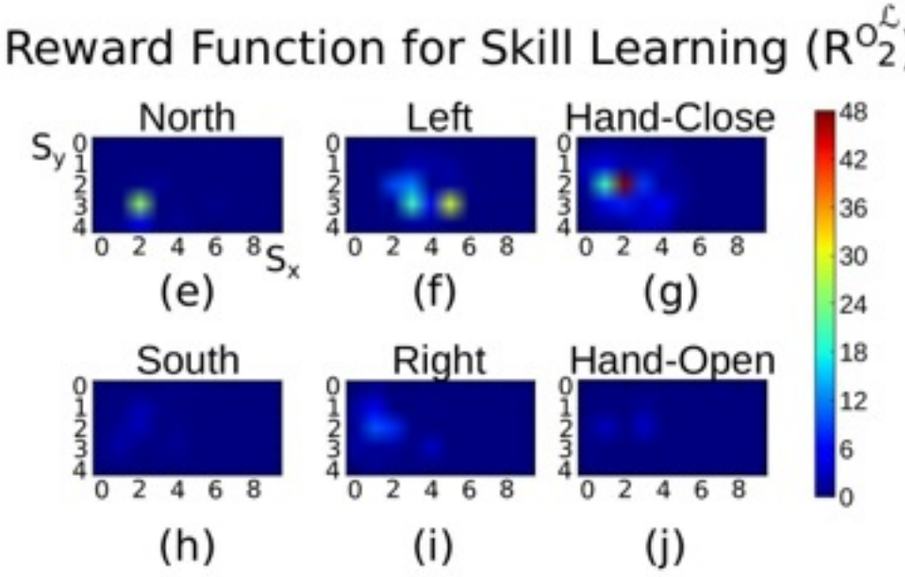
Experiment 2

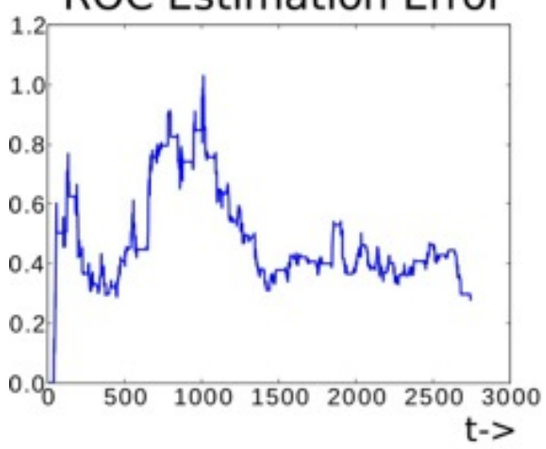
iCub Learns to Grasp a Cup

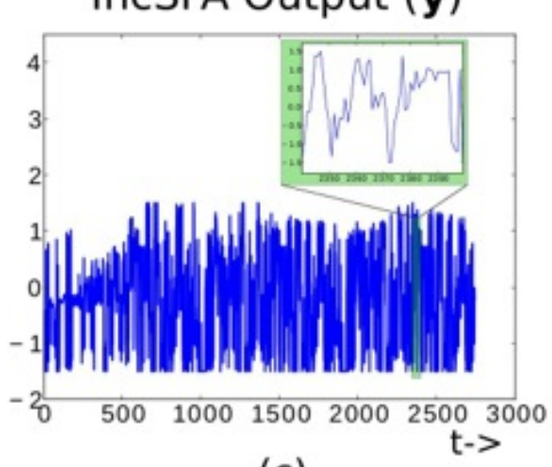
https://www.youtube.com/watch?v=OTqdXbTEZpE

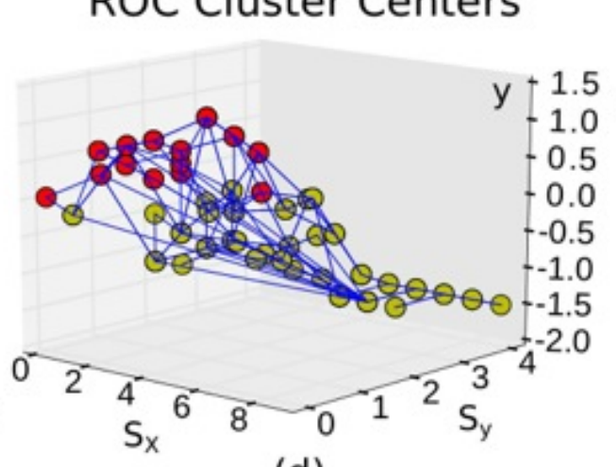
https://www.youtube.com/watch?v=OTqdXbTEZpE

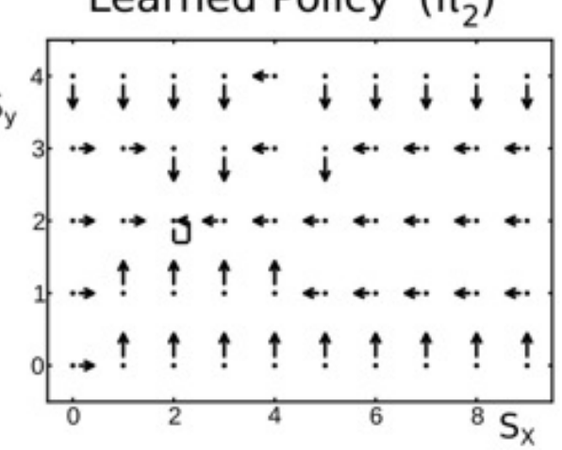

Sample Input Observation Streams (X) (left camera)
Successful Grasp
Unsuccessful Grasp
(a)


Reward Function for Skill Learning ($R^{O^{\mathcal{L}}_2}$)
North
Left
Hand-Close
South
Right
Hand-Open
S_y
S_x
(e)
(f)
(g)
(h)
(i)
(j)


ROC Estimation Error
t->
(b)


IncSFA Output (y)
t->
(c)


ROC Cluster Centers
S_x
S_y
y
(d)


Learned Policy ($\pi^{\mathcal{L}}_2$)
S_y
S_x
(k)

Related work
(Skill Acquisition in RL via Intrinsic Motivation)

- Simple Domains:
  - Bakker & Schmidhuber [2004] - (Hassle; Hierarchical RL)
  - Stout & Barto [2010] - (Competence-based IM)
  - Pape et al. [2012] - (tactile skills on a biomimetic finger)

Related work
(Skill Acquisition in RL via Intrinsic Motivation)

- Attempts to find skills using feature-abstractions in the domain of Humanoid robots:
  - Hart [2009] - (DEDS + task-specific IM)

Related work
(Skill Acquisition in RL via Intrinsic Motivation)

- Attempts to find skills using feature-abstractions in the domain of Humanoid robots:
  - Hart [2009] - (DEDS + task-specific IM)
  - Konidaris [2009-2011] - (Options + ART tags + pre-existing abstraction library), uBot.

Related work
(Skill Acquisition in RL via Intrinsic Motivation)

- Attempts to find skills using feature-abstractions in the domain of Humanoid robots:
  - Hart [2009] - (DEDS + task-specific IM)
  - Konidaris [2009-2011] - (Options + ART tags + pre-existing abstraction library), uBot.
  - Mugan & Kuipers [2012] - (QLAP, assumes low-level object tracking models)

Related work
(Skill Acquisition in RL via Intrinsic Motivation)

- Attempts to find skills using feature-abstractions in the domain of Humanoid robots:
  - Hart [2009] - (DEDS + task-specific IM)
  - Konidaris [2009-2011] - (Options + ART tags + pre-existing abstraction library), uBot.
  - Mugan & Kuipers [2012] - (QLAP, assumes low-level object tracking models)
  - Barnes & Oudeyer [2013] - (SAGG-RIAC, Competence progress + heuristics + subgoals, task-dependent)

Related work
(Skill Acquisition in RL via Intrinsic Motivation)

- Attempts to find skills using feature-abstractions in the domain of Humanoid robots:
  - Hart [2009] - (DEDS + task-specific IM)
  - Konidaris [2009-2011] - (Options + ART tags + pre-existing abstraction library), uBot.
  - Mugan & Kuipers [2012] - (QLAP, assumes low-level object tracking models)
  - Barnes & Oudeyer [2013] - (SAGG-RIAC, Competence progress + heuristics + subgoals, task-dependent)
  - Ngo et al. [2012-2013] - (selective sampling + Curiosity, Overhead camera + pre-designed sensori-motor abstractions)

Take-Home Message

Take-Home Message

- Curious Dr. MISFA is an online active modular incremental SFA.

Take-Home Message

- Curious Dr. MISFA is an online active modular incremental SFA.
- Skill acquisition using Curious Dr. MISFA is a **task-independent** method that demonstrates **intrinsically motivated skill acquisition** from **raw-pixel streams** on a **real** humanoid robot.

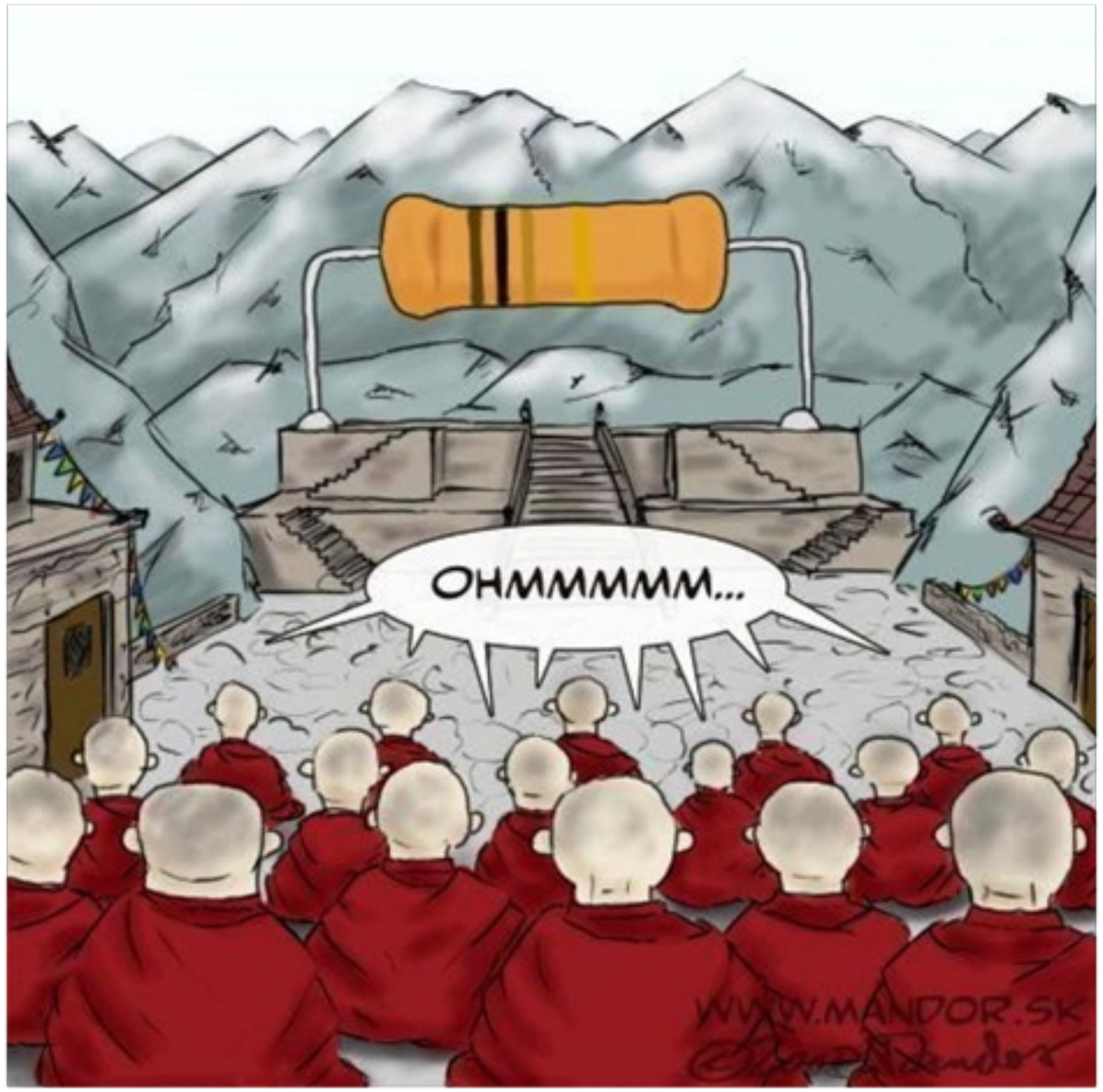
Acknowledgement

- This tutorial is funded through SNF grant #138219 (Theory and Practice of Reinforcement Learning II)

- I personally thank:
  Dr. Jan Koutnik (http://www.idsia.ch/~koutnik/),
  Dr. Faustino Gomez (http://www.idsia.ch/~tino/) and
  Dr. Matthew Luciw (http://www.idsia.ch/~luciw/)
  for helping me prepare for this tutorial.

Disclaimer

- Some of the artwork used in this presentation was downloaded from Google Images.
- Presenter does not wish to use them for any commercial purposes.



Thank you