Continual Learning and Robotics: an overview

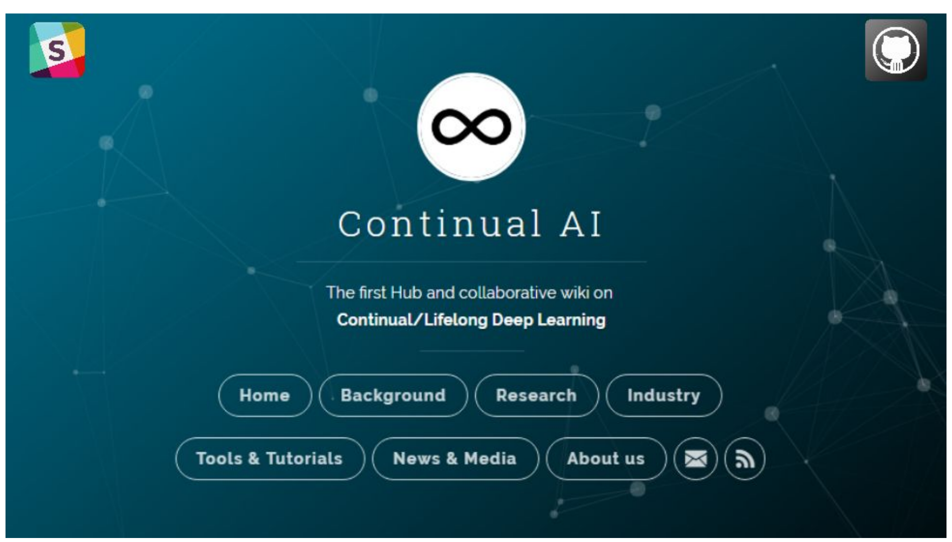
Natalia Díaz Rodríguez, PhD

ENSTA
Inria
INVENTEURS DU MONDE NUMÉRIQUE
Autonomous Systems and Robotics
Computer Science and System Engineering Laboratory

INSTITUT POLYTECHNIQUE DE PARIS

DREAM

www.robotsthatdream.eu



Special thanks to: **T. Lesort, V. Lomonaco, M. Seurin, A. Raffin, R. Traoré, A. Hill, T. Sun, H. Caselles-Dupre, D. Filliat**

International Conference on Machine Learning (ICML) - Workshop on Multi-Task and Lifelong Reinforcement Learning.
Long Beach, California, 15th June 2019

About me

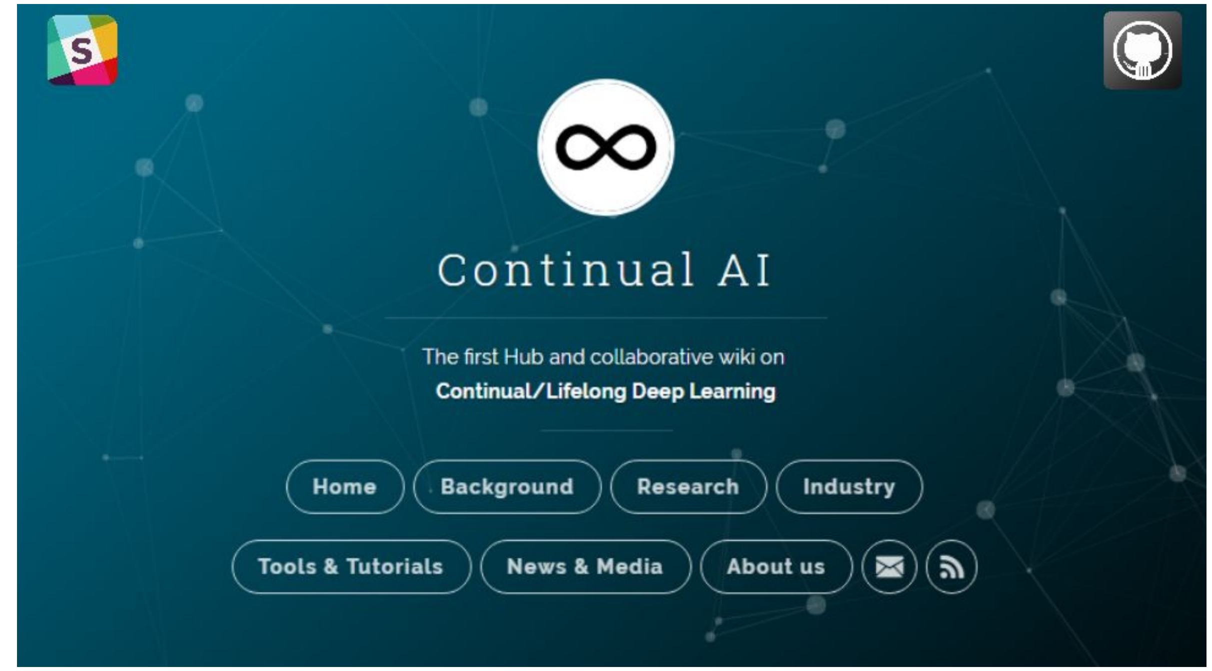
INVENTEURS DU MONDE NUMÉRIQUE

ENSTA

Autonomous Systems and Robotics

Computer Science and System Engineering Laboratory

INSTITUT POLYTECHNIQUE DE PARIS



Flowers Laboratory

FLOWing Epigenetic Robots and Systems

HOME RESEARCH TOPICS + PUBLICATIONS + TEAM ROBOTS + SOFTWARE MEDIA +

LIFELONG LEARNING AND DEVELOPMENT IN ROBOTS AND HUMANS

The Flowers project-team, at Inria and Ensta ParisTech, studies mechanisms that can allow robots and humans to acquire autonomously and cumulatively repertoires of novel skills over extended periods of time.

This includes mechanisms for learning by self-exploration, as well as learning through interaction with peers, for the acquisition of both sensorimotor and social skills. Sensorimotor skills include locomotion, affordance learning, active manipulation. Interactive skills include grounded language use and understanding, adaptive interaction protocols, and human-robot collaboration.

Our approach is organized along two strands of research:

Artificial intelligence: constructing machines and robots, inspired by animal cognitive development, and capable of lifelong development, adaptation and interaction with the physical and social world.

Cognitive Science: Elaborating computer and robotic models as tools for understanding developmental processes in humans.

Our project-team, headed by Pierre-Yves Oudeyer (Inria) and co-started with David Filliat (Ensta ParisTech Cognitive Robotics Group), focuses in particular on the study of developmental mechanisms that guide efficient open-ended learning of novel skills in large real world environments. In particular, we study:

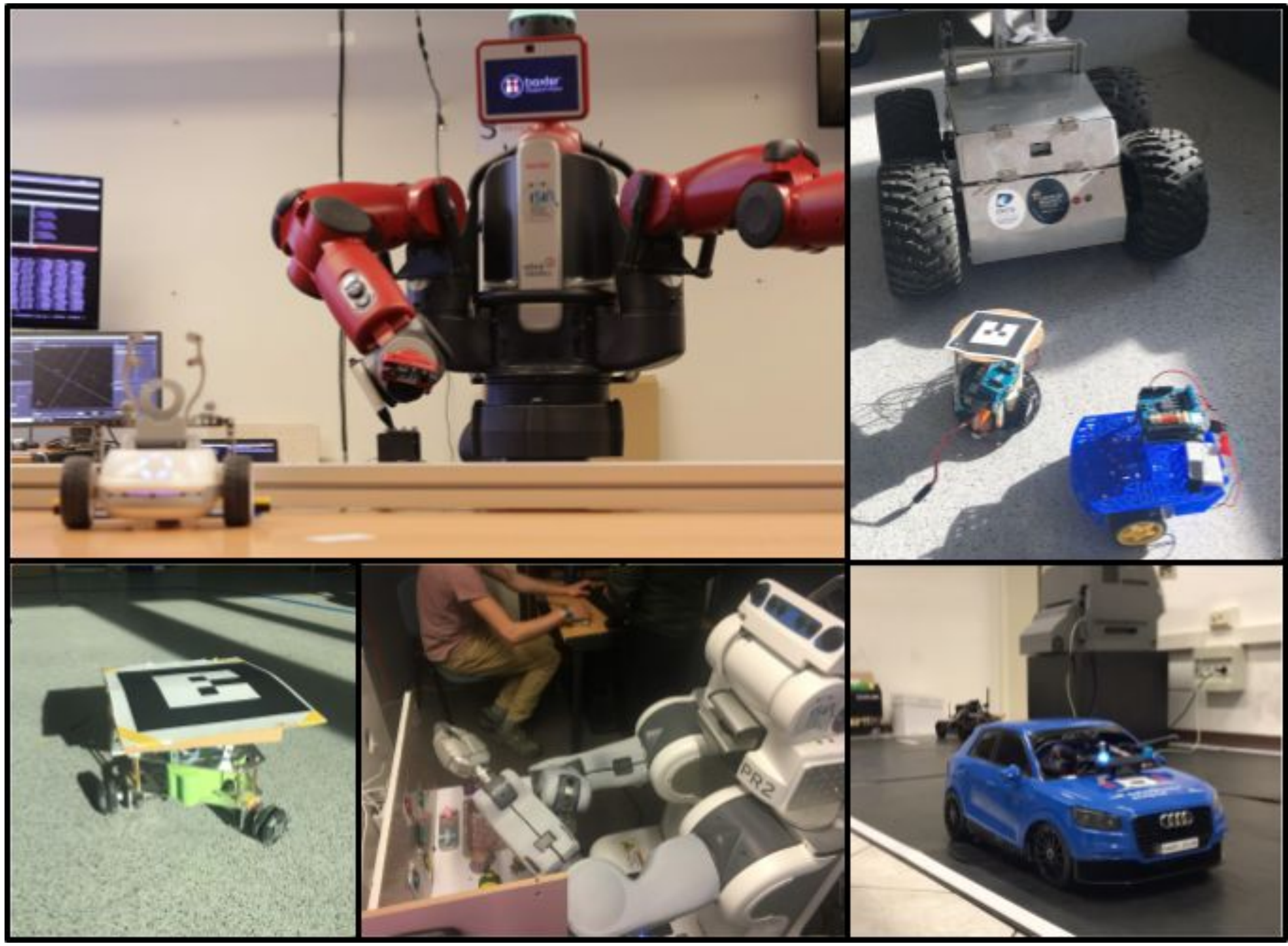
- **Intrinsically motivated multitask learning and exploration, information seeking and active learning, including artificial curiosity;**
- **Social learning**, e.g. learning by imitation or demonstration, which implies both issues related to machine learning and human-robot interaction;
- Mechanisms for learning to **sequence and compose actions to reach goals**, especially within the framework or reinforcement learning;
- The role of **embodiment**, in particular through the concept of morphological computation, as well as the structure of motor primitives/muscle synegies that can leverage the properties of morphology and physics;
- **Maturational constraints** which can allow the progressive release of novel sensorimotor degrees of freedom to be explored;

The FLOWERS project-team is located on two sites: Inria Bordeaux Sud-Ouest and ENSTA-Paristech in Paris.

Lifelong or Continual Learning (CL)

A model's ability to learn new skills

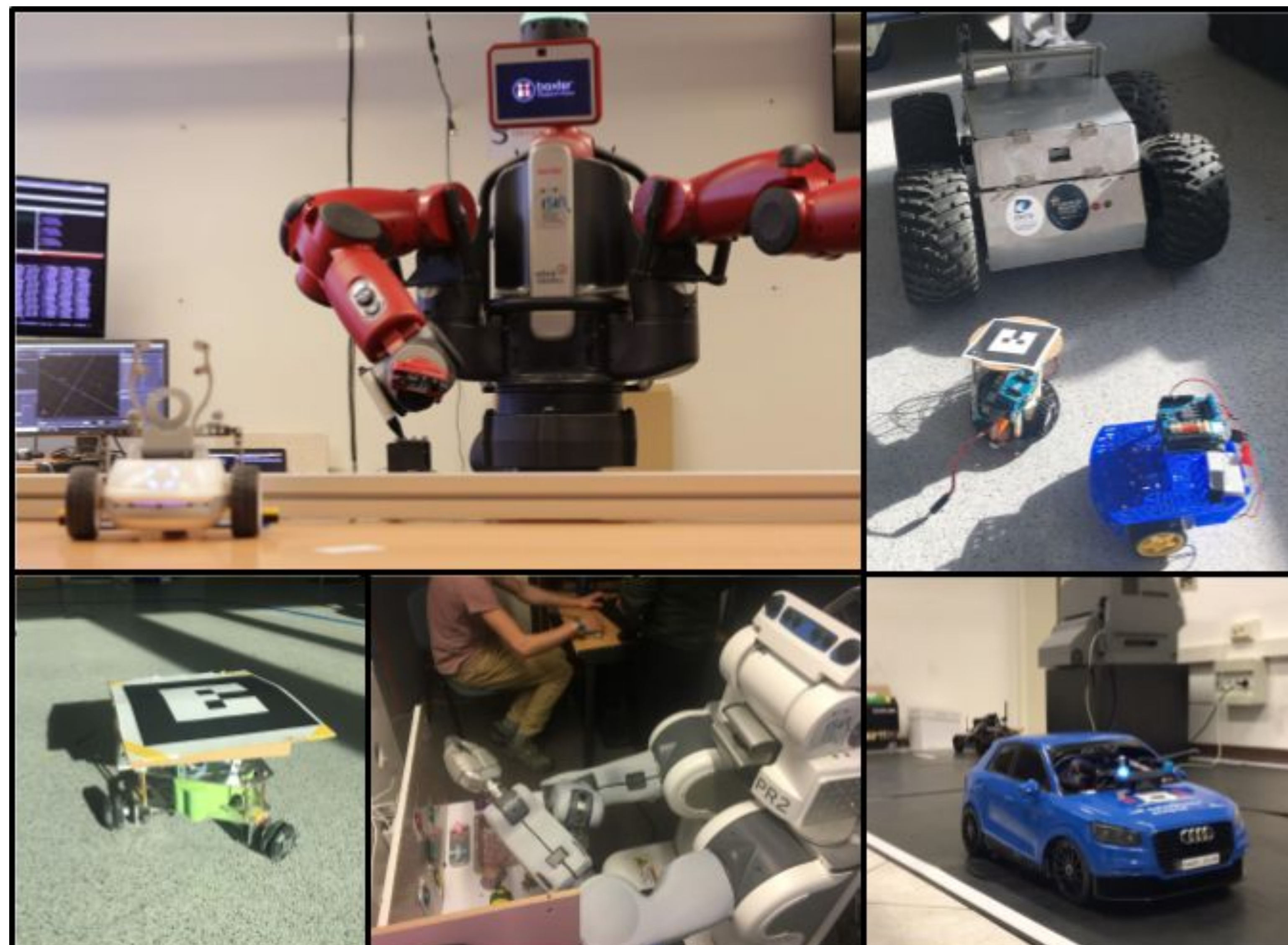
- without forgetting previous knowledge
- **sequentially**, from a data stream
- being able to solve **any** task at the end



Lifelong or Continual Learning (CL)

A model's ability to learn new skills
- without forgetting previous knowledge
- **sequentially**, from a data stream
- being able to solve **any** task at the end

How it differs from multi-task learning and meta-learning?
- Here: No data picking & shuffling (no i.i.d)



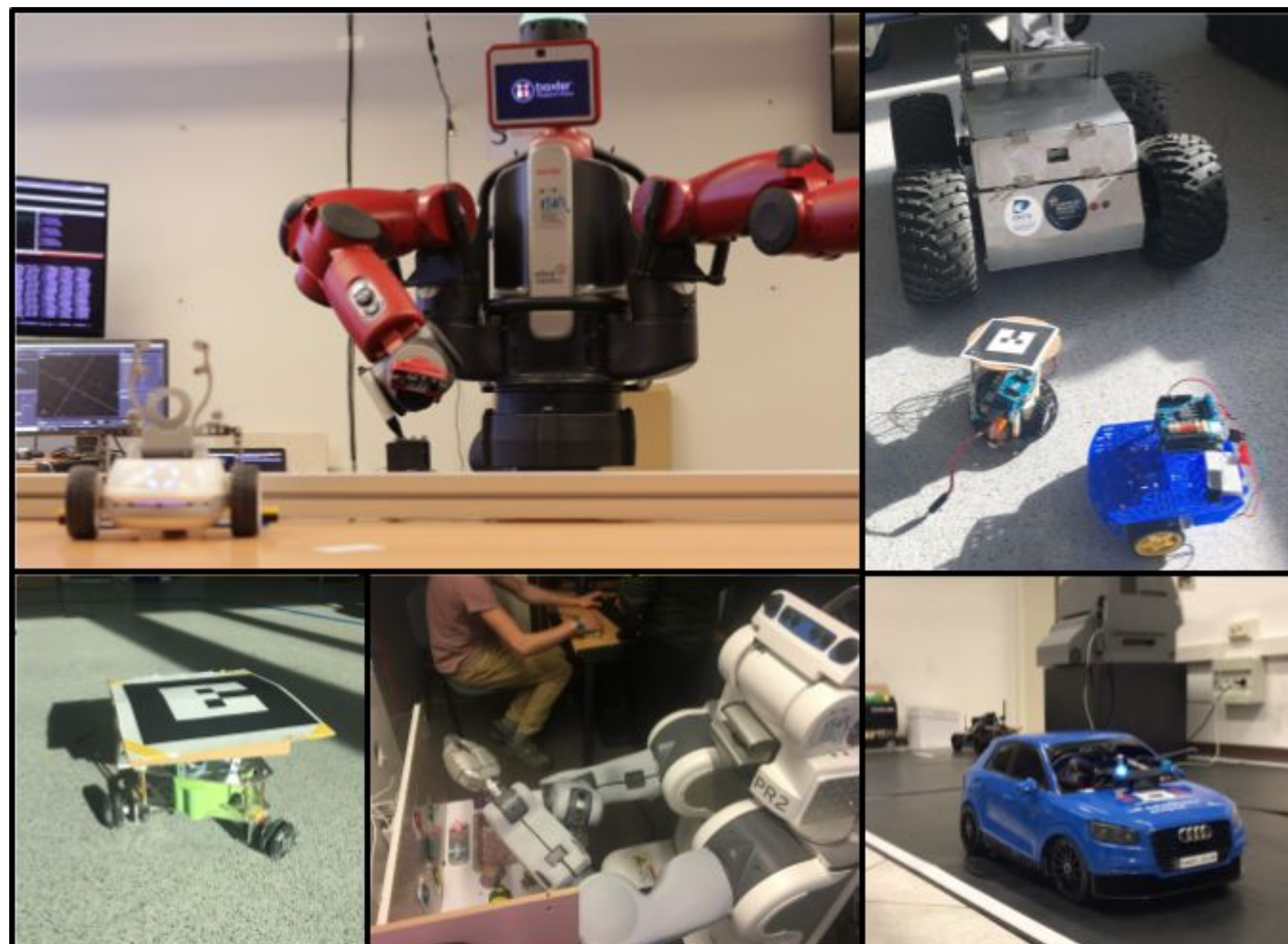
Lifelong or Continual Learning (CL)

A model's ability to learn new skills

- without forgetting previous knowledge
- **sequentially**, from a data stream
- being able to solve **any** task at the end

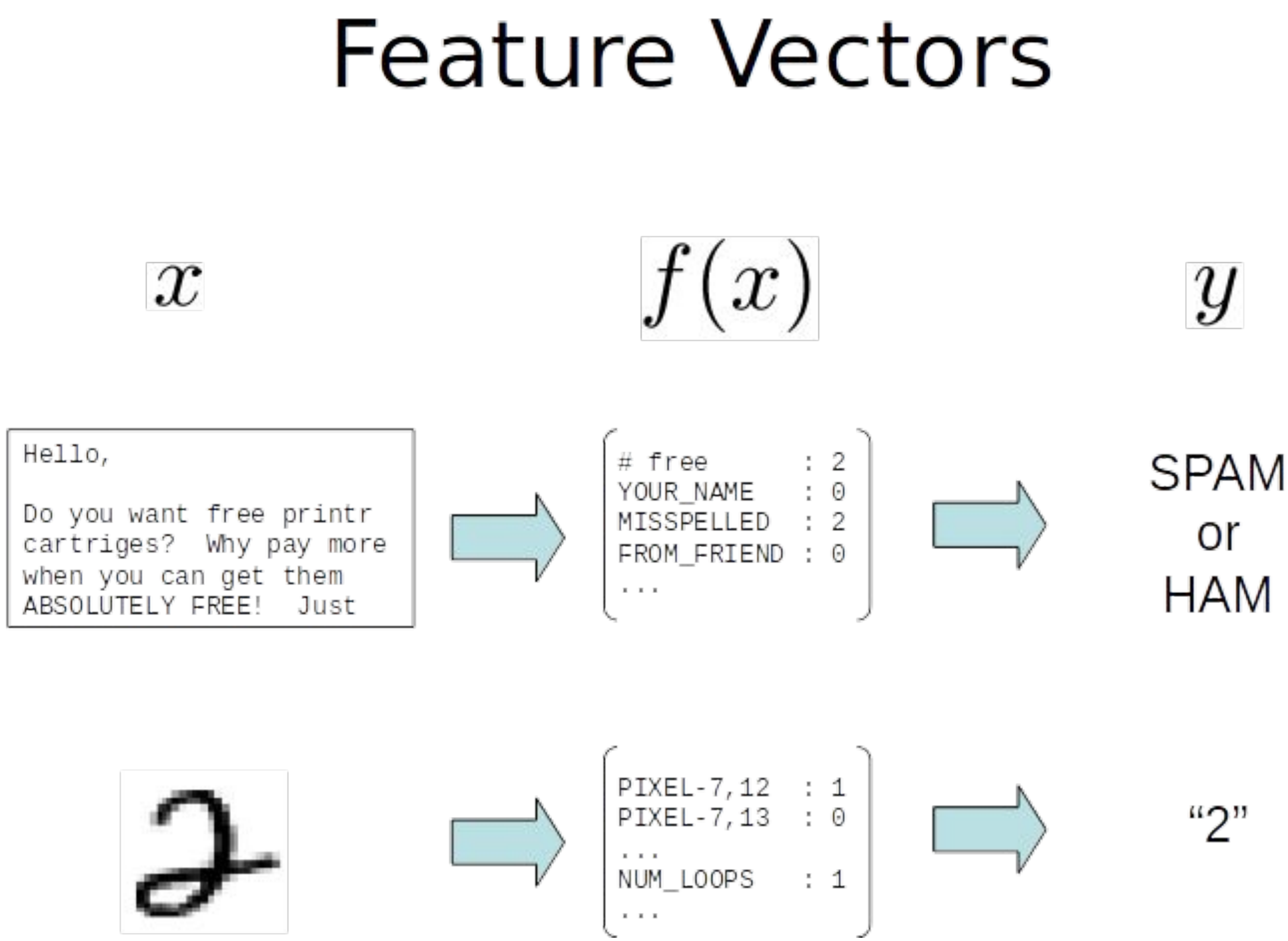
How it differs from **Never Ending Learning**?

- Here: Forgetting to be minimized, generalization: maximized



A super crash course in Artificial Neural Networks

Artificial Neural Networks Example: Predicting SPAM email

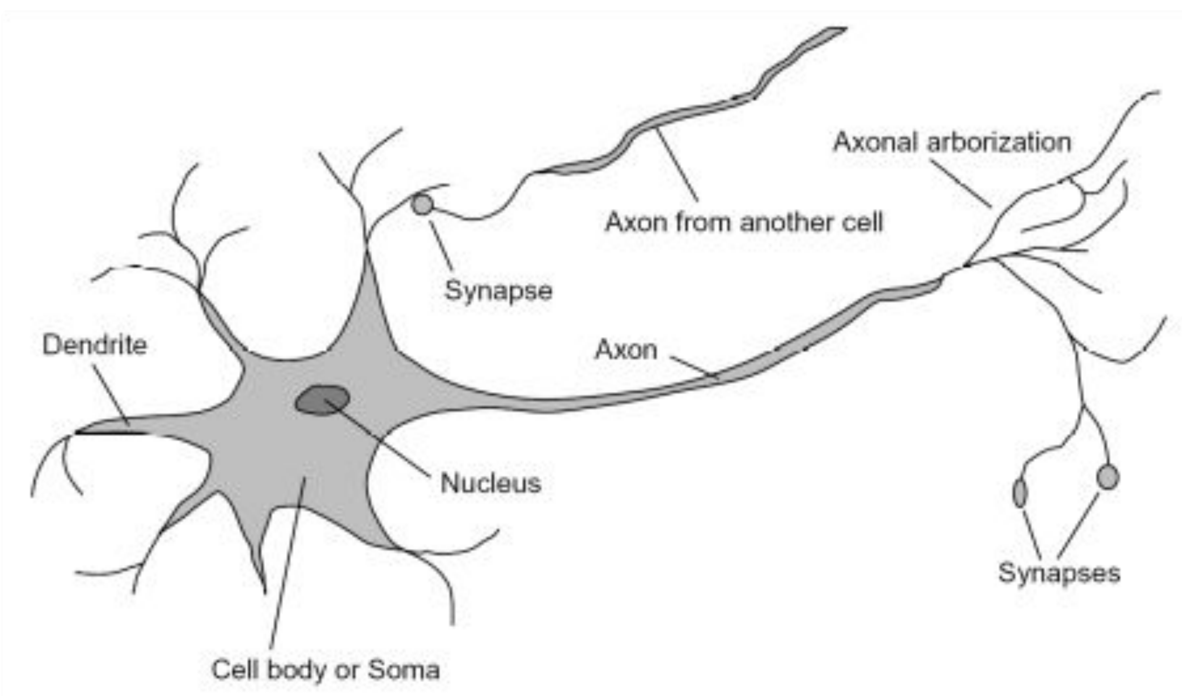


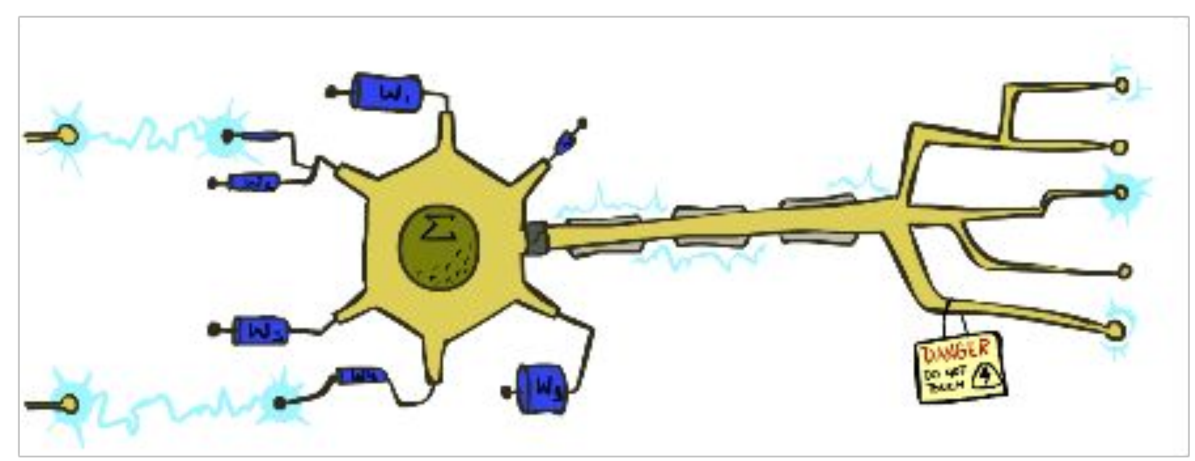
http://ai.berkeley.edu

Artificial Neural Networks

Some (Simplified) Biology

- Very loose inspiration: human neurons



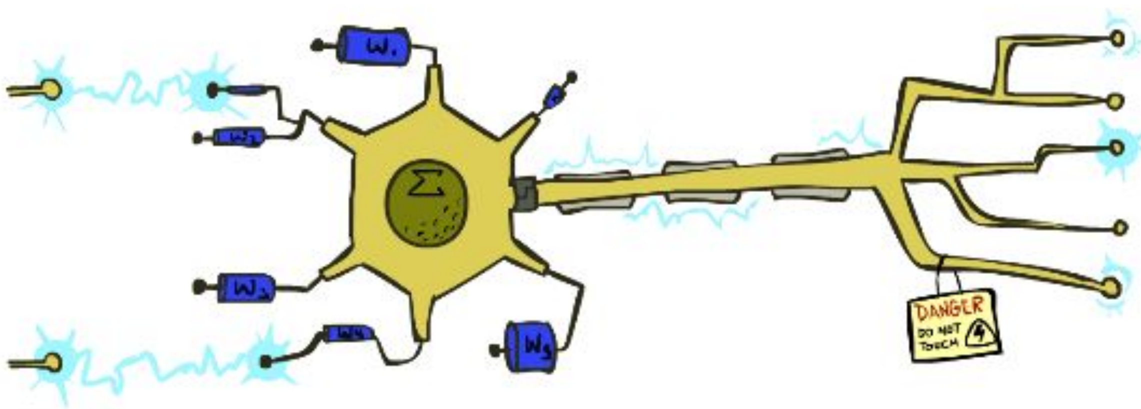


http://ai.berkeley.edu

Artificial Neural Networks

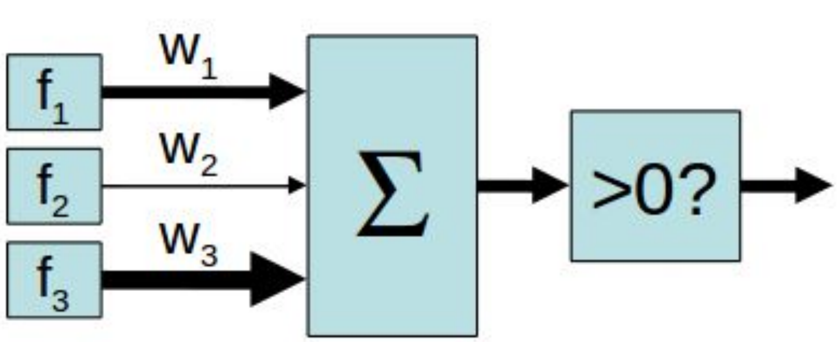
Linear Classifier

- Inputs are feature values
- Each feature has a weight
- Sum is the activation



$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
  - Positive, output +1
  - Negative, output -1

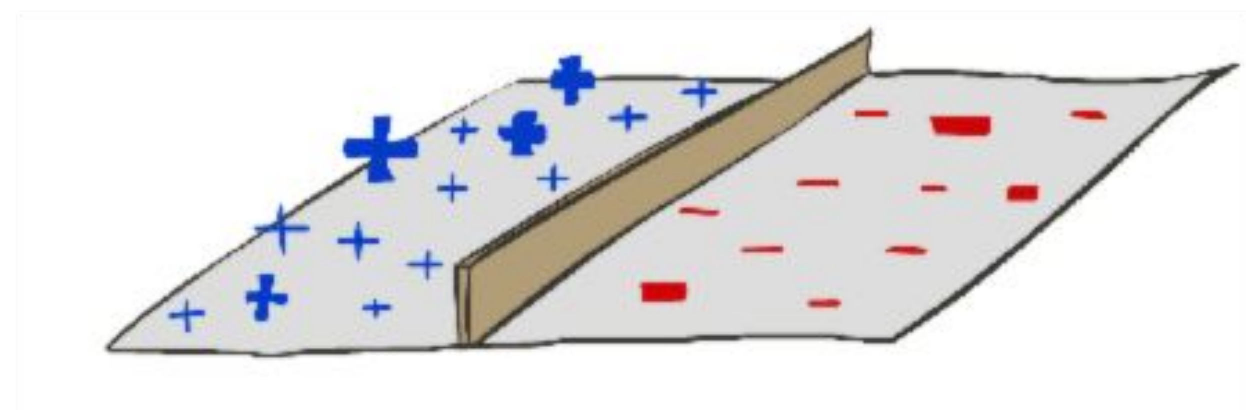


http://ai.berkeley.edu

Artificial Neural Networks

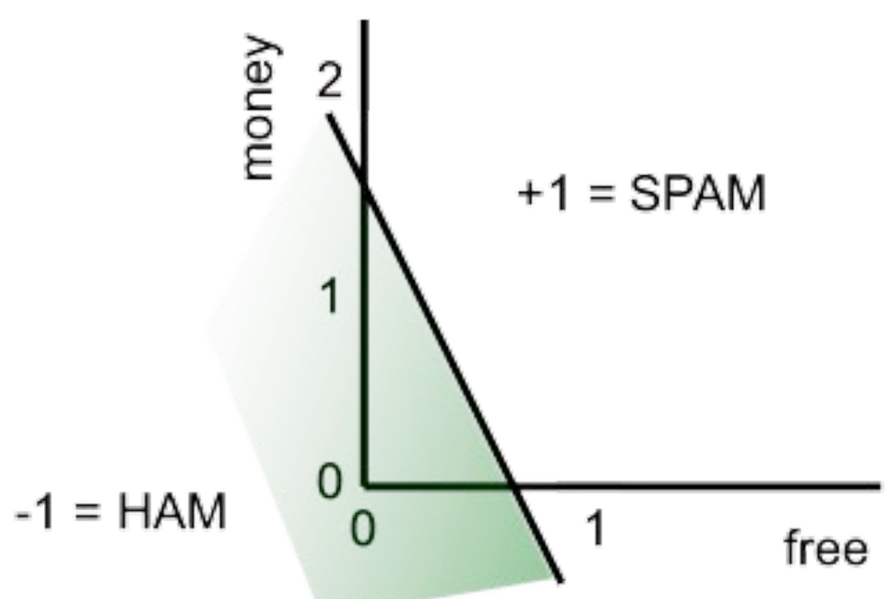
Binary Decision Rule

- In the space of feature vectors
  - Examples are points
  - Any weight vector is a hyperplane
  - One side corresponds to Y=+1
  - Other corresponds to Y=-1



w

```
BIAS  : -3
free  :  4
money :  2
...
```



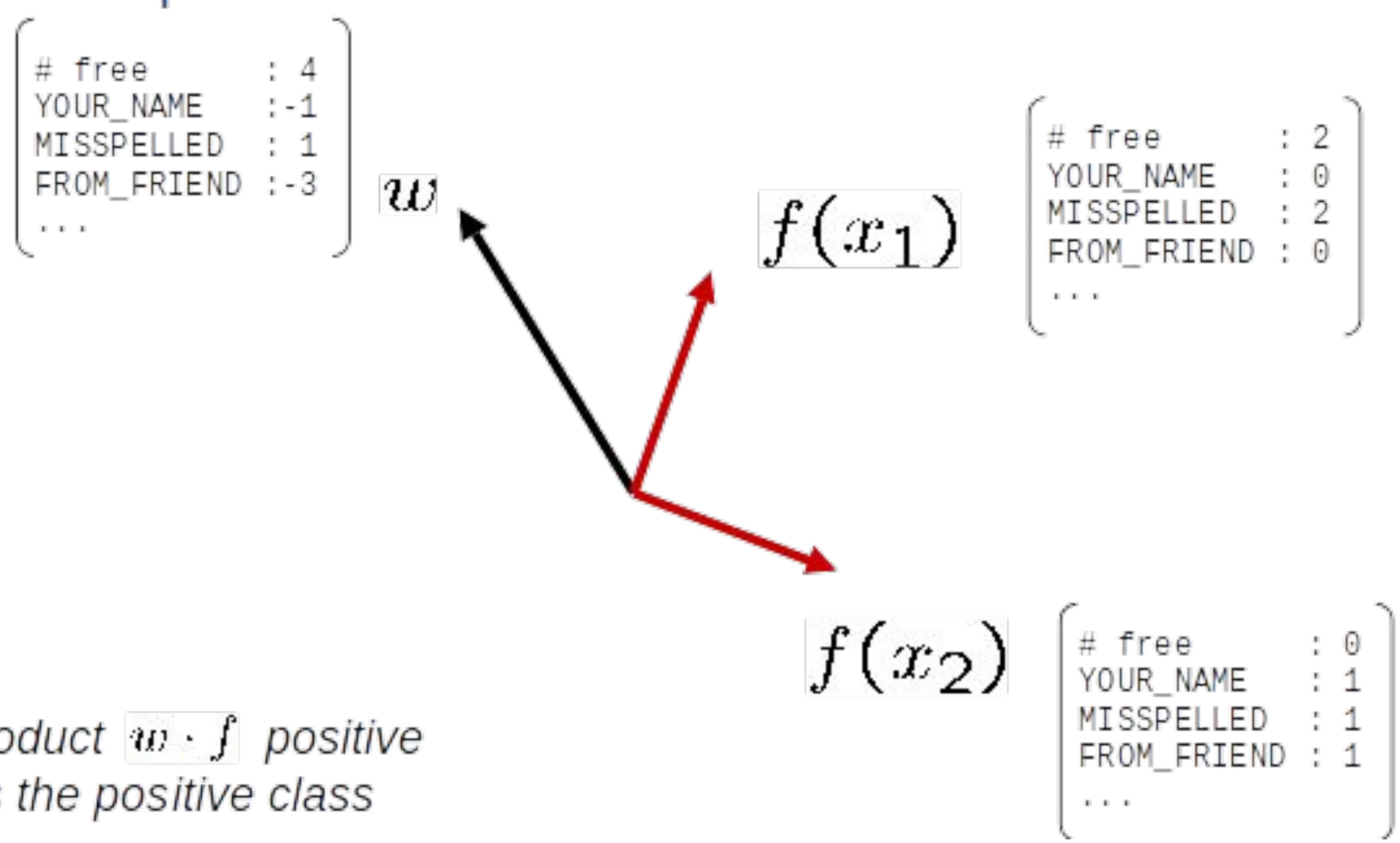
$f \cdot w = 0$

http://ai.berkeley.edu

Artificial Neural Networks

Weights

- Binary case: compare features to a weight vector
- Learning: figure out the weight vector from examples

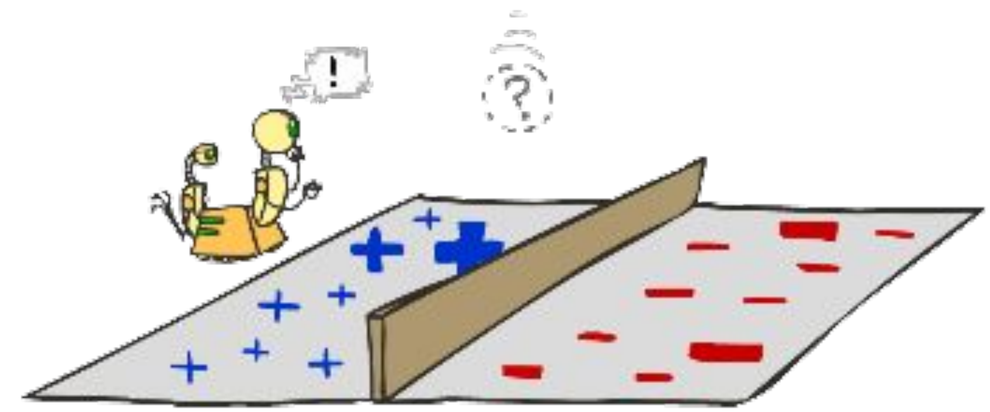


Dot product $w \cdot f$ *positive means the positive class*

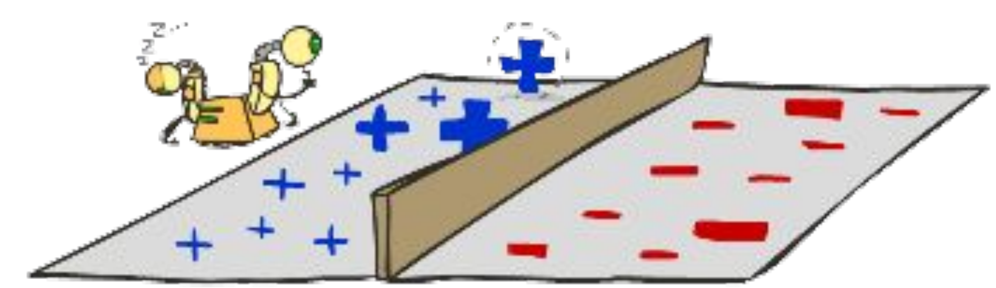
Artificial Neural Networks

Learning: Binary Perceptron

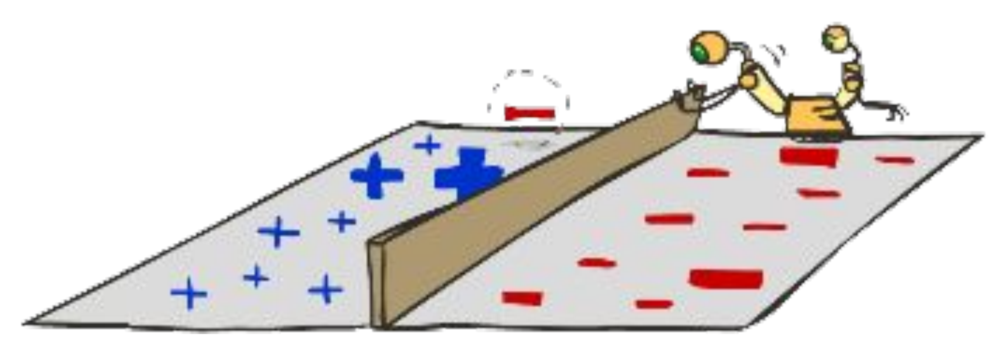
- Start with weights = 0
- For each training instance:
  - Classify with current weights



  - If correct (i.e., y=y*), no change!



  - If wrong: adjust the weight vector



http://ai.berkeley.edu

Catastrophic Forgetting

Catastrophic Forgetting (CF) [McCloskey and Cohen'89; French'99]

- Largest problem in Deep Neural Nets
- Training with new info interferes with previously learned knowledge

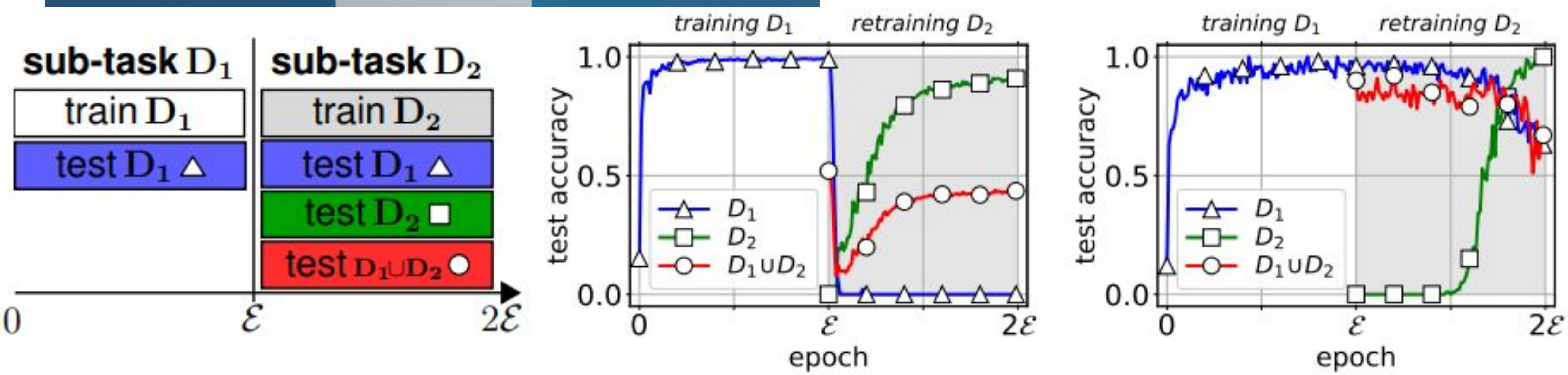
Catastrophic Forgetting (CF) [McCloskey and Cohen'89; French'99]

- Largest problem in Deep Neural Nets
- Training with new info interferes with previously learned knowledge
  - Abrupt performance decrease

[Pfulf & Gepperth'19]

Catastrophic Forgetting (CF) [McCloskey and Cohen'89; French'99]

- Largest problem in Deep Neural Nets
- Training with new info interferes with previously learned knowledge
    - Abrupt performance decrease
    - Knowledge overwritten

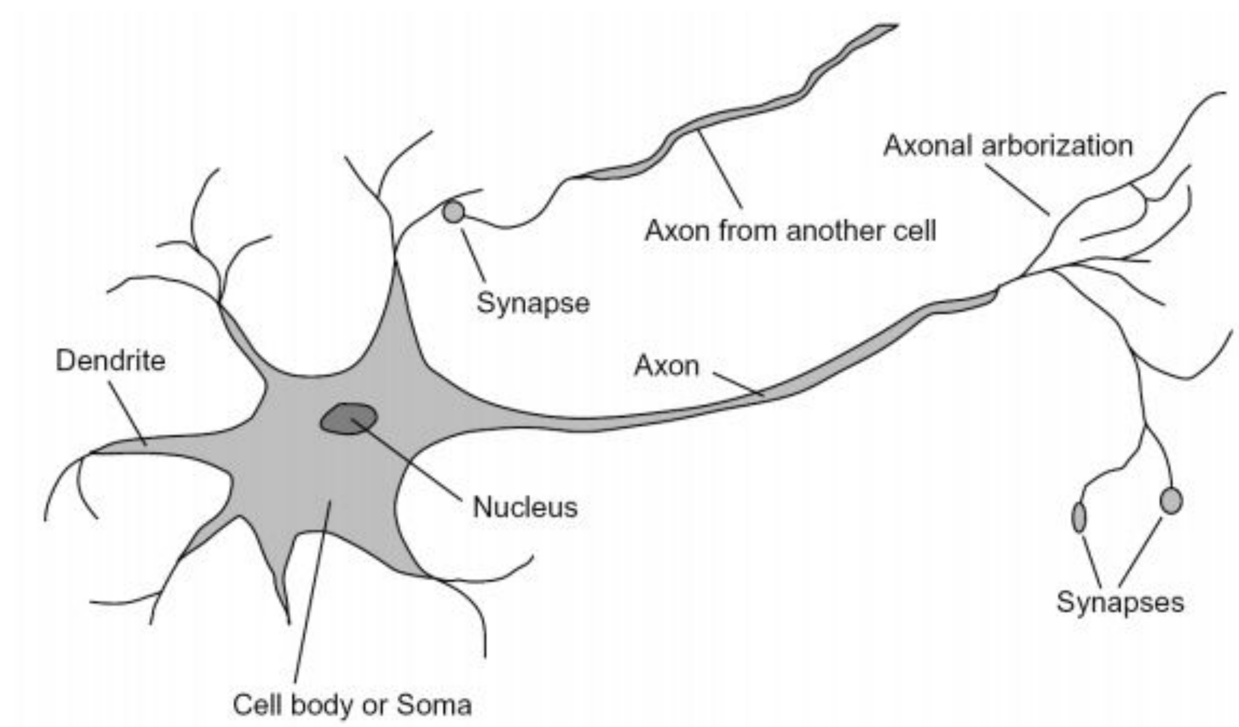


(a) Training scheme (b) with CF (c) without CF

[Pfulf & Gepperth'19]

Catastrophic Forgetting: Biological Factors

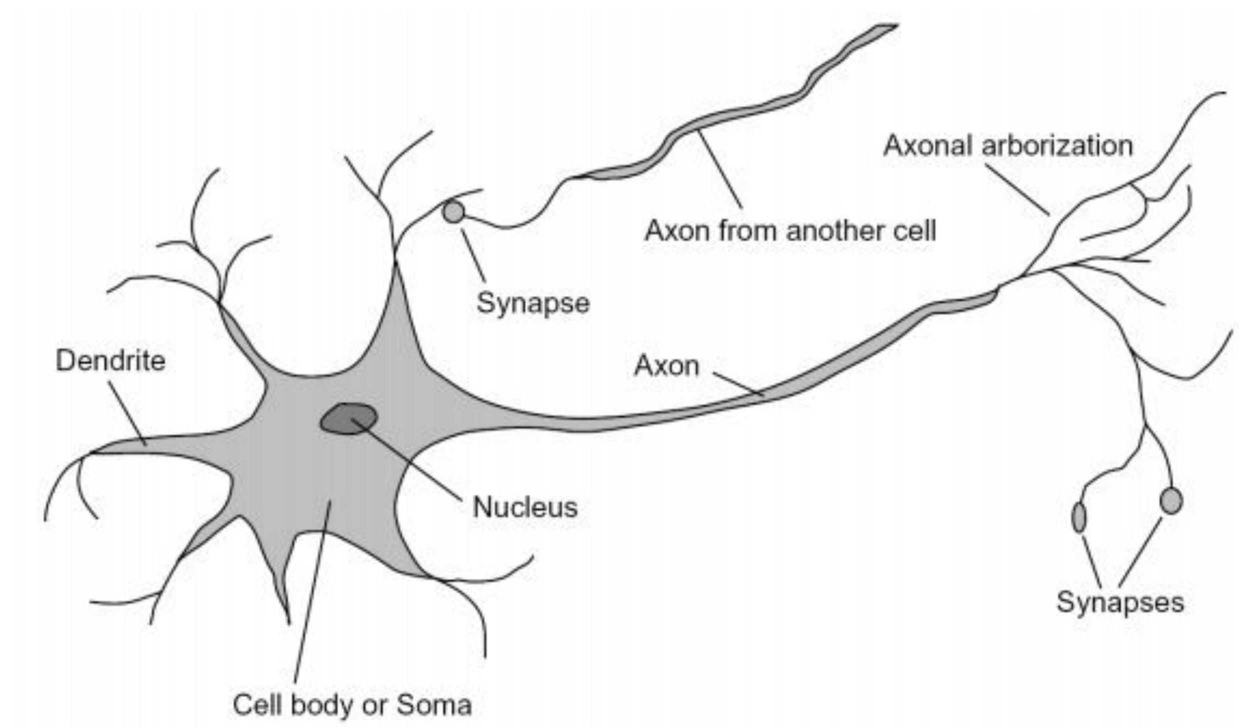
- (Structural) Stability-Plasticity Dilemma:
  - How to handle memories?
  - Remember past concepts?
  - Learn new concepts?
  - Generalize?



http://ai.berkeley.edu

Catastrophic Forgetting: Biological Factors

- (Structural) Stability-Plasticity Dilemma:
  - How to handle memories?
  - Remember past concepts?
  - Learn new concepts?
  - Generalize?
- Complementary Learning Systems (CLS) Theory



http://ai.berkeley.edu

A Continual Learning Framework

A Continual Learning Framework

Definition 1 ***Continual Distributions and Training Sets***

In Continual Learning, $\mathcal{D}$ can be thought of a potentially infinite sequence of unknown distributions $\mathcal{D} = \{D_1, \ldots, D_N\}$ over $X \times Y$, encountered over time, with X and Y as input and output

A Continual Learning Framework

Definition 1 ***Continual Distributions and Training Sets***

In Continual Learning, $\mathcal{D}$ can be thought of a potentially infinite sequence of unknown distributions $\mathcal{D} = \{D_1, \ldots, D_N\}$ over $X \times Y$, encountered over time, with X and Y as input and output

Definition 2 ***Task***

A task is a learning experience characterized by a unique task label t and its target function $g^*_{\hat{t}}(x) \equiv h^*(x, t = \hat{t})$, i.e., the objective of its learning.

A Continual Learning Framework

Definition 1 ***Continual Distributions and Training Sets***
In Continual Learning, $\mathcal{D}$ can be thought of a potentially infinite sequence of unknown distributions $\mathcal{D} = \{D_1, \ldots, D_N\}$ over $X \times Y$, encountered over time, with X and Y as input and output

Definition 2 ***Task***
A task is a learning experience characterized by a unique task label t and its target function $g^*_{\hat{t}}(x) \equiv h^*(x, t = \hat{t})$, i.e., the objective of its learning.

Definition 3 ***Continual Learning Algorithm*** *Given h^* as the general target function (i.e. our ideal prediction model), and a task label t, a continual learning algorithm A^{CL} is an algorithm with the following signature:*

$$\forall D_i \in \mathcal{D}, \qquad A_i^{CL}: \; < h_{i-1}, Tr_i, M_{i-1}, t_i > \rightarrow < h_i, M_i > \tag{1}$$

A Continual Learning Framework

A CL algorithm A^{CL} is an algorithm with the following signature:

$$\forall D_i \in \mathcal{D}, \quad A_i^{CL}: \; < h_{i-1}, Tr_i, M_{i-1}, t > \rightarrow < h_i, M_i > \tag{1}$$

Where h_i is the model, Tr_i is the training set of examples drawn from the respective D_i distribution, M_i is an external memory where we can store previous training examples and t is a task label. For simplicity, we can assume N as the number of tasks, one for each Tr_i.

A Continual Learning Framework

In Continual Learning,

- $\mathcal{D} = \{D_1, \ldots, D_N\}$: a potentially infinite sequence of unknown distributions over $X \times Y$ encountered over time
- X and Y input and output r.v.
- h^*: general target function (i.e. our ideal prediction model)
- Task: defined by a unique task label t and its target function

A Continual Learning Framework

In Continual Learning,

- $\mathcal{D} = \{D_1, \ldots, D_N\}$: a potentially infinite sequence of unknown distributions over $X \times Y$ encountered over time
- X and Y input and output r.v.
- h^*: general target function (i.e. our ideal prediction model)
- Task: defined by a unique task label t and its target function

A CL algorithm A^{CL} has the signature:

$$\forall D_i \in \mathcal{D}, \qquad A_i^{CL}: \ < h_{i-1}, Tr_i, M_{i-1}, t > \rightarrow < h_i, M_i > \tag{1}$$

A Continual Learning Framework

In Continual Learning,

- $\mathcal{D} = \{D_1, \ldots, D_N\}$: a potentially infinite sequence of unknown distributions over $X \times Y$ encountered over time
- X and Y input and output r.v.
- h^*: general target function (i.e. our ideal prediction model)
- Task: defined by a unique task label t and its target function

A CL algorithm A^{CL} has the signature:

$$\forall D_i \in \mathcal{D}, \qquad A_i^{CL}: \; < h_{i-1}, Tr_i, M_{i-1}, t > \rightarrow < h_i, M_i > \tag{1}$$

where

- h_i: the model
- Tr_i: training set of examples drawn from distribution D_i
- M_i: external memory (can store previous training examples)
- N: nr of tasks

Definition 4 ***Continual Learning scenarios***

- *Single-Incremental-Task (SIT):* $t_1 = t_2 = \cdots = t_N$.
- *Multi-Task (MT):* $\forall i, j \in [1, .., n]^2 : t_i \neq t_j$.
- *Multi-Incremental-Task (MIT):* $\exists\ i, j, k :\ t_i = t_j$ *and* $t_j \neq t_k$.

Task/Session	**CL settings**		
Task ID	**SIT**	**MT**	**MIT**
t_1	0	1	0
t_2	0	2	1
t_3	0	3	0
...	...	...	...
t_i	0	i	...

Definition 4 ***Continual Learning scenarios***

- *Single-Incremental-Task (SIT):* $t_1 = t_2 = \cdots = t_N$.
- *Multi-Task (MT):* $\forall i, j \in [1, .., n]^2 : t_i \neq t_j$.
- *Multi-Incremental-Task (MIT):* $\exists\ i, j, k :\ t_i = t_j$ *and* $t_j \neq t_k$.

Task/Session	**CL settings**		
Task ID	**SIT**	**MT**	**MIT**
t_1	0	1	0
t_2	0	2	1
t_3	0	3	0
...	...	...	...
t_i	0	i	...

Definition 4 ***Continual Learning scenarios***

A CL scenario is a specific CL setting in which the sequence of N task labels respects a certain "task structure" over time. Based on the proposed framework, we can define three different common scenarios:

- *Single-Incremental-Task (SIT):* $t_1 = t_2 = \cdots = t_N$.
- *Multi-Task (MT):* $\forall i, j \in [1, .., n]^2 : t_i \neq t_j$.
- *Multi-Incremental-Task (MIT):* $\exists\ i, j, k :\ t_i = t_j$ *and* $t_j \neq t_k$.

Task/Session	**CL settings**		
Task ID	**SIT**	**MT**	**MIT**
t_1	0	1	0
t_2	0	2	1
t_3	0	3	0
...	...	...	...
t_i	0	i	...

Example of task IDs

Definition 5 ***Content Update Type*** *The nature of the data samples or observations*

- *New Instances (NI): Data samples or observations contained in the training set at time-step i are relative to the same dependent variable used in the past.*
- *New Concepts (NC): Data samples or observations contained in the training set at time-step i are relative to a new dependent variable to be learned from the model.*
- *New Instances and New Concepts (NIC): Data samples or observations contained in the training set at time-step i are relative to both, already encountered dependent variables, and new ones.*

Definition 5 ***Content Update Type*** *The nature of the data samples or observations contained in each Tr_i can be conveniently framed in three different categories:*

- *New Instances (NI): Data samples or observations contained in the training set at time-step i are relative to the same dependent variable used in the past.*
- *New Concepts (NC): Data samples or observations contained in the training set at time-step i are relative to a new dependent variable to be learned from the model.*
- *New Instances and New Concepts (NIC): Data samples or observations contained in the training set at time-step i are relative to both, already encountered dependent variables, and new ones.*

Continual
Learning
Strategies

CL Strategies

- Regularization
  - Penalty computing
  - Knowledge distillation

CL Strategies

- Regularization
  - Penalty computing
  - Knowledge distillation
- Dynamic architectures
  - Implicit/Explicit architecture modification
  - Dual Architectures

CL Strategies

- Regularization
  - Penalty computing
  - Knowledge distillation
- Dynamic architectures
  - Implicit/Explicit architecture modification
  - Dual Architectures
- Replay
  - Rehearsal
  - Generative replay

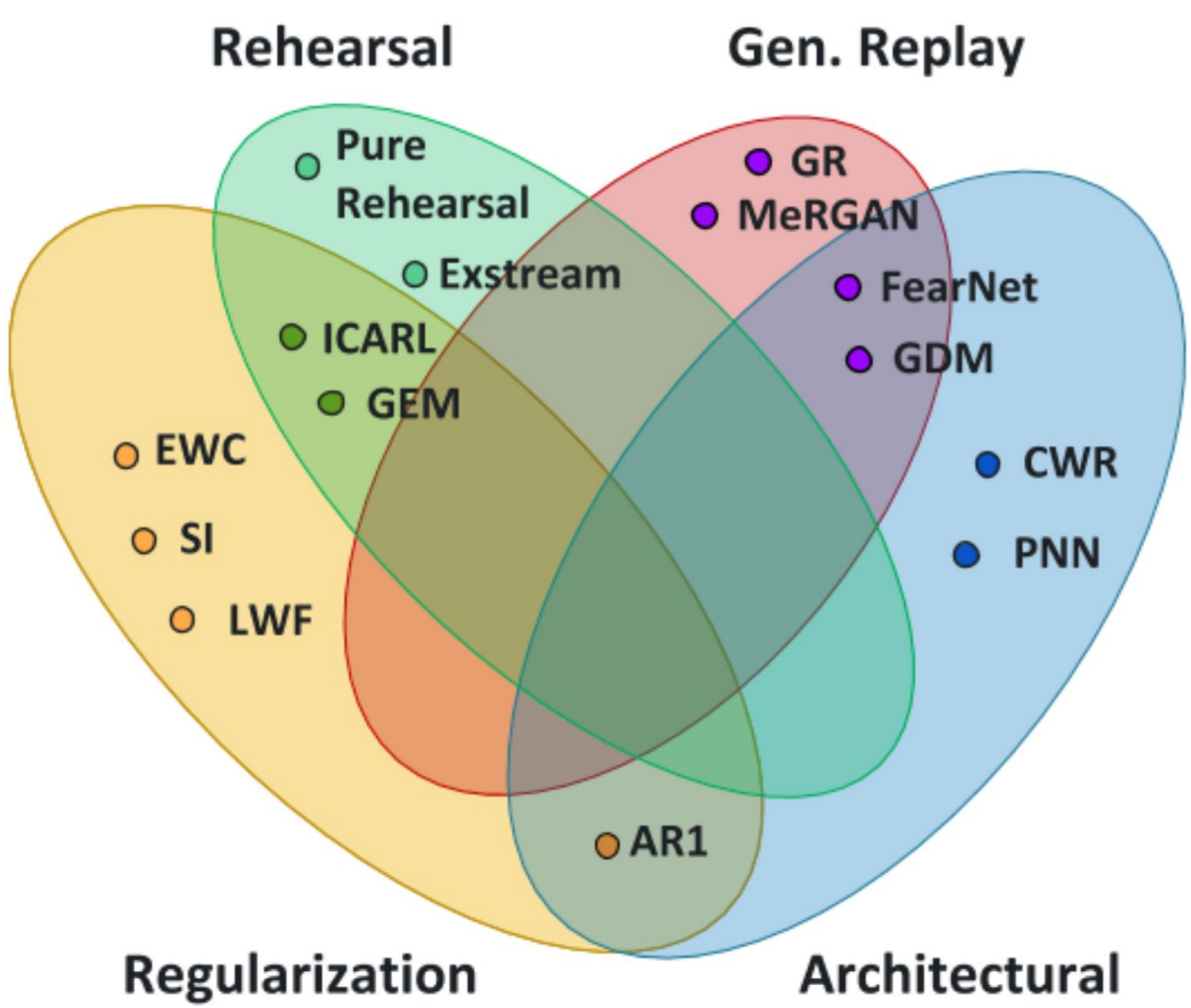
Popular Continual Learning Strategies

References	Regularization	Rehearsal	Architectural	Generative-Replay
Zhou et al. (2012)				
Goodfellow et al. (2013)	✓			
Lyubova et al. (2015)				
Rusu et al. (2016)	✓			
Camoriano et al. (2016)				
Furlanello et al. (2016)	✓		✓	
Li and Hoiem (2017a)	✓		✓	
Rusu et al. (2016)			✓	
Jung et al. (2016)	✓		✓	
Aljundi et al. (2016)				
Rebuffi et al. (2016)	✓	✓		
Kirkpatrick et al. (2017)	✓			
Fernando et al. (2017)				
Lee et al. (2017)	✓			
Lee et al.	✓			
Triki et al. (2017)	✓			
Seff et al. (2017)	✓			
Shin et al. (2017)				✓
Velez and Clune (2017)	✓			
Lopez-Paz and Ranzato (2017)	✓	✓		
Zenke et al. (2017)	✓			
Kemker et al. (2017)				
Nguyen et al. (2017)				
Ramapuram et al. (2017)	✓			✓
Mallya and Lazebnik			✓	
Kamra et al. (2017)				✓
Draelos et al. (2017)				✓
Serra et al. (2018)	✓			
Mallya and Lazebnik (2018)			✓	
Parisi et al. (2018)	✓		✓	✓
He and Jaeger (2018)	✓		✓	
Wu et al. (2018b)		✓		✓
Ritter et al. (2018)	✓			
Schwarz et al. (2018)		✓		
Maltoni and Lomonaco (2018)	✓		✓	
Achille et al. (2018)			✓	✓
Wu et al. (2018a)	✓			✓
Lesort et al. (2018b)				✓
Caselles-Dupré et al. (2019)				✓
Lesort et al. (2018a)				✓
Sprechmann et al. (2018)		✓	✓	
Kemker and Kanan (2018)			✓	✓
Chaudhry et al. (2019)	✓	✓		

[Díaz-Rodríguez, Lesort, Lomonaco et al.'19]

Popular Continual Learning Strategies

References	Regularization	Rehearsal	Architectural	Generative-Replay
Zhou et al. (2012)				
Goodfellow et al. (2013)	✓			
Lyubova et al. (2015)				
Rusu et al. (2016)	✓			
Camoriano et al. (2016)				
Furlanello et al. (2016)	✓		✓	
Li and Hoiem (2017a)	✓		✓	
Rusu et al. (2016)			✓	
Jung et al. (2016)	✓		✓	
Aljundi et al. (2016)				
Rebuffi et al. (2016)	✓	✓		
Kirkpatrick et al. (2017)	✓			
Fernando et al. (2017)				
Lee et al. (2017)	✓			
Lee et al.	✓			
Triki et al. (2017)	✓			
Seff et al. (2017)	✓			
Shin et al. (2017)				✓
Velez and Clune (2017)	✓			
Lopez-Paz and Ranzato (2017)	✓	✓		
Zenke et al. (2017)	✓			
Kemker et al. (2017)				
Nguyen et al. (2017)				
Ramapuram et al. (2017)	✓			✓
Mallya and Lazebnik			✓	
Kamra et al. (2017)				✓
Draelos et al. (2017)				✓
Serra et al. (2018)	✓			
Mallya and Lazebnik (2018)			✓	
Parisi et al. (2018)	✓		✓	✓
He and Jaeger (2018)	✓		✓	
Wu et al. (2018b)		✓		✓
Ritter et al. (2018)	✓			
Schwarz et al. (2018)		✓		
Maltoni and Lomonaco (2018)	✓		✓	
Achille et al. (2018)			✓	✓
Wu et al. (2018a)	✓			✓
Lesort et al. (2018b)				✓
Caselles-Dupré et al. (2019)				✓
Lesort et al. (2018a)				✓
Sprechmann et al. (2018)		✓	✓	
Kemker and Kanan (2018)			✓	✓
Chaudhry et al. (2019)	✓	✓		

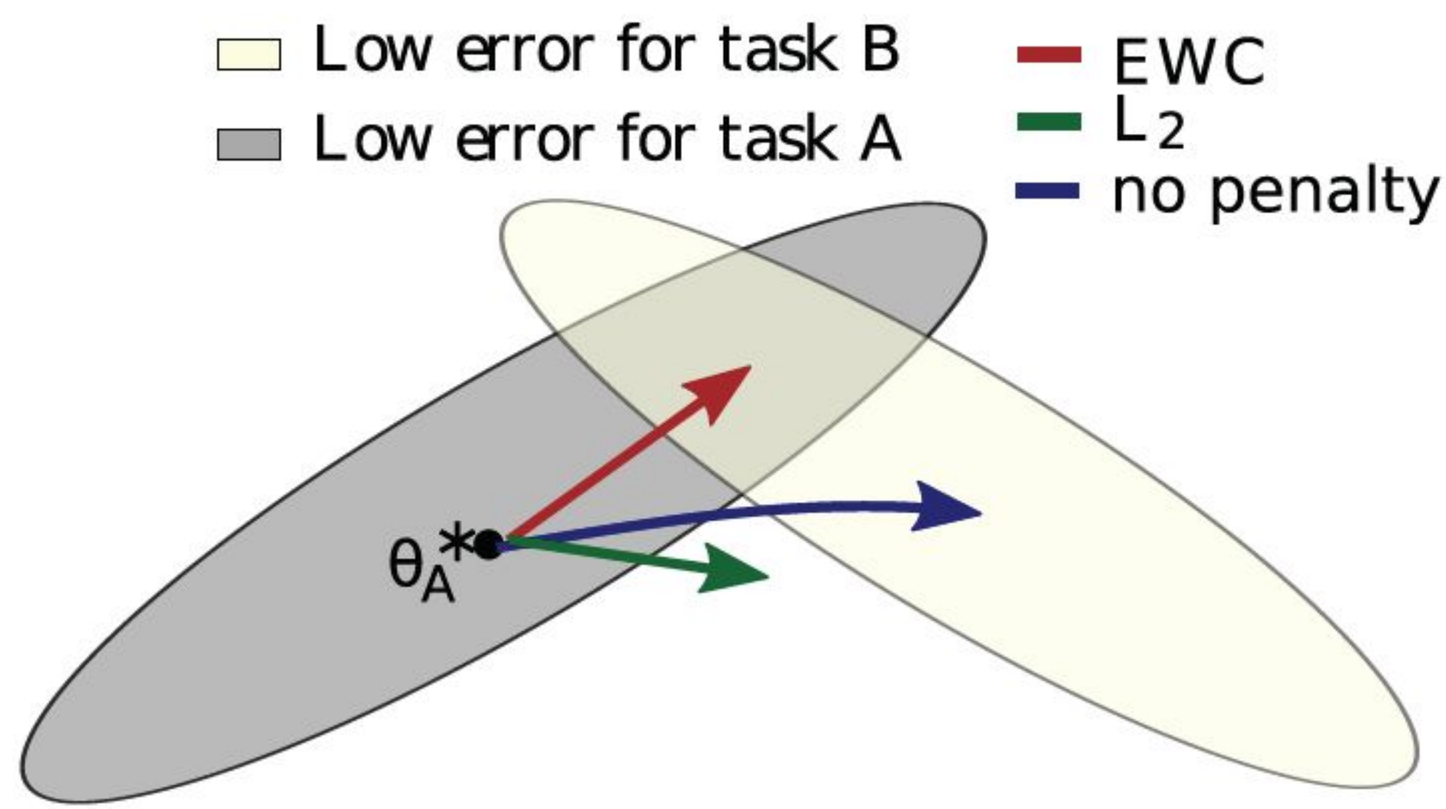


[Díaz-Rodríguez, Lesort, Lomonaco et al.'19]

Regularization approaches

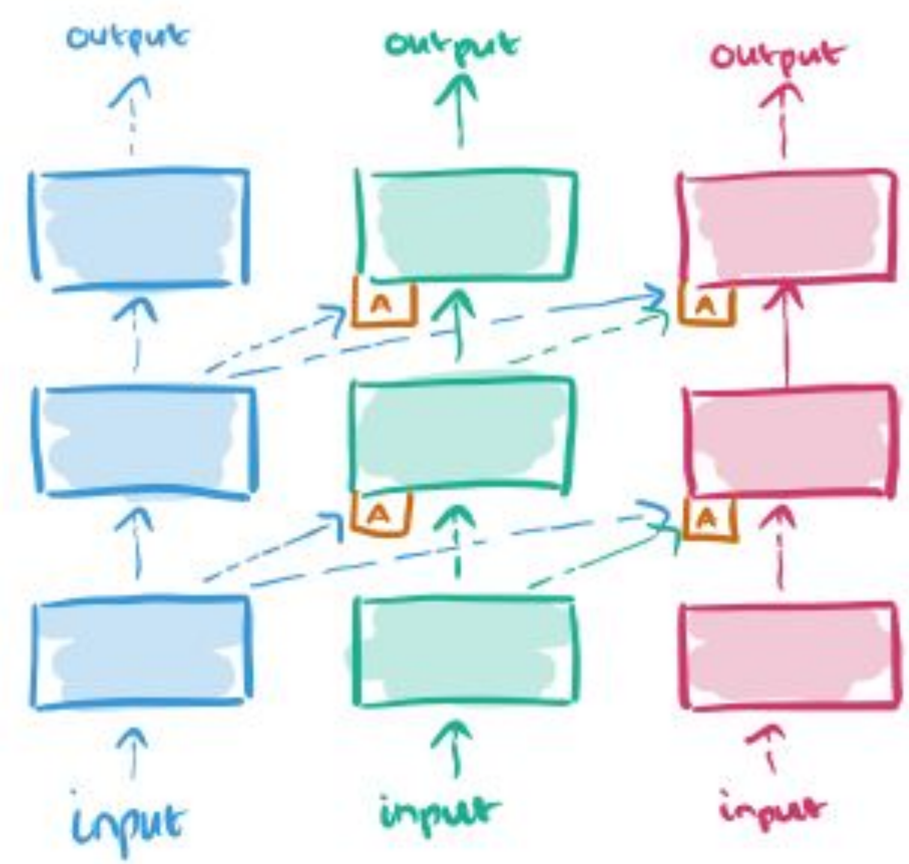
Regularization Strategies

- **Penalty** computing Synaptic intelligence [Zenke'17], AR1 [Maltoni18]
  - E.g.: EWC [Kirkpatrick'17]: Estimates weights' importance to regularize accordingly (λ)



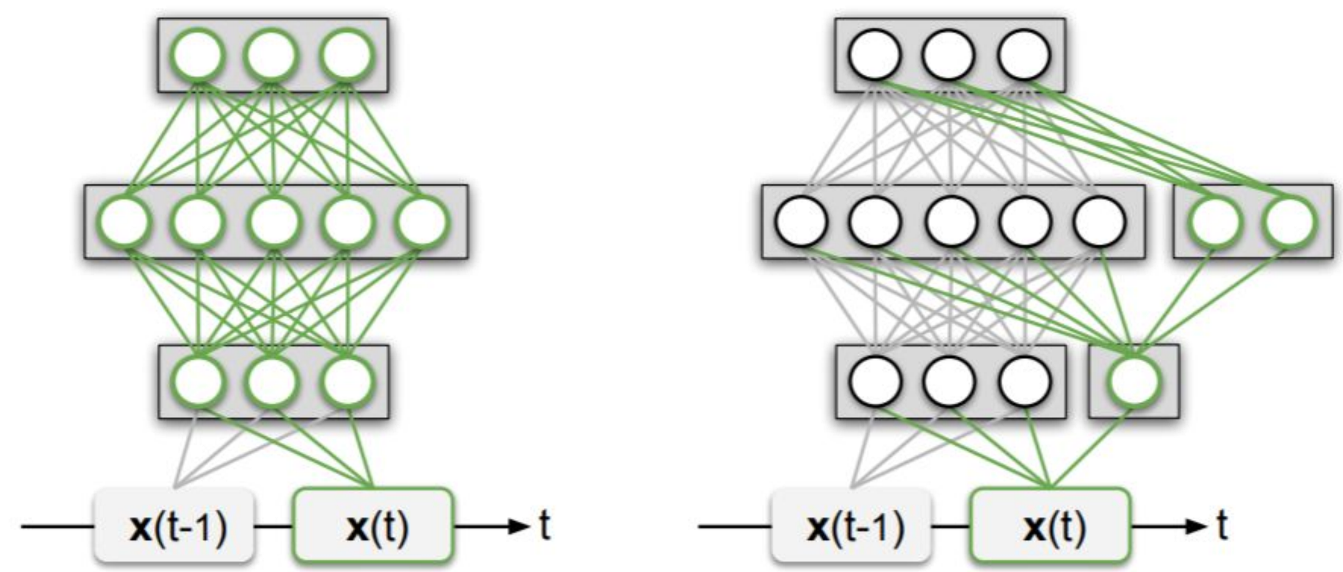
Regularization Strategies

- **Penalty** computing Synaptic intelligence [Zenke'17], AR1 [Maltoni18]
  - E.g.: EWC [Kirkpatrick'17]: Estimates weights' importance to regularize accordingly (λ)
- **Distillation:** *Transfer* previously learned tasks to a new model [Hinton'15], Distral [Teh'17]



Dynamic architecture approaches

Dynamic Architectures



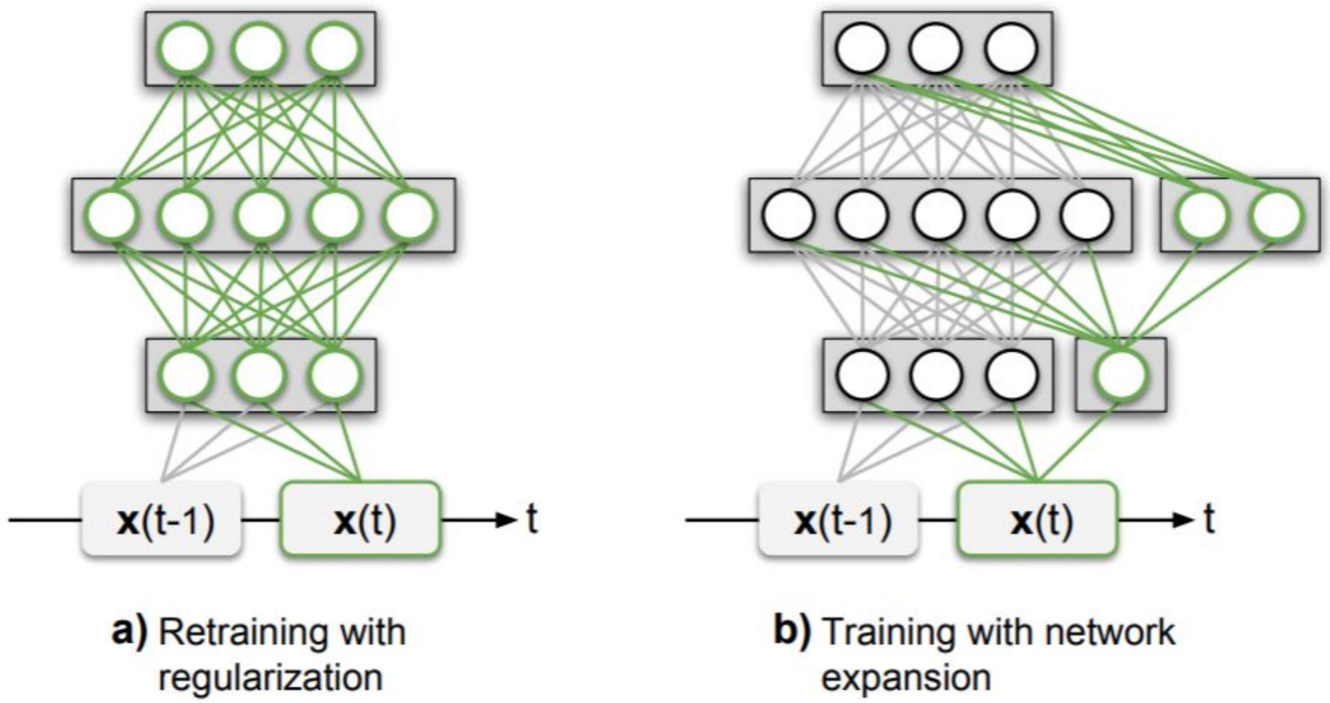
a) Retraining with regularization

b) Training with network expansion

[Parisi & Ji'19]

Dynamic Architectures

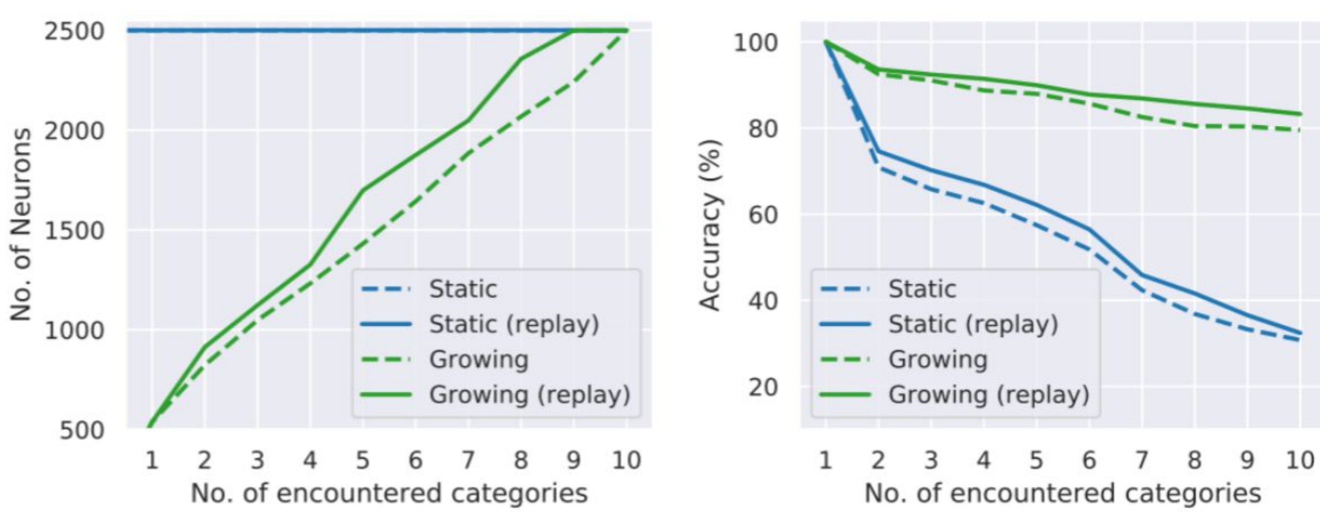
- Outperform static nets in incremental settings (under fix memory budget)
  - Progressive NN [Rusu'16], Progress & Compress [Schwarz'18], LwF [Li'16], GWR [Parisi'17], A-LTM [Furlanello'16], [Li'19]

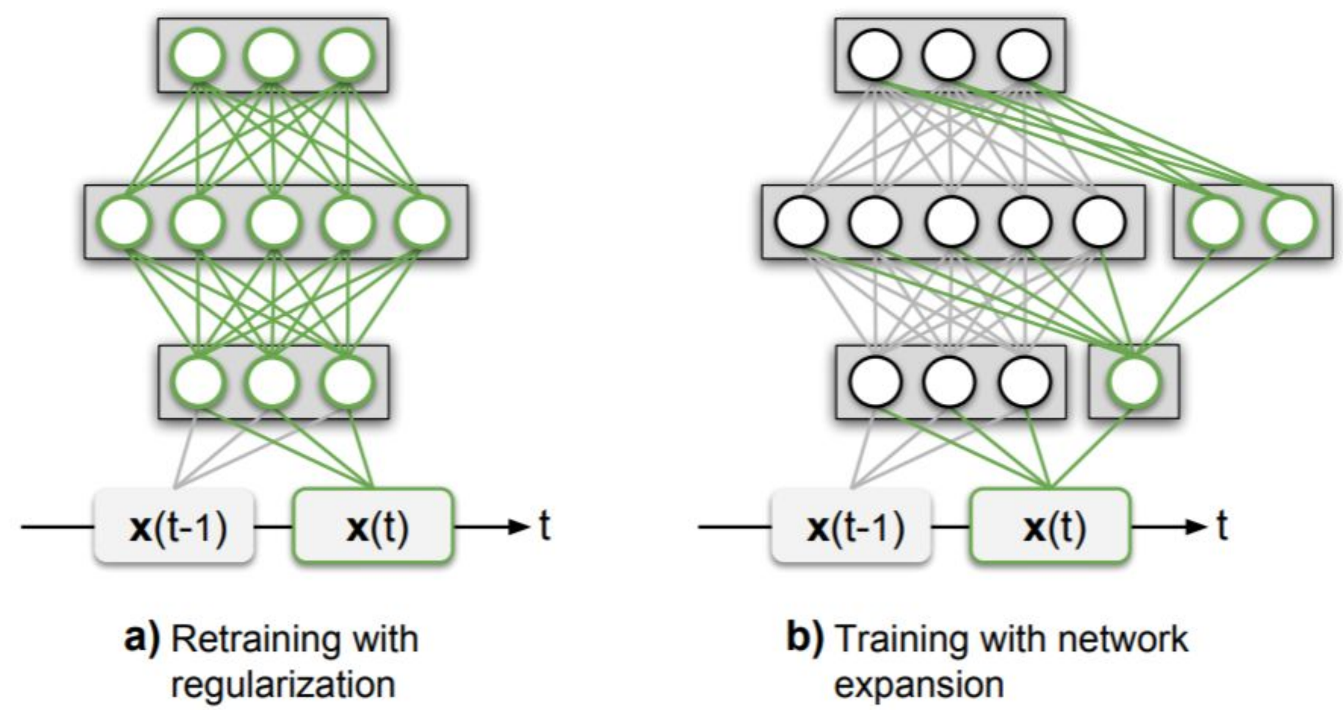


[Parisi & Ji'19]

Dynamic Architectures

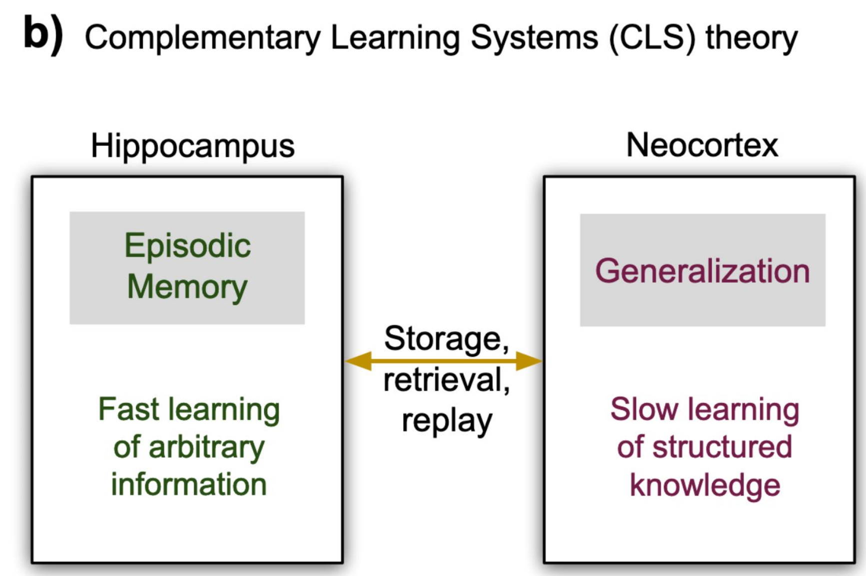
- Outperform static nets in incremental settings (under fix memory budget)
  - Progressive NN [Rusu’16], Progress & Compress [Schwarz’18], LwF [Li’16], A-LTM [Furlanello’16], [Li’19]
- **Neurogenesis**: structural plasticity prevents CF in non-stationary environments GWR [Parisi’17]





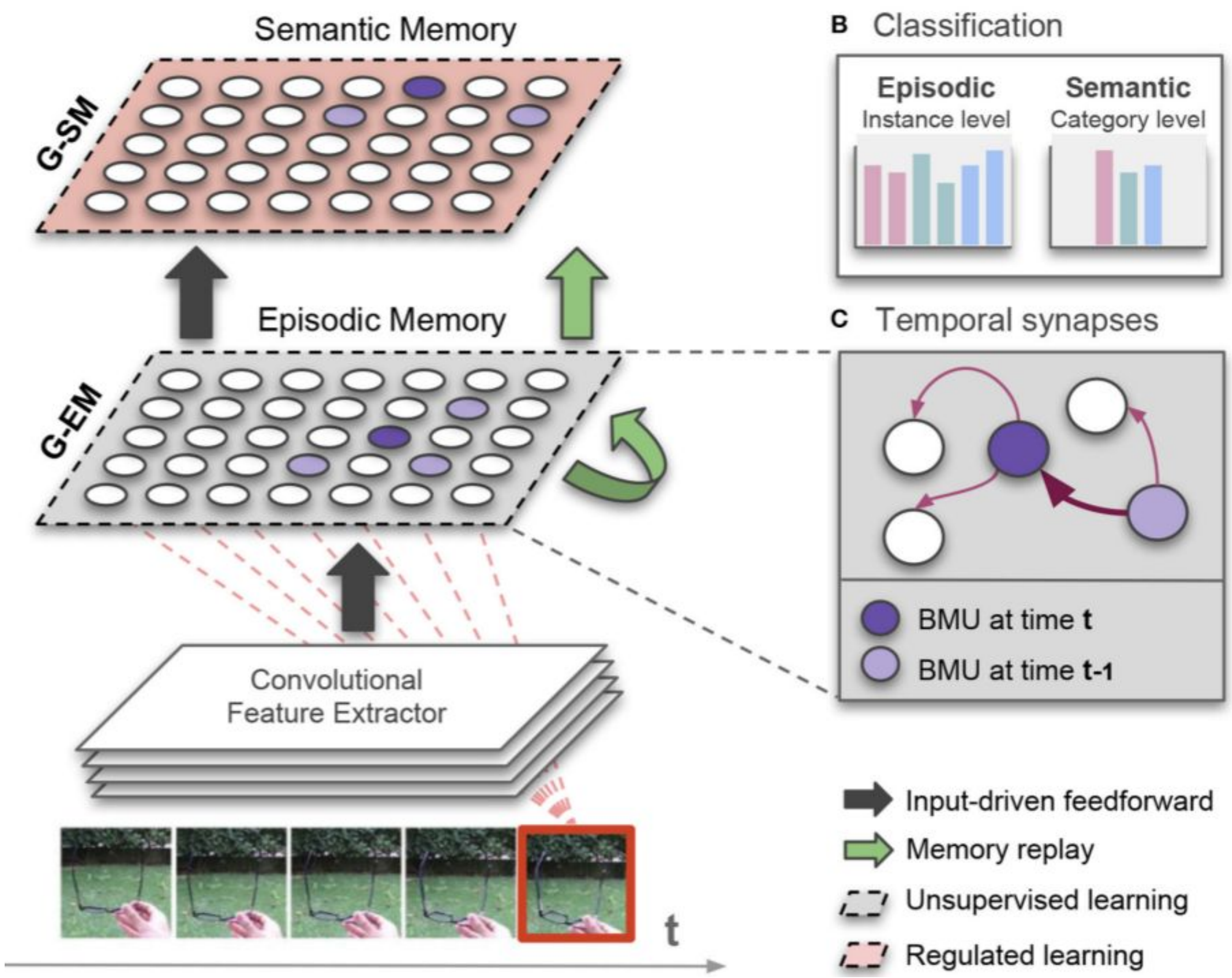
[Parisi & Ji'19]

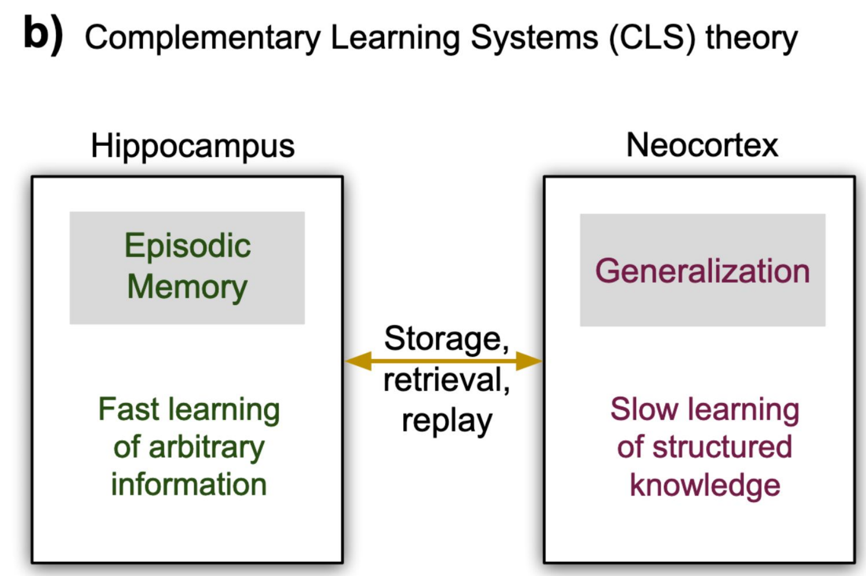
Dual Memories



[McClelland'95, Parisi'18]

Dual Memories

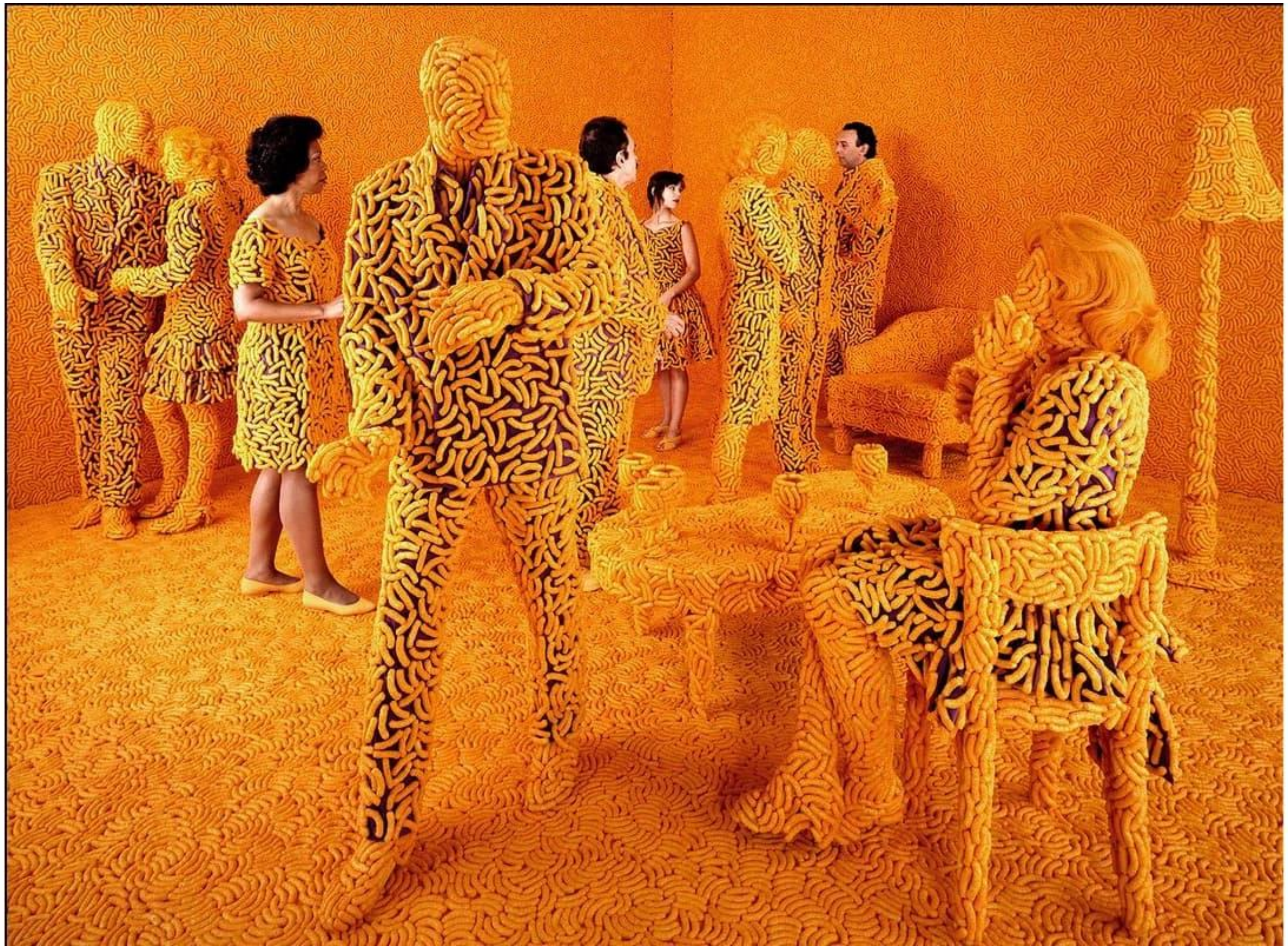




[McClelland'95, Parisi'18]

Lifelong Learning of Spatio-temporal Representations with Dual-Memory Recurrent Self-Organization. Frontiers in Neurorobotics.[Parisi, Tani, Weber, Wermter'19],[Kumaran'16]

Replay approaches

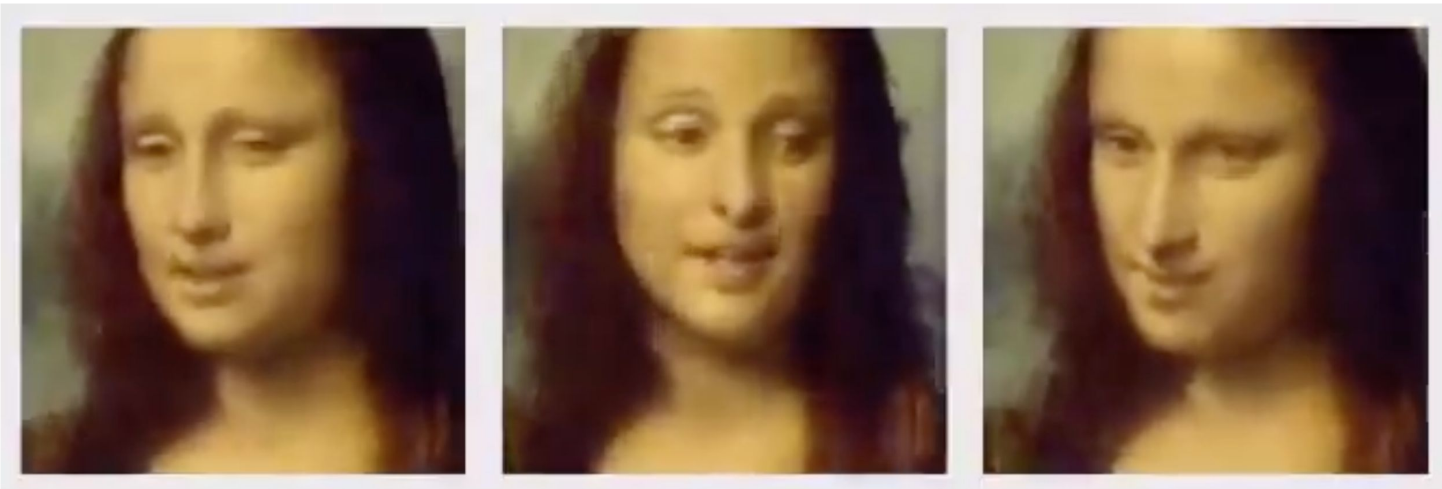


Rehearsal approaches [Robins'95]

- **Rehearsal**: "Buying souvenirs"
  a. Learn task A
  b. Fine-tune params with batches from task A + task B sampled uniformly e.g., ICARL [Rebuffi'16]

Rehearsal approaches [Robins'95]

- **Rehearsal**: "Buying souvenirs"
  a. Learn task A
  b. Fine-tune params with batches from task A + task B sampled uniformly e.g., ICARL [Rebuffi'16]
- **Generative Replay** [Wu18, Shin17] of *fake* samples

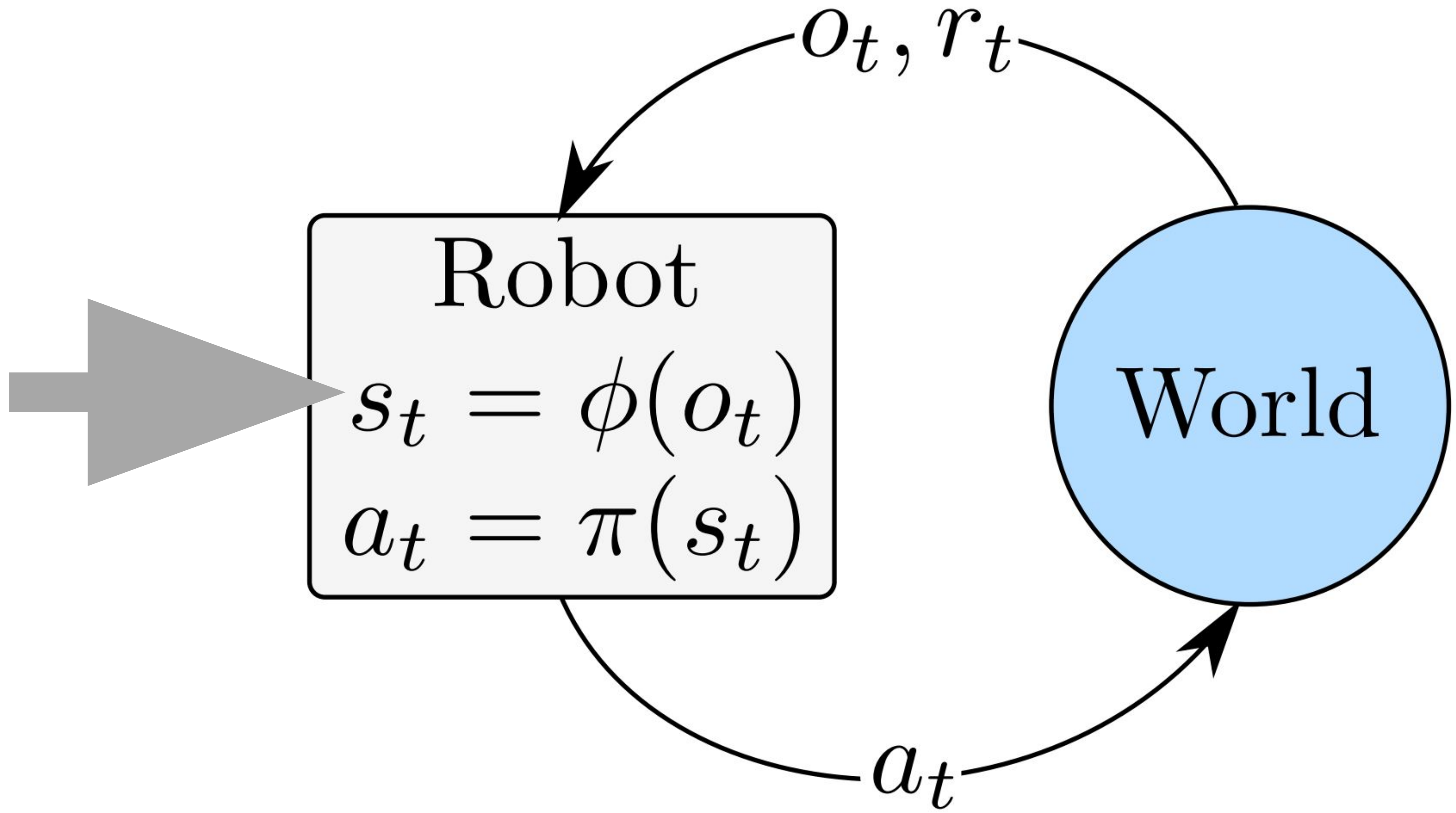


Mona Lisa to speak [Zakharov, Shysheya et al'19]

Example:
Continual
State
Representation
Learning (SRL)

State Representation Learning (SRL) in RL context

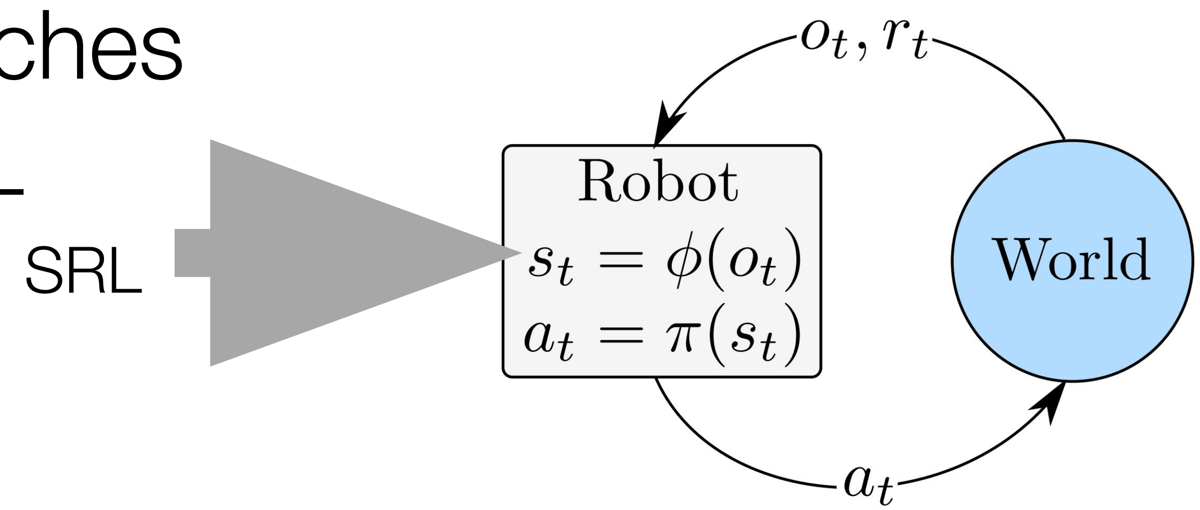
SRL



[Lesort'18, State representation learning for control: An overview. Neural Networks]

Learning states vs learning representations

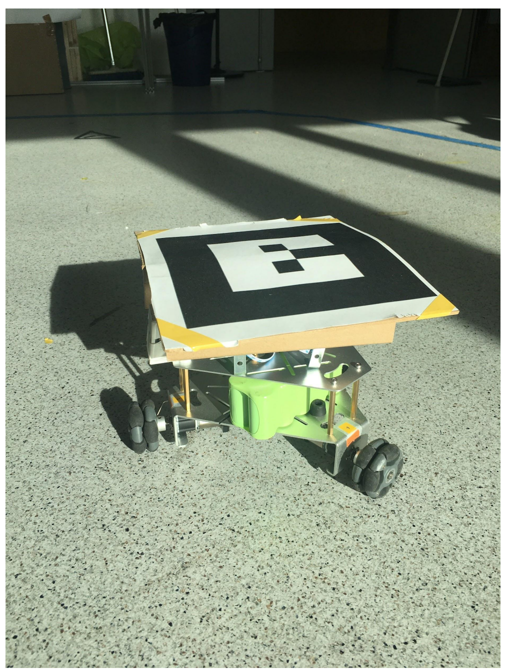
- Learning a state:
  - A particular case of representation learning
  - Entails a control/robotics context where actions are possible
  - Often: look for interpretable info./ info with a *physical* meaning
- Learning a state can be an objective by itself, but is often present in more general approaches
  - E.g., as an auxiliary task in RL



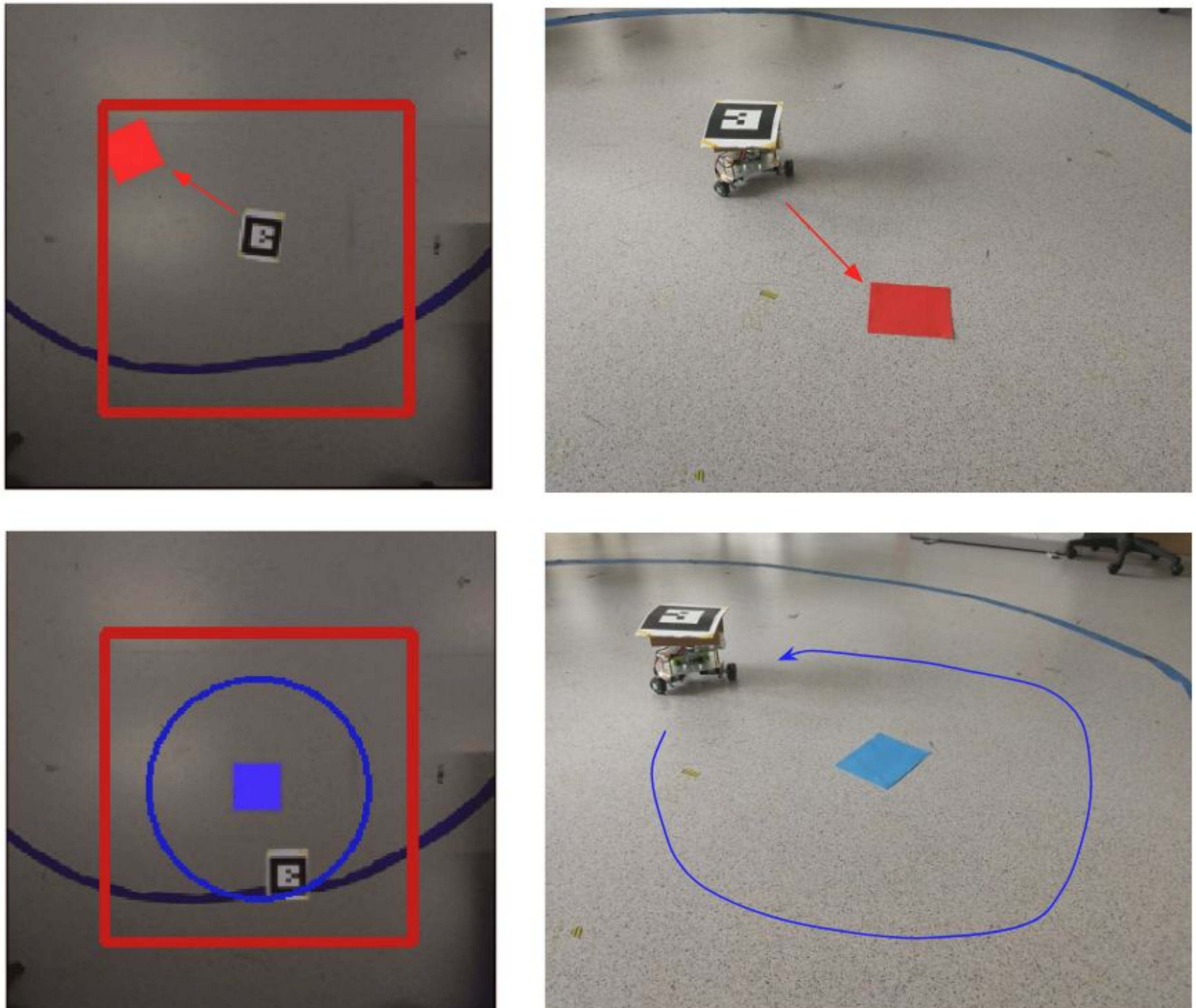
[Lesort'18, State representation learning for control: An overview. Neural Networks]

Continual RL in multi-task, lifelong, real life setting

Task 1



Task 2

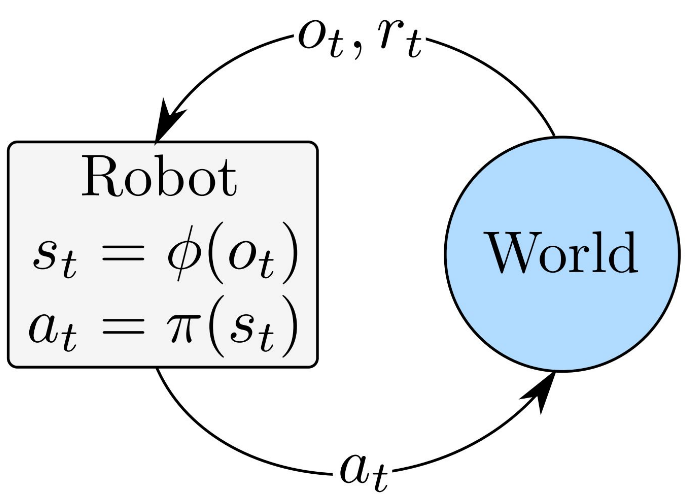


Simulation

Real Life

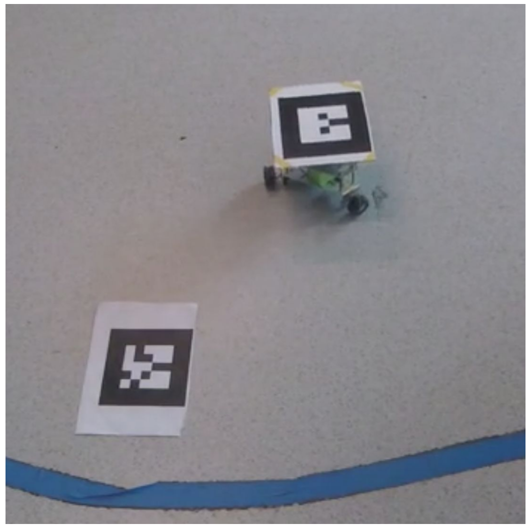
Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer. [Traoré'19]
State representation learning for control: An overview. Neural Networks. [Lesort'18]
DREAM EU H2020 Project: www.robotsthatdream.eu

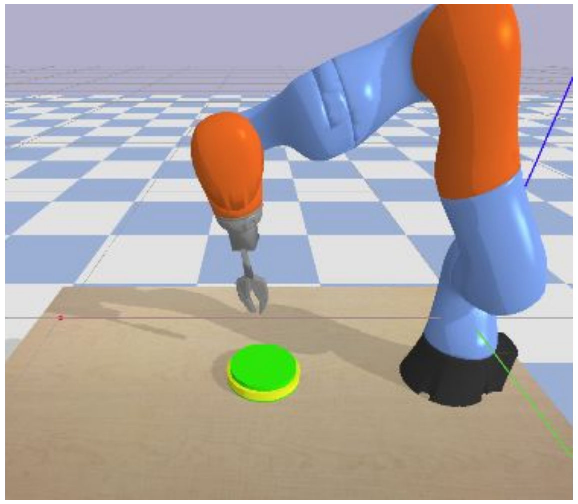
Continual State Representation Learning (SRL)



Vision goal-based tasks: a mobile, visible goal to reach

- Can't be solved using robot states alone
- Should encode goal position



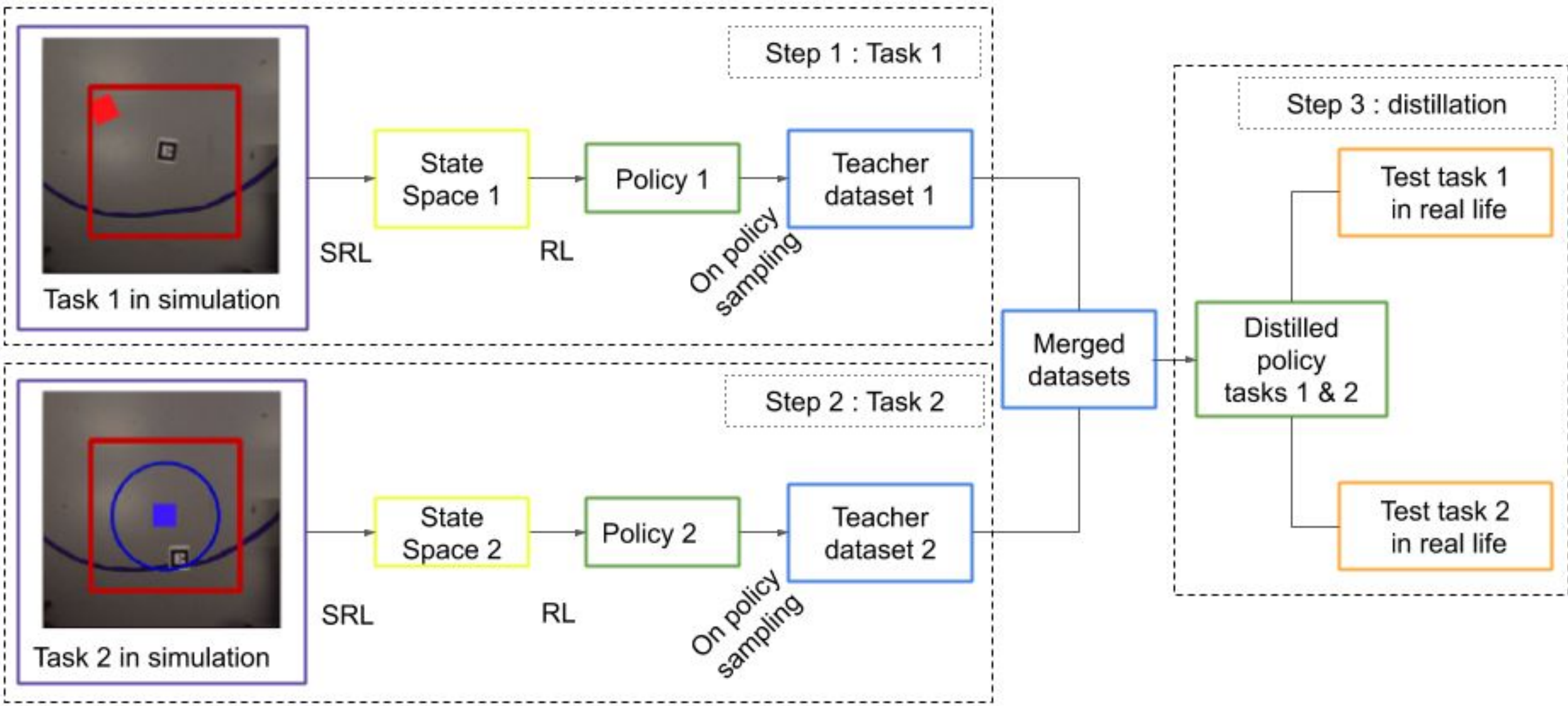


[Lesort'18, State representation learning for control: An overview. Neural Networks]

DREAM EU H2020 Project: www.robotsthatdream.eu

Continual RL in multi-task, lifelong & real life settings

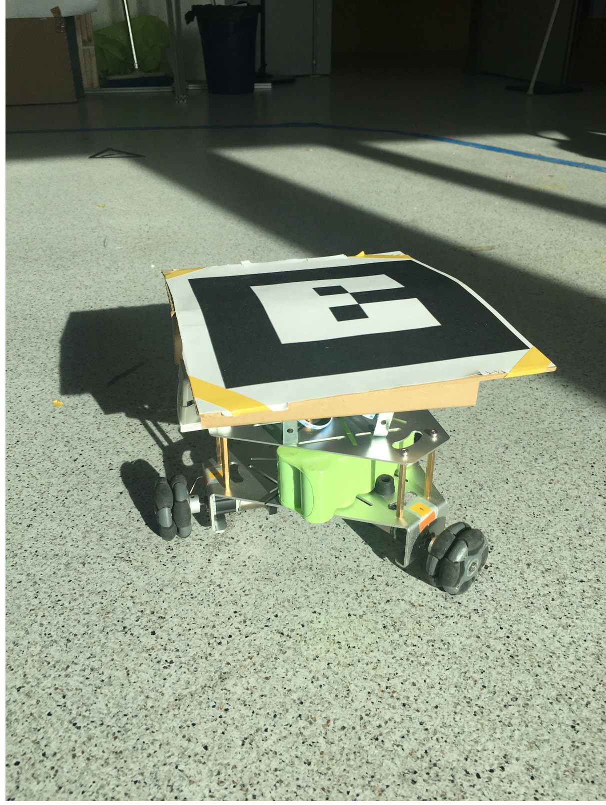
- Distilling a teacher network's knowledge (=each task policy) to a student
  => A single network to perform all tasks
- Deploying the multi-task policy sim2real
- No task label at test time!

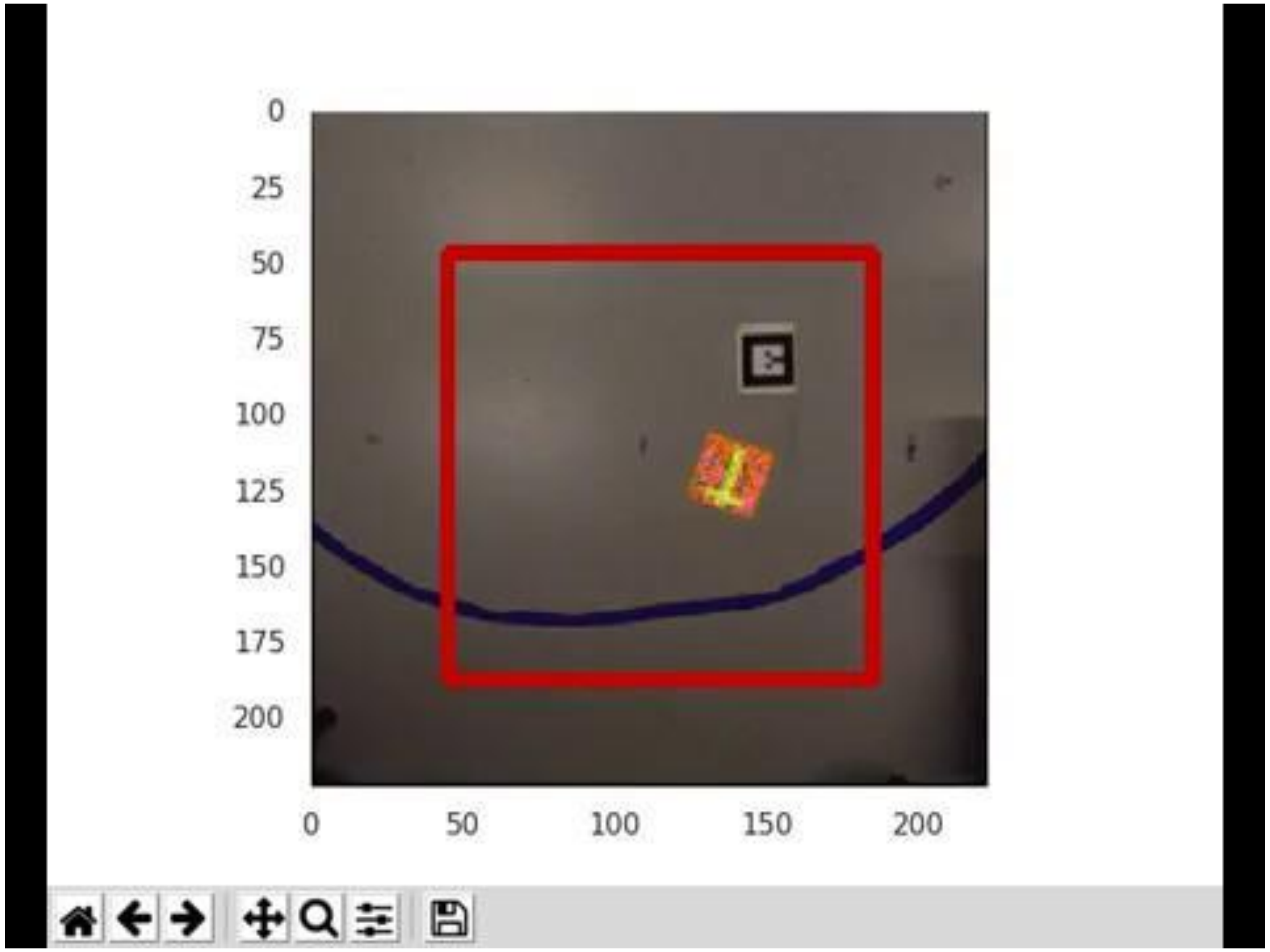


Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer. [Traoré'19]

Results

- Sim2Real policy distillation
- SRL + Policy learning continually, sequentially
- No test time task labels

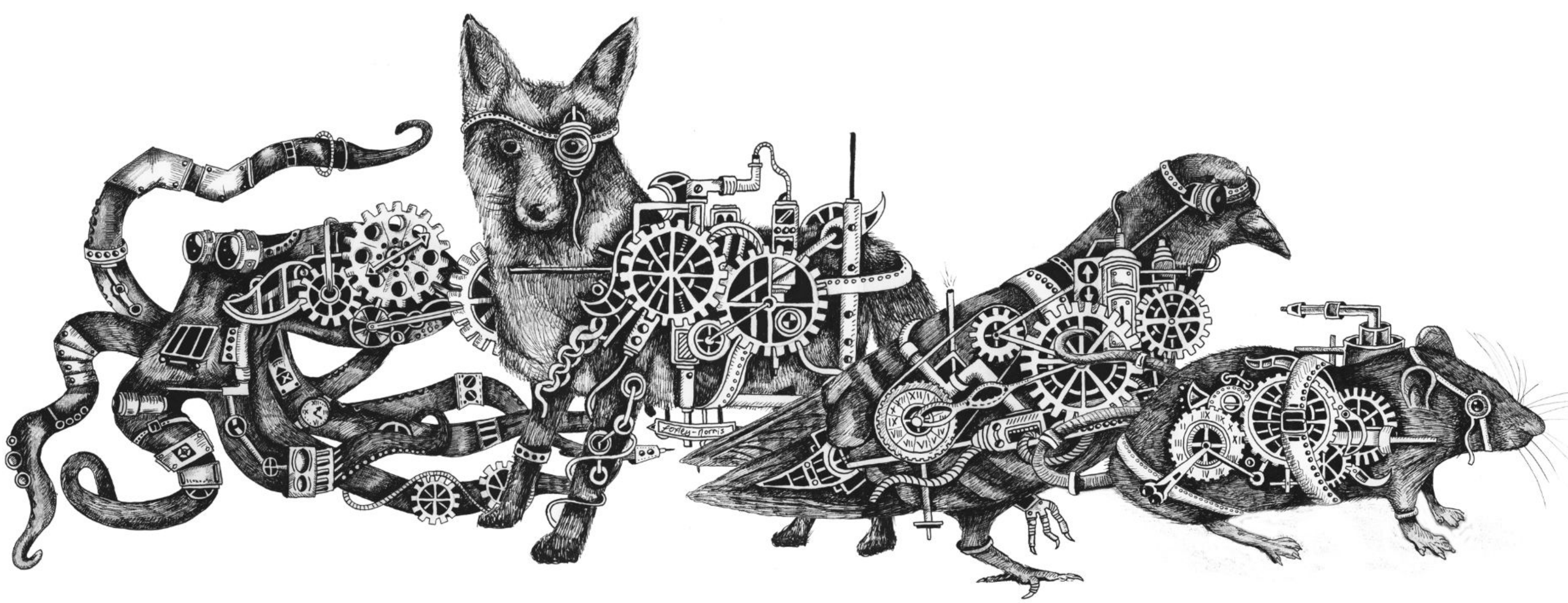




Continual Learning Benchmarks

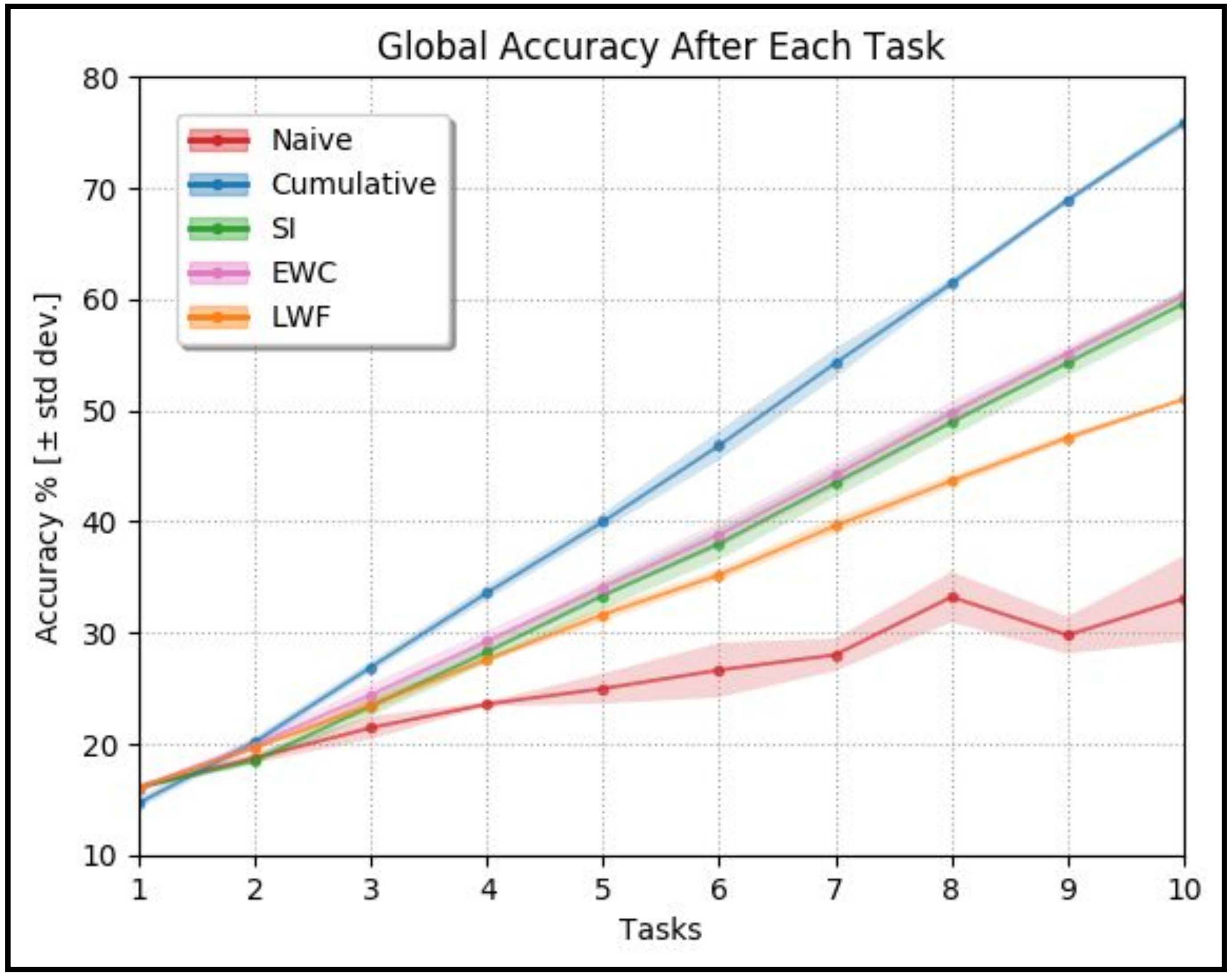
Evaluation setups [Chaudhry18]

- **Single-head** evaluation: **1** output space over labels for **all** tasks (+ difficult, + realistic)
- **Multi-head** predictions (restricted to when task labels provided).



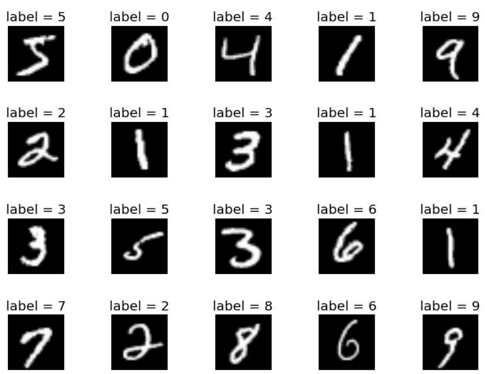
Reasonable baselines [Lomonaco & Maltoni'18]

- **Naïve** strategy (Lower bound): fine-tuning
- **Cumulative** strategy (Upper bound): from scratch each new Tr_i



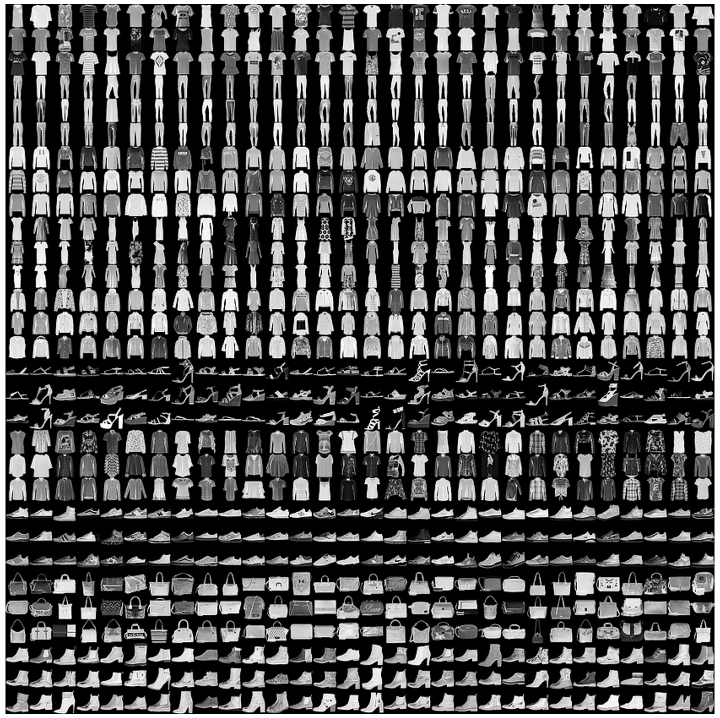
[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting: new metrics for Continual Learning. CL Workshop, NeurIPS 2018].

Evaluation Benchmarks



MNIST Dataset [LeCun & Cortes'98]

CORe50 Dataset [Lomonaco & Maltoni'18]



Fashion MNIST [Xiao'17]

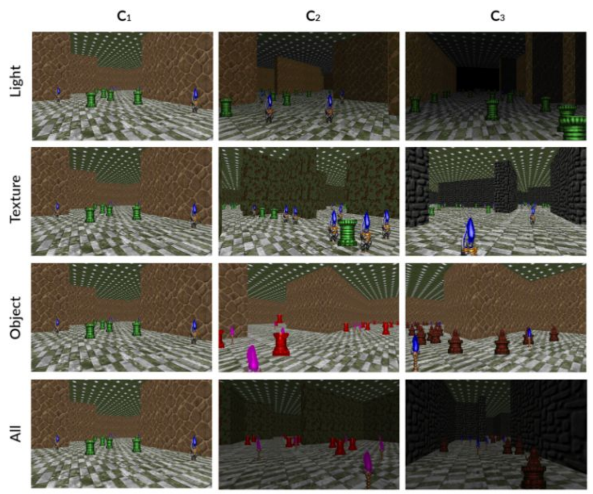
Benchmark	NI	NC	NIC	CL Use case example
Split MNIST		✓		Lesort et al. (2018a,b); He and Jaeger (2018)
Rotation MNIST	✓			Lopez-Paz and Ranzato (2017); Lesort et al. (2018a)
Permutation MNIST	✓			Goodfellow et al. (2013); Kirkpatrick et al. (2017); Fernando et al. (2017); Shin et al. (2017); Zenke et al. (2017); Lesort et al. (2018a); He and Jaeger (2018)
iCIFAR10/100		✓		Rebuffi et al. (2016); Maltoni and Lomonaco (2018); Kemker and Kanan (2018)
SVHN		✓		Kemker et al. (2017); Seff et al. (2017)
CUB200	✓			Lee et al. (2017)
CORe50	✓	✓	✓	Lomonaco and Maltoni (2017); Parisi et al. (2018); Maltoni and Lomonaco (2018)
Atari		✓		Rusu et al. (2016); Kirkpatrick et al. (2017); Schwarz et al. (2018)
LSUN		✓		Wu et al. (2018a)
ImageNet		✓		Rebuffi et al. (2016); Mallya and Lazebnik (2018)

Content Update Type: **NI**: New Instances, **NC**:New Concepts, **NIC**: New Instances & Concepts

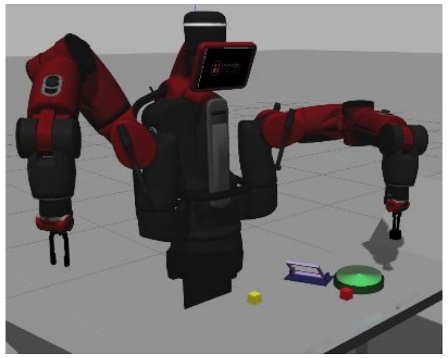
[Diaz-Rodriguez, Lesort, Lomonaco et al. 19]

Evaluation Environments

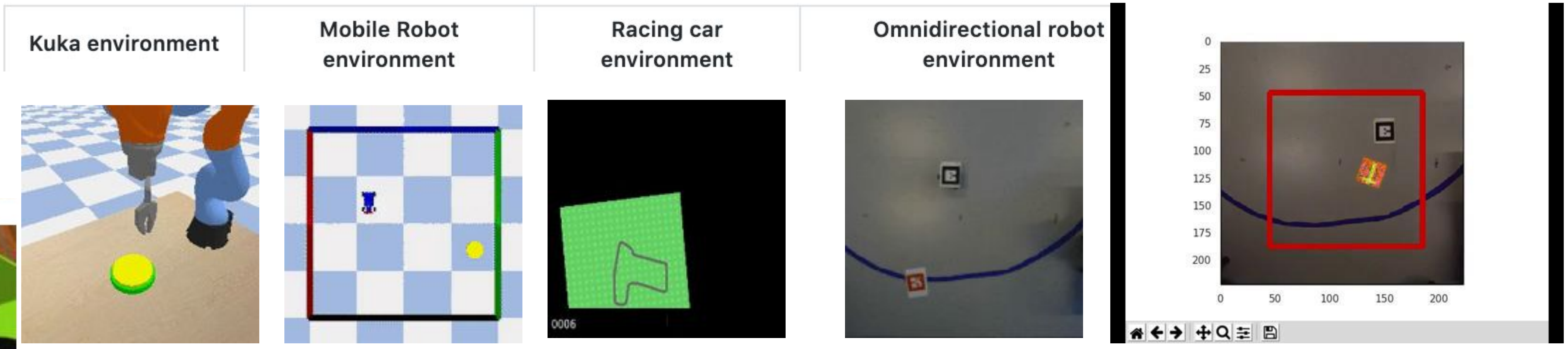
VizDoom



- **RL**: Atari (2D), VizDoom & Vizdoom (3D), DeepMind Lab, Malmo, OpenAI Gym, MuJoCo, Unity 3D, StarCraft II,
- **Robotics** manipulation skills



Colorful75



S-RL Toolbox: Environments, Datasets and Evaluation Metrics for State Representation Learning [Raffin'18] https://arxiv.org/abs/1809.09369

Continual
Learning
Metrics

Continual Learning Metrics

- **Accuracy (A)**: Given matrix $R \in \mathbb{R}^{N \times N}$, which contains in each entry $R_{i,j}$ the test classification accuracy of the model on task t_j after observing the last sample from task t_i [3], Accuracy metric is:

$$A = \frac{\sum_{i \geq j}^{N} R_{i,j}}{\frac{N(N+1)}{2}} \tag{2}$$

Table 1: Accuracy matrix R: elements accounted to compute A (white & cyan), BWT (cyan), and FWT (gray) Tr_i = training, Te_i= test tasks.

R	Te_1	Te_2	Te_3
Tr_1	R^*	R_{ij}	R_{ij}
Tr_2	R_{ij}	R^*	R_{ij}
Tr_3	R_{ij}	R_{ij}	R^*

Backward Transfer (BWT) measures the influence that learning a task has on the performance on previous tasks [3]:

$$BWT = \frac{\sum_{i=2}^{N} \sum_{j=1}^{i-1} (R_{i,j} - R_{j,j})}{\frac{N(N-1)}{2}} \tag{3}$$

- **Remembering (REM)**:

$$REM = 1 - |min(BWT, 0)| \tag{4}$$

is the originally negative BWT while the originally positive BWT, i.e., improvement over time, is:

- **Positive Backward Transfer (BWT^+)**:

$$BWT^+ = max(BWT, 0) \tag{5}$$

- **Forward Transfer (FWT)**: measures the influence that learning a task has on the performance of future tasks [3].

$$FWT = \frac{\sum_{i<j}^{N} R_{i,j}}{\frac{N(N-1)}{2}} \tag{6}$$

[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting: new metrics for Continual Learning. CL Workshop, NeurIPS 2018].

Continual Learning Metrics

$$A = \frac{\sum_{i \geq j}^{N} R_{i,j}}{\frac{N(N+1)}{2}} \tag{2}$$

where $R_{i,j}$ in Accuracy matrix $R \in \mathbb{R}^{N \times N}$ is test classification accuracy on task t_j after observing the last sample from task t_i.

R	Te_1	Te_2	Te_3
Tr_1	R^*	R_{ij}	R_{ij}
Tr_2	R_{ij}	R^*	R_{ij}
Tr_3	R_{ij}	R_{ij}	R^*

[4] [Lopez-Paz and Ranzato, 2017]

[5] Accuracy matrix R: elements accounted to compute A (white & cyan), BWT (cyan), and FWT (gray). $R^* = R_{ii}$, Tr_i = training, Te_i= test tasks.

[Lopez-Paz & Ranzato'17], [Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting: new metrics for Continual Learning. CL Workshop, NeurIPS 2018].

Continual Learning Metrics

- **Model size efficiency (MS)**: The memory size of model h_i quantified in terms of parameters θ at each task i, $Mem(\theta_i)$, should not grow too rapidly with respect to the size of the model that learned the first task, $Mem(\theta_1)$.

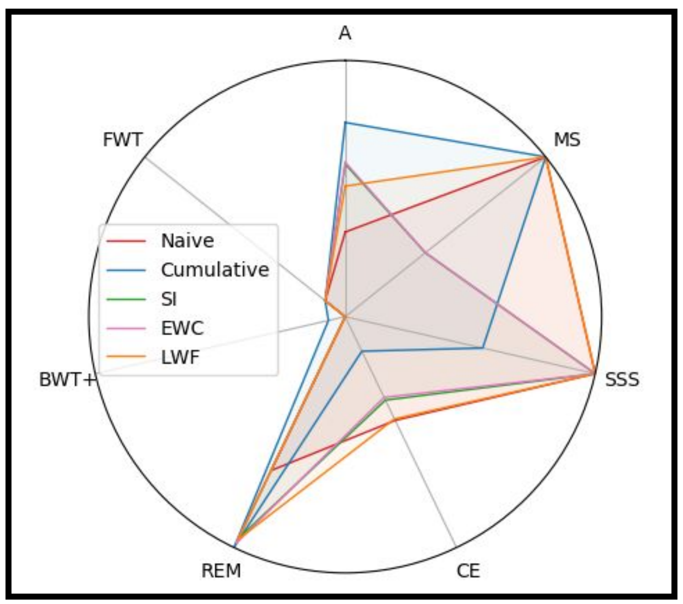
$$MS = min(1, \frac{\sum_{i=1}^{N} \frac{Mem(\theta_1)}{Mem(\theta_i)}}{N}) \tag{7}$$

- **Samples storage size efficiency (SSS)**: The memory occupation in bits by the samples storage memory M, $Mem(M)$, should be bounded by the memory occupation of the total nr of examples encountered at the end of the last task (D is the lifetime dataset associated to all distributions $\mathcal{D}$):

$$SSS = 1 - min(1, \frac{\sum_{i=1}^{N} \frac{Mem(M_i)}{Mem(D)}}{N}) \tag{8}$$

- **Computational efficiency (CE)**: it is bounded by the nr of operations for training set Tr_i. $Ops(Tr_i)$ is the number of (mul-adds) operations needed to learn Tr_i and $Ops\uparrow\downarrow(Tr_i)$ is the nr of operations required to do one forward and one backward (backprop) pass on Tr_i.

$$CE = min(1, \frac{\sum_{i=1}^{N} \frac{Ops\uparrow\downarrow(Tr_i)\cdot\varepsilon}{1+Ops(Tr_i)}}{N}) \tag{9}$$



[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting: new metrics for Continual Learning. CL Workshop, NeurIPS 2018].

Continual Learning Metrics

- **Model size efficiency (MS)**: The memory size of model h_i quantified in terms of parameters θ at each task i, $Mem(\theta_i)$, should not grow too rapidly with respect to the size of the model that learned the first task, $Mem(\theta_1)$.

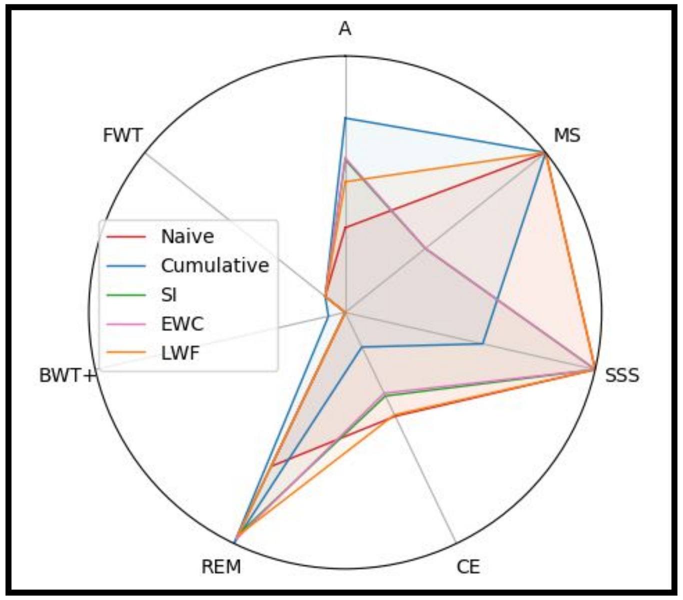
$$MS = min(1, \frac{\sum_{i=1}^{N} \frac{Mem(\theta_1)}{Mem(\theta_i)}}{N}) \tag{7}$$

- **Samples storage size efficiency (SSS)**: The memory occupation in bits by the samples storage memory M, $Mem(M)$, should be bounded by the memory occupation of the total nr of examples encountered at the end of the last task (D is the lifetime dataset associated to all distributions $\mathcal{D}$):

$$SSS = 1 - min(1, \frac{\sum_{i=1}^{N} \frac{Mem(M_i)}{Mem(D)}}{N}) \tag{8}$$

- **Computational efficiency (CE)**: it is bounded by the nr of operations for training set Tr_i. $Ops(Tr_i)$ is the number of (mul-adds) operations needed to learn Tr_i and $Ops\uparrow\downarrow(Tr_i)$ is the nr of operations required to do one forward and one backward (backprop) pass on Tr_i).

$$CE = min(1, \frac{\sum_{i=1}^{N} \frac{Ops\uparrow\downarrow(Tr_i)\cdot\varepsilon}{1+Ops(Tr_i)}}{N}) \tag{9}$$



[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting: new metrics for Continual Learning. CL Workshop, NeurIPS 2018].

Continual Learning Metrics

- **Model size efficiency (MS)**: The memory size of model h_i quantified in terms of parameters θ at each task i, $Mem(\theta_i)$, should not grow too rapidly with respect to the size of the model that learned the first task, $Mem(\theta_1)$.

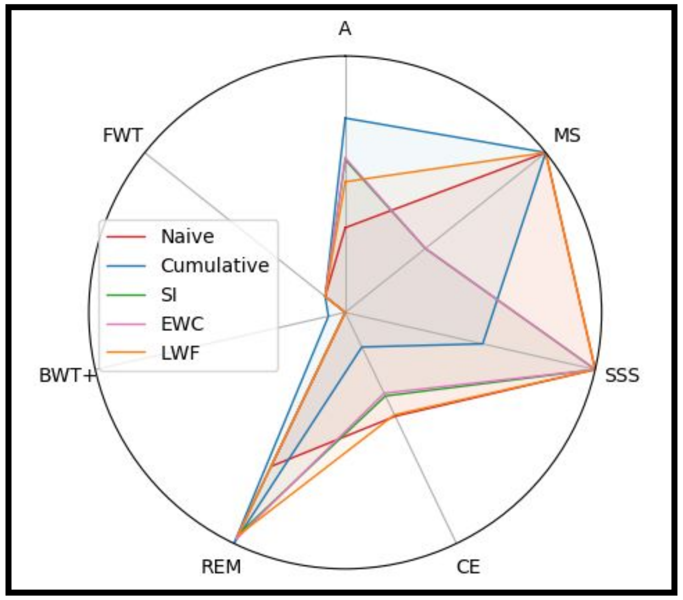
$$MS = min(1, \frac{\sum_{i=1}^{N} \frac{Mem(\theta_1)}{Mem(\theta_i)}}{N}) \tag{7}$$

- **Samples storage size efficiency (SSS)**: The memory occupation in bits by the samples storage memory M, $Mem(M)$, should be bounded by the memory occupation of the total nr of examples encountered at the end of the last task (D is the lifetime dataset associated to all distributions $\mathcal{D}$):

$$SSS = 1 - min(1, \frac{\sum_{i=1}^{N} \frac{Mem(M_i)}{Mem(D)}}{N}) \tag{8}$$

- **Computational efficiency (CE)**: it is bounded by the nr of operations for training set Tr_i. $Ops(Tr_i)$ is the number of (mul-adds) operations needed to learn Tr_i and $Ops\uparrow\downarrow(Tr_i)$ is the nr of operations required to do one forward and one backward (backprop) pass on Tr_i.

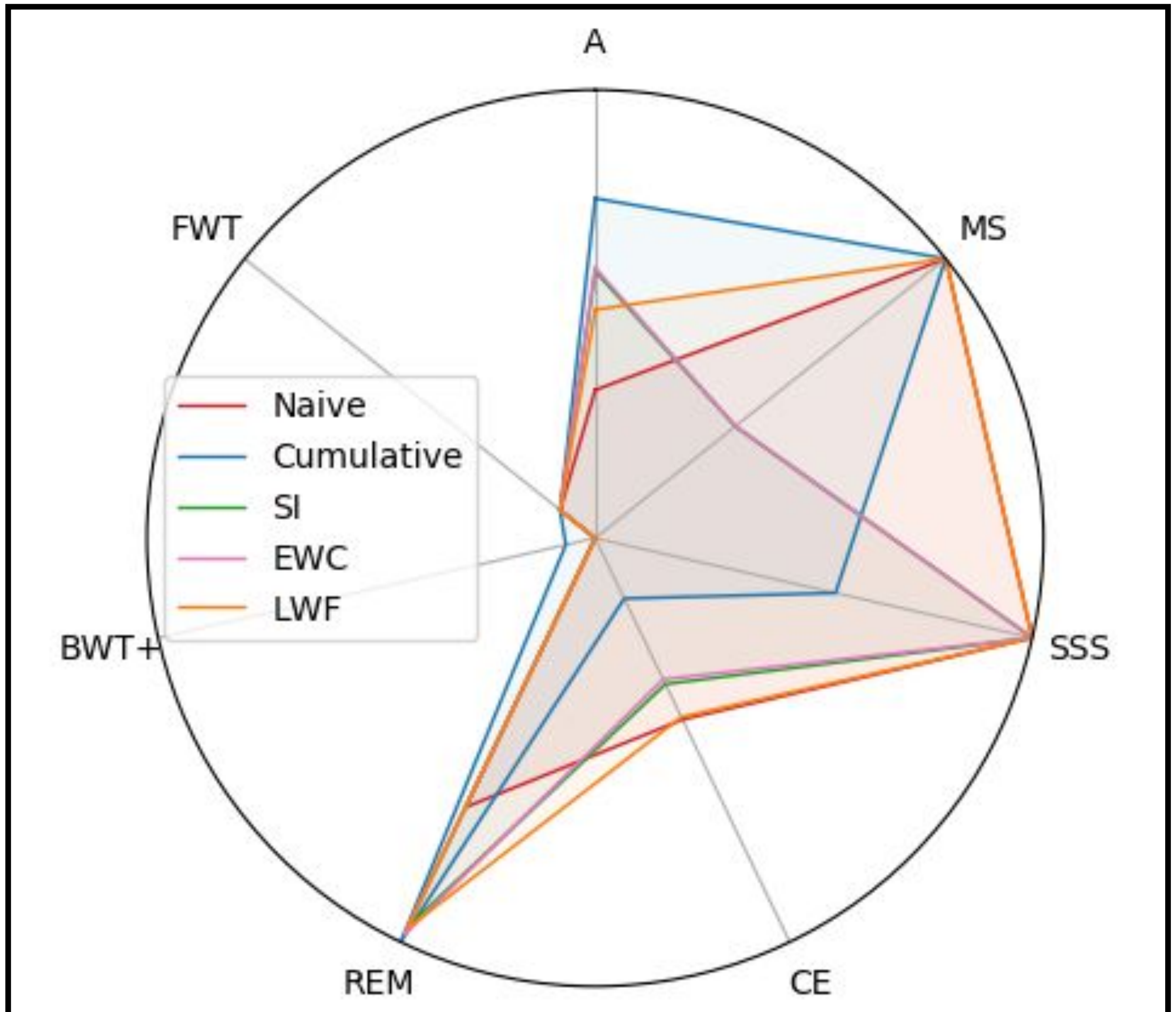
$$CE = min(1, \frac{\sum_{i=1}^{N} \frac{Ops\uparrow\downarrow(Tr_i)\cdot\varepsilon}{1+Ops(Tr_i)}}{N}) \tag{9}$$

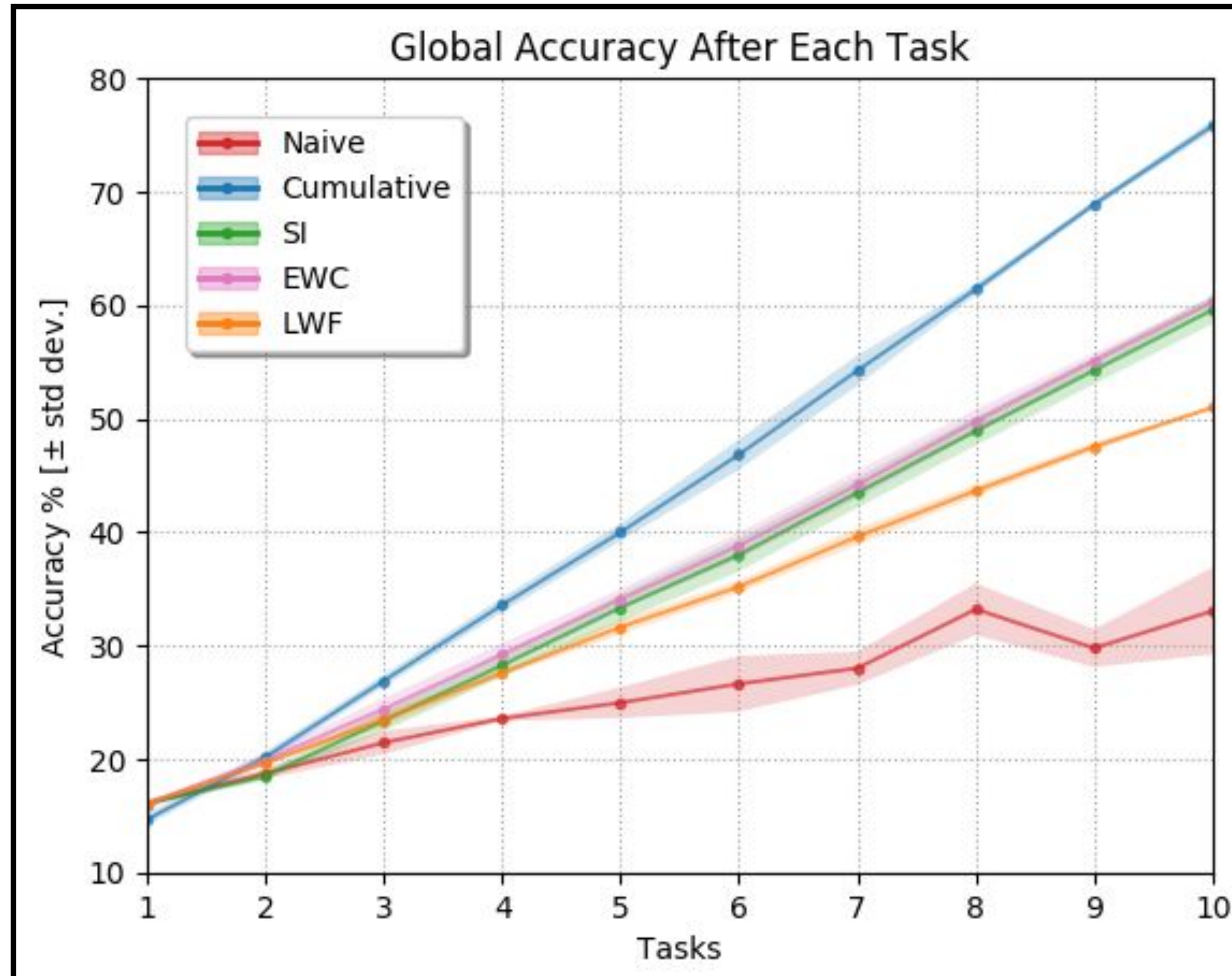


$$CL_{score} = \sum_{i=1}^{\#C} w_i c_i$$

[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting: new metrics for Continual Learning. CL Workshop, NeurIPS 2018].

Continual Learning Metrics





[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting: new metrics for Continual Learning. CL Workshop, NeurIPS 2018].

CL Constraints, Desiderata, Recommendations

CL Constraints

Constraint 1 *For every step in time, the number of current examples contained into the memory is lower than the total number of previously seen examples*[4]*:* $\forall i \in [1, ..., n], |M_i| \ll \left|\bigcup_{i=1}^{i-1} Tr_i\right|$

Constraint 2 *Memory and computation for each iteration step i are bounded. Given two functions* $ops()$ *and* $mem()$ *that compute the number of operations and memory occupation required by* A_i^{CL}*, two reasonably small values* max_ops *and* max_mem *should exist, such that, for each* i*,* $ops(A_i^{CL}) < max\_ops$ *and* $mem(h_{i-1}, M_{i-1}) < max\_mem$*.*

Relaxations over CL constraints

Relaxation 1 *Memory relaxation: Removes the fixed memory bound over* $ops()$ *and* $mem()$.

Relaxation 2 *Computation relaxation: Removes the fixed computational bound* $ops(h_i) < max\_ops$.

CL Recommendations

CL Recommendations

Recommendation 1 *On-line capabilities: CL algorithms should not assume the number of total tasks is given beforehand.*

Recommendation 2 *Learning complexity: We recommend keeping the learning model complexity below an upper bound of a linear growth in terms of the number of parameter growth when performing architectural dynamic changes.*

Recommendation 3 *Memory limitation: In order for realistic CL industrial systems to be practical, they should not assume unlimited memory resources. This is a feature often not quantified in some CL architectural strategies.*

Recommendation 4 *Reporting metrics: We recommend reporting as many metrics as possible and at least final performance, forward and backward (learning) transfer, the model's remembering capacity, model memory size, samples storage size, computational efficiency, CL score and stability metrics as described in Section 5.2.*

Recommendation 5 *Offline baselines: we recommend the usage of publicly available baselines for metrics computation and fair assessment for reproducibility purposes.*

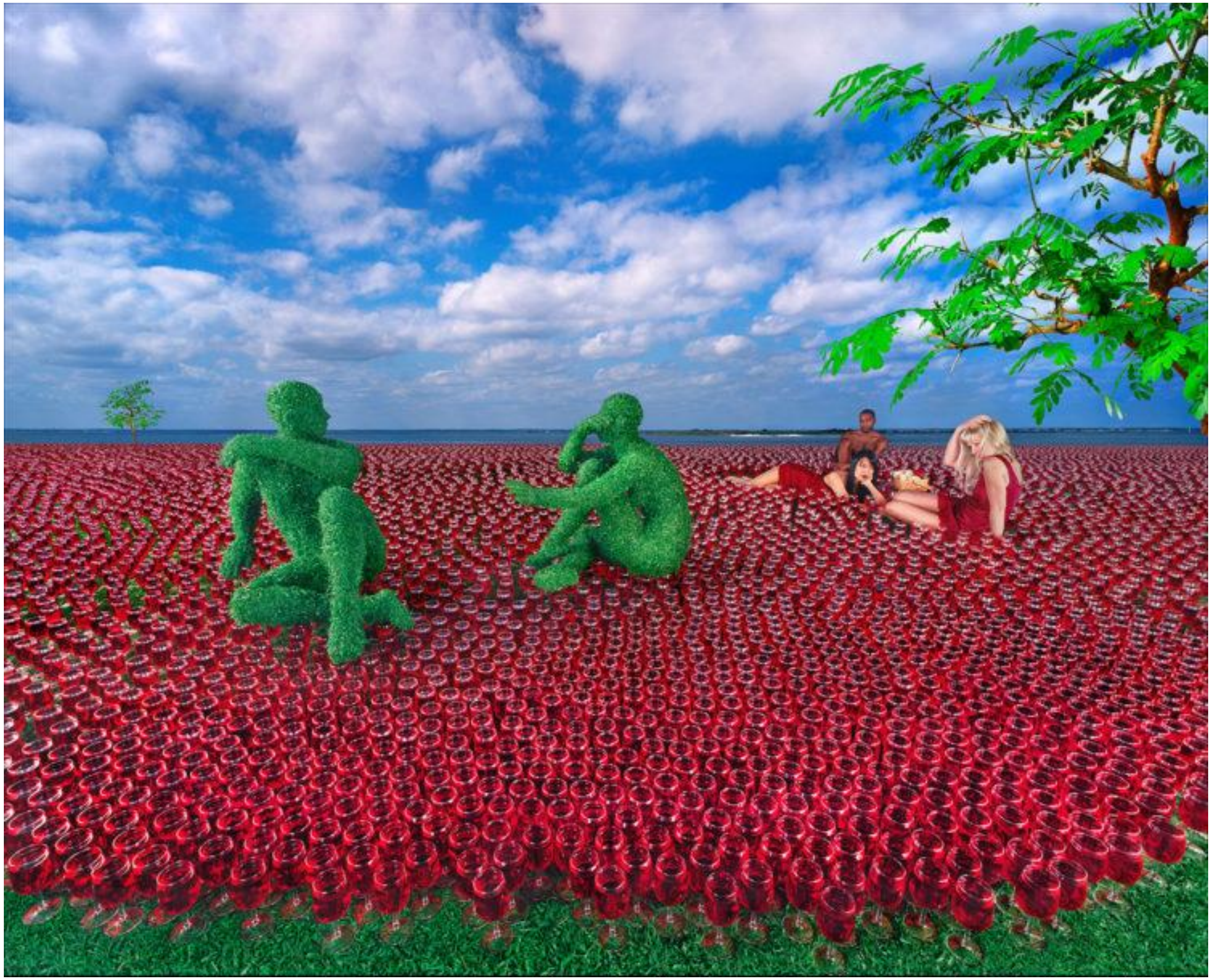
Recommendation 6 *Distributional shifts: We recommend to formally describe how is handling the distributional shifts, not only when tasks change, but also among batches where data points conform to different distributions.*

Recommendation 7 *Benchmarks: We recommend the use complex datasets with realistic and higher resolution scales than MNIST and CIFAR100; the use of the former is seen as a limiting factor and not a realistic robustness assessment method for CL (see Section 5.1).*

Recommendation 8 *Report precisely and clearly how an approach learns and the assumptions it implies as described in the framework (Section 3).*

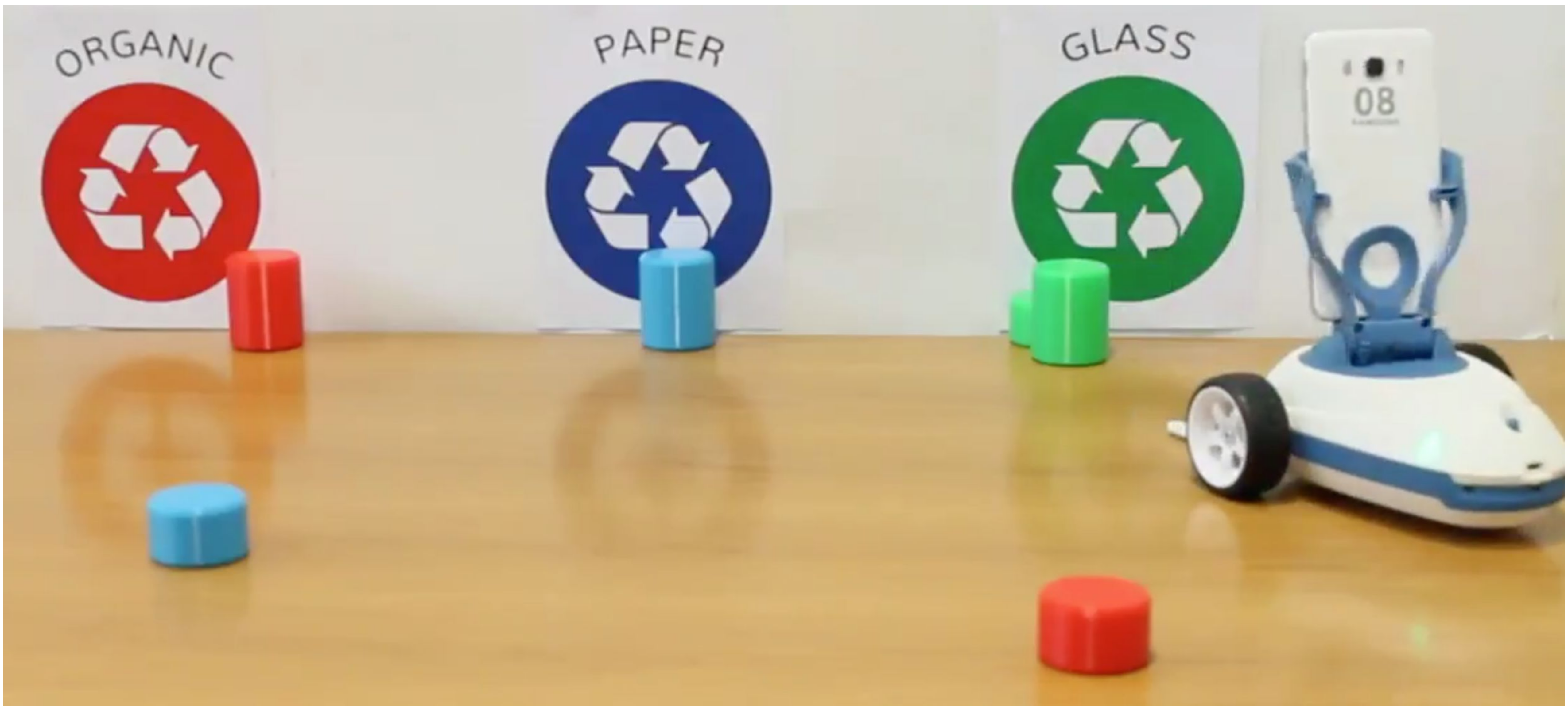
[Diaz-Rodriguez, Lesort, Lomonaco et al. 19]

Future Perspectives



Future perspectives

- More & More complex tasks



Future perspectives

- More & More complex tasks
- Multimodality

IMAGES **CAPTIONS**

A man is sitting on a curb with a bike.

A man holding a child in a park with a kite.

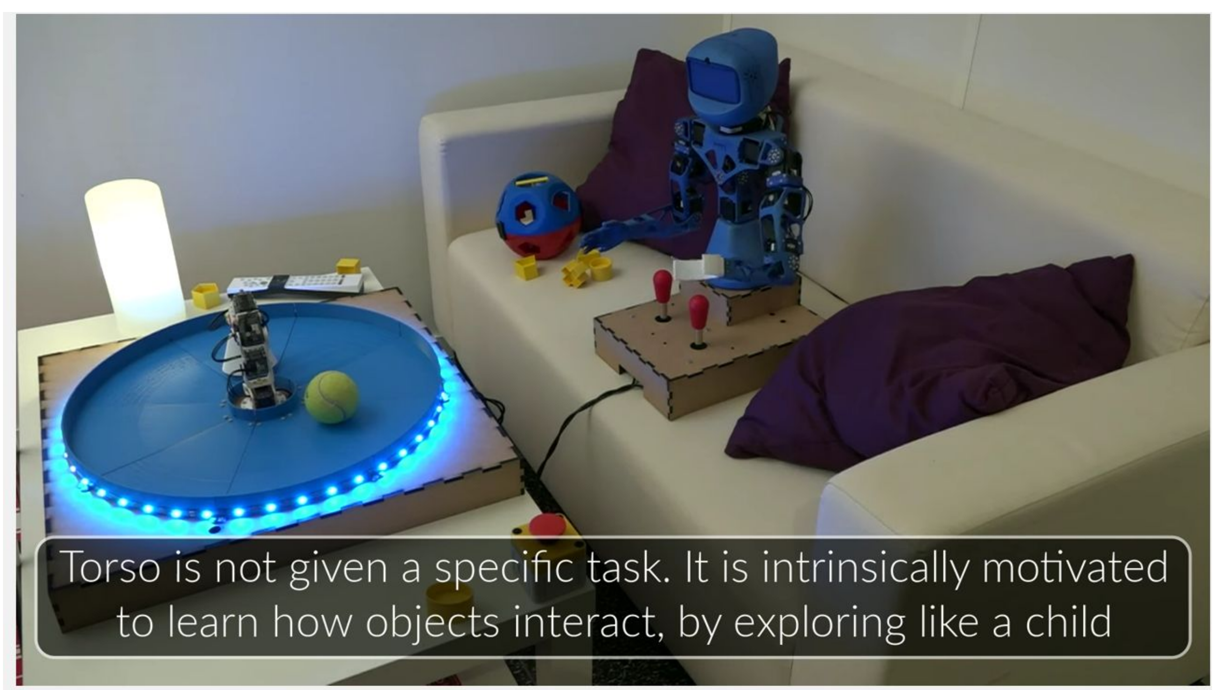
EGOSHOTS Dataset processed with pretrained network on MSCOCO “Captioning Images with Diverse Objects”, CVPR 2017. [P. Agarwal]

Future perspectives

- Wider, + comprehensive metrics:
  - no more benchmarking on MNIST!
- + Ablation studies
  - Stating relaxations, constraints

Future perspectives

- Wider, + comprehensive metrics:
  - no more benchmarking on MNIST!
- + Ablation studies
  - Stating relaxations, constraints
- + Efficient RL
- Compositionality



Intrinsically Motivated Multi-Task Reinforcement Learning

FEW REFERENCES

- Lesort et. al. *Unsupervised state representation learning with robotic priors: a robustness benchmark*, IJCNN19, DEMO video
- Lesort et al. *State Representation learning for control: an Overview*, 2018, Neural Networks.
- Doncieux et al. *Open-Ended Learning: A Conceptual Framework Based on Representational Redescription*. Frontiers in Neurorobotics, 2018
- Díaz-Rodríguez et al., *Don't forget, there is more than forgetting: new metrics for Continual Learning*, NeurIPS18 Continual Learning workshop.
- Díaz-Rodríguez, Lesort, Lomonaco et al, *Continual Learning and Robotics (submitted). 2019.*
- Traoré et al. *Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer, MTLRL*@ICML19 workshop.
- Raffin et al. *S-RL Toolbox: Environments, Datasets and Evaluation Metrics for State Representation Learning*. NeurIPS18 Deep RL workshop.
- Raffin et al. *Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics*, *SPIRL*@ICLR19 workshop.

Acknowledgment: Sandy Skoglund Photography and Camera Centro Italiano Fotografia, Torino.

Thank you!

COME SEE OUR POSTER: *Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer.* **[Traoré'19]**

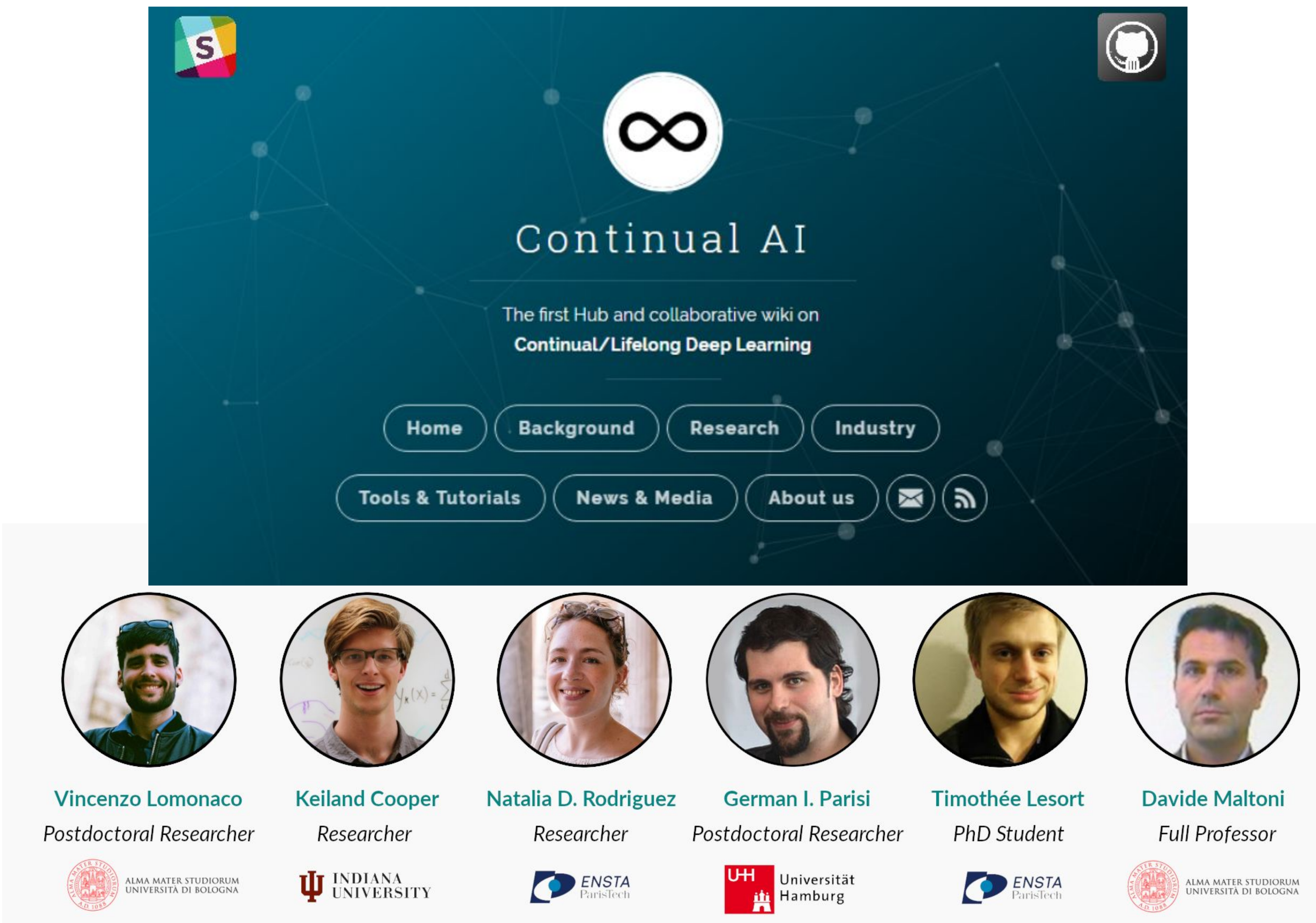
https://nataliadiaz.github.io/

@NataliaDiazRodr natalia.diaz@ensta-paristech.fr

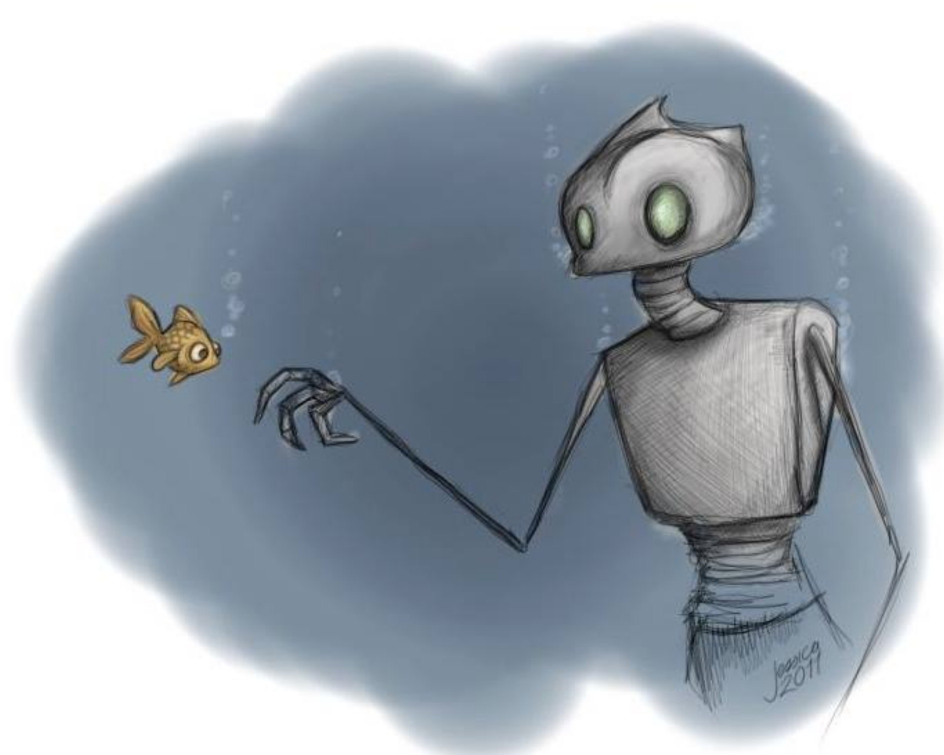
Special thanks to collaborators: **Timothée Lesort, Vincenzo Lomonaco, Kalifou René Traoré, Hugo Caselles-Dupré, Te Sun, Antonin Raffin, Ashley Hill, David Filliat, Pieter Abbeel, INRIA Flowers team et al.**

Join! https://www.continualai.org/

Slack channel: https://continualai.herokuapp.com/



APPENDIX



Catastrophic Forgetting in DNNs [Pfulf & Gepperth'19]

Different Image classification schemes for 10 class datasets

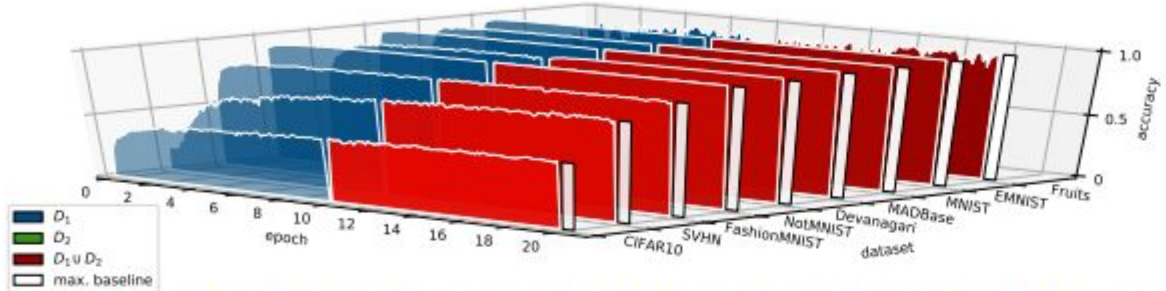


Figure 2: Best FC experiments for DP10-10. The blue surfaces (epochs 0-10) represent the accuracy on D_1, the green (covered here) and red surfaces the accuracy on D_2 and $D_1 \cup D_2$ during re-training (epochs 10-20). The white bars indicate baseline performance. See also Appendix B for 2D plots.

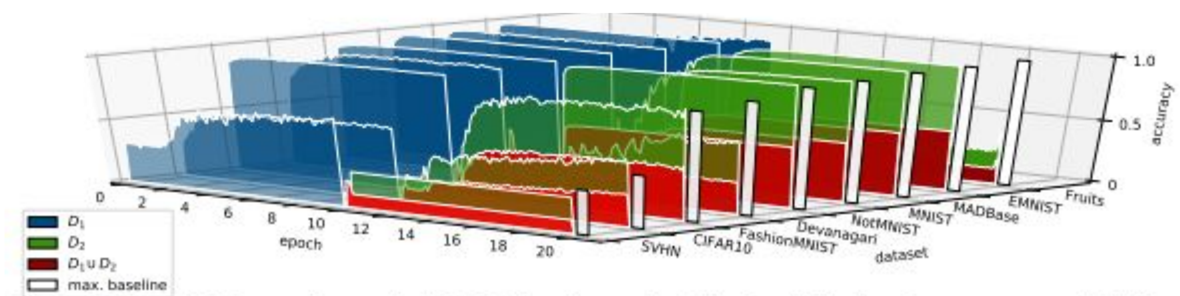


Figure 3: Best D-FC experiments for SLT D5-5, to be read as Fig. 2 and showing the occurrence of CF. See also Appendix B for 2D plots.

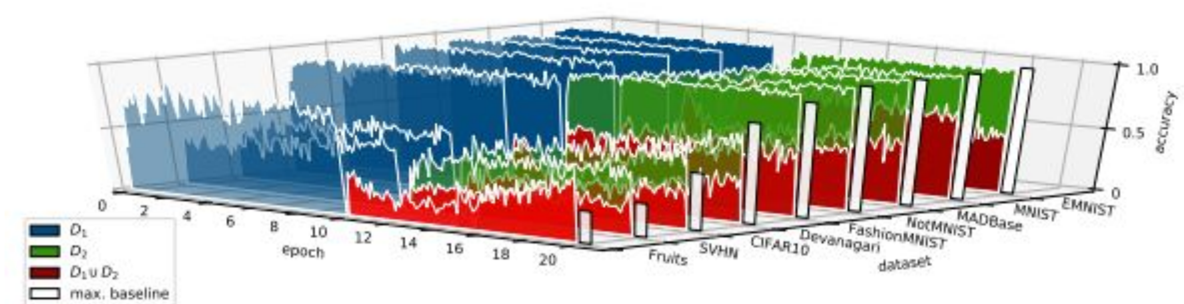


Figure 5: Best EWC experiments for SLT D5-5d constructed from all datasets, to be read as Fig. 2. We observe that CF happens for all datasets. See also Appendix B for 2D plots.

Recommendations [Pfülb & Gepperth'19]

- No dataset immune to CF (1)
  => Permutation-based datasets shouldn't be used to investigate CF (2)
- EWC: ineffective (for complex problems, D5-5 tasks) (3)
  - Knowledge Pre-conditions

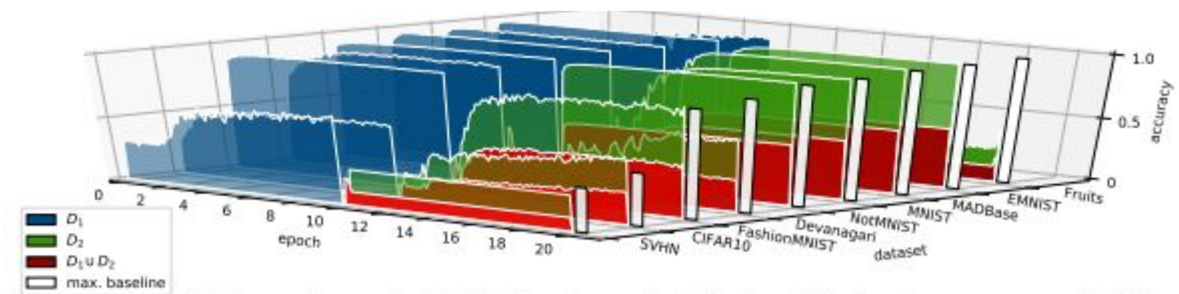


Figure 3: Best D-FC experiments for SLT D5-5, to be read as Fig. 2 and showing the occurrence of CF. See also Appendix B for 2D plots.

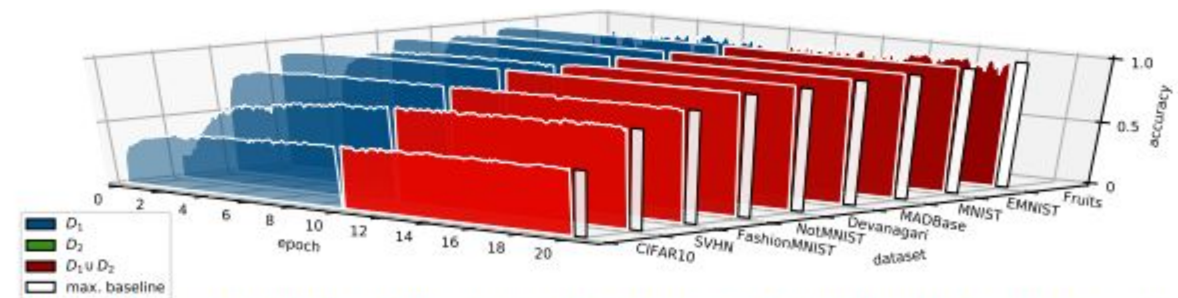


Figure 2: Best FC experiments for DP10-10. The blue surfaces (epochs 0-10) represent the accuracy on D_1, the green (covered here) and red surfaces the accuracy on D_2 and $D_1 \cup D_2$ during re-training (epochs 10-20). The white bars indicate baseline performance. See also Appendix B for 2D plots.

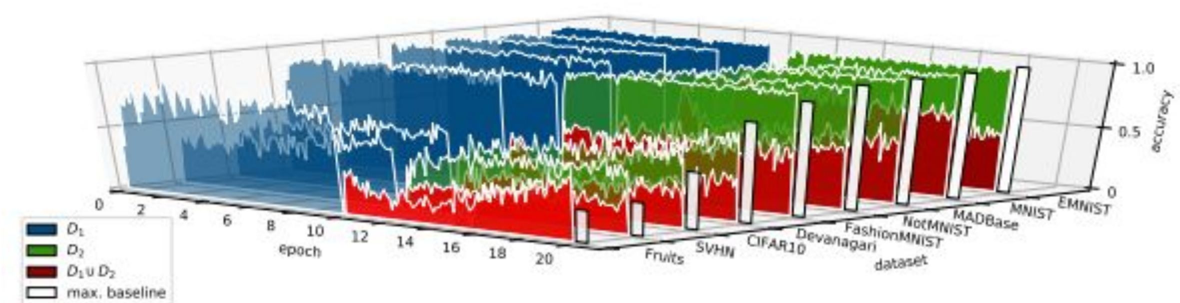
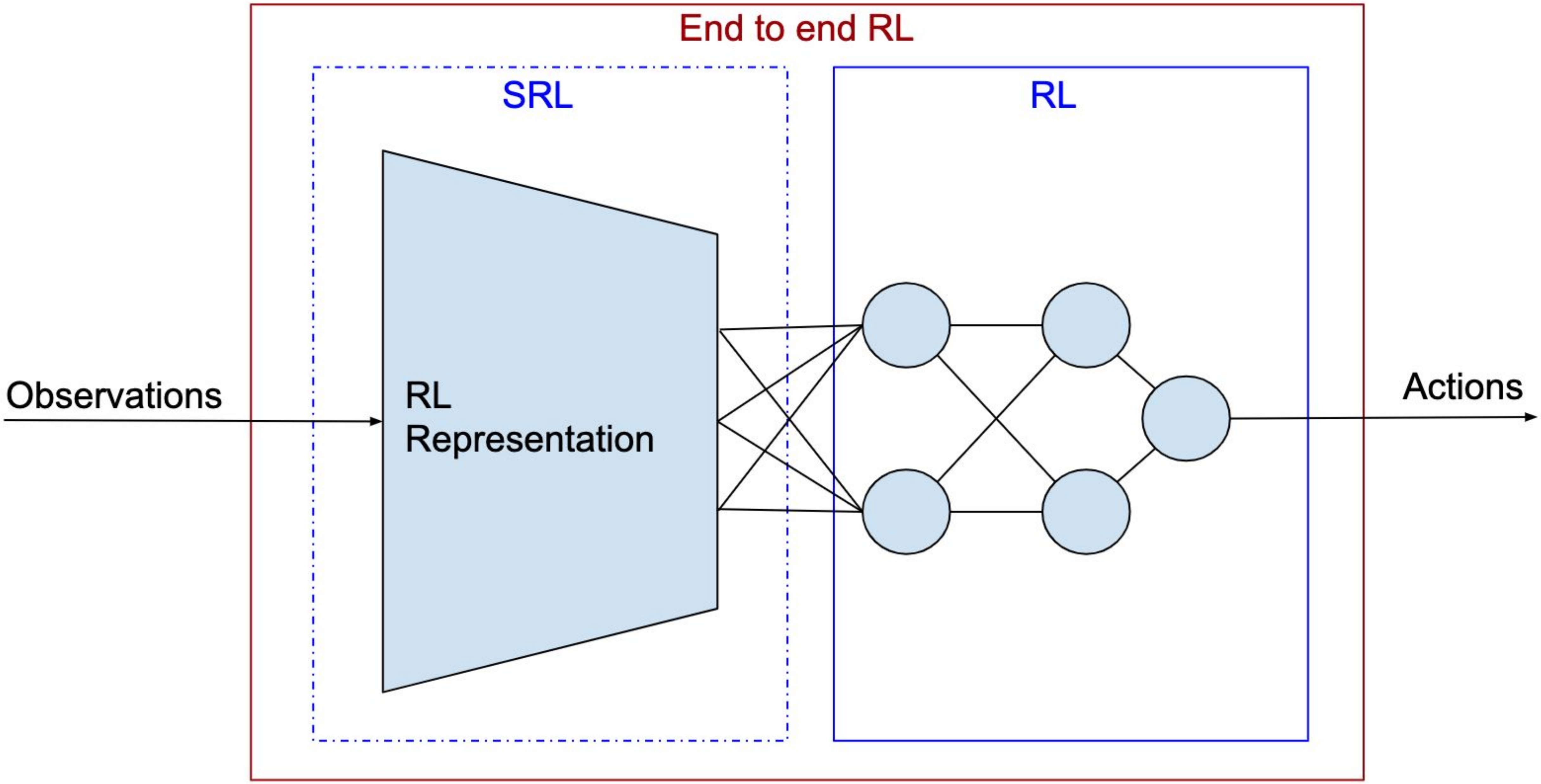


Figure 5: Best EWC experiments for SLT D5-5d constructed from all datasets, to be read as Fig. 2. We observe that CF happens for all datasets. See also Appendix B for 2D plots.

State Representation Learning (SRL) in RL context



Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics [Raffin et al. 2018]

What can SRL be used for?
E.g. computer vision for robotics control

Our goal is to learn a relevant state from images, actions and rewards. We train a neural network in unsupervised manner using prior knowledge about the physical world in form of robotic priors.

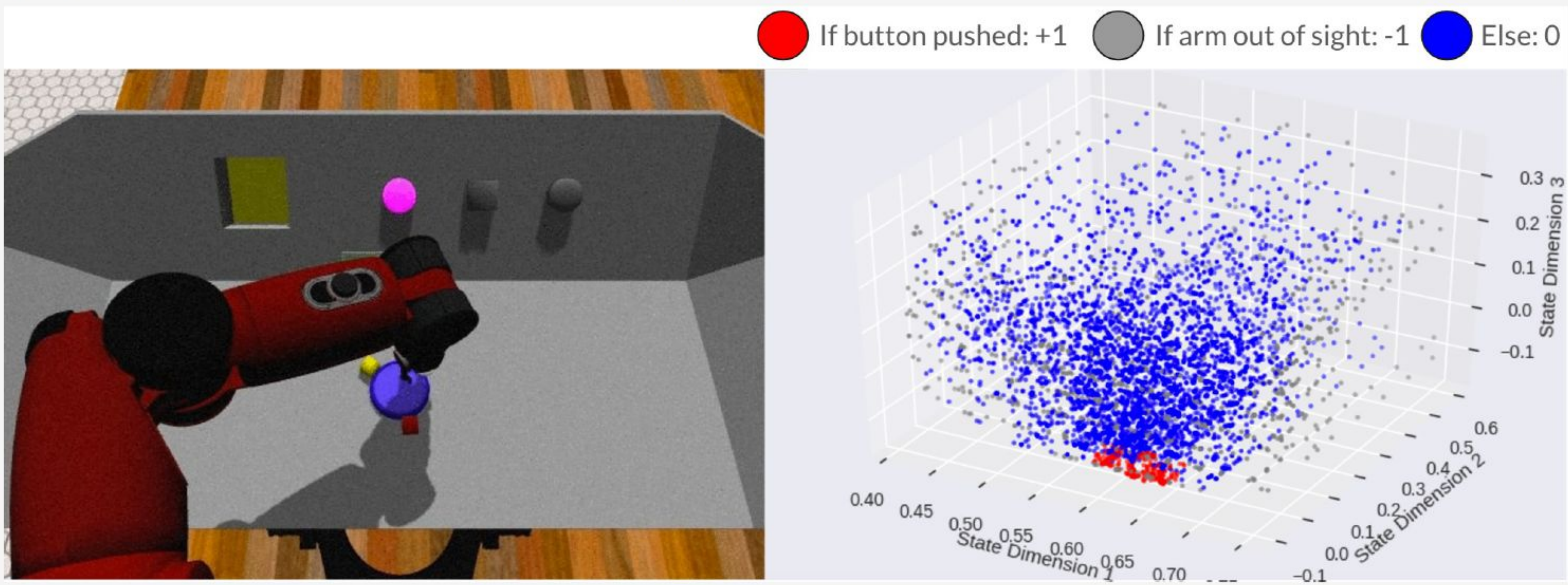


Figure 1: Left: Baxter's camera view for Static-Button-Distractors dataset 2. Right: Baxter's left hand ground truth position and its coded reward

Learning to grow layers in NLP

In NLP [Li19]: After the structure is searched, the model parameters are then estimated.
1) explicitly continual structure learning makes more effective use of parameters among tasks, which leads to better performance and sensible structures for different tasks;
2) separating structure and parameter learning significantly reduced catastrophic forgetting as compared to other baseline methods with similar model complexities.

Evaluation Benchmarks

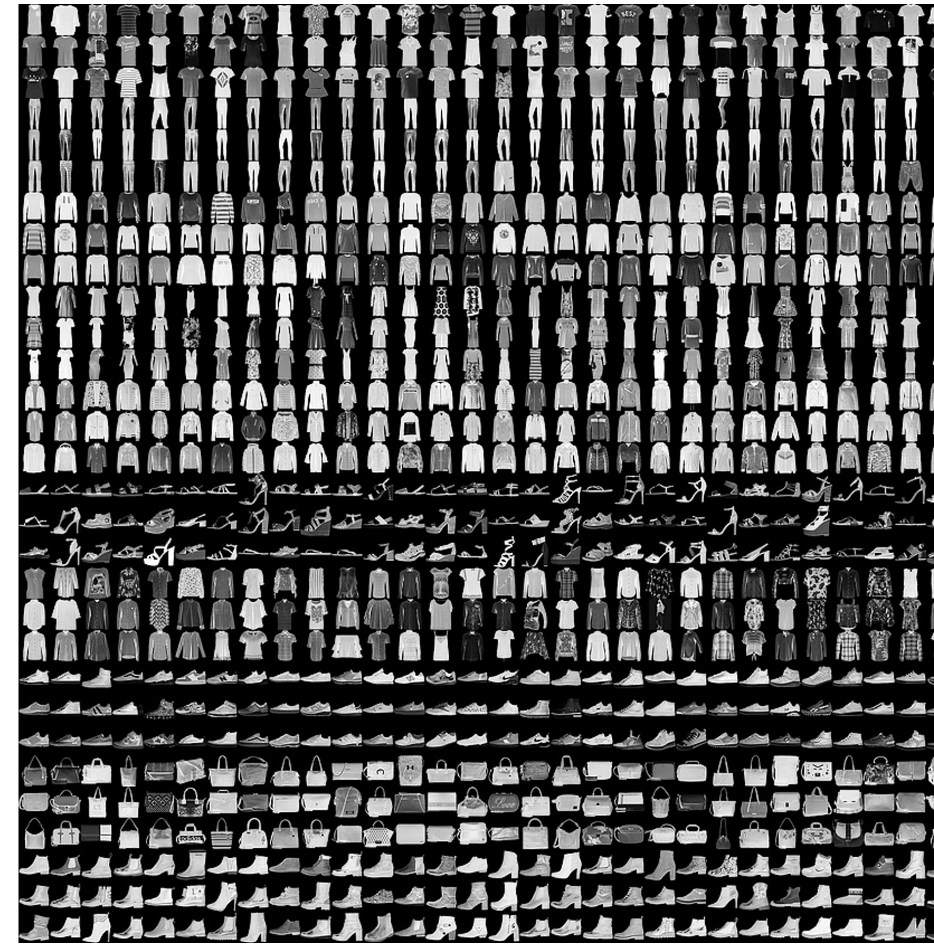
Table 1: Overview of each dataset's detailed properties. Image dimensions are given as width × height × channels. Concerning data imbalance, the largest percentual difference in sample count between any two classes is given for training and test data, a value of 0 indicating a perfectly balanced dataset.

Dataset \ **Properties**	image size	number of elements train	number of elements test	class balance (%) train	class balance (%) test
CIFAR10	32 × 32 ×3	50.000	10.000	0	0
Devanagari	32 × 32 ×1	18.000	2.000	0.3	2.7
EMNIST	28 × 28 ×1	345.035	57.918	2.0	2.0
FashionMNIST	28 × 28 ×1	60.000	10.000	0	0
Fruits 360	100 ×100 ×3	6.148	2.052	4.0	4.2
MADBase	28 × 28 ×1	60.000	10.000	0	0
MNIST	28 × 28 ×1	55.000	10.000	2.2	2.4
NotMNIST	28 × 28 ×1	529.114	18.724	∼0	∼0
SVHN	32 × 32 ×3	73.257	26.032	12.6	13.5

MNIST (LeCun et al., 1998) is the common benchmark for computer vision systems and classification problems. It consist of gray scale images of handwritten digits (0-9).
EMNIST (Cohen et al., 2017) is an extended version of MNIST with additional classes of handwritten letters. There are different variations of this dataset: we extract the ten best-represented classes from the *By_Class* variation containing 62 classes.
Fruits 360 (Murean & Oltean, 2017) is a dataset comprising fruit color images from different rotation angles spread over 75 classes, from which we extract the ten best-represented ones.
Devanagari (Acharya et al., 2015) contains gray scale images of Devanagari handwritten letters. From the 46 character classes (1.700 images per class) we extract 10 random classes.
FashionMNIST (Xiao et al., 2017) consists of images of clothes in 10 classes and is structured like the MNIST dataset. We use this dataset for our investigations because it is a "more challenging classification task than the simple MNIST digits data (Xiao et al., 2017)".
SVHN (Netzer et al., 2011) is a 10-class dataset based on photos of house numbers (0-9). We use the cropped digit format, where the number is centered in the color image.
CIFAR10 (Krizhevsky, 2009) contains color images of real-world objects e.g, dogs, airplanes etc.
NotMNIST (Bulatov Yaroslav) contains grayscale images of the 10 letter classes from "A" to "J", taken from different publicly available fonts.
MADBase (Abdelazeem Sherif & El-Sherif Ezzat) is a modified version of the "Arabic Digits dataBase", containing grayscale images of handwritten digits written by 700 different persons.

[Pfülb and Gepperth'19]

Evaluation Benchmarks

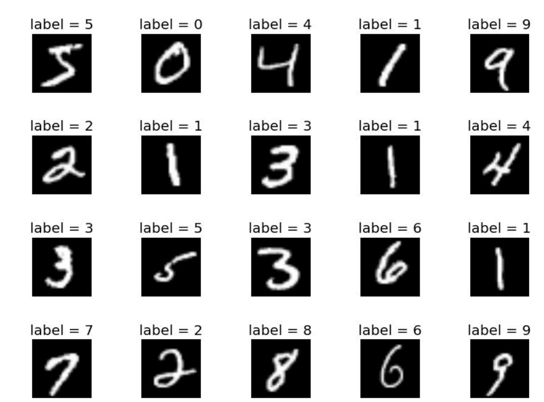


Fashion MNIST [Xiao'17]

Table 1: Comparison of datasets (with temporal coherent sessions) for continuous object recognition.

Dataset	Cat.	Obj.	Sess.	Frames per sess.	Format	Acquisition setting	Outdoor sessions
NORB [12]	5	25	20	20*	grayscale	turntable	no
COIL-100 [18]	-	100	20	54*	RGB	turntable	no
iLab-20M [2]	15	704	-	-	RGB	turntable	no
RGB-D [24]	51	300	-	-	RGB-D	turntable	no
BigBIRD [25]	-	100	-	-	RGB-D	turntable	no
ALOI [8]	-	1000	-	-	RGB	turntable	no
BigBrother [6]	-	7	54	~20	RGB	wall cameras	no
iCubWorld28 [19]	7	28	4	~150	RGB	hand hold	no
iCubWorld-Transf [20]	15	150	6	~150	RGB	hand hold	no
CORe50	**10**	**50**	**11**	**~300**	**RGB-D**	**hand hold**	**yes (3)**

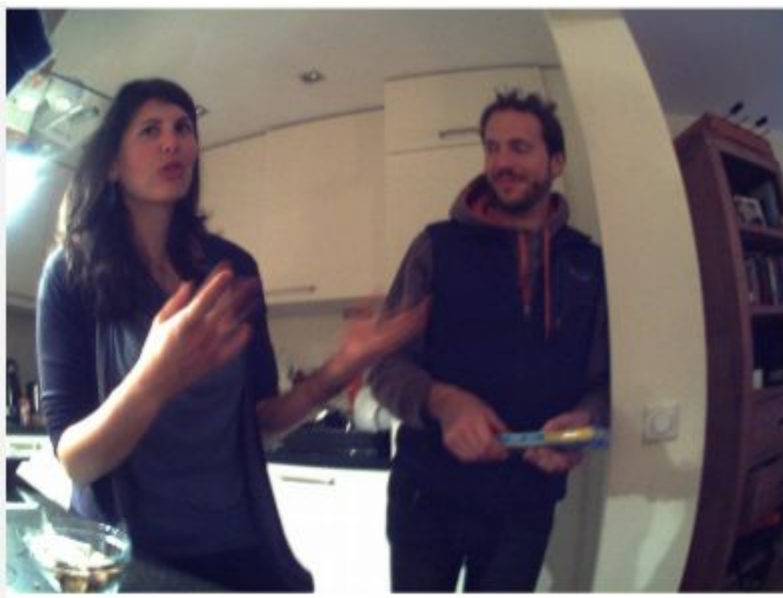
* Temporal coherent training/test sessions for NORB and COIL-100 have been defined in [16] and [17].



MNIST Dataset [LeCun & Cortes'98]

Future perspectives

E.g. Domain transfer in other domains: Captioning



A woman taking a picture of a lady in a mirror

a young lady posing in a bathroom with a lady holding a phone

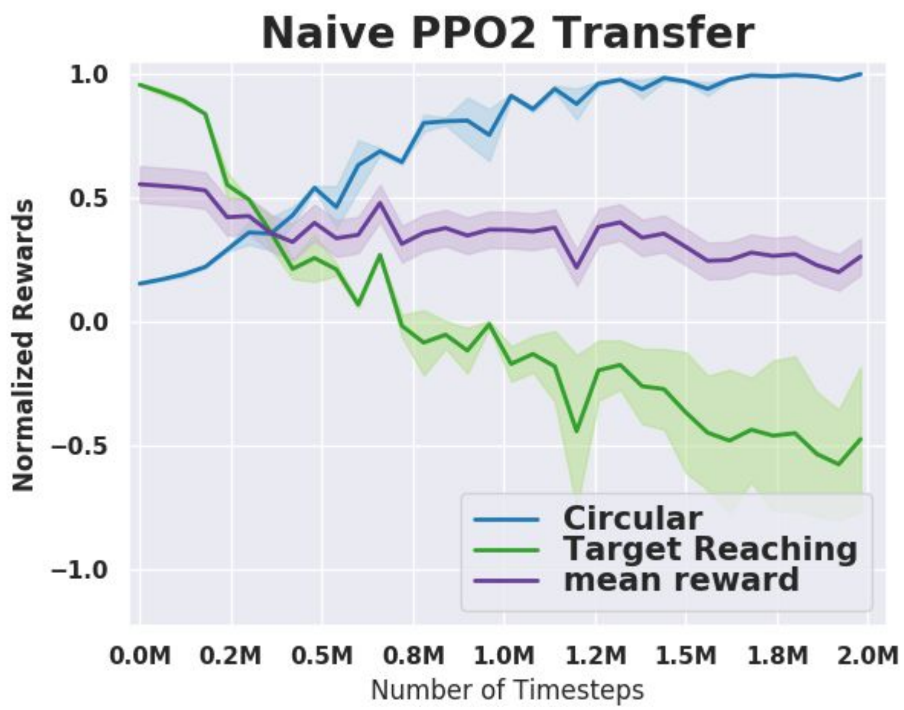
A man in a black shirt and a white shirt and a black and white street sign.

EGOSHOTS Dataset processed with pretrained network on MSCOCO “Captioning Images with Diverse Object”, CVPR 2017. (P. Agarwal)

Result

For a quick evaluation, the tests are based on **3 random seeds** with a evaluation interval length of **4 or 5 episodes**

Purple line stands for the mean rewards of two tasks, with std equals to the mean of stds of the two tasks as well (suppose the evaluations are independent)



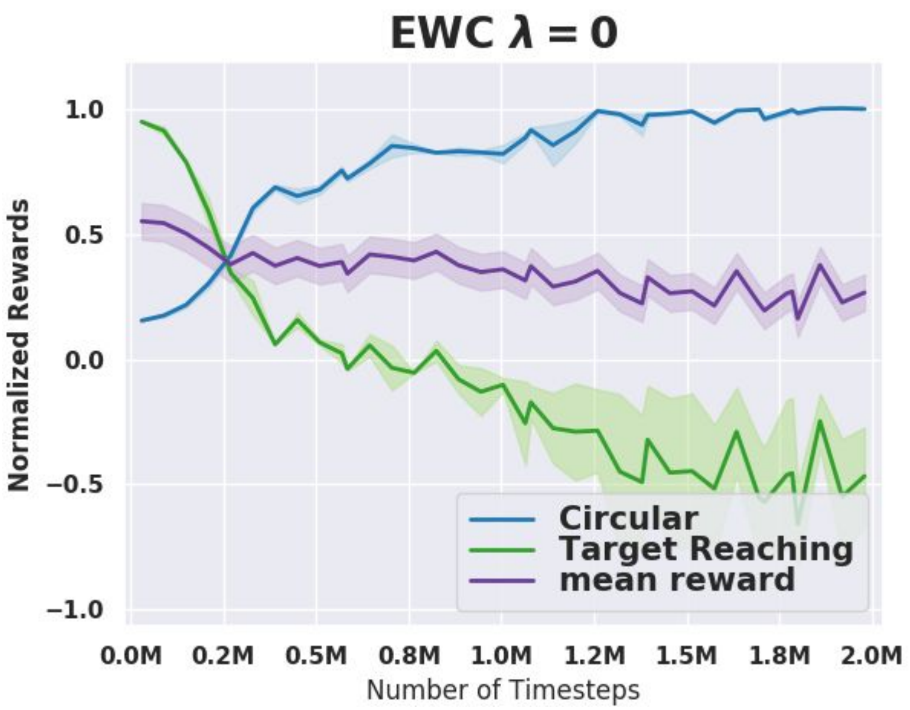


Fig 1: Implementation validation of EWC when lambda is zero (the result should be very similar to a simple continual transfer from PPO2)

Essential Tools to face CL challenges in Robotics

- SRL: allows to build strong representations of the world since agents should be able to understand their surroundings, and extract general concepts from sensory inputs of complex scenes. An agent better sees a chair as an object, not as a bunch of pixels together in an image.
- Helps scaling end-to-end RL for robotics control in terms of sample efficiency.
- Against end-to-end learning
    - helps learn a compact, efficient and relevant representation of states.
    - speeds up policy learning
    - reduce the number of samples needed
    - easier to interpret representations
- CL allows to learn such representation without forgetting in settings where the distribution of data change through time and is needed for agents that learn in the real-world and are required to adapt to changes. Combining CL and SRL would then allow to create strong representation robust to catastrophic forgetting.
- RL:
    - To learn robot controllers,
    - Can take advantage of SRL to learn faster and to produce more robust policies.

[Lesort'18, State representation learning for control: An overview. Neural Networks]

SRL for Robotics: Challenges

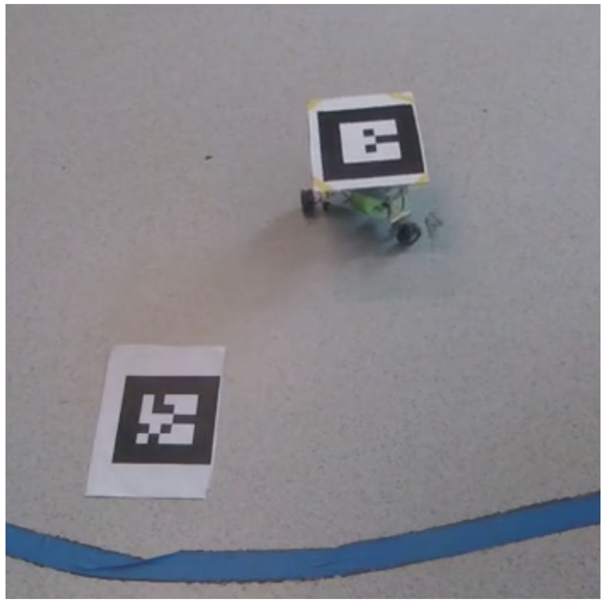
Vision goal-based tasks (mobile, visible goal to reach -vs open-ended tasks s.t. *walking*-)

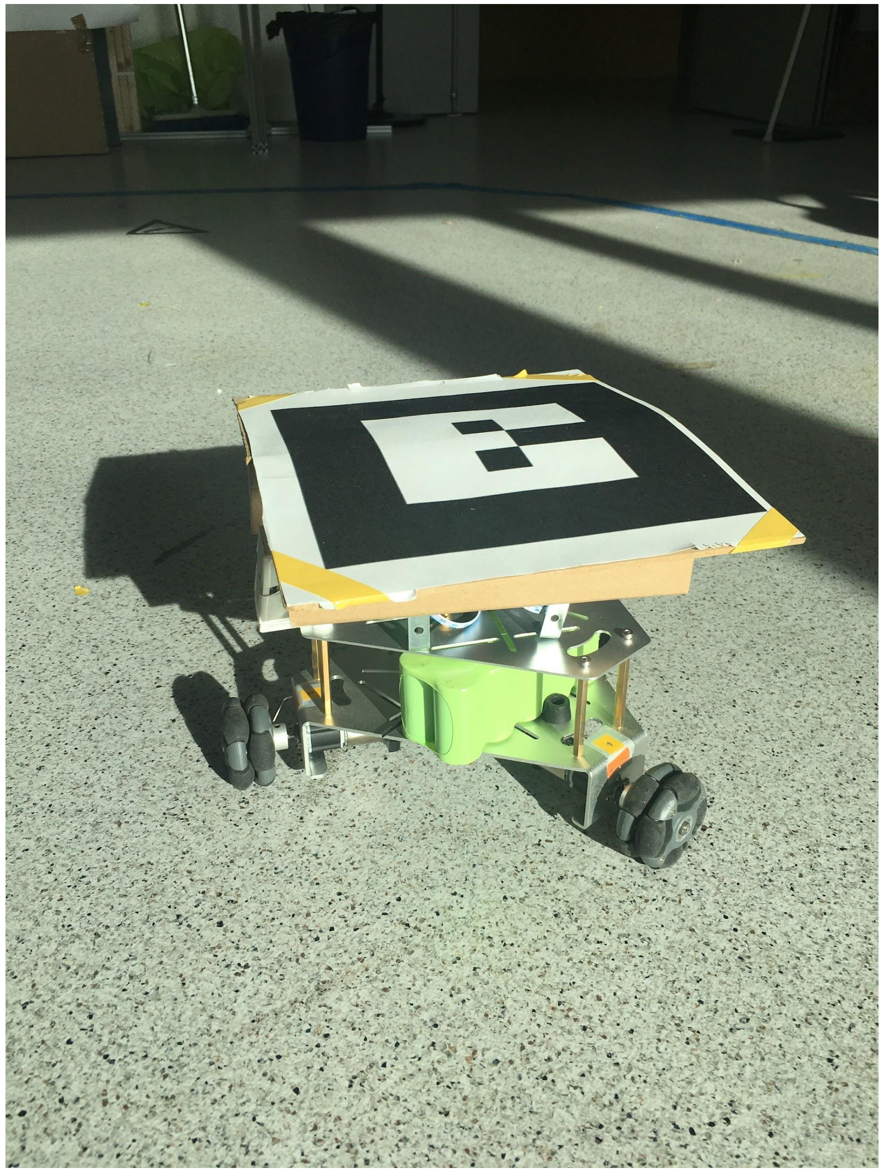
- Can’t be solved using robot states alone (*priors, fwd/inv* models)
- Should encode goal position (*(V)AE*,...)
- => Contradictory objectives (combination non efficient)

[Lesort’18, State representation learning for control: An overview. Neural Networks]

DREAM EU H2020 Project: www.robotsthatdream.eu

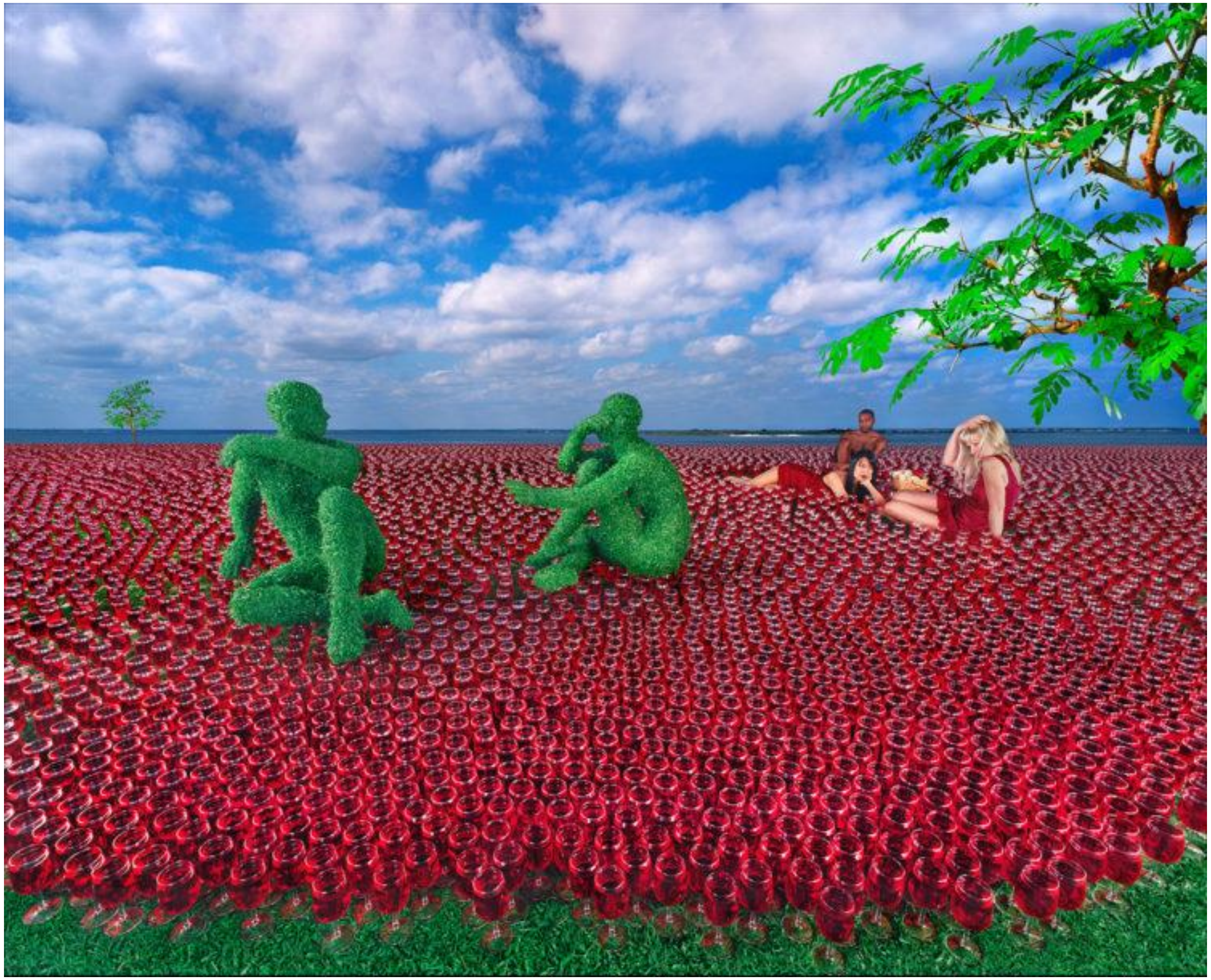
Evaluating learned representations: Environments: Omnirobot





[Kalifou'19]

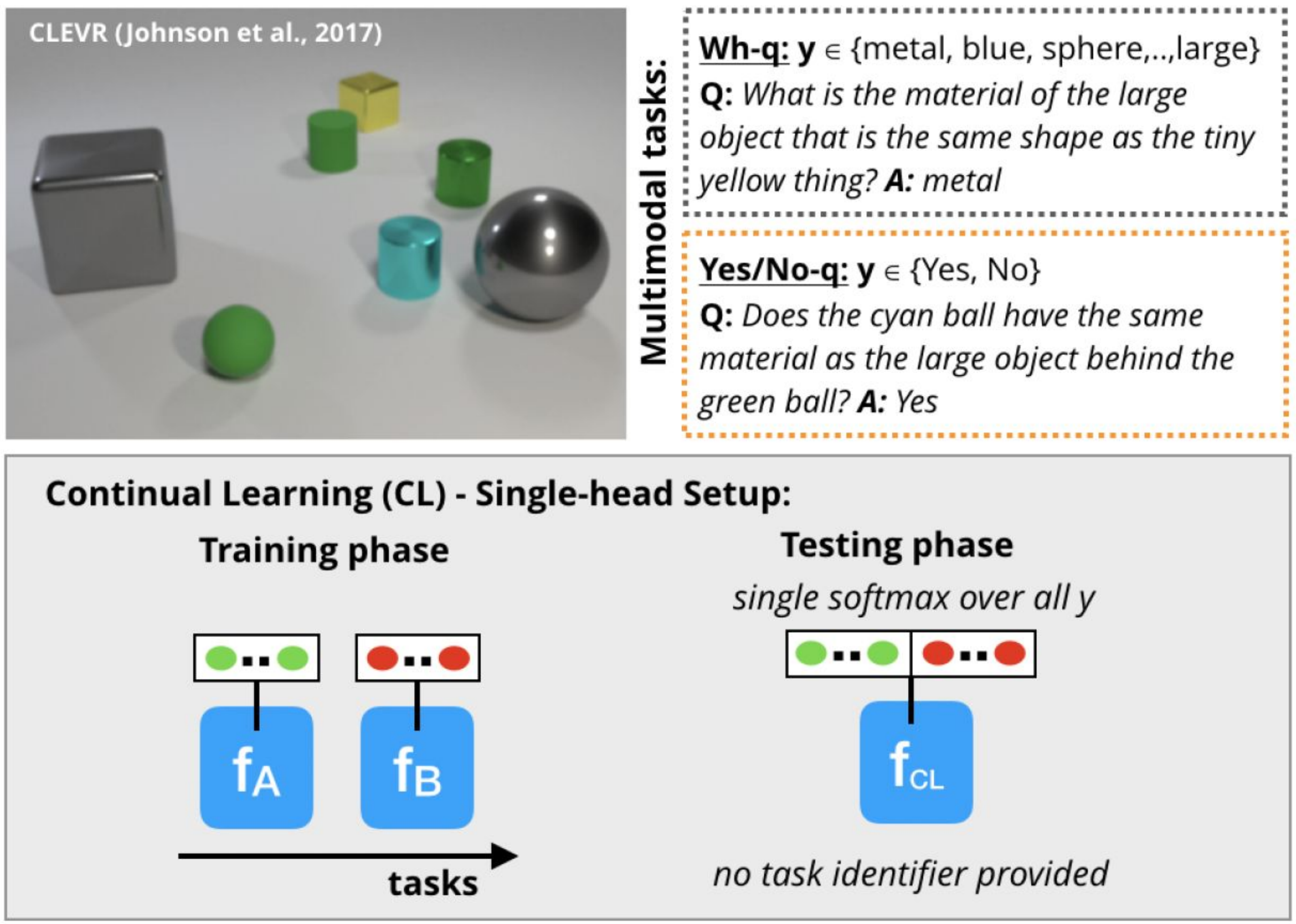
Hybrid Approaches



Catastrophic Forgetting in NLP

When applied to tasks with different difficulty:

- Regularization based approaches less suitable than rehearsal ones
- Order matters
  - WH -> Y/N question learning facilitates CL more than the opposite
  - Confirms psycholinguistic evidence



[Greco et al.'19]

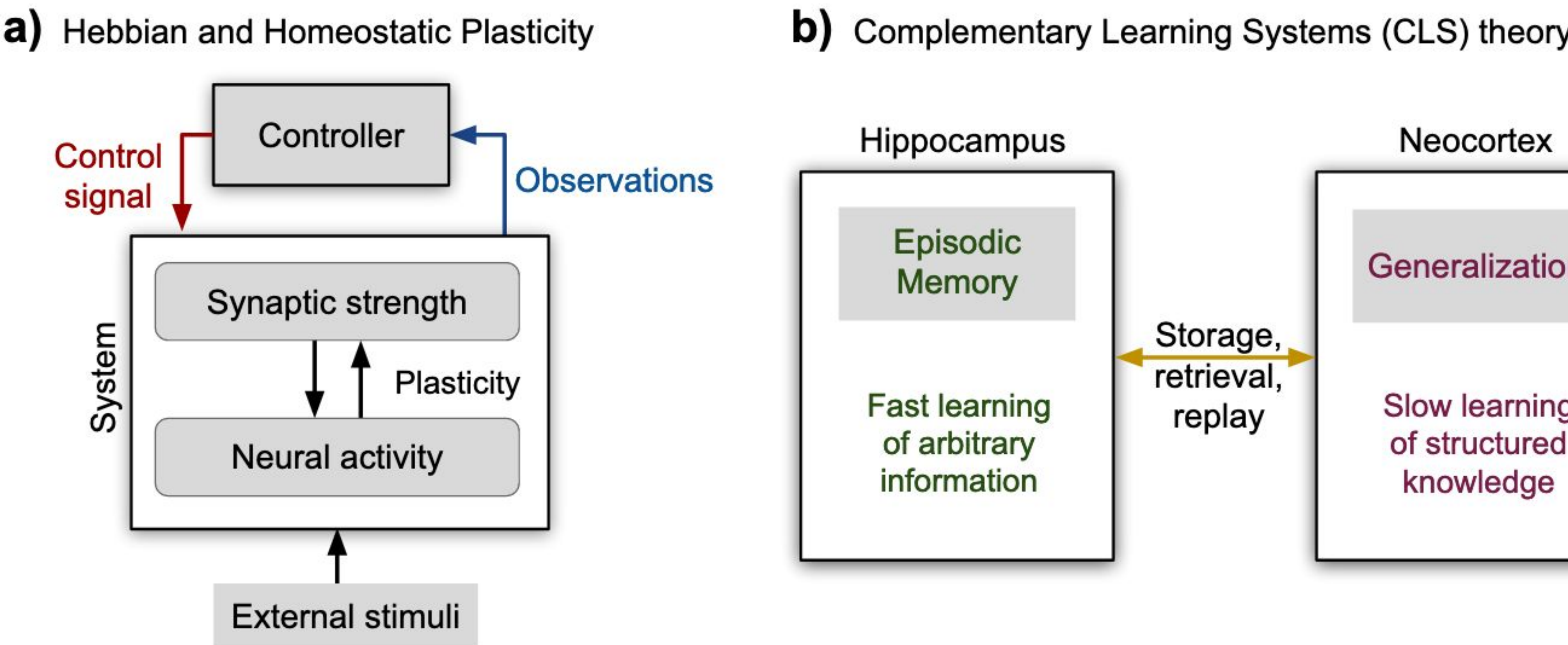


Figure 1: Schematic view of two aspects of neurosynaptic adaptation: a) Hebbian learning with homeostatic plasticity as a compensatory mechanism that uses observations to compute a feedback control signal (Adapted with permission from Zenke, Gerstner & Ganguli (2017)). b) The complementary learning systems (CLS) theory (McClelland et al. 1995) comprising the hippocampus for the fast learning of episodic information and the neocortex for the slow learning of structured knowledge.

[Parisi'19]

Continual RL in multi-task, lifelong & real life settings

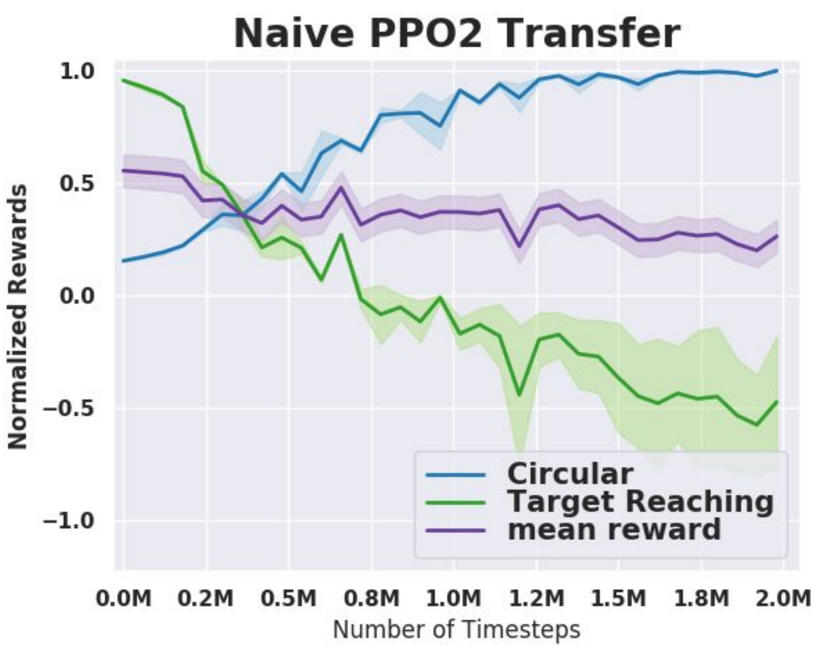
Reward (for k last steps. z: the 2D pos. wrt the circle center.)

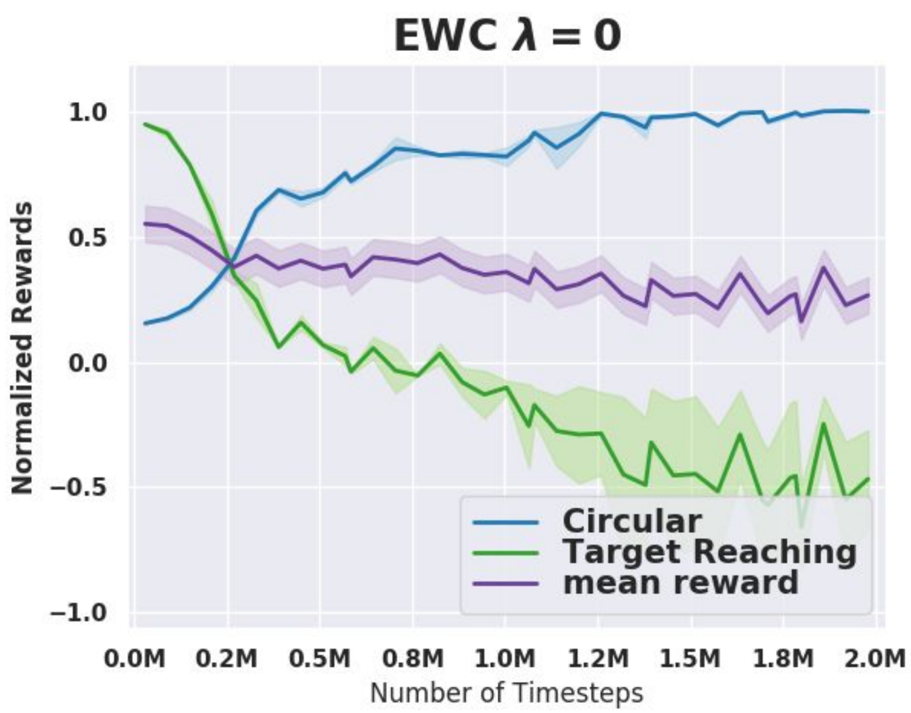
$$R_{t,circle} = 1 - \lambda(\|z_t\| - R_{circle})^2 \quad (1)$$

$$R_{t,movement} = \|z_t - z_{t-k}\|_2^2 \quad (2)$$

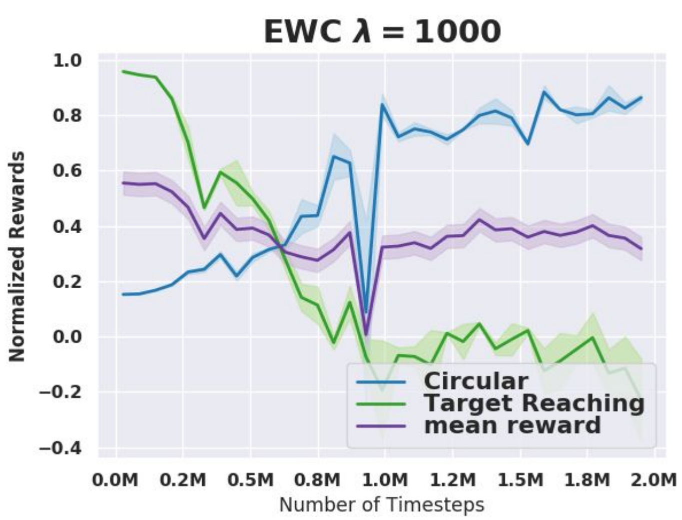
$$\lambda R_{t,circle} * R_{t,movement} + \lambda^2 R_{t,bump} \quad (3)$$

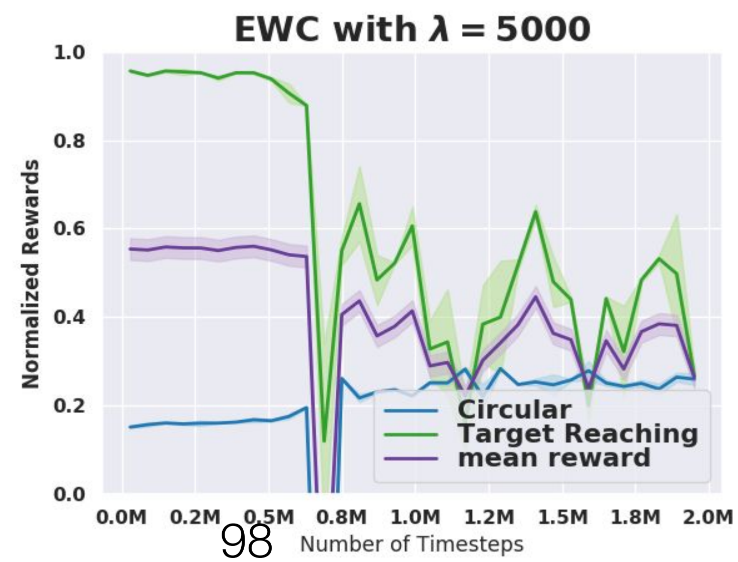
Without regularization:

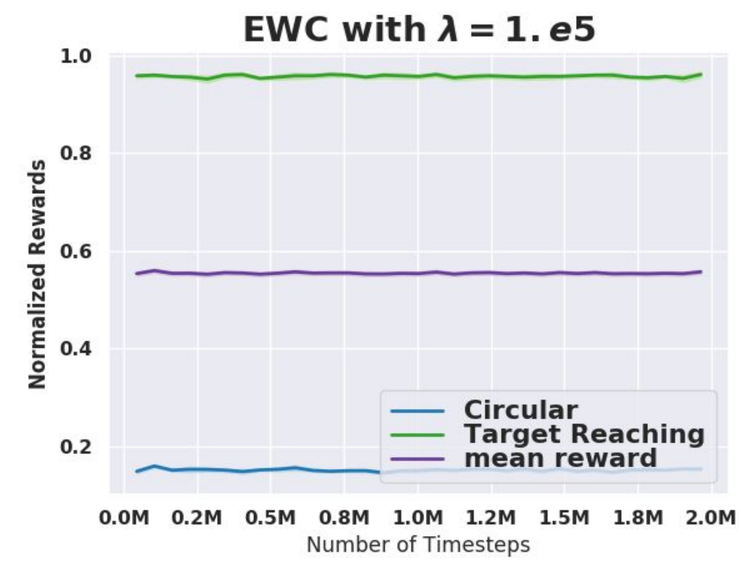




With regularization (+lambda -> + emphasis on the importance of prev. task)

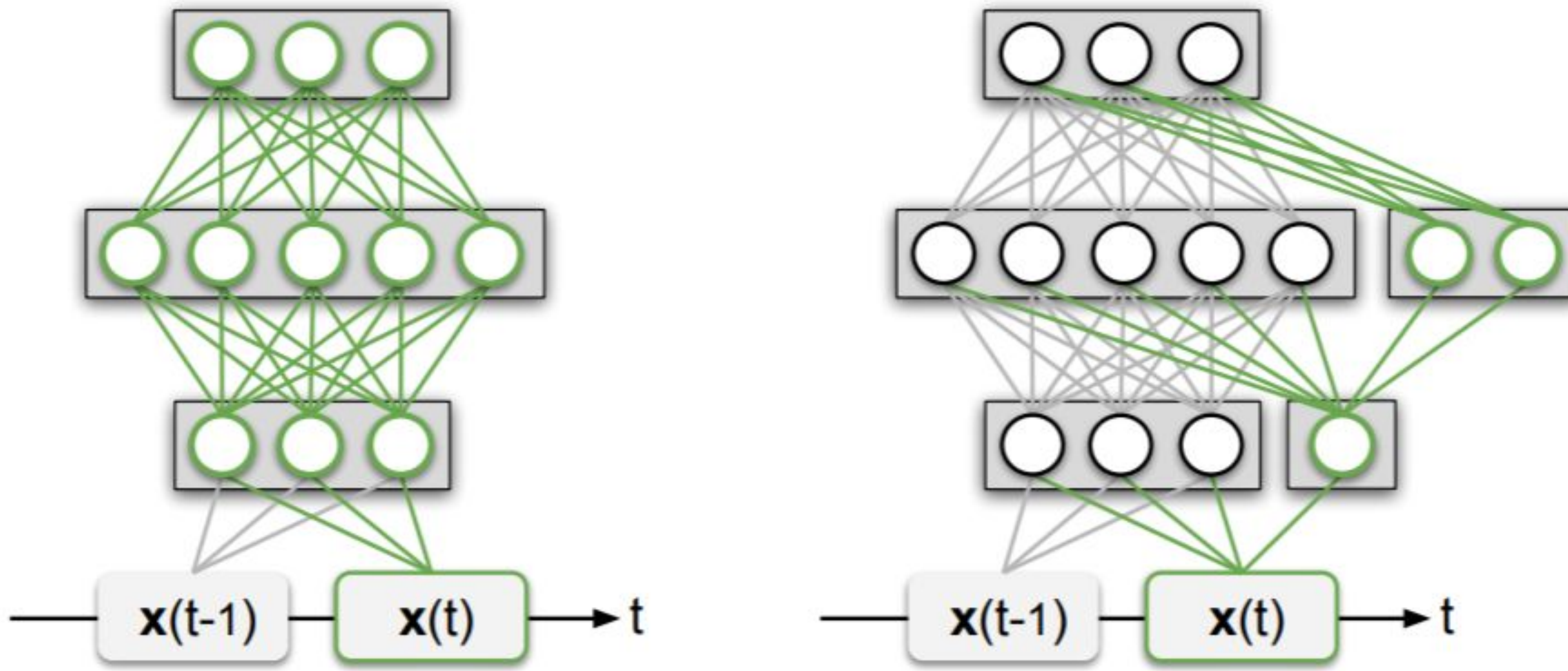






[Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer. Traoré et al. 2019]

Regularization & Architectural Strategies



a) Retraining with regularization

b) Training with network expansion

[Parisi'18]

Evaluation Benchmarks: RL & Robotics

Acrobot-v1
Swing up a two-link robot.

CartPole-v1
Balance a pole on a cart.

MountainCar-v0
Drive up a big hill.

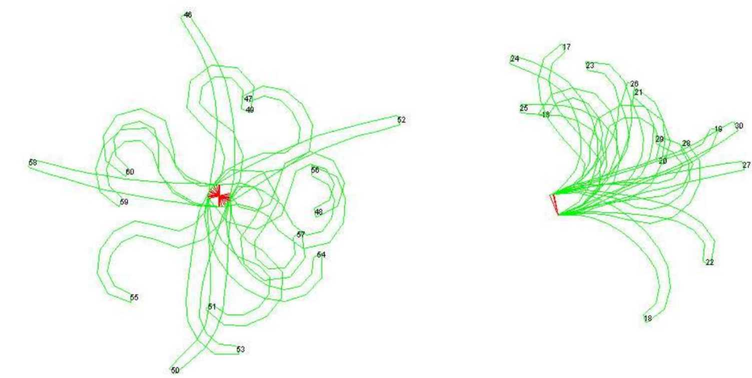


Figure 3: Examples of initial states for the rotating-base experiments (left) and the fixed-base experiments (right). Starting states also include velocities, which are not shown.

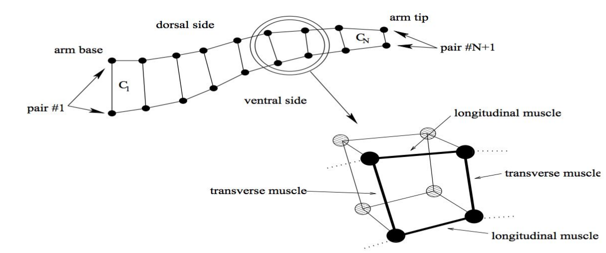
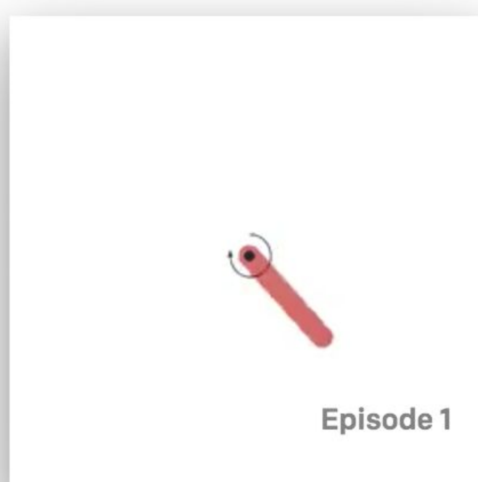
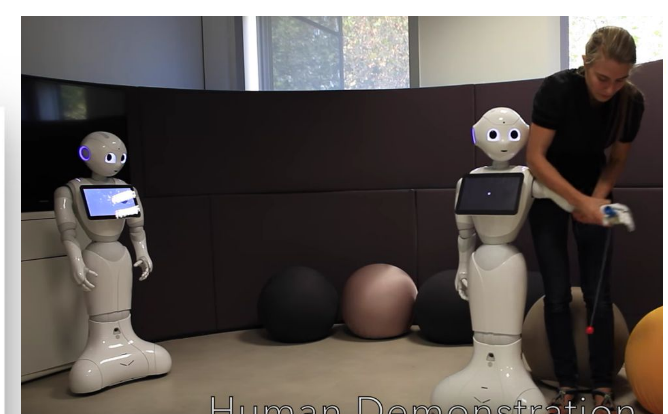


Figure 1: An N compartment simulated Octopus arm. Each constant area compartment C_i is defined by its surrounding 2 longitudinal muscles (ventral and dorsal) and 2 transverse muscles. Circles mark the $2N + 2$ point masses in which the arm's mass is distributed. In the bottom right one compartment is magnified with additional detail.

MountainCarContinuous-v0
Drive up a big hill with continuous control.



Pendulum-v0
Swing up a pendulum.



Ball in cup

Figure : Ball-in-cup skill, which the Meka acquired through reinforcement learning

https://gym.openai.com/envs/#classic_control
[Octopus arm [Engel'06, Munk'16]]()

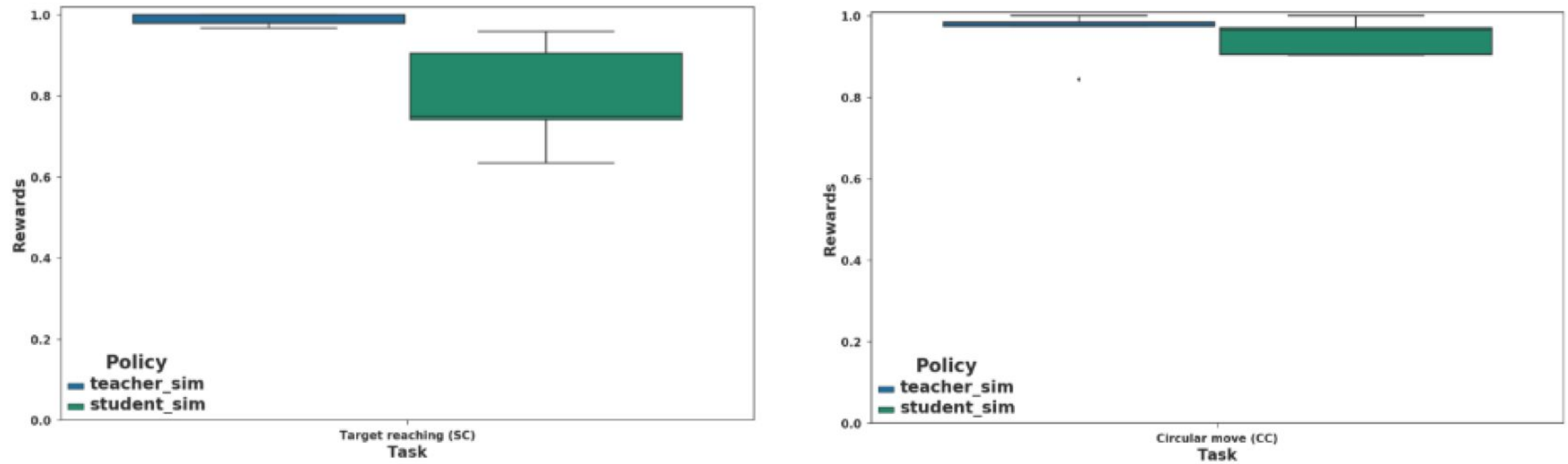
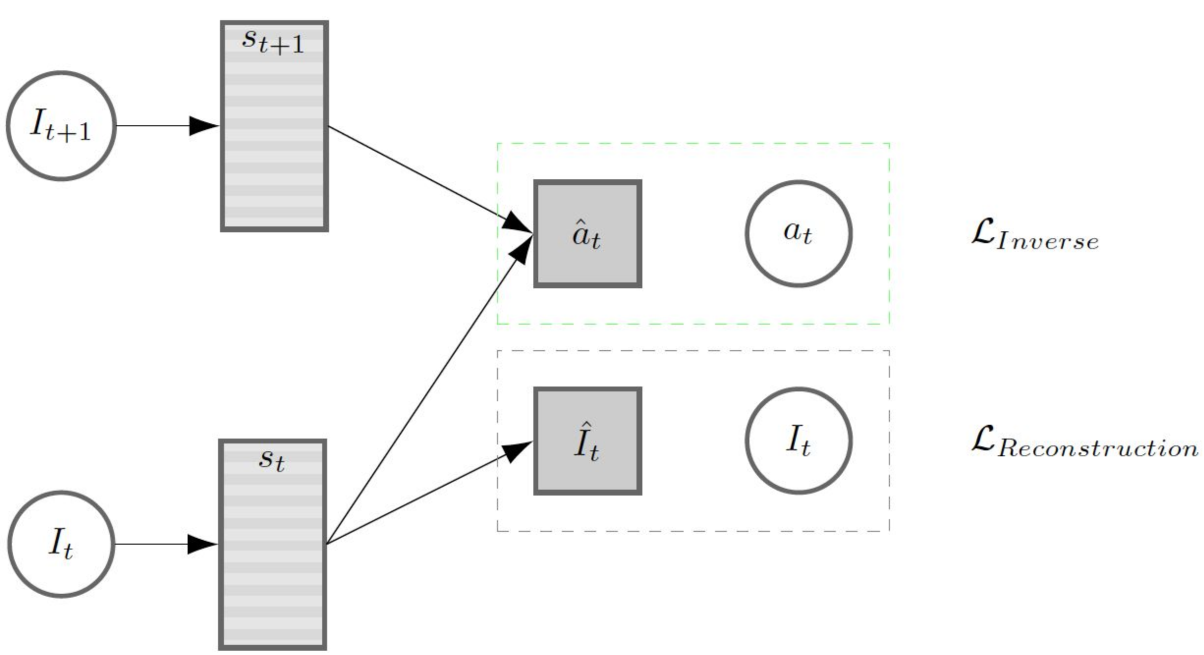
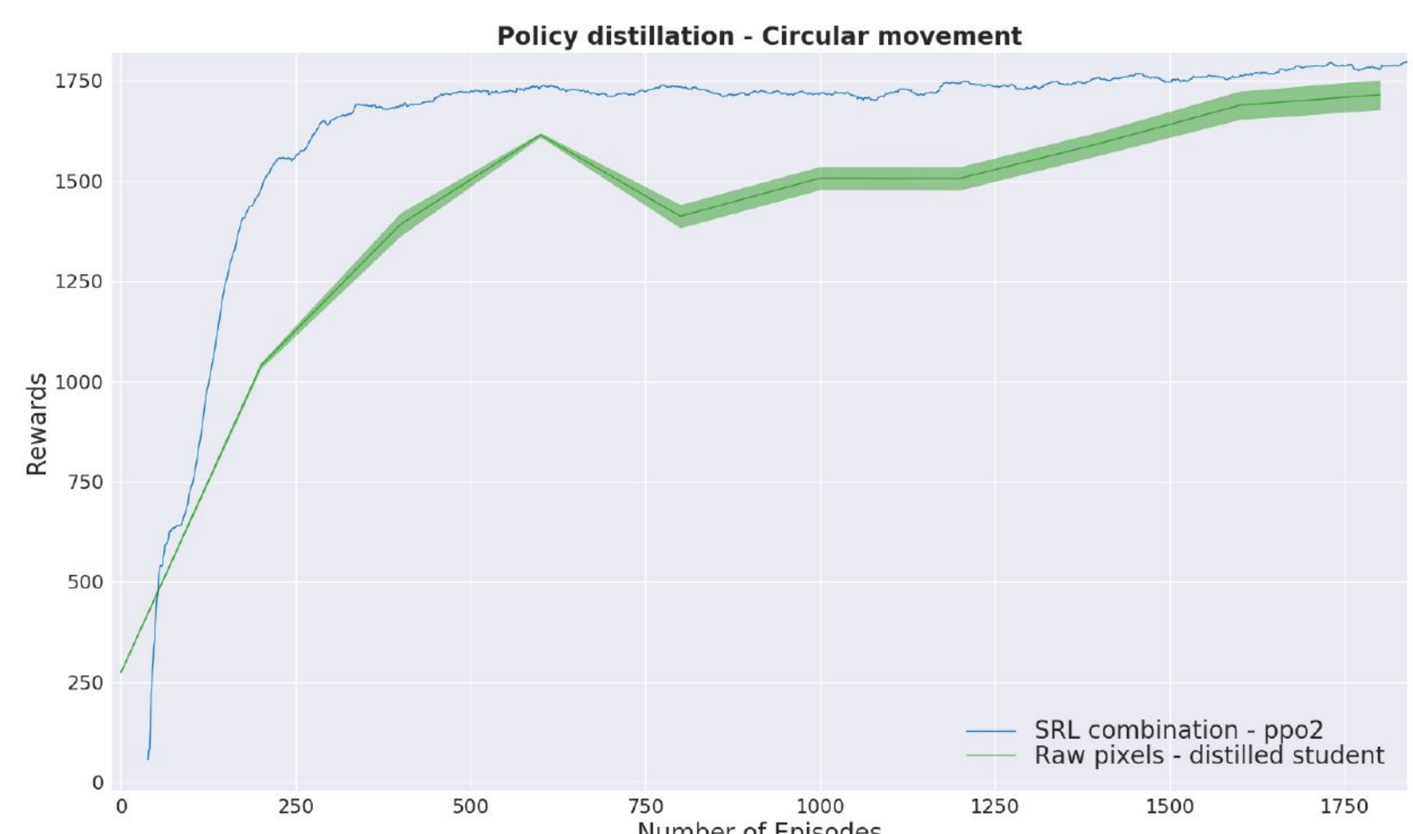
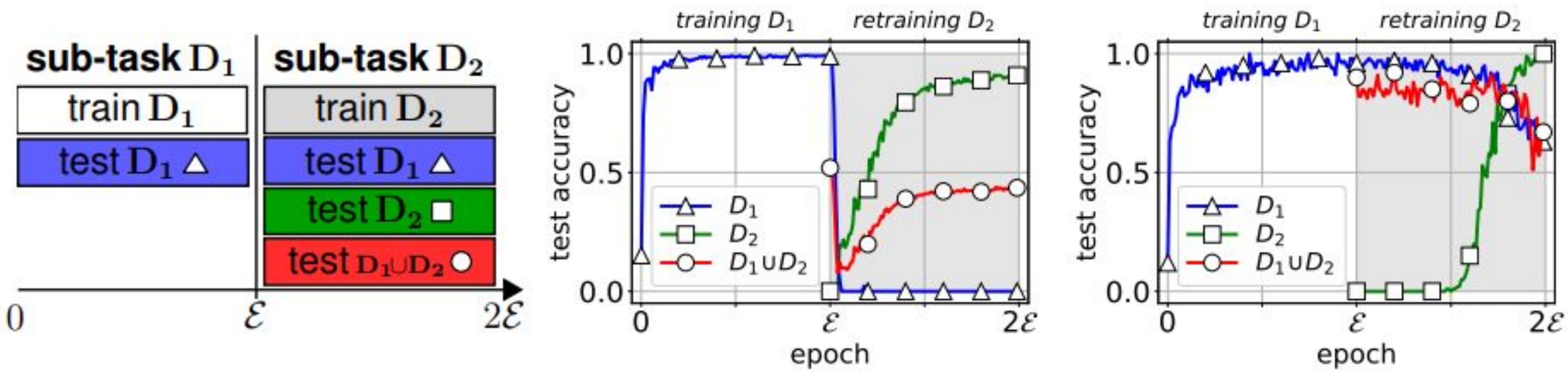


Figure 4. Comparison between performance (normalized mean reward and standard error) of policy trained on one task only to distilled student policy on the two tasks. The student policy has similar performance on both tasks. **Left:** Target Reaching (TR). **Right:** Target Circling (TC) task.





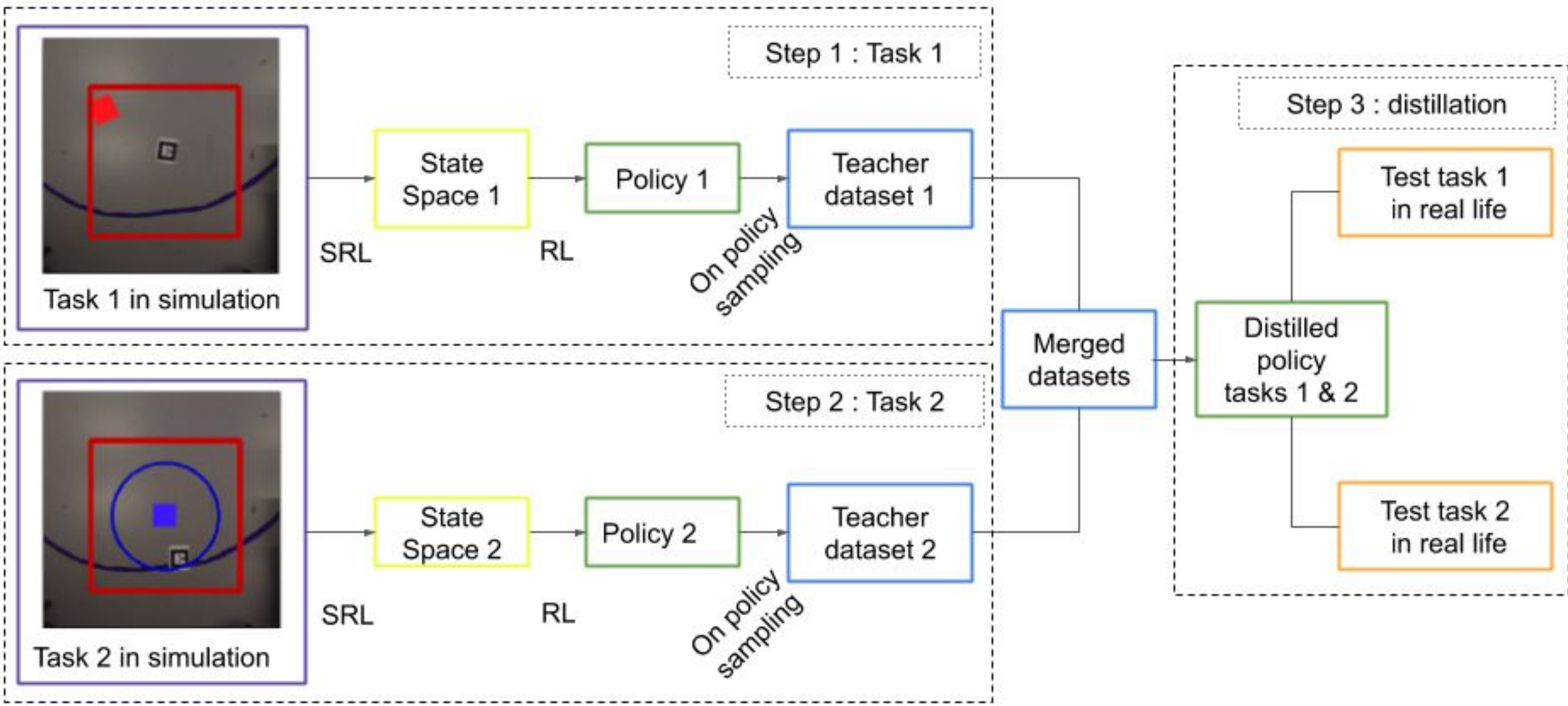
Catastrophic Forgetting in DNNs [Pfulf & Gepperth'19]



(a) Training scheme (b) with CF (c) without CF

Continual RL in multi-task, lifelong & real life settings

- Distilling a teacher network's knowledge (=each task policy) to a student
  => A single network to perform all tasks
- Deploying the multi-task policy sim2real
- No task label at test time!



Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer. [Traoré'19]

Evaluation setups [Chaudhry18]

- **Single-head** evaluation: **1** output space over labels for **all** tasks (+ difficult, + realistic)
- **Multi-head** predictions (restricted to when task labels provided).

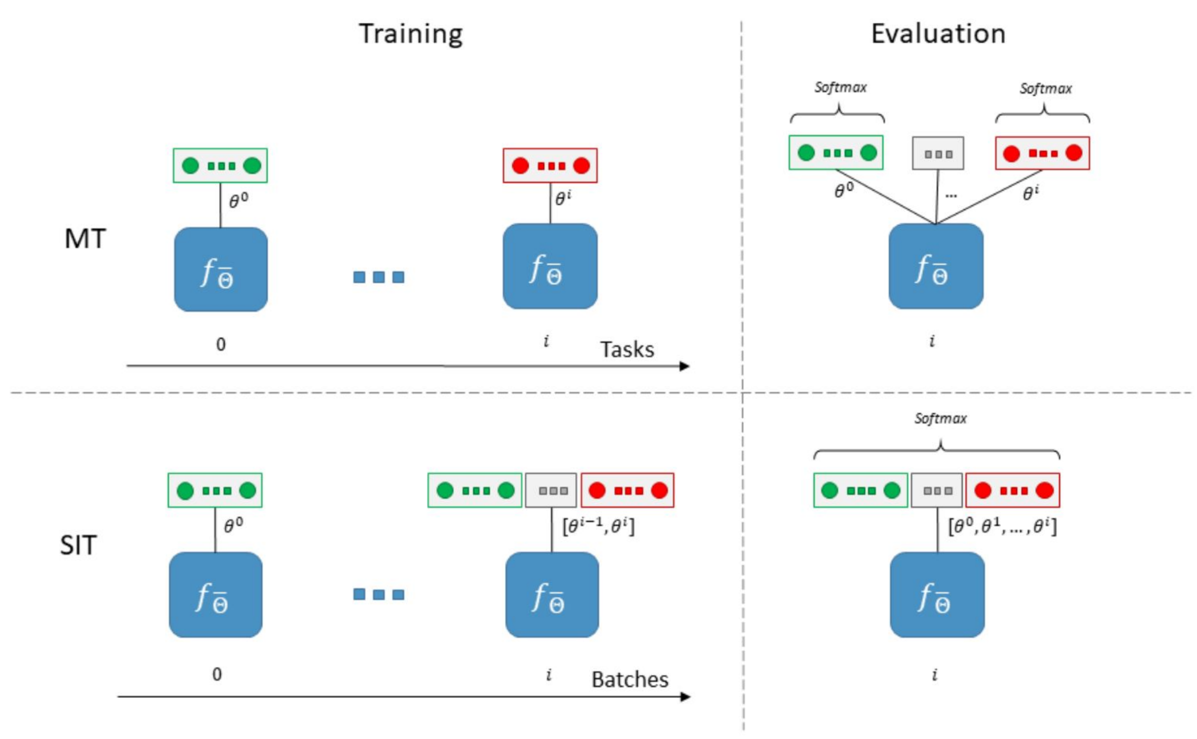


Figure 2: Key architectural differences between MT and SIT scenarios: a disjoint output layer (also denoted as "head") is used in MT for each independent task, while a single (dynamically expanded) output layer is used in SIT to include all the classes encountered so far. Better viewed in color.

[Lomonaco & Maltoni'18]