



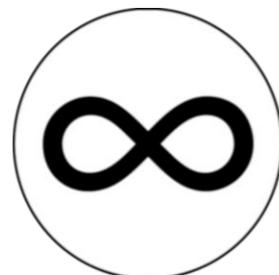
inria
INVENTEURS DU MONDE NUMÉRIQUE

ENSTA

Autonomous Systems and Robotics
Computer Science and System Engineering Laboratory



INSTITUT
POLYTECHNIQUE
DE PARIS



Continual **AI**



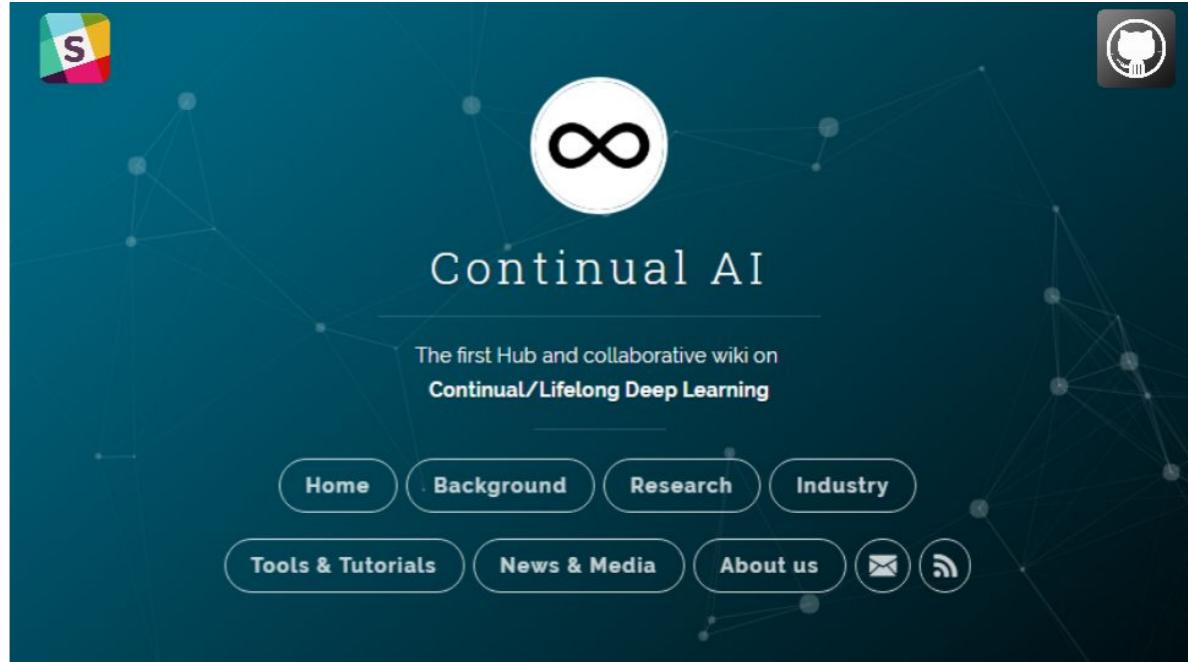
www.robots that dream.eu

Continual Learning and Robotics: an overview

Natalia Díaz Rodríguez, PhD



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



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.

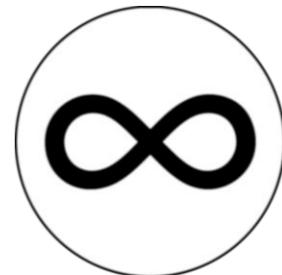
Cognitive Science: Elaborating computer and robotic models as tools for understanding developmental processes in humans.

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 of reinforcement learning;
- The role of **embodiment**, in particular through the concept of morphological computation, as well as the structure of motor primitives/muscle synergies 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.

About me

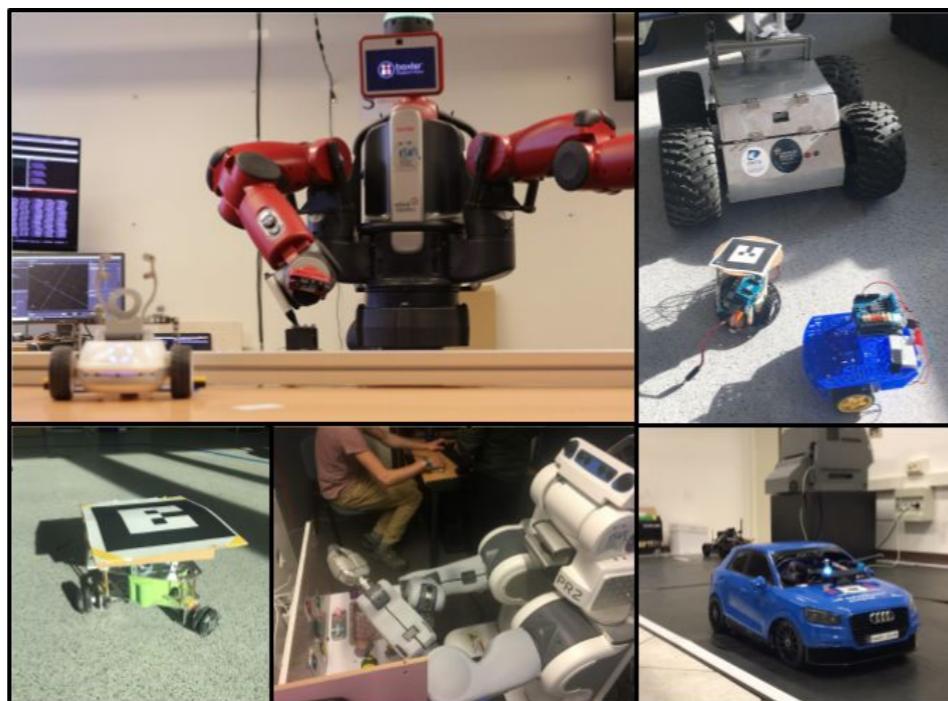


ContinualAI

Lifelong or Continual Learning (CL)

A model's ability to learn new skills

- without forgetting previous knowledge
- **sequentially**, from non i.i.d. data streams
- being able to solve **any** task at the end



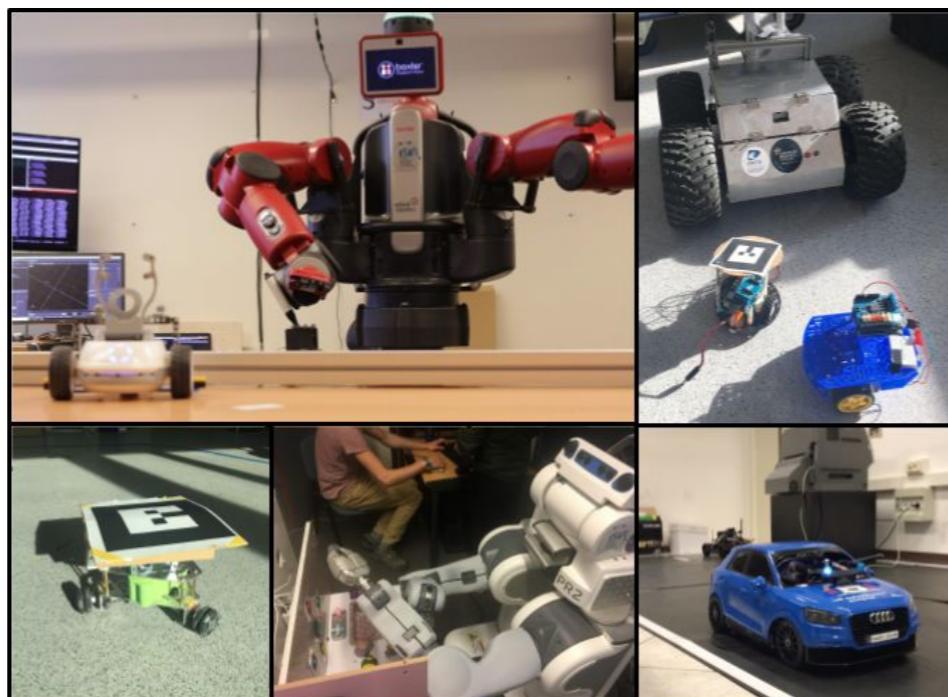
Lifelong or Continual Learning (CL)

A model's ability to learn new skills

- without forgetting previous knowledge
- **sequentially**, from non i.i.d. data streams
- being able to solve **any** task at the end

How it differs from **Never-Ending Learning** [Mitchell'09,14] ?

- Synonyms! Here: Forgetting & memory (to be minimized)



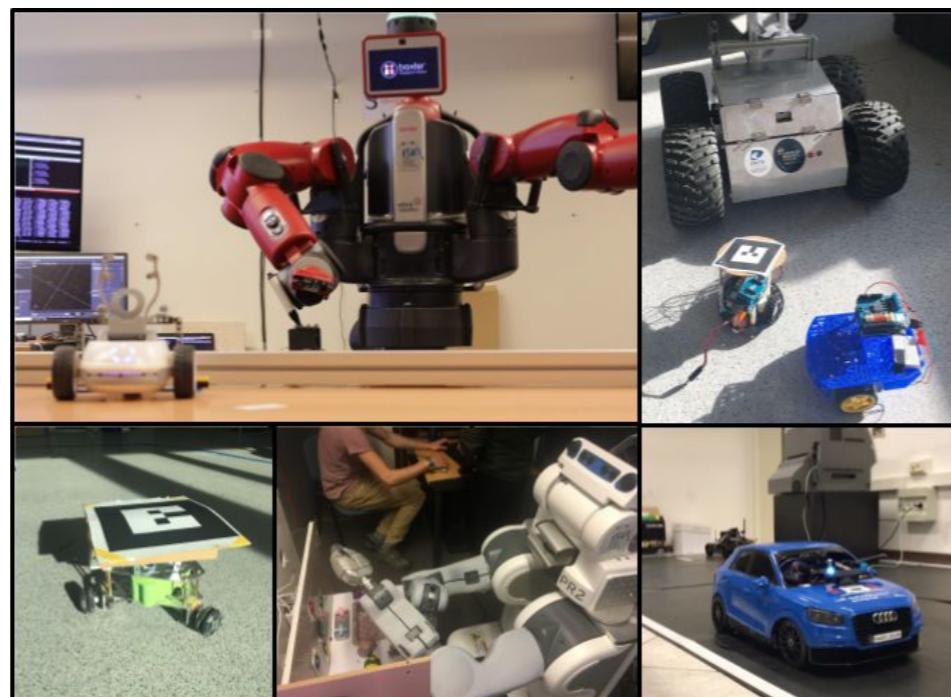
Lifelong or Continual Learning (CL)

A model's ability to learn new skills

- without forgetting previous knowledge
- **sequentially**, from non i.i.d. data streams
- being able to solve **any** task at the end

How it differs from **Meta-Learning** [Schmidhuber'87] ?

- Here: Shuffling & picking not possible!



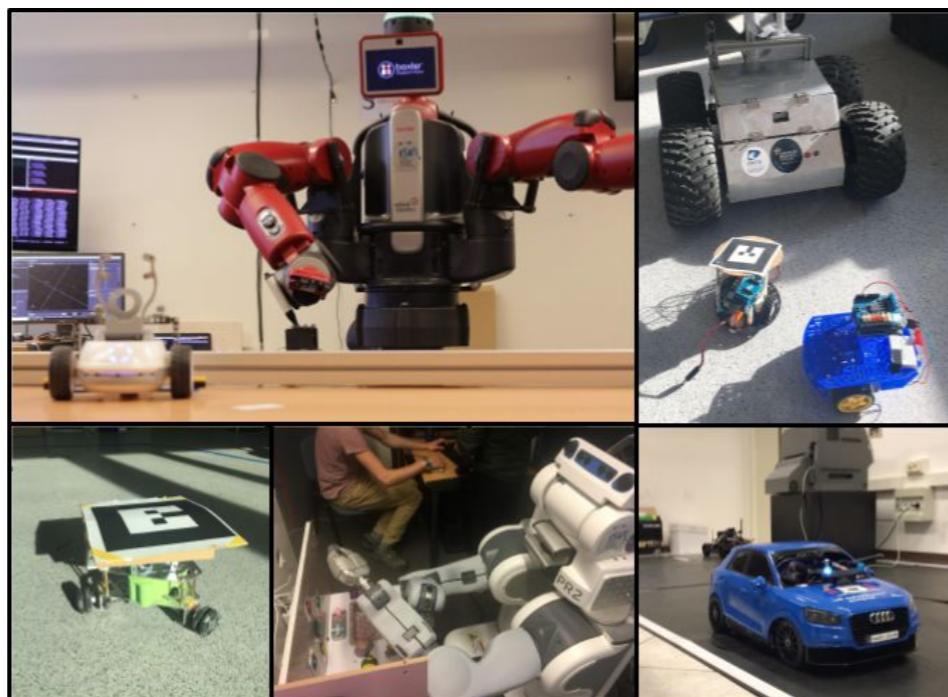
Lifelong or Continual Learning (CL)

A model's ability to learn new skills

- without forgetting previous knowledge
- **sequentially**, from non i.i.d. data streams
- being able to solve **any** task at the end

How it differs from **Transfer-Learning**?

- Transfer learning may not care about past tasks



Catastrophic Forgetting



Catastrophic Forgetting (CF)

[McCloskey and Cohen'89; French'99]

- Largest problem of DNNs in non-static settings
- Training with new data **interferes** with previously learned knowledge



Task 1



Task 2



Task 3

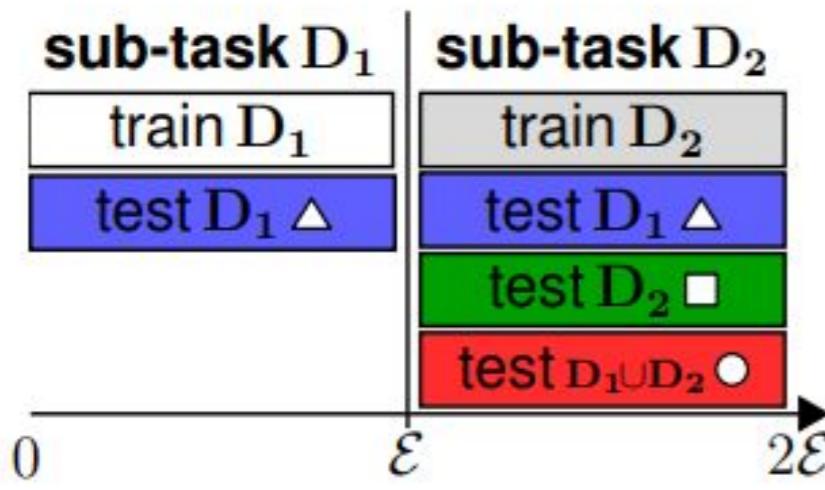


Task 2

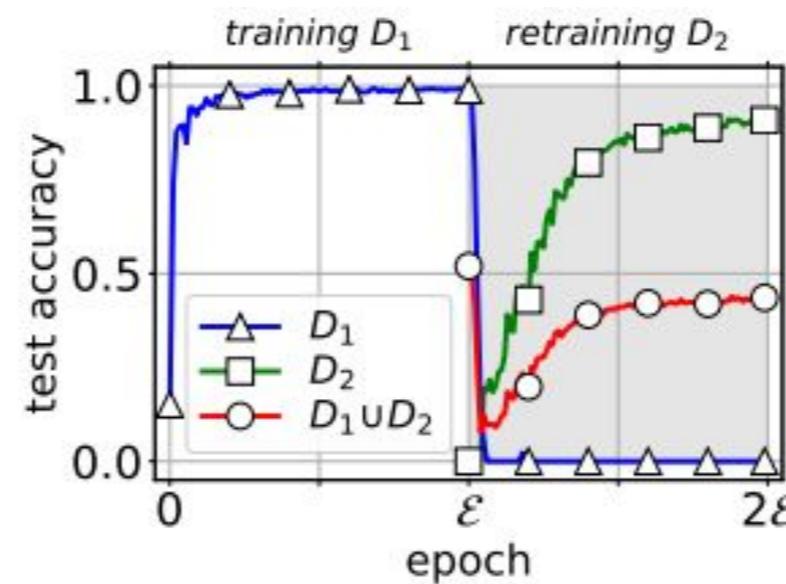
Catastrophic Forgetting (CF)

[McCloskey and Cohen'89; French'99]

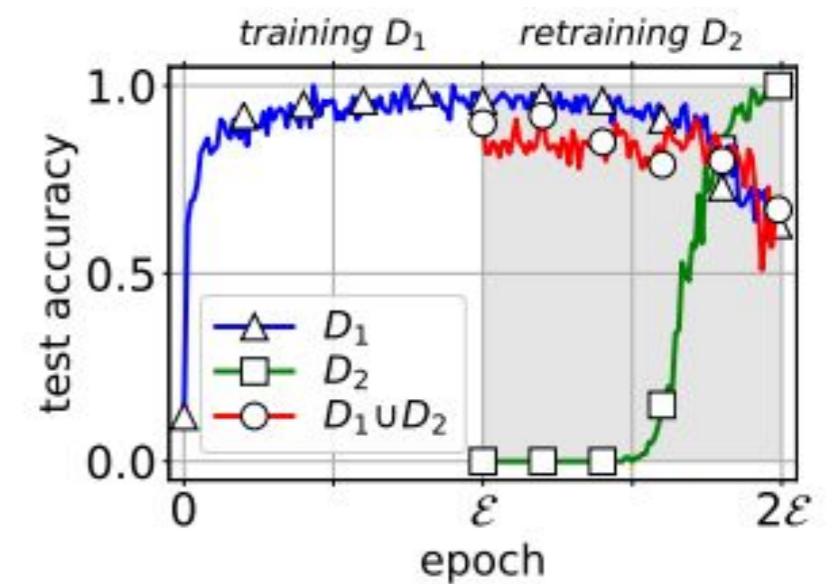
- Largest problem of DNNs in non-static settings
- Training with new data **interferes** with previously learned knowledge
 - Abrupt performance decrease
 - Knowledge overwritten



(a) Training scheme



(b) with CF



(c) without CF

Catastrophic Forgetting in DNNs

[Pföhl & Gepperth'19]

Different Image classification schemes for 10 class datasets

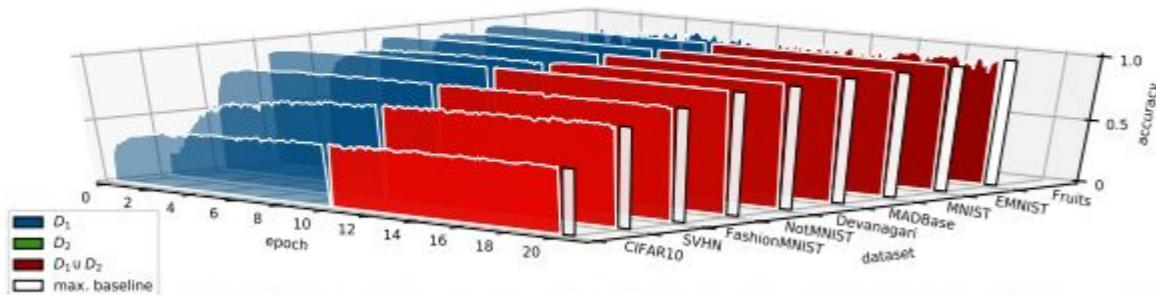


Figure 2: Best FC experiments for DP10-10. The blue surfaces (epochs 0-10) represent the accuracy on D_1 , the green (covered here) and red surfaces the accuracy on D_2 and $D_1 \cup D_2$ during re-training (epochs 10-20). The white bars indicate baseline performance. See also Appendix B for 2D plots.

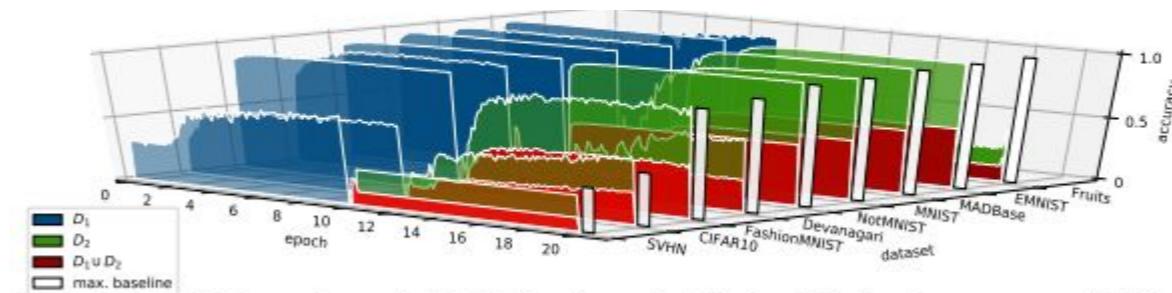


Figure 3: Best D-FC experiments for SLT D5-5, to be read as Fig. 2 and showing the occurrence of CF. See also Appendix B for 2D plots.

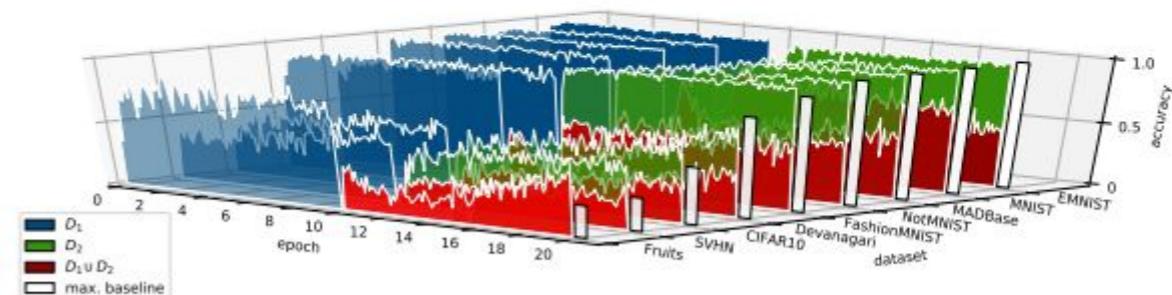
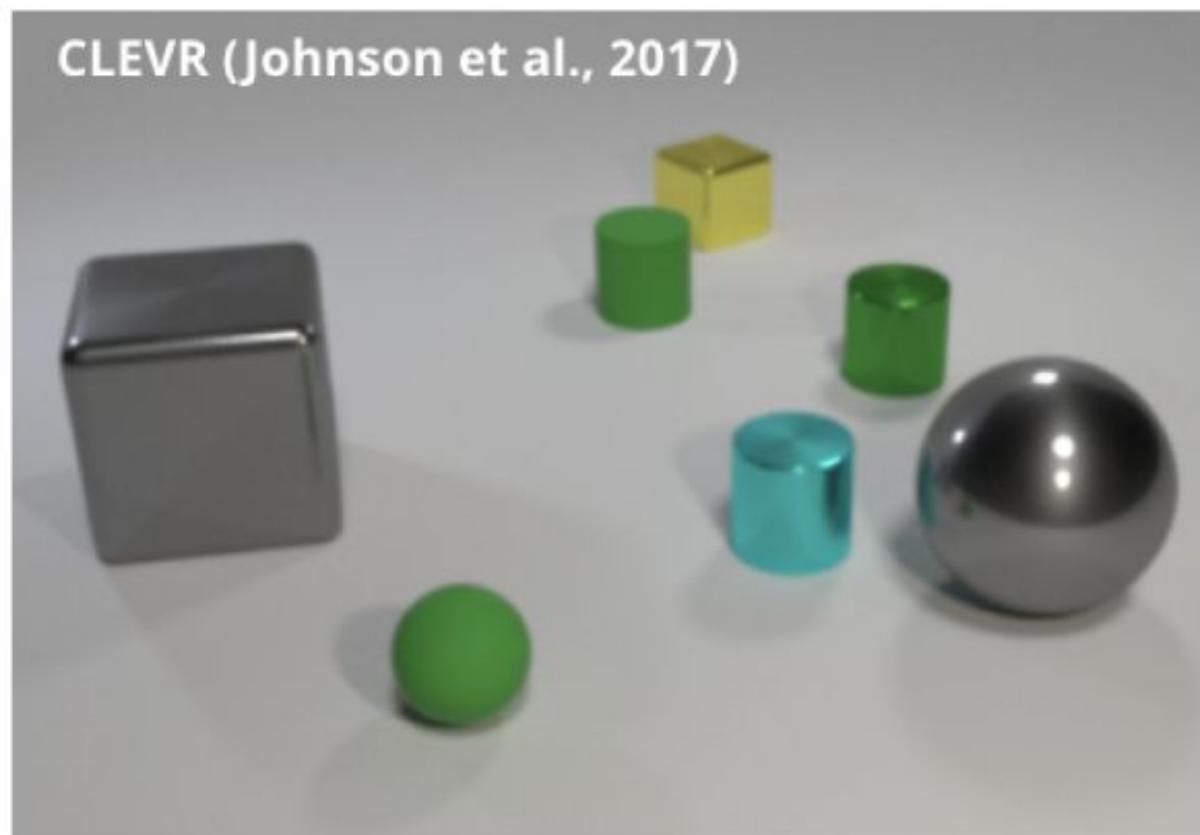


Figure 5: Best EWC experiments for SLT D5-5d constructed from all datasets, to be read as Fig. 2. We observe that CF happens for all datasets. See also Appendix B for 2D plots.

Catastrophic Forgetting (CF)

[McCloskey and Cohen'89; French'99]

- Order and difficulty matter (confirming psycholinguistic evidence)
 - Current CL methods: limited/no effect



Multimodal tasks:

Wh-q: $\mathbf{y} \in \{\text{metal, blue, sphere,..,large}\}$

Q: *What is the material of the large object that is the same shape as the tiny yellow thing?* **A:** metal

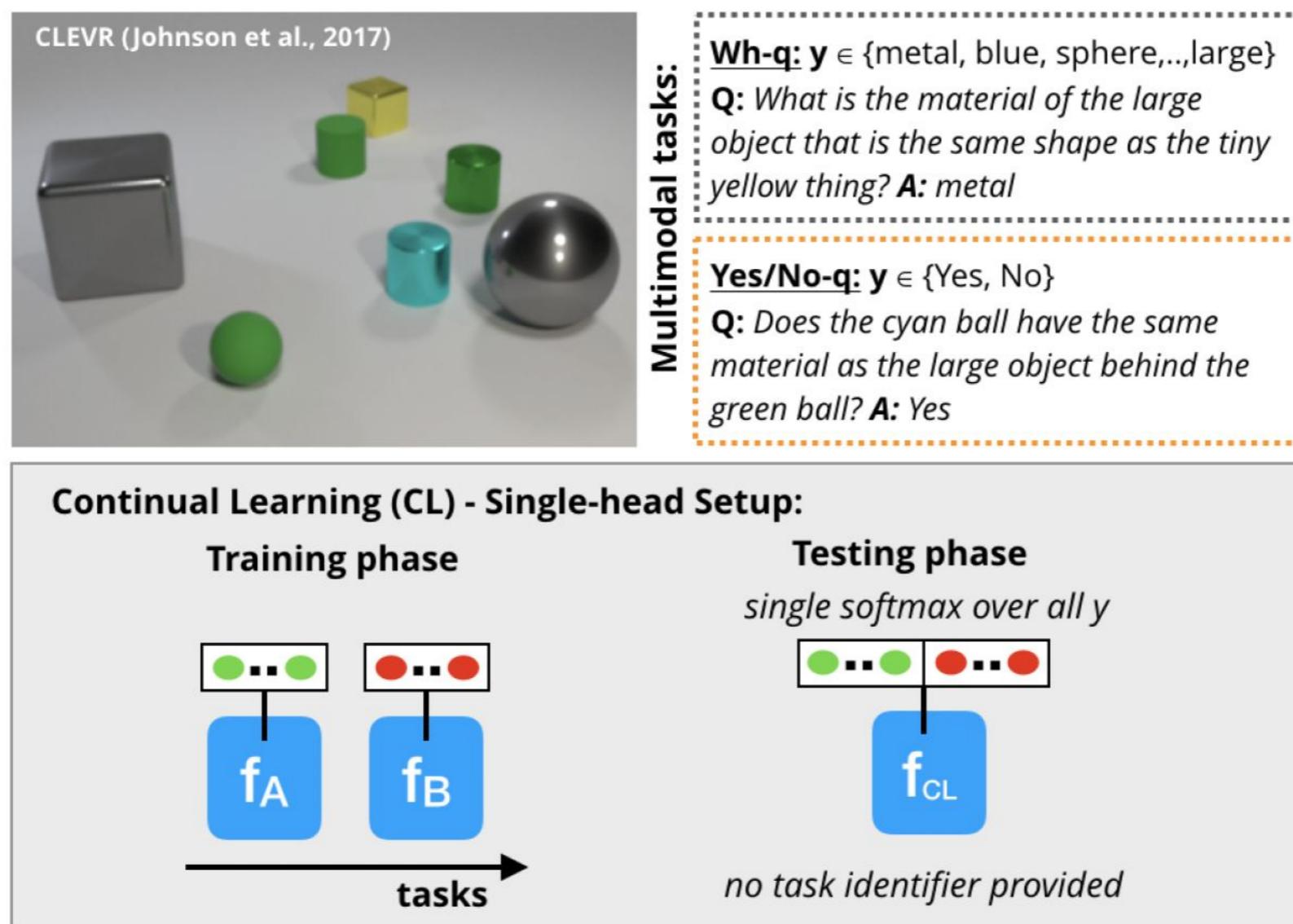
Yes/No-q: $\mathbf{y} \in \{\text{Yes, No}\}$

Q: *Does the cyan ball have the same material as the large object behind the green ball?* **A:** Yes

Catastrophic Forgetting in NLP

When applied to tasks with different difficulty:

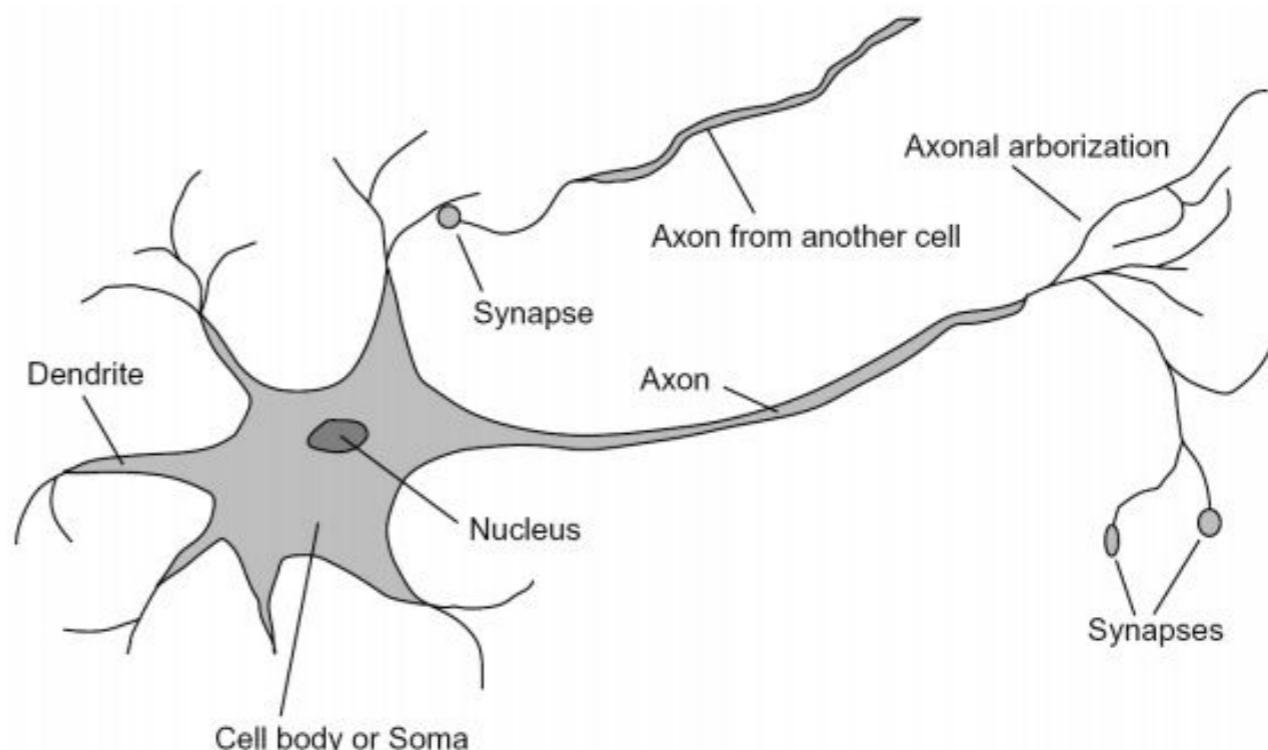
- Regularization based approaches less suitable than rehearsal ones
- Order matters
 - WH -> Y/N question learning facilitates CL more than the opposite
 - Confirms psycholinguistic evidence



[Greco et al.'19]

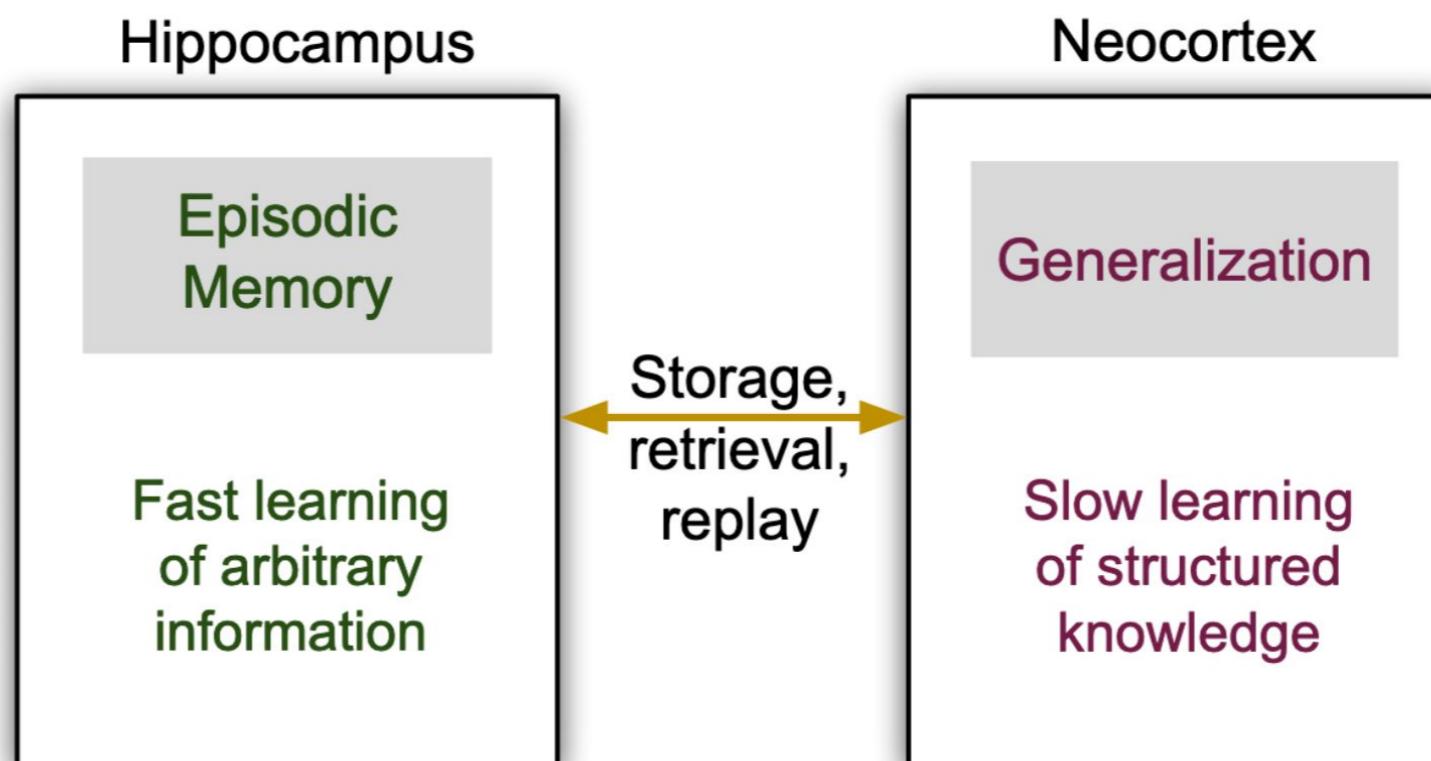
Catastrophic Forgetting: Biological Factors

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



Catastrophic Forgetting: Biological Factors

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



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 $\mathcal{D} = \{D_1, \dots, D_N\}$ 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 $\mathcal{D} = \{D_1, \dots, D_N\}$ 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 $\mathcal{D} = \{D_1, \dots, D_N\}$ 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:*

$$\forall D_i \in \mathcal{D}, \quad A_i^{CL} : \langle h_{i-1}, Tr_i, M_{i-1}, t_i \rangle \rightarrow \langle h_i, M_i \rangle \quad (1)$$

A Continual Learning Framework

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

$$\forall D_i \in \mathcal{D}, \quad A_i^{CL} : \langle h_{i-1}, Tr_i, M_{i-1}, t \rangle \rightarrow \langle h_i, M_i \rangle \quad (1)$$

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,

- $\mathcal{D} = \{D_1, \dots, D_N\}$: 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,

- $\mathcal{D} = \{D_1, \dots, D_N\}$: 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:

$$\forall D_i \in \mathcal{D}, \quad A_i^{CL} : \langle h_{i-1}, Tr_i, M_{i-1}, t \rangle \rightarrow \langle h_i, M_i \rangle \quad (1)$$

A Continual Learning Framework

In Continual Learning,

- $\mathcal{D} = \{D_1, \dots, D_N\}$: 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:

$$\forall D_i \in \mathcal{D}, \quad A_i^{CL} : \langle h_{i-1}, Tr_i, M_{i-1}, t \rangle \rightarrow \langle h_i, M_i \rangle \quad (1)$$

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

A Continual Learning Framework



Definition 4 *Continual Learning scenarios*

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

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	...



$t = 1$



$t = 2$



$t = 3$

A Continual Learning Framework

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

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

A Continual Learning Framework

Definition 5 *Content Update Type*

- *New Instances (NI)*.
- *New Concepts (NC)*
- *New Instances and New Concepts (NIC)*



A Continual Learning Framework

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

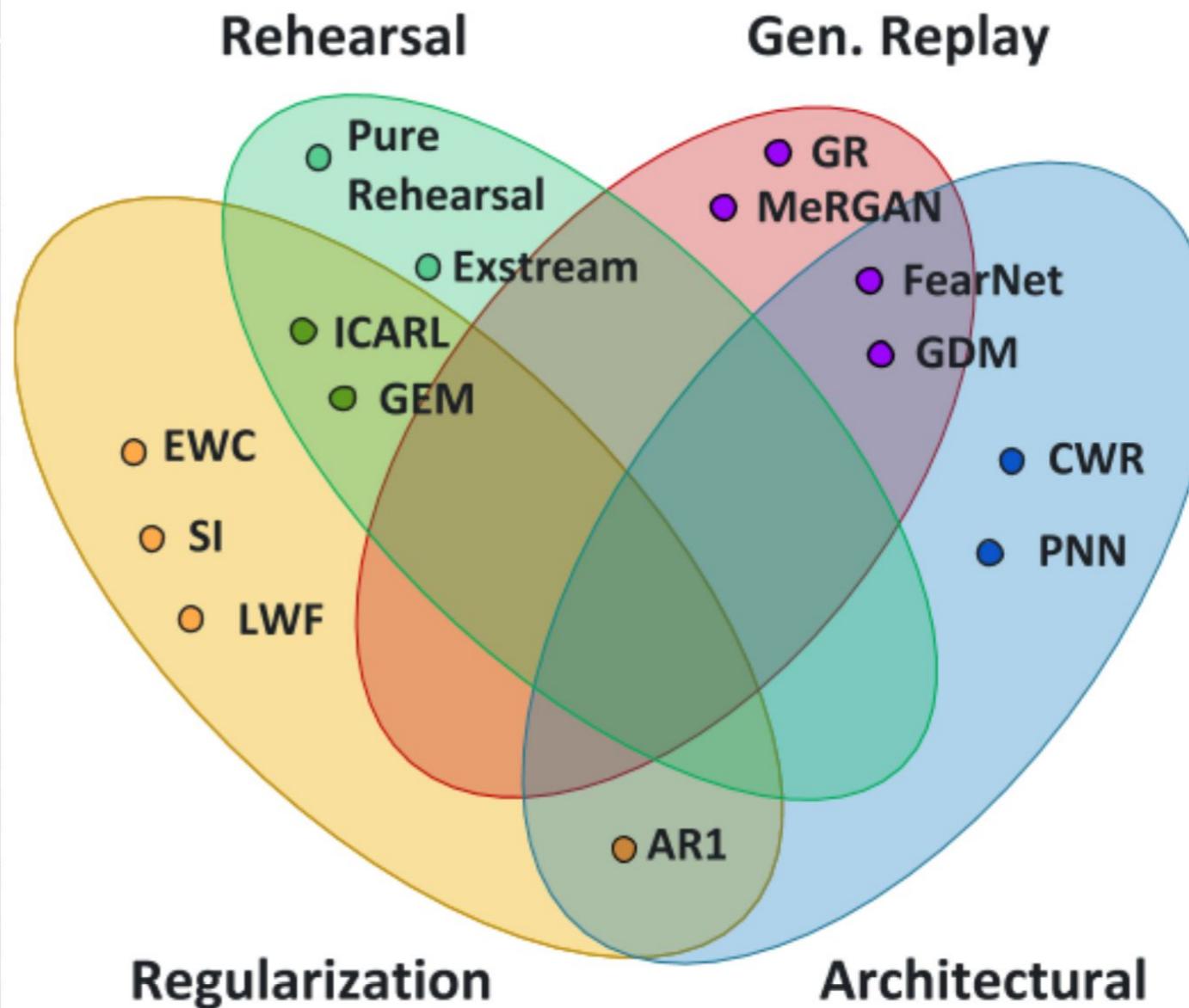


Popular Continual Learning Strategies

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)	✓	✓		

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)	✓	✓		

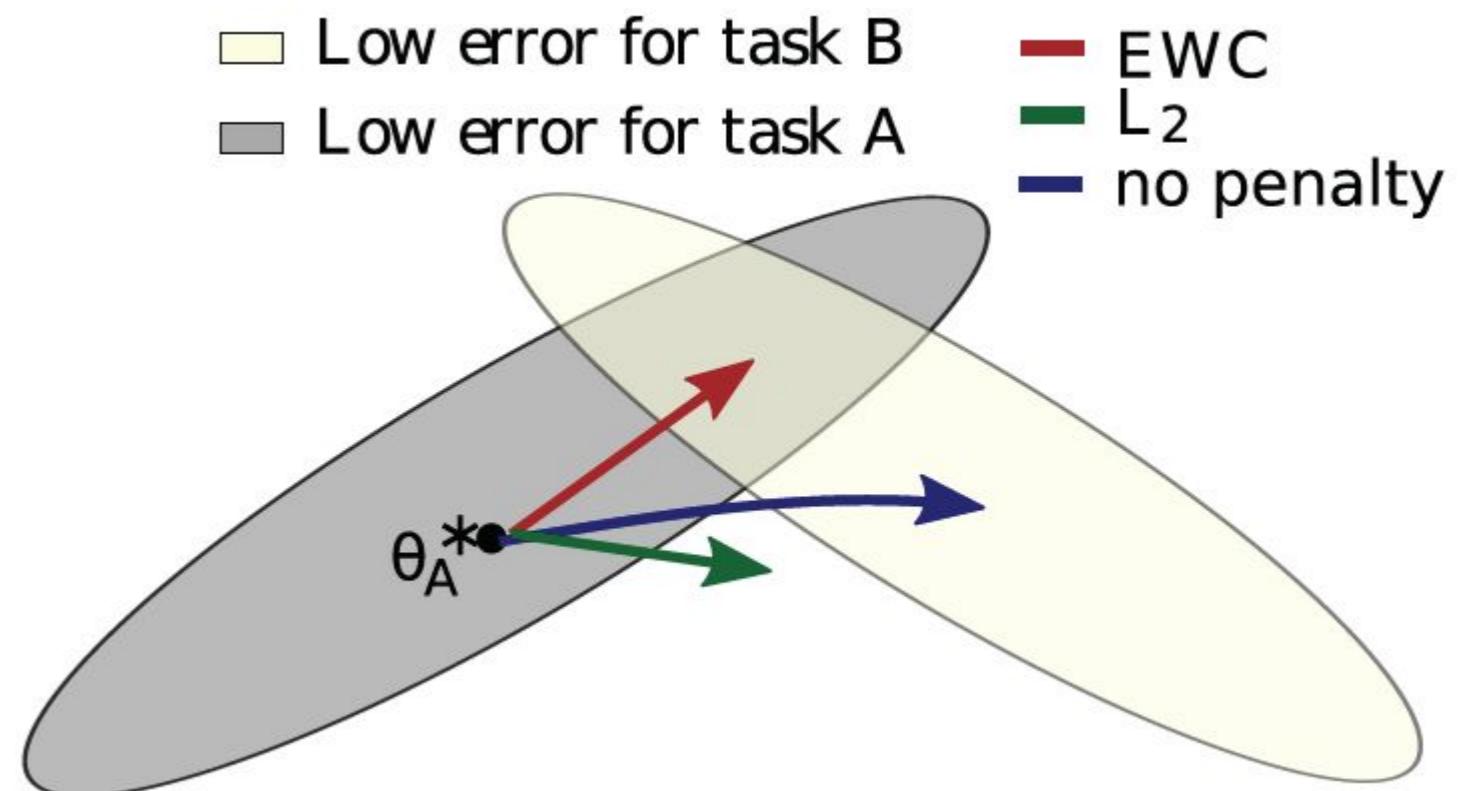


Regularization approaches



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]

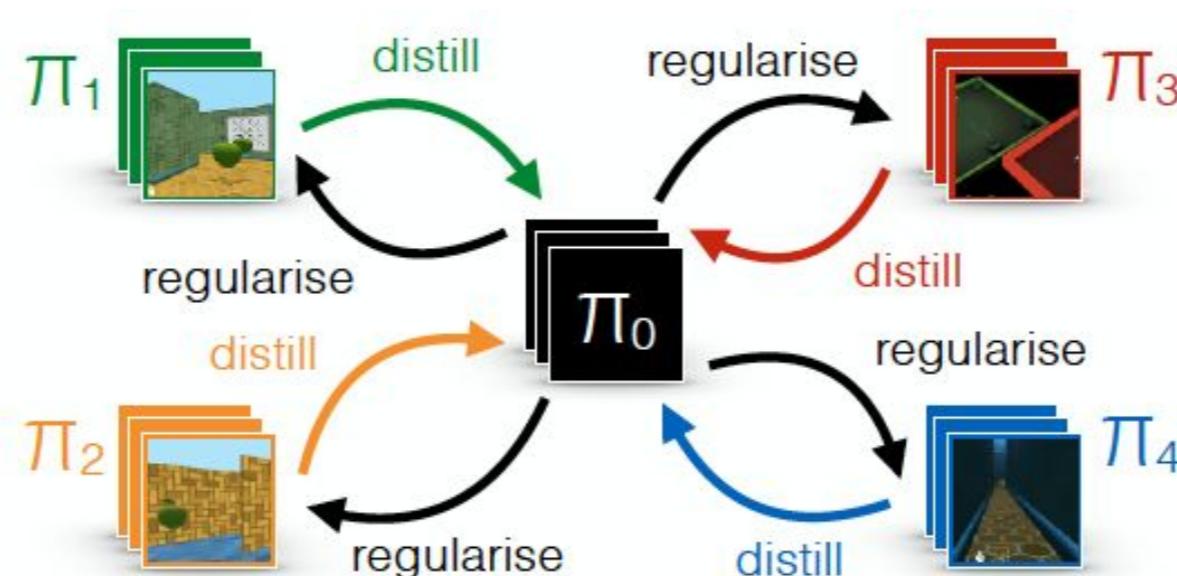
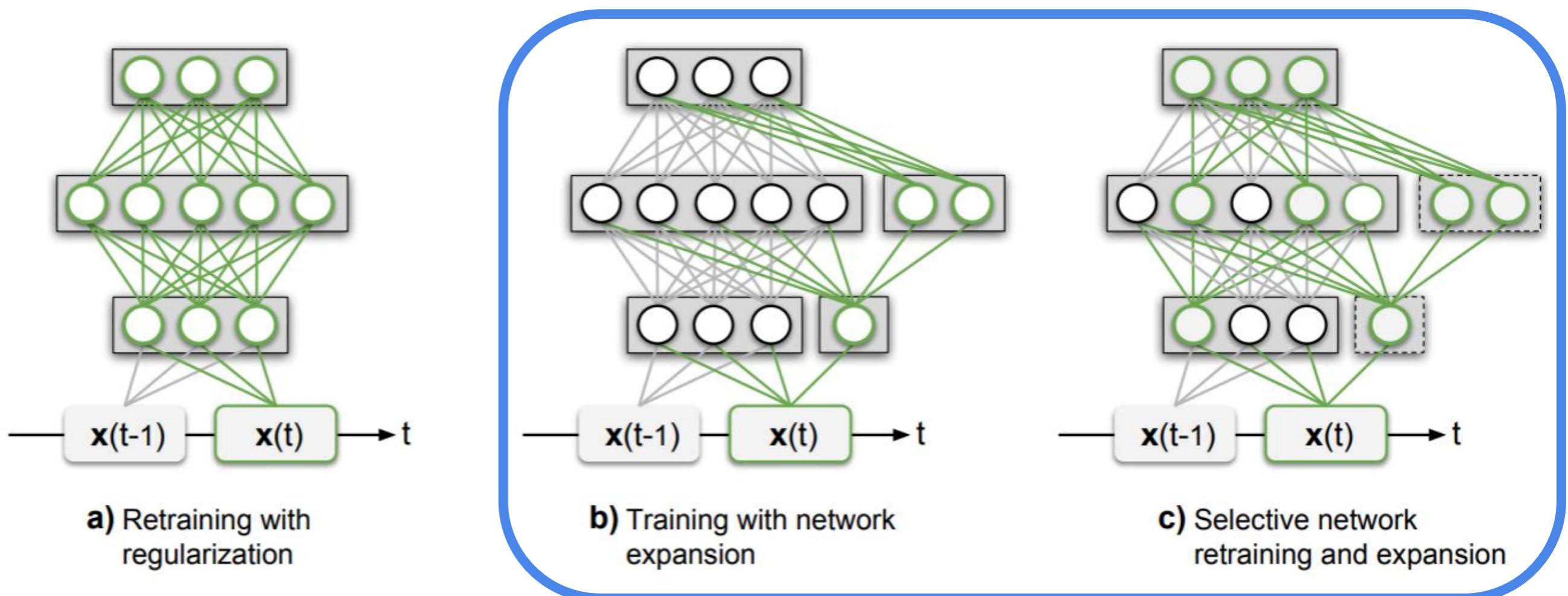


Figure 1: Illustration of the Distral framework.

Dynamic architecture approaches

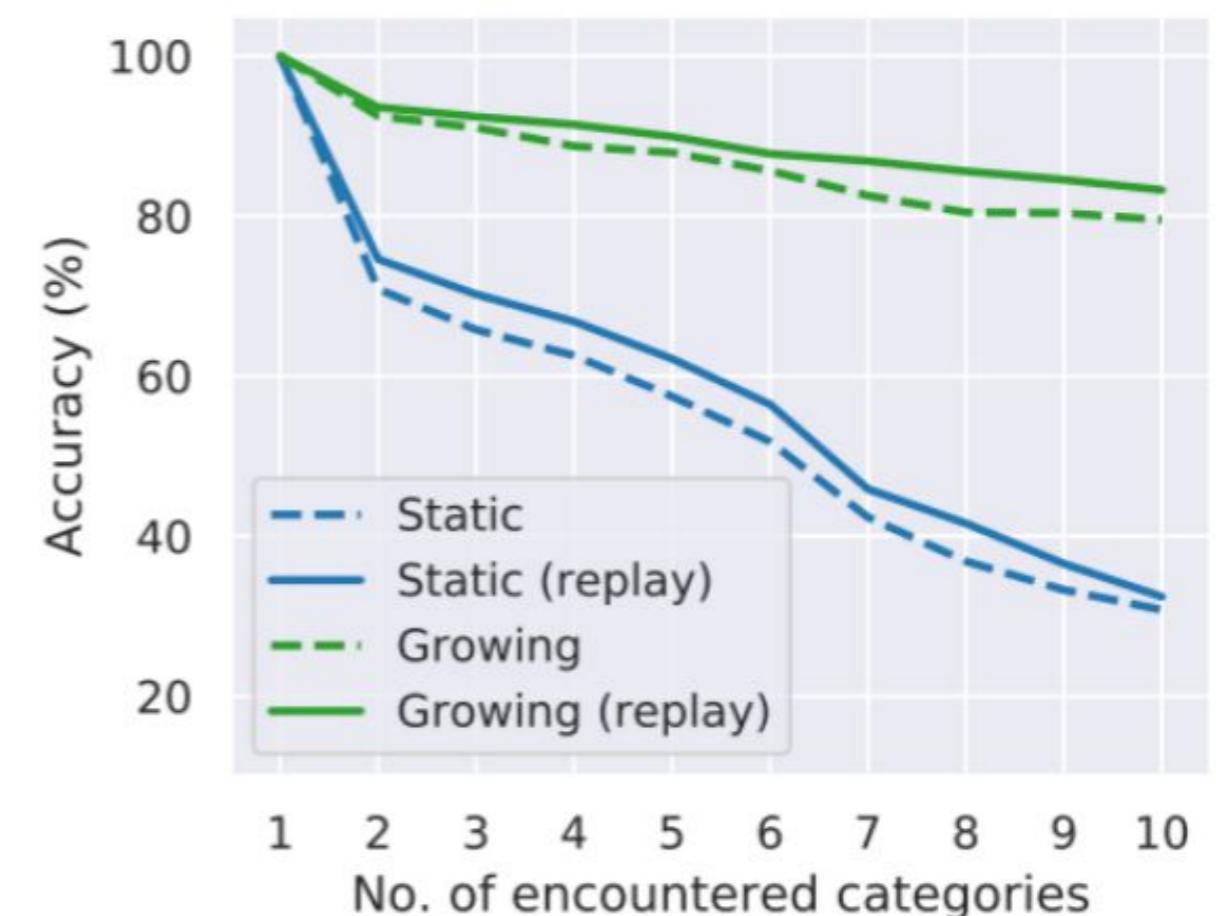
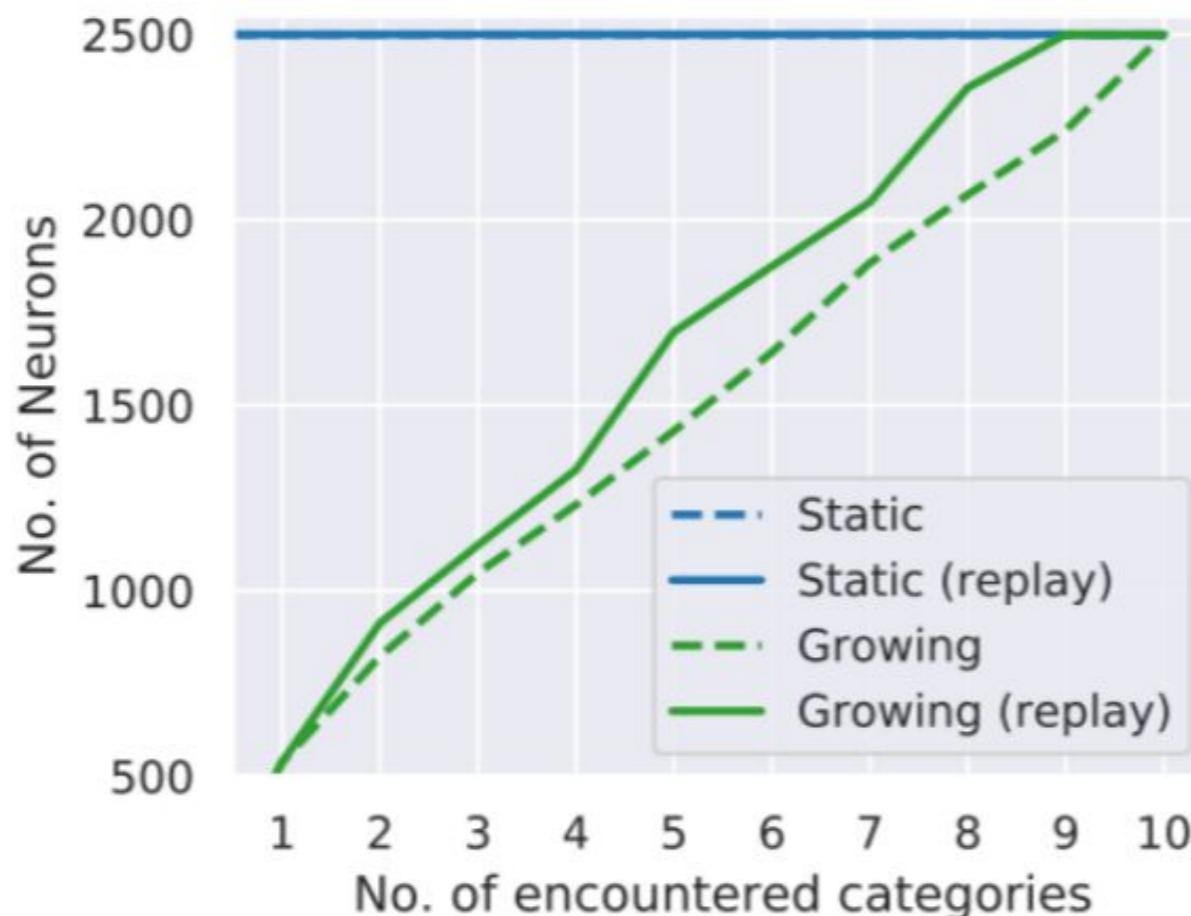


Dynamic Architectures



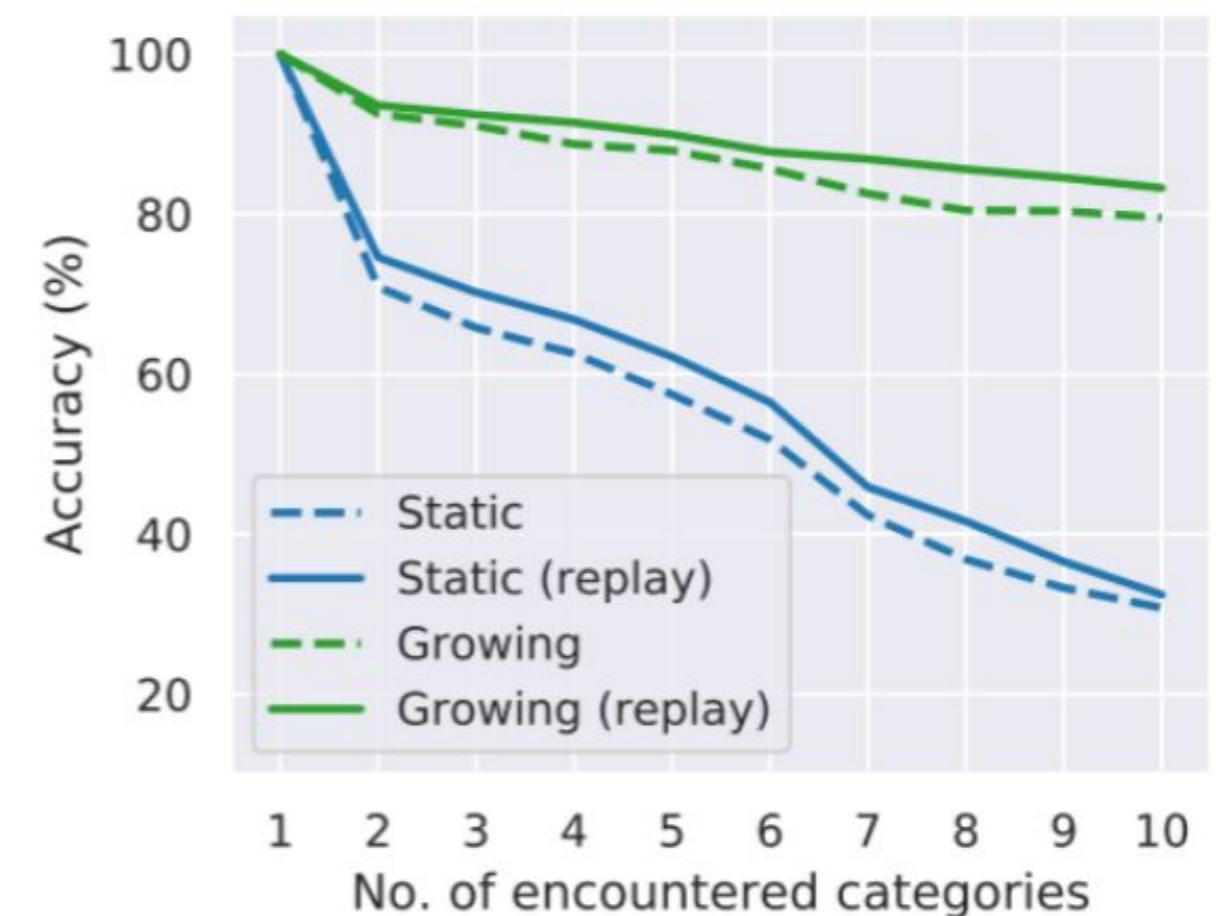
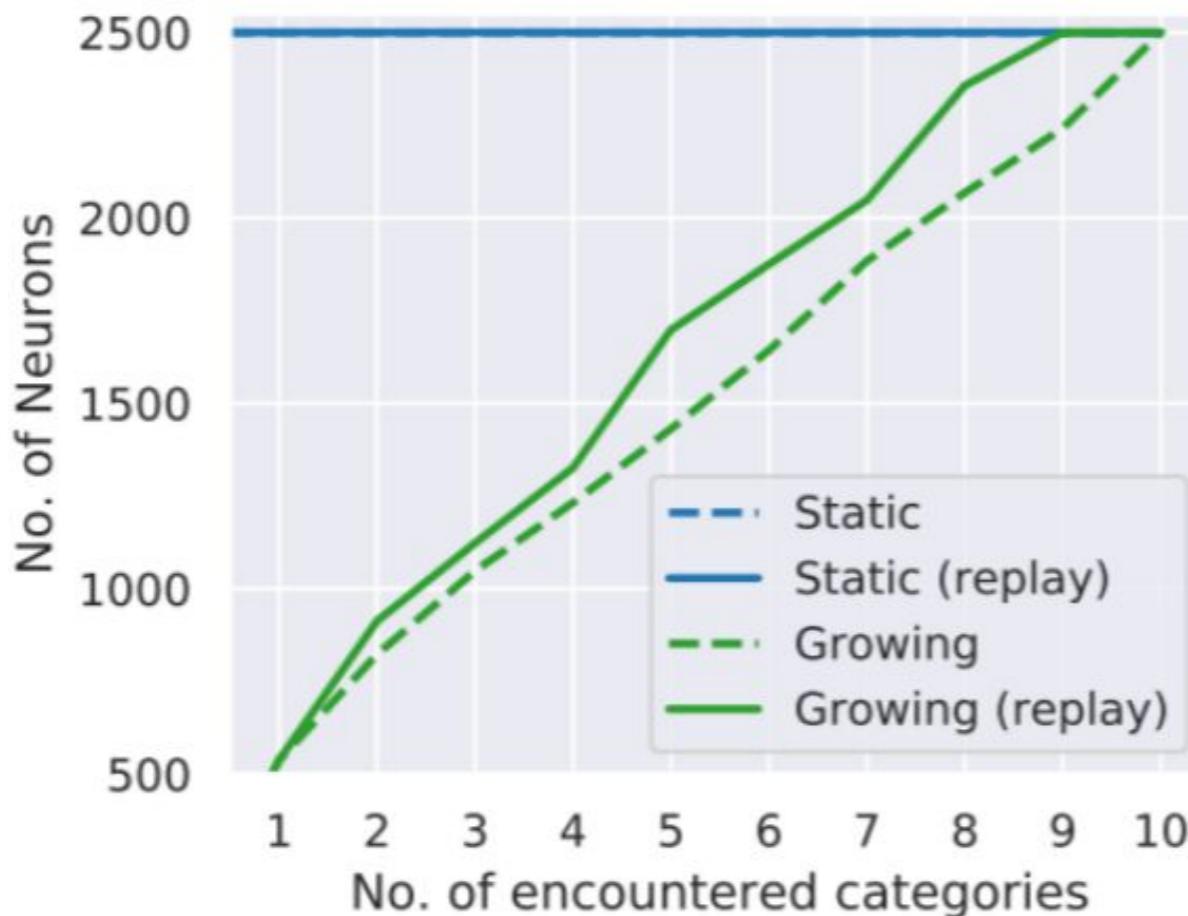
Dynamic Architectures

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

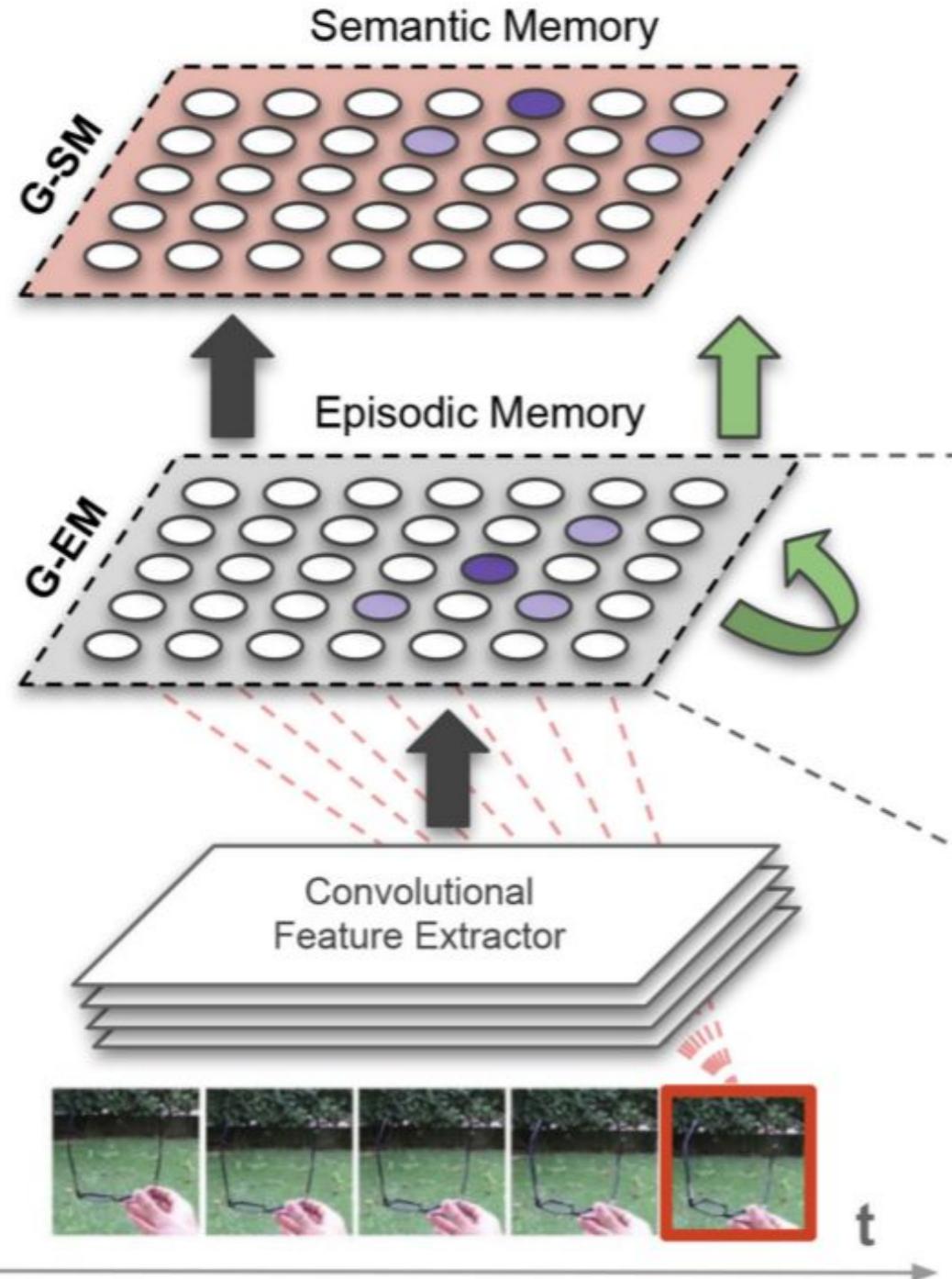


Dynamic Architectures

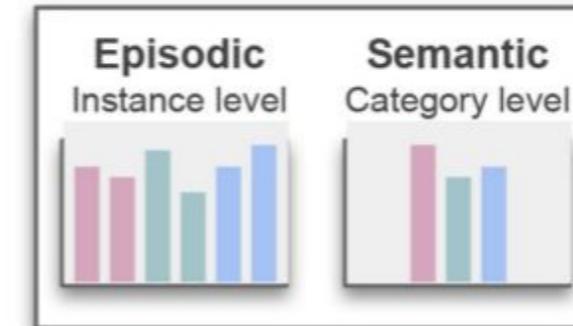
- Outperform static nets in CL settings (under fix memory budget)
 - Progressive NN [Rusu'16], Progress & Compress [Schwarz'18], LwF [Li'16], A-LTM [Furlanello'16], [Li'19]
- **Neurogenesis:** structural plasticity prevents CF in non-stationary environments. GWR (Grow-When-Required) [Parisi'17]



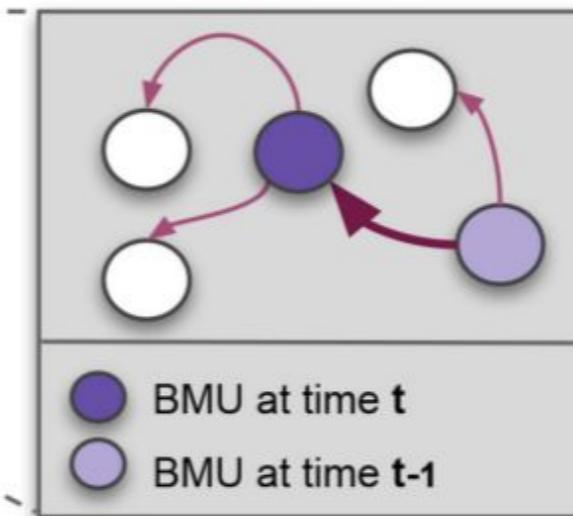
Dual Memories



B Classification



C Temporal synapses



- Input-driven feedforward
- Memory replay
- ↔ Unsupervised learning
- ↔ Regulated learning

Hippocampus

Episodic Memory

Fast learning of arbitrary information

Storage, retrieval, replay

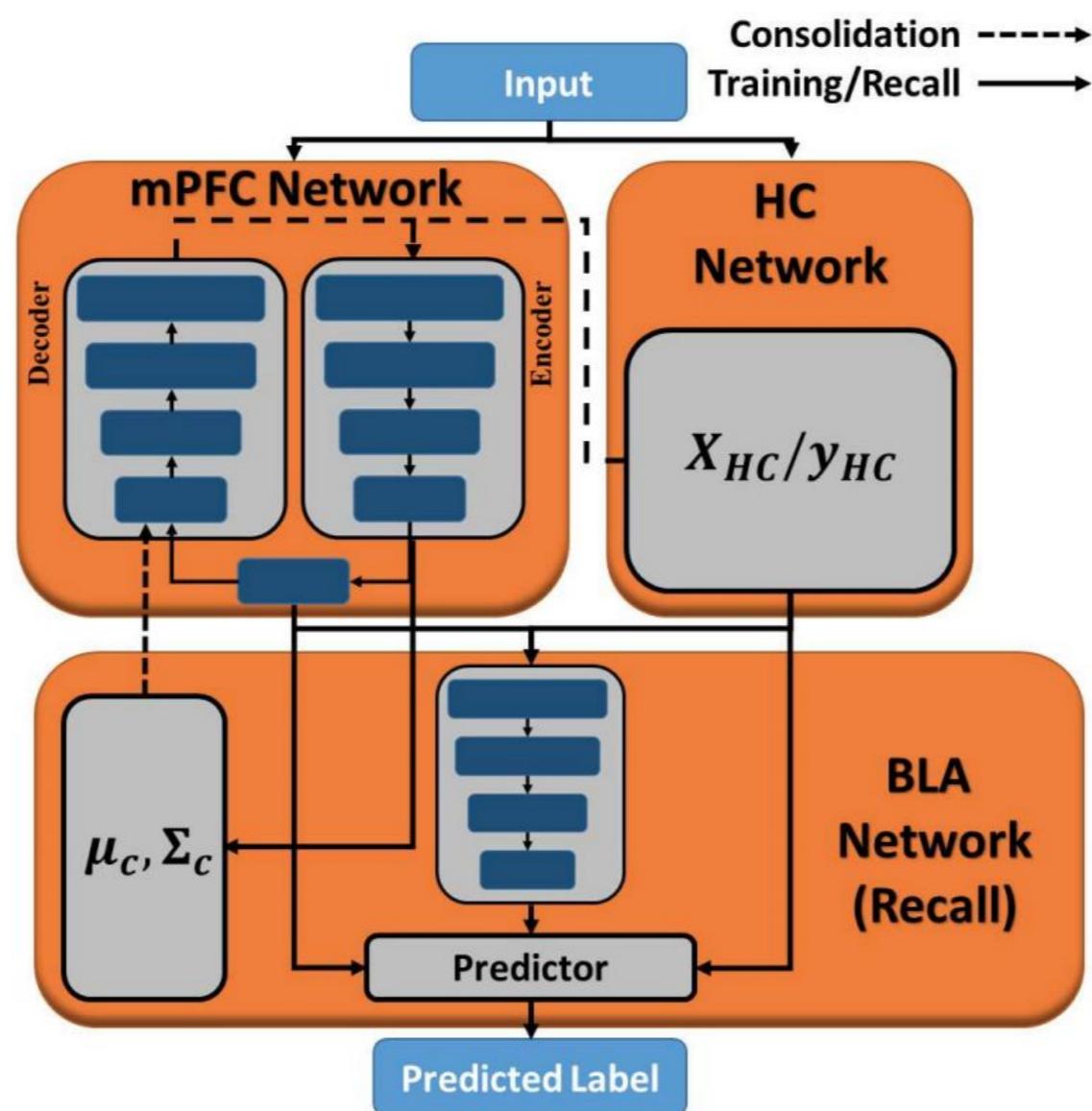
Neocortex

Generalization

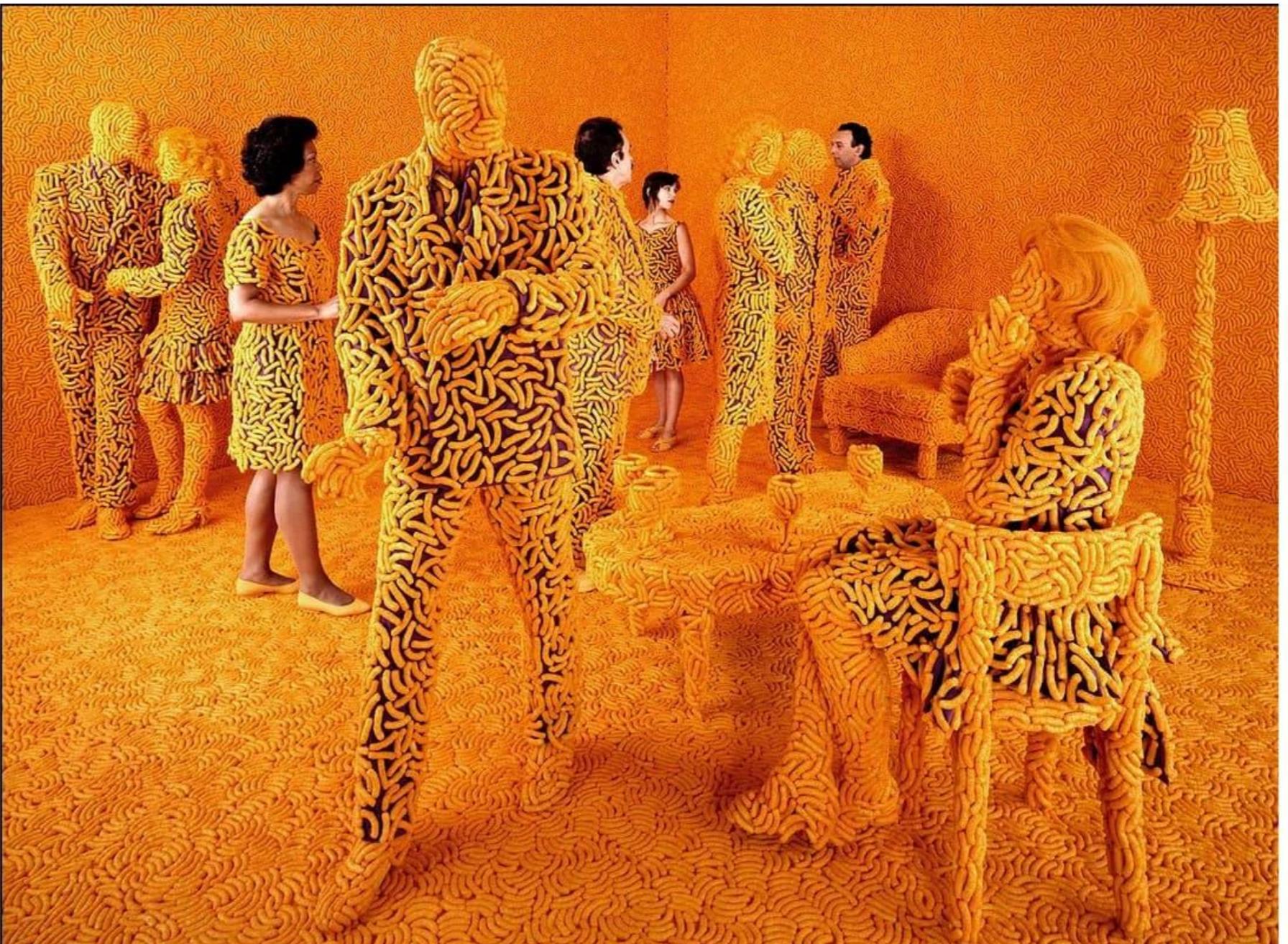
Slow learning of structured knowledge

CLS Theory [McClelland'95, Parisi'18]

Dynamic architectures: Extension of dual architecture



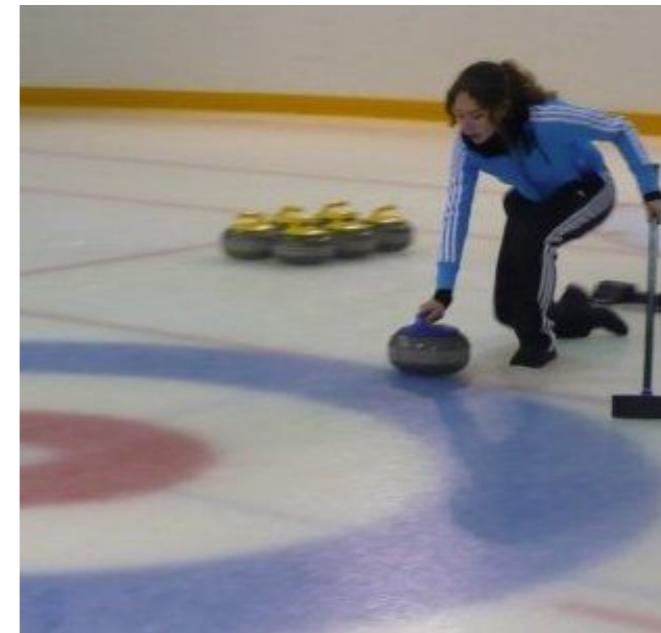
Replay Approaches



Replay Approaches

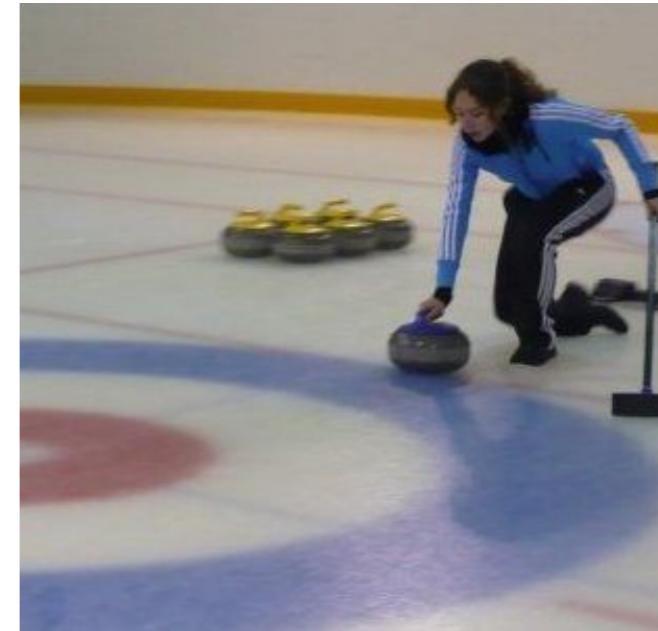
[Robins'95]

- **Rehearsal:** “Buying souvenirs”



Replay Approaches

[Robins'95]



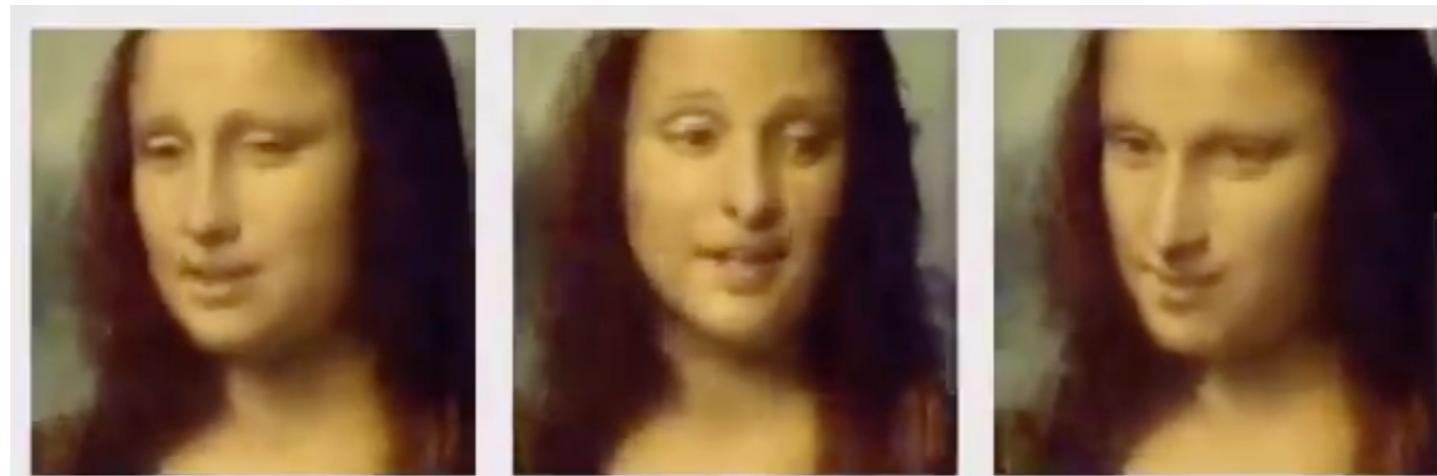
- **Rehearsal:** “Buying souvenirs”
 - Learn task A
 - Fine-tune params with batches from task A + task B uniformly e.g., ICARL [Rebuffi'16]



Replay Approaches

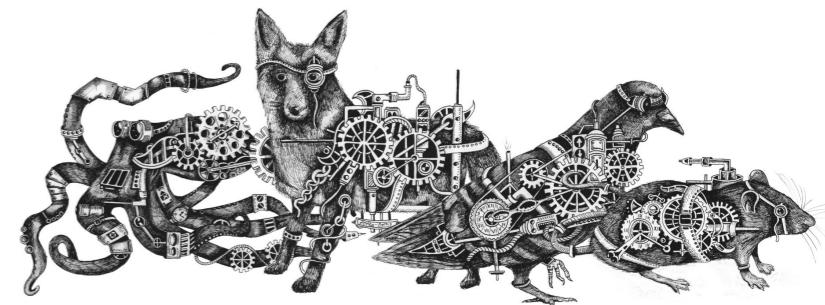
[Robins'95]

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



Continual Learning Benchmarks

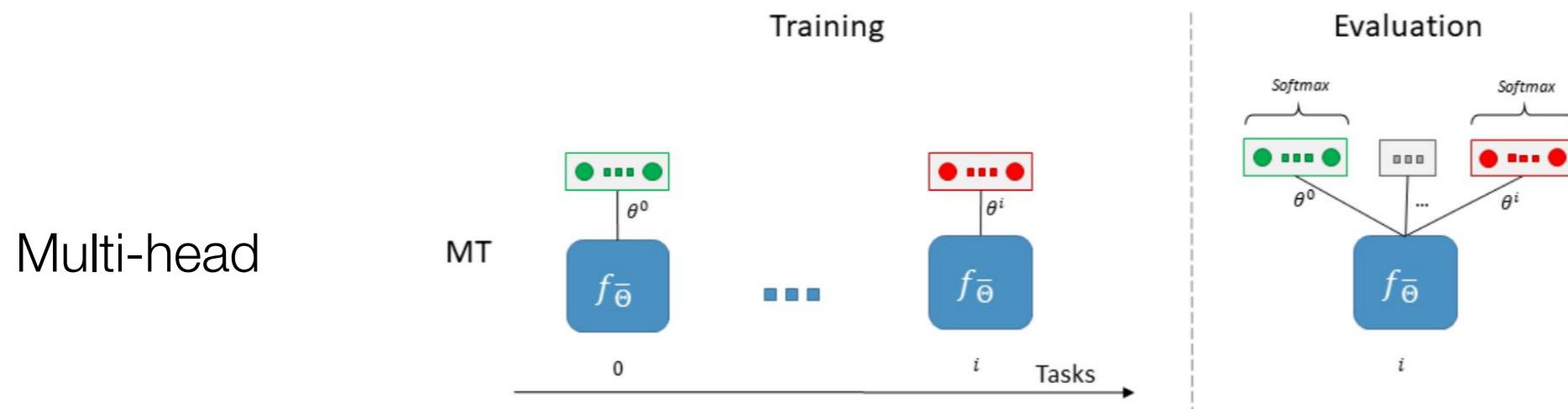


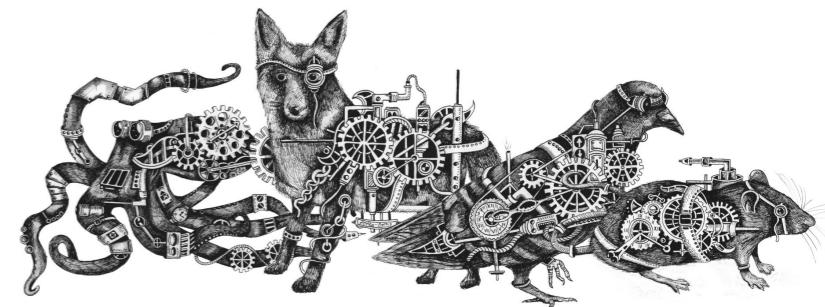


Evaluation Setups

[Chaudhry18]

- **Multi-head** predictions (requires task label to select output subset)
-

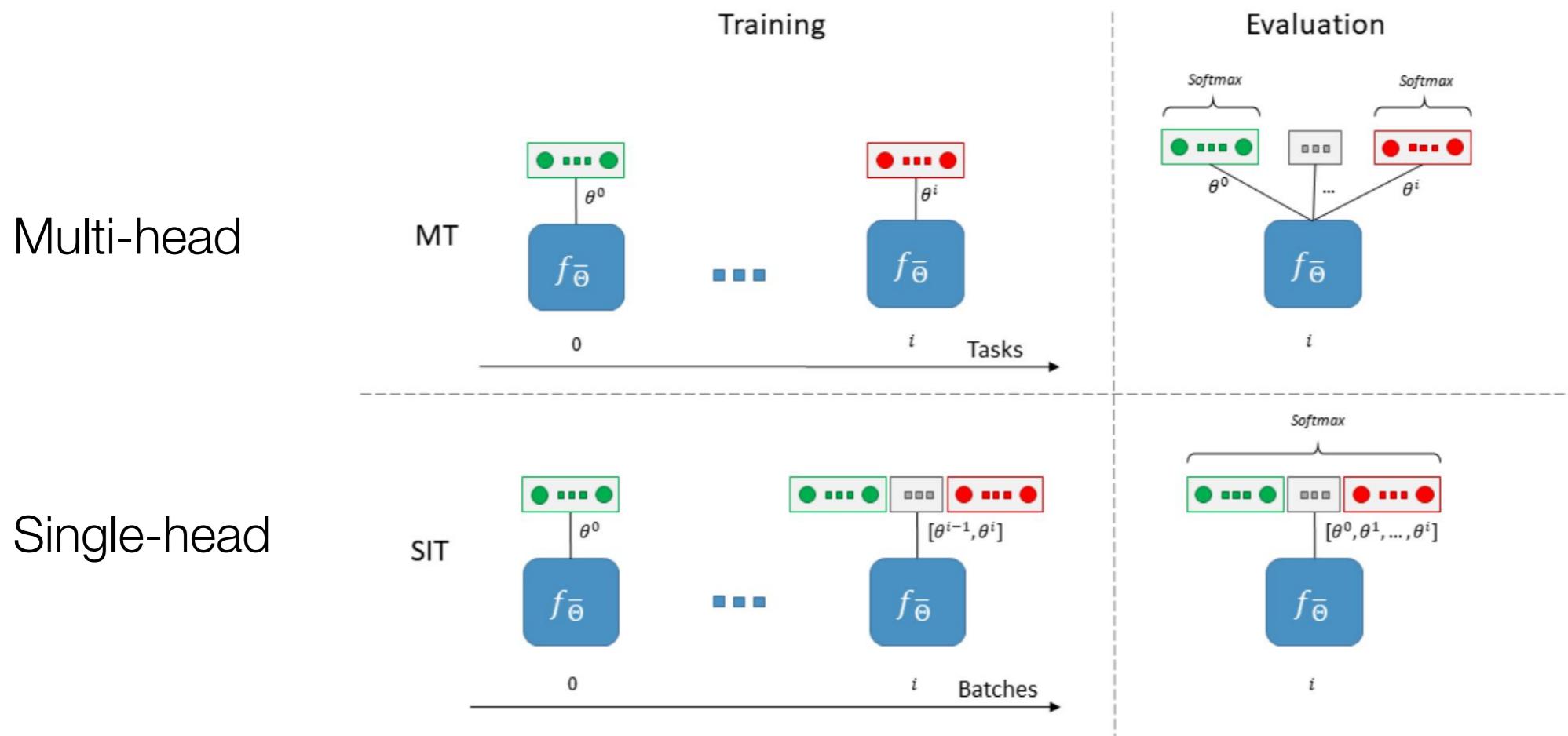




Evaluation Setups

[Chaudhry18]

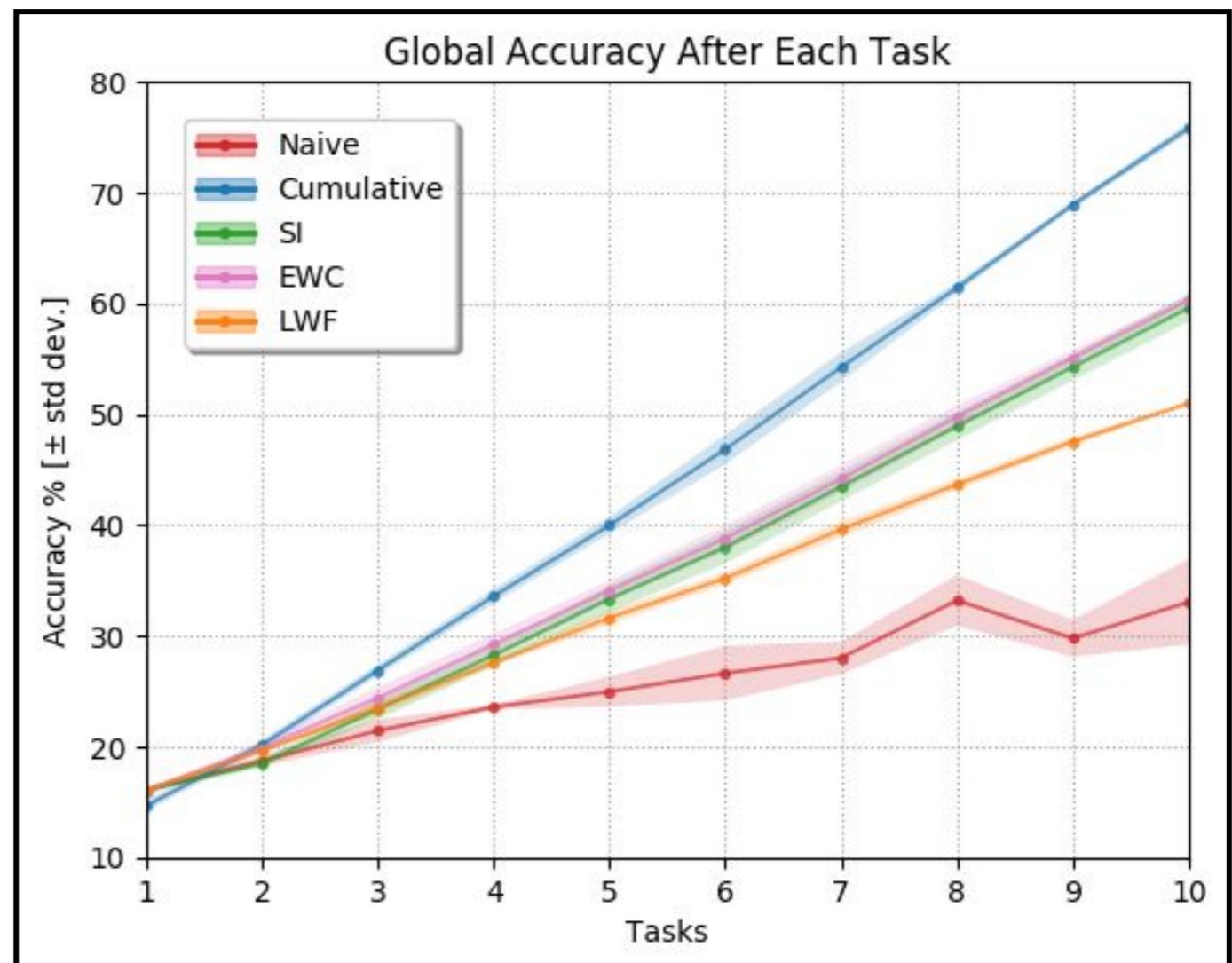
- **Multi-head** predictions (requires task label to select output subset)
- **Single-head** evaluation: 1 output space over **all** tasks labels (+ realistic & difficult -requires dynamic expansion of output layer-)



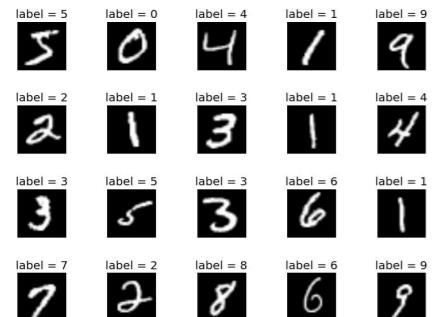
Reasonable Baselines

[Lomonaco & Maltoni'18]

- **Naïve** strategy (Lower bound): fine-tuning
- **Cumulative** strategy (Upper bound): from scratch each new T_{r_i}



Evaluation Benchmarks



MNIST Dataset [LeCun & Cortes'98]



CORe50 Dataset [Lomonaco & Maltoni'18]

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



NICO (Non-I.I.D. Image dataset with Contexts) [He'19]



Fashion MNIST [Xiao'17]

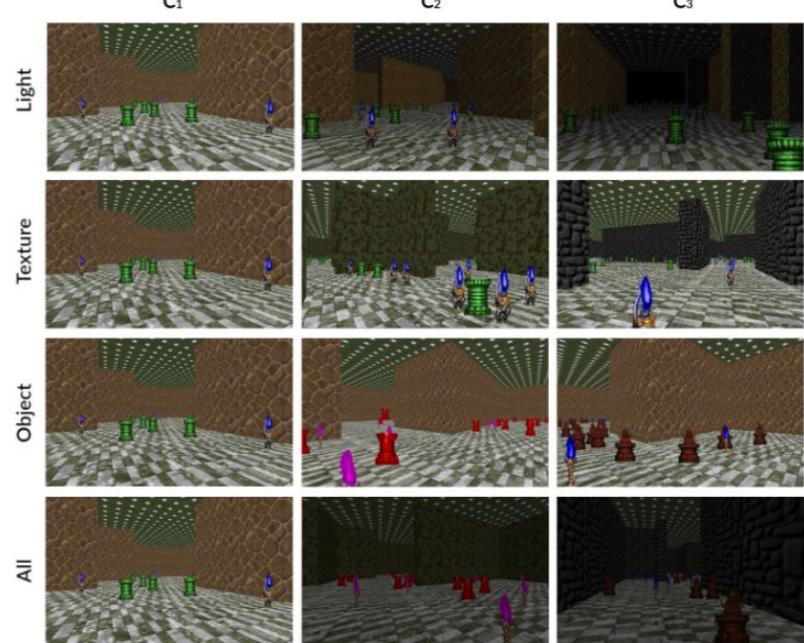
Evaluation Environments



CRLMaze [Lomonaco'19], ViZDoom

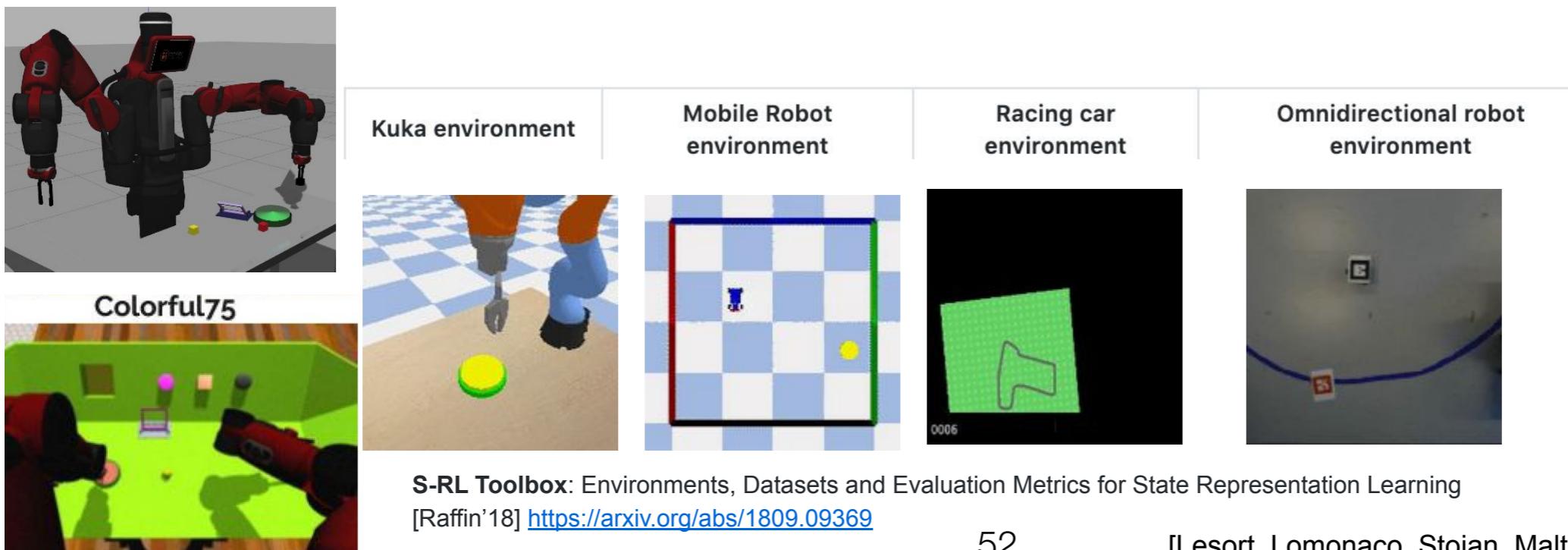
- **RL:** Atari (2D), ViZDoom & CRLMaze (3D), DeepMind Lab, Malmo, OpenAI Gym, MuJoCo, Unity 3D, StarCraft II

Evaluation Environments



CRLMaze [Lomonaco'19], ViZDoom

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



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:

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

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]:

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

- ▶ **Remembering (REM):**

$$REM = 1 - |min(BWT, 0)| \quad (4)$$

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

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

$$BWT^+ = max(BWT, 0) \quad (5)$$

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

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

Continual Learning Metrics

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

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

- ▶ **Samples storage size efficiency (SSS)**: The memory occupation in bits by the samples storage memory M , $\text{Mem}(M)$, 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}):

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

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

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

Continual Learning Metrics

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

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

- ▶ **Samples storage size efficiency (SSS)**: The memory occupation in bits by the samples storage memory M , $\text{Mem}(M)$, 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}):

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

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

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

Continual Learning Metrics

We fuse¹¹ these metrics into a single score:

$$CL_{score} = \sum_{i=1}^{\#C} w_i c_i \quad (10)$$

where

- $c_i \in [0, 1]$: avg. of r runs of c_i , assigned a weight $w_i \in [0, 1]$ s.t. $\sum_i^C w_i = 1$
- As each c_i , the final CL_{score} :
 - $\in [0, 1]$
 - is to be maximized
 - can rank CL strategies

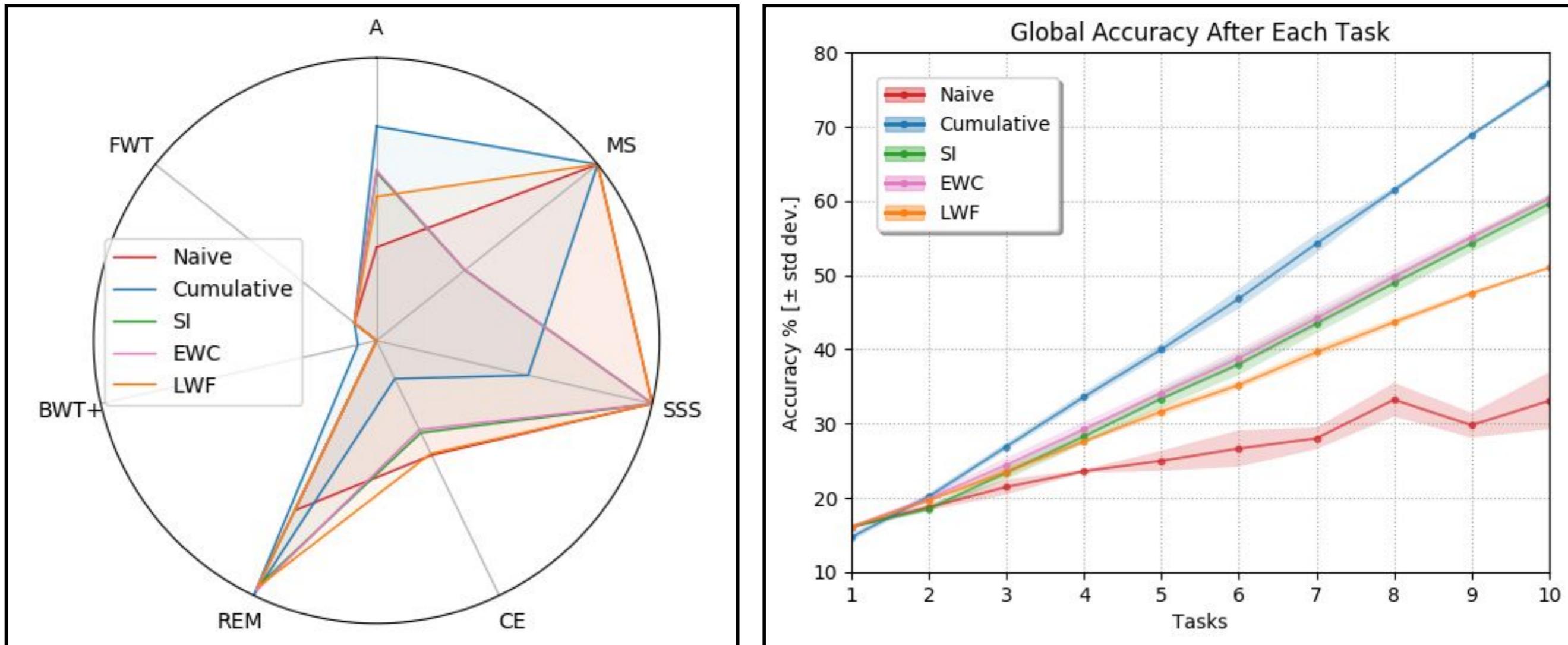
Continual Learning Metrics

The mean std. deviation from all previous criteria c_i :

$$CL_{stability} = 1 - \sum_{i=1}^{\#C} w_i \sigma_{c_i} \quad (11)$$

- $c_i \in [0, 1]$: avg. of r runs assigned a weight $w_i \in [0, 1]$ s.t. $\sum_i^C w_i = 1$
- σ_{c_i} : std. deviation of criterion c_i

Continual Learning Metrics



Continual Learning Metrics

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

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^*

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

$$REM = 1 - |min(BWT, 0)|$$

$$BWT^+ = max(BWT, 0)$$

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

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

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

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

[Lopez-Paz & Ranzato'17]
[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting:
60 new metrics for Continual Learning. CL Workshop, NeurIPS 2018].
[Lesort, Lomonaco, Stoian, Maltoni, Filliat, Diaz-Rodriguez. 19]

Continual Learning Metrics

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

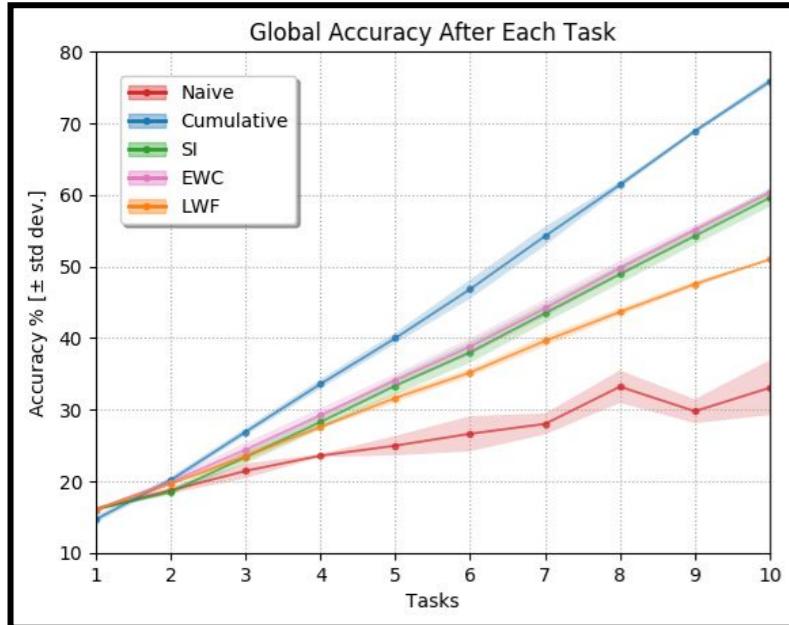
$$SSS = 1 - \min(1, \frac{\sum_{i=1}^N \frac{Mem(M_i)}{Mem(D)}}{N})$$

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

[Lopez-Paz & Ranzato'17]

[Díaz-Rodríguez, Lomonaco et al. Don't forget, there is more than forgetting:
61 new metrics for Continual Learning. CL Workshop, NeurIPS 2018].
[Lesort, Lomonaco, Stoian, Maltoni, Filliat, Diaz-Rodriguez. 19]

Continual Learning Metrics



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

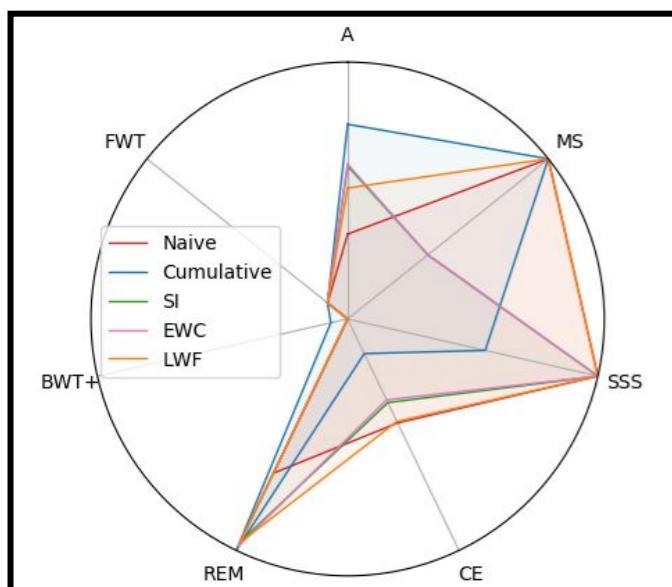
$$MS = \min(1, \frac{\sum_{i=1}^N \frac{\text{Mem}(\theta_1)}{\text{Mem}(\theta_i)}}{N}) \quad (7)$$

- **Samples storage size efficiency (SSS)**: The memory occupation in bits by the samples storage memory M , $\text{Mem}(M)$, 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}):

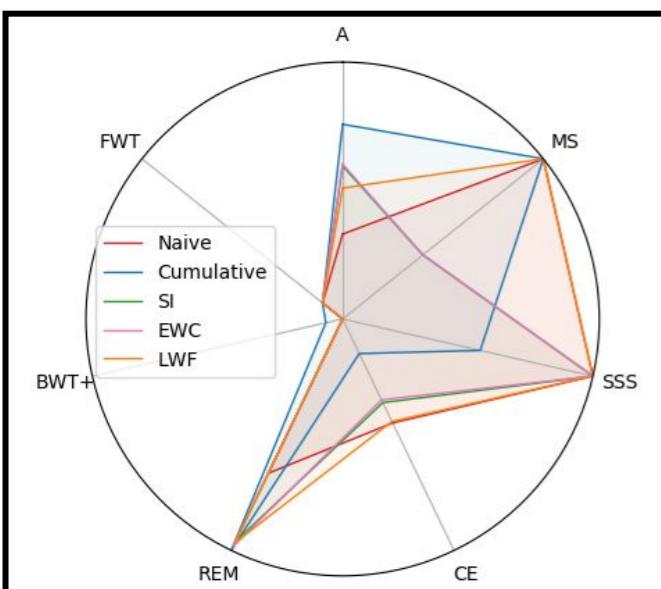
$$SSS = 1 - \min(1, \frac{\sum_{i=1}^N \frac{\text{Mem}(M_i)}{\text{Mem}(D)}}{N}) \quad (8)$$

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

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



Continual Learning Metrics



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

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

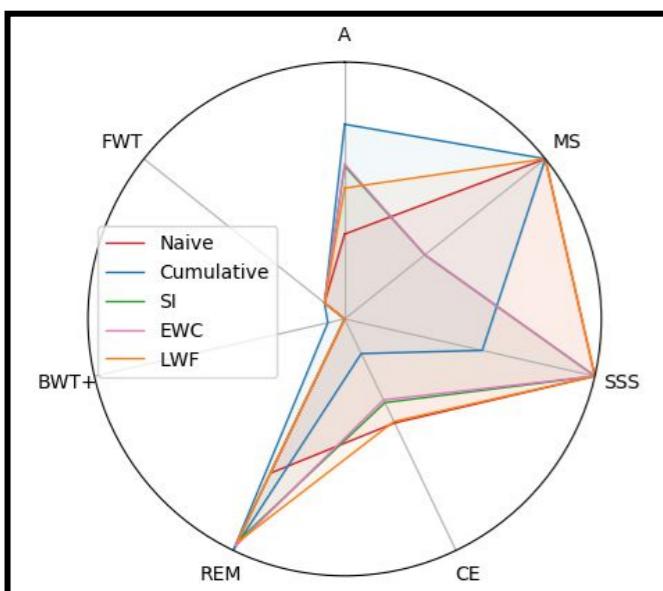
- **Samples storage size efficiency (SSS)**: The memory occupation in bits by the samples storage memory M , $\text{Mem}(M)$, should be bounded by the memory occupation of the total nr of examples encountered at the end of the last task (D is the lifetime dataset associated to all distributions \mathcal{D}):

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

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

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

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, $Mem(\theta_1)$.

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

- **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}):

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

- **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 .

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

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

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 examples⁴:* $\forall i \in [1, \dots, n], |M_i| \ll \left| \bigcup_{i=1}^{i-1} Tr_i \right|$

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

Constraints, Desiderata, Recommendations



CL Recommendations



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

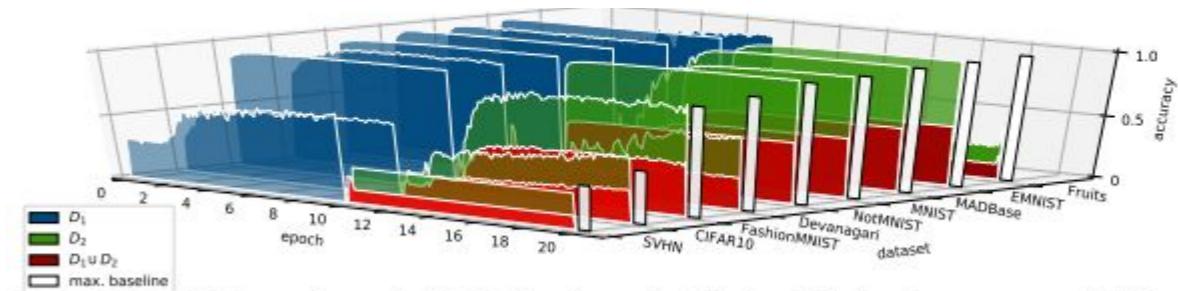


Figure 3: Best D-FC experiments for SLT D5-5, to be read as Fig. 2 and showing the occurrence of CF. See also Appendix B for 2D plots.

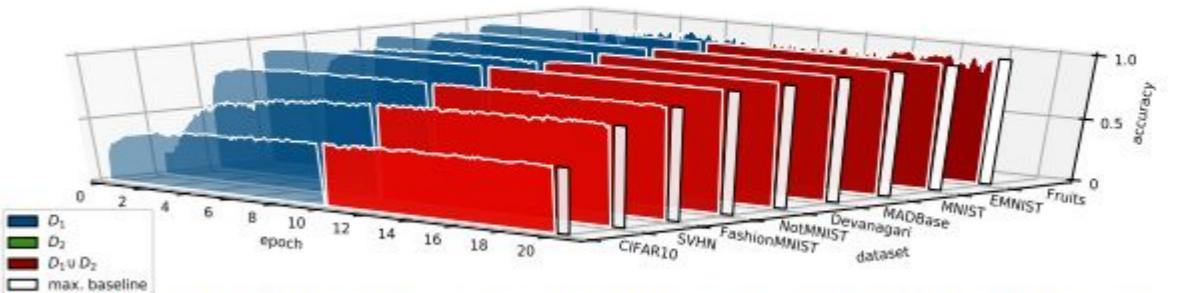


Figure 2: Best FC experiments for DP10-10. The blue surfaces (epochs 0-10) represent the accuracy on D_1 , the green (covered here) and red surfaces the accuracy on D_2 and $D_1 \cup D_2$ during re-training (epochs 10-20). The white bars indicate baseline performance. See also Appendix B for 2D plots.

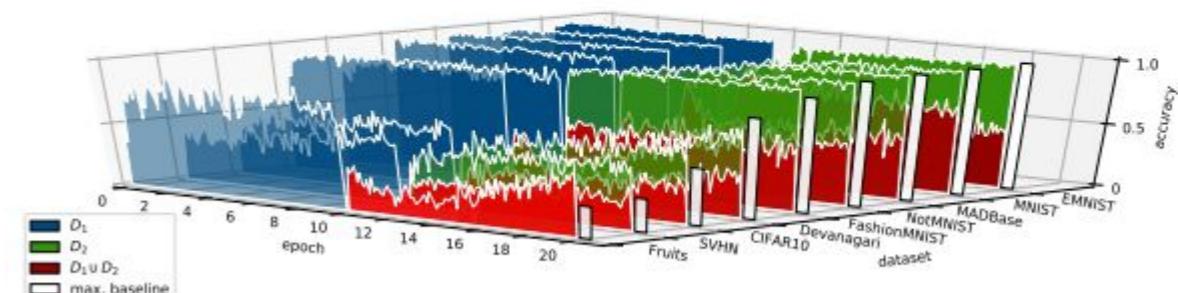


Figure 5: Best EWC experiments for SLT D5-5d constructed from all datasets, to be read as Fig. 2. We observe that CF happens for all datasets. See also Appendix B for 2D plots.

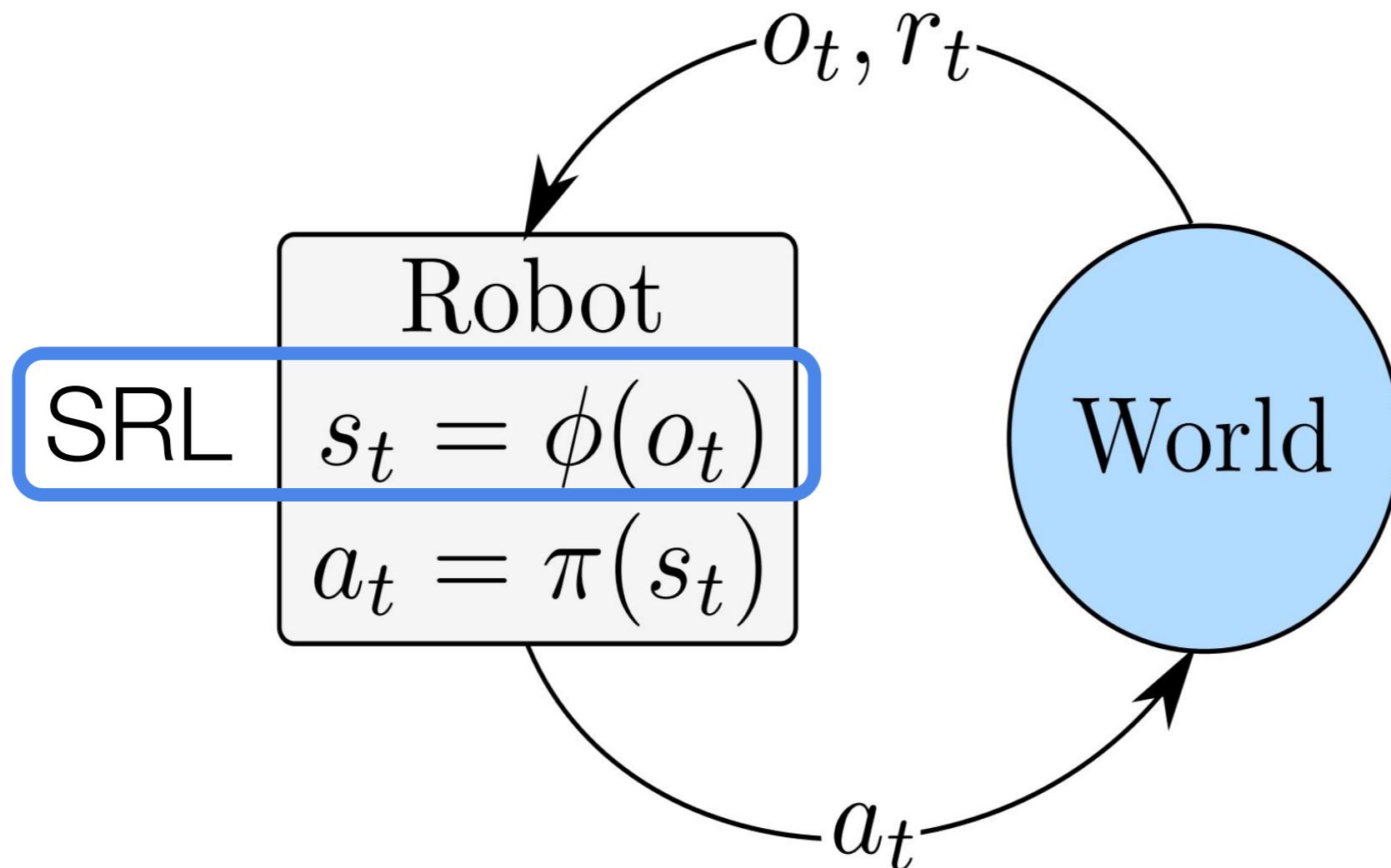
CL Recommendations

- On-line capabilities: a feature of CL models
- Do not assume:
 - **#tasks** is given
 - Unlimited:
 - Memory
 - Computation
 - Data storage

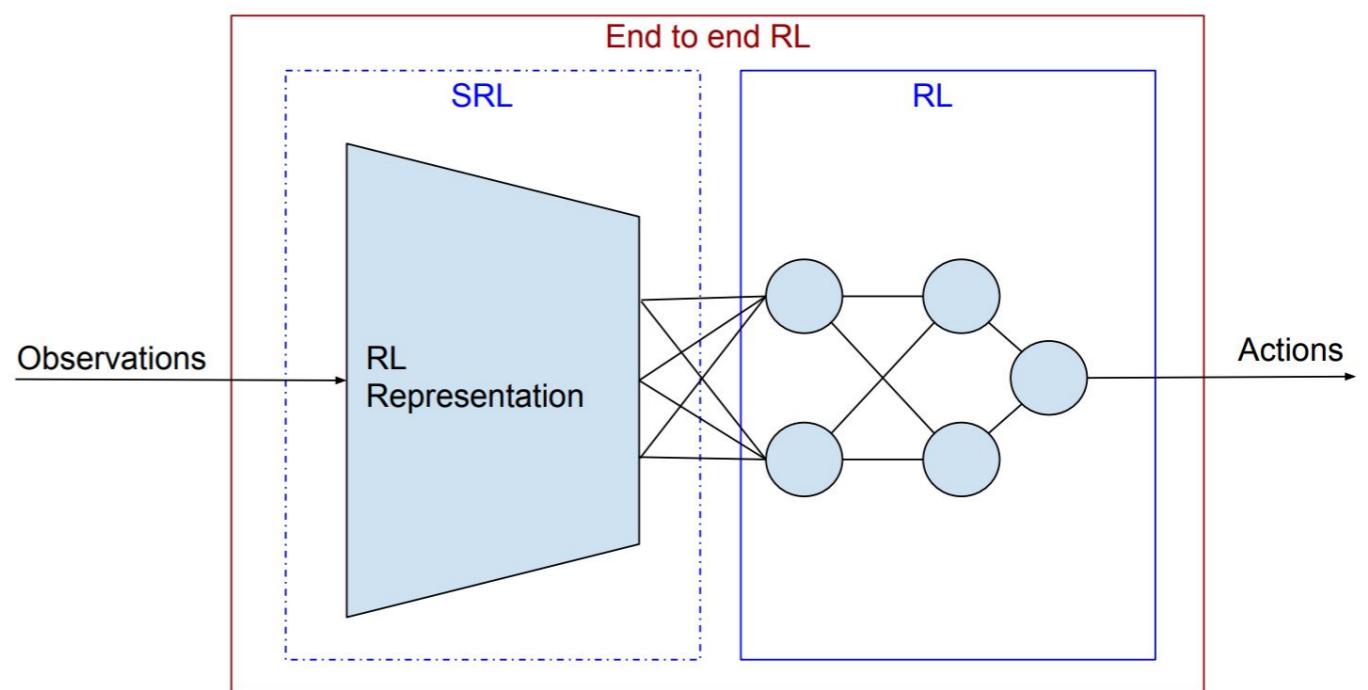
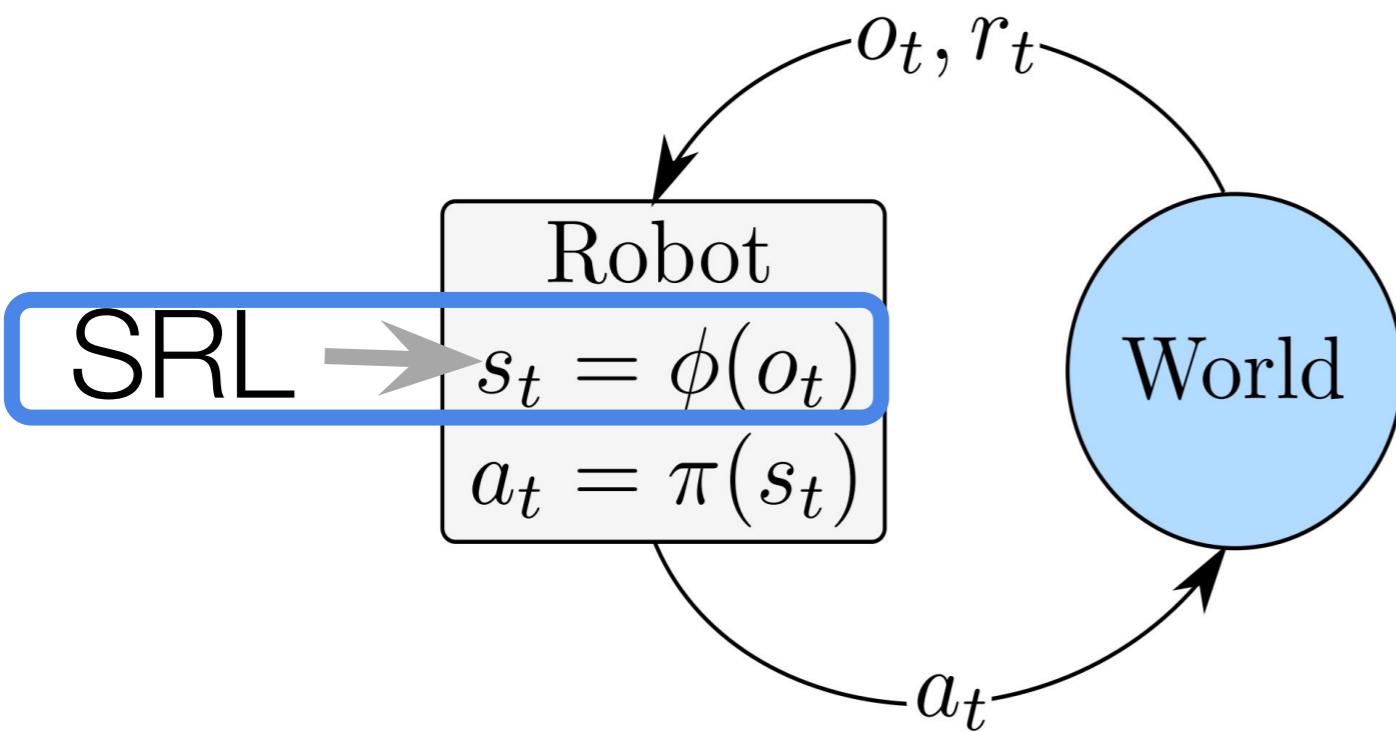
Example:
Continual
State
Representation
Learning (SRL)



State Representation Learning (SRL) in RL context

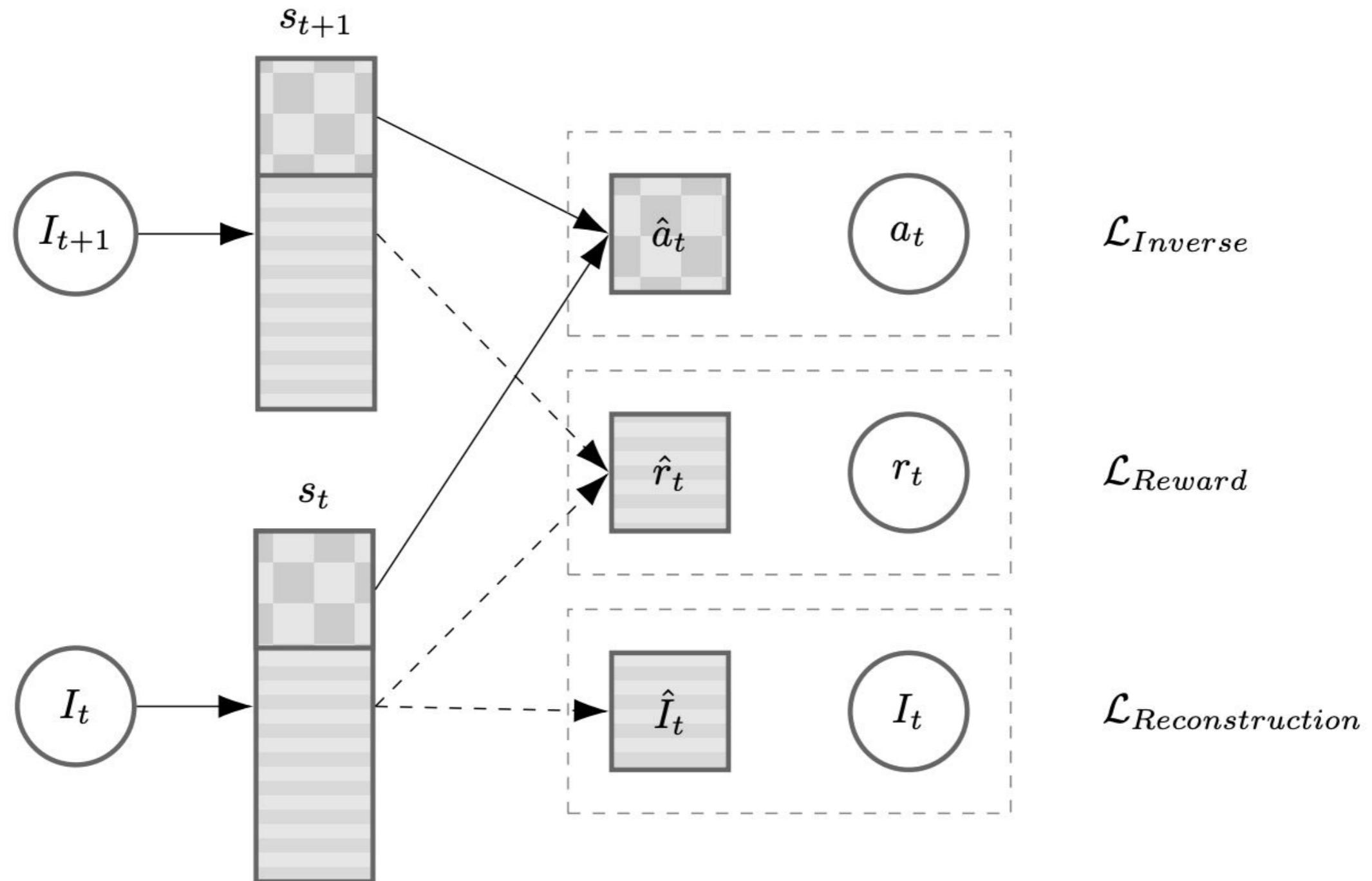


State Representation Learning (SRL) in RL context



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

Learning 1 task: Decoupling feature extraction from policy learning



Name	Action space (discrete)	Action space (continuous)	Rewards	ground truth
Kuka Button	6 actions (3D cardinal direction)	3 axis (3D cardinal direction) ⁽¹⁾	1 when target reached, -1 when too far from target or when table is hit, otherwise 0 ^{(2) (3)}	the X,Y,Z position of the effector (4)
Kuka RandButton	6 actions (3D cardinal direction)	3 axis (3D cardinal direction) ⁽¹⁾	1 when target reached, -1 when too far from target or when table is hit, otherwise 0 ^{(2) (3)}	the X,Y,Z position of the effector (4)
Kuka 2Button	6 actions (3D cardinal direction)	3 axis (3D cardinal direction) ⁽¹⁾	1 when the first target is reached, 1 when the second target is reached, -1 when too far from target or when table is hit, otherwise 0 ⁽²⁾	the X,Y,Z position of the effector (4)
Kuka MovingButton	6 actions (3D cardinal direction)	3 axis (3D cardinal direction) ⁽¹⁾	1 when target reached, -1 when too far from target or when table is hit, otherwise 0 ^{(2) (3)}	the X,Y,Z position of the effector (4)
MobileRobot	4 actions (2D cardinal direction)	2 axis (2D cardinal direction)	1 when target reached, -1 for a wall hit, otherwise 0 ⁽²⁾	the X,Y position of the robot (4)
MobileRobot 2Target	4 actions (2D cardinal direction)	2 axis (2D cardinal direction)	1 when target reached, -1 for a wall hit, otherwise 0 ⁽²⁾	the X,Y position of the robot (4)
MobileRobot 1D	2 actions (1D cardinal direction)	1 axis (1D cardinal direction)	1 when target reached, -1 for a wall hit, otherwise 0 ⁽²⁾	the X position of the robot (4)
MobileRobot LineTarget	4 actions (2D cardinal direction)	2 axis (2D cardinal direction)	1 when target reached, -1 for a wall hit, otherwise 0 ⁽²⁾	the X,Y position of the robot (4)
CarRacing	4 actions (left, right, accelerate, brake)	3 axis (steering, accelerate, brake)	-100 when out of bounds, otherwise -0.1	the X,Y position of the car (4)
OmniRobot	4 actions (2D cardinal direction)	2 axis (2D cardinal direction)	1 when target reached, -1 for a wall hit, otherwise 0 ⁽²⁾	the X,Y position of the robot (4)

S-RL Toolbox Environments [Raffin'18]

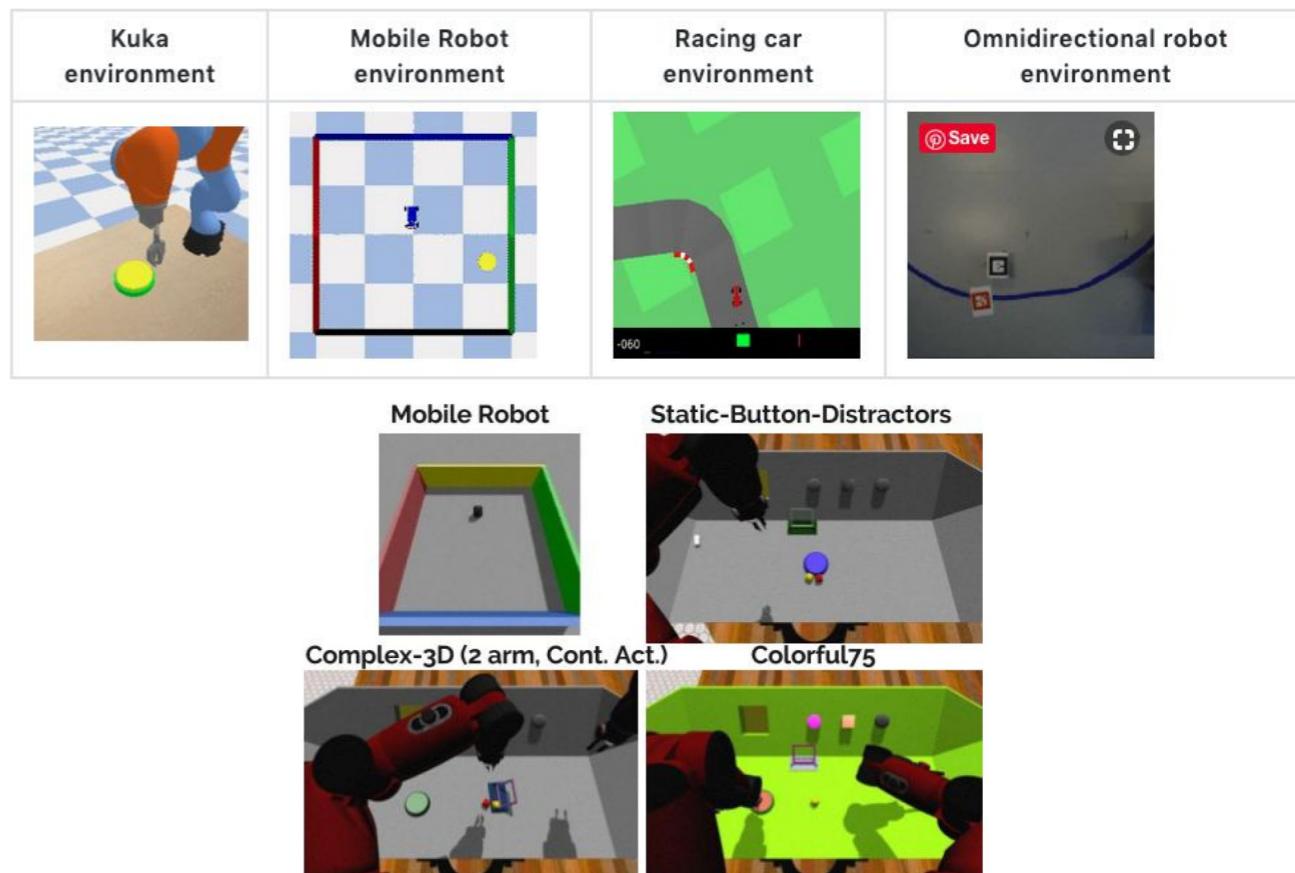


Fig. 3. A sample of each dataset (1-4), created for our benchmark



Decoupling feature extraction from policy learning

- Surpass supervised, reaching GT

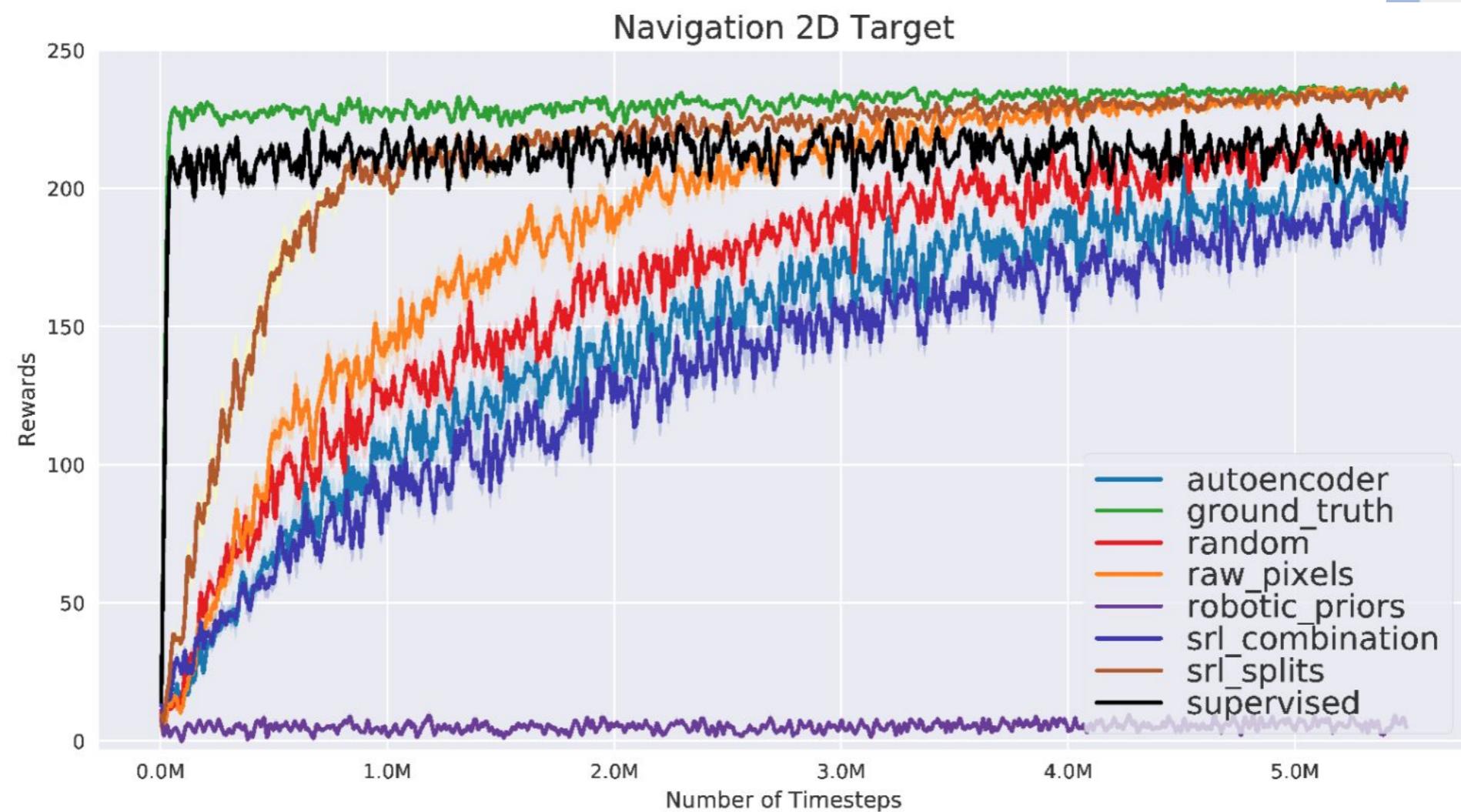
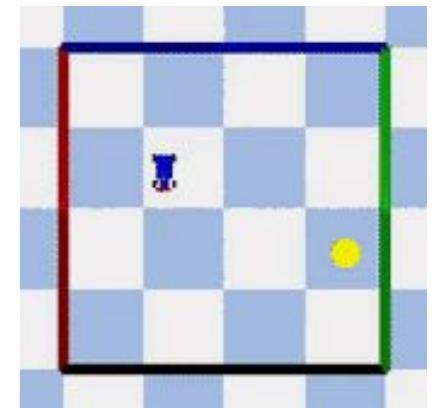


Figure 6: Performance (mean and standard error for 10 runs) for PPO algorithm for different state representations learned in Navigation 2D random target environment.

Decoupling feature extraction from policy learning

- Surpass supervised, reaching GT



Budget (in timesteps)	1 Million	2 Million	3 Million	5 Million
Ground Truth	4.1 ± 0.5	4.1 ± 0.6	4.1 ± 0.6	4.2 ± 0.5
Supervised	4.0 ± 0.3	3.8 ± 0.3	3.4 ± 0.3	3.1 ± 0.3
Raw Pixels	0.6 ± 0.3	0.8 ± 0.3	1.2 ± 0.3	2.6 ± 0.3
Random Features	1.5 ± 0.4	2.8 ± 0.3	3.5 ± 0.3	4.1 ± 0.3
Auto-Encoder	0.92 ± 0.3	1.6 ± 0.3	2.2 ± 0.3	3.4 ± 0.3
SRL Combination	1.0 ± 0.3	1.5 ± 0.3	2.0 ± 0.3	2.9 ± 0.3
SRL Splits	1.1 ± 0.3	2.1 ± 0.3	2.7 ± 0.4	3.7 ± 0.3

Table 3: Mean reward performance in RL (using PPO) per episode (average on 100 episodes) for different budgets, with standard error in robotic arm with random target environment.

Decoupling feature extraction from policy learning

- Surpass supervised, reaching GT
- +Sample efficient & stable

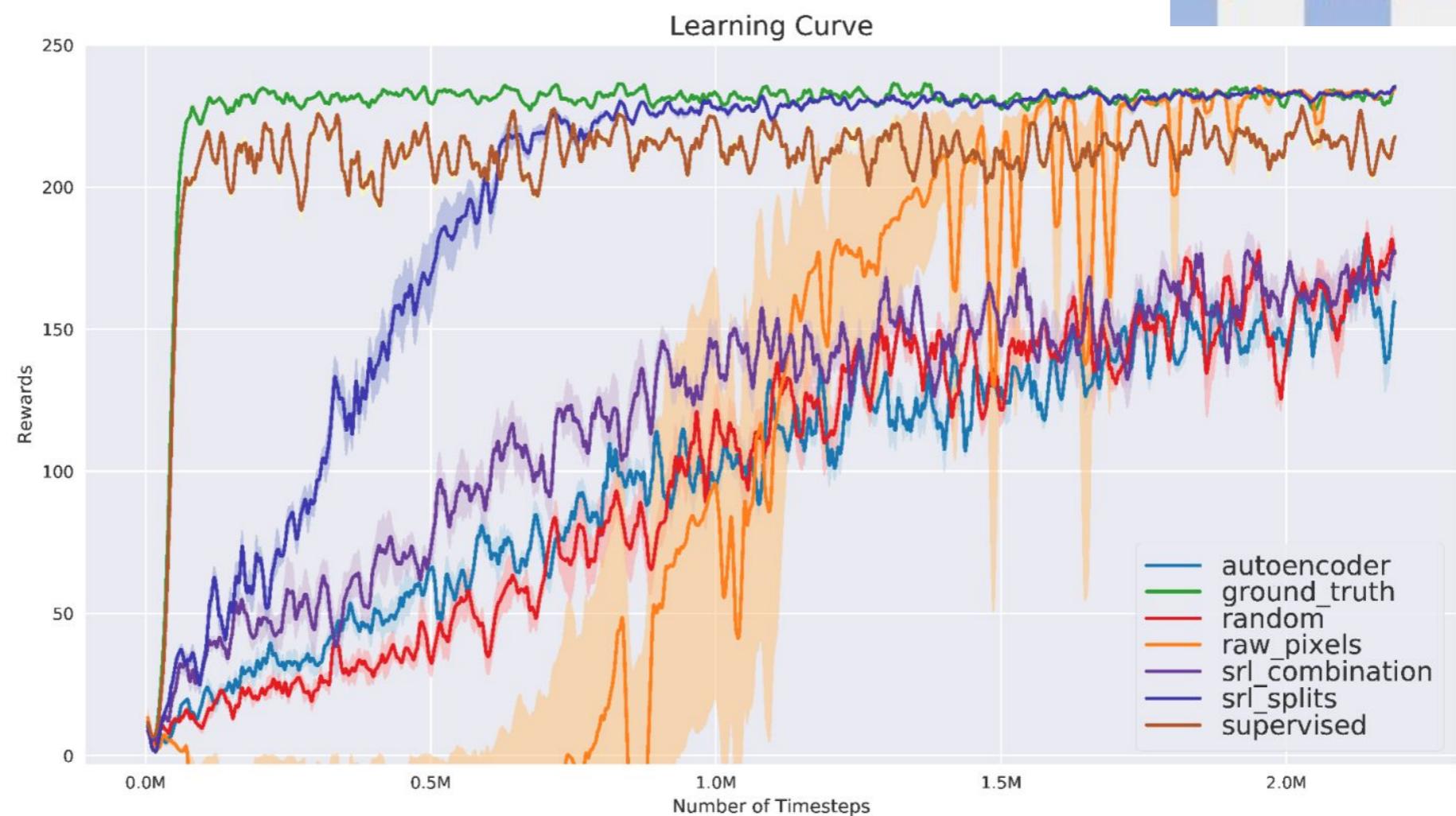
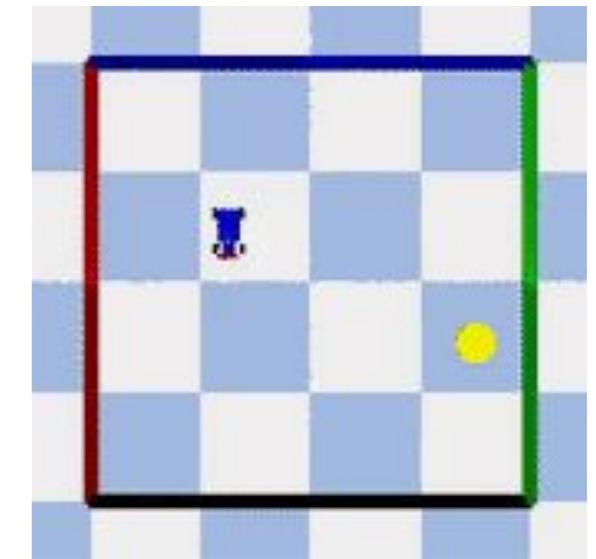
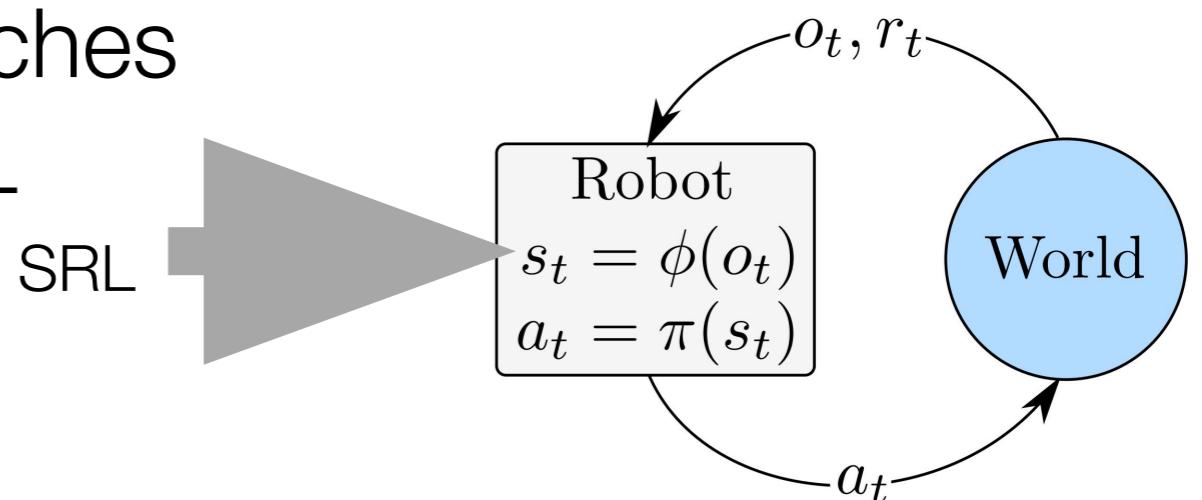


Figure 7: Performance (mean and standard error for 10 runs) for A2C algorithm for different state representations learned in Navigation 2D random target environment.

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



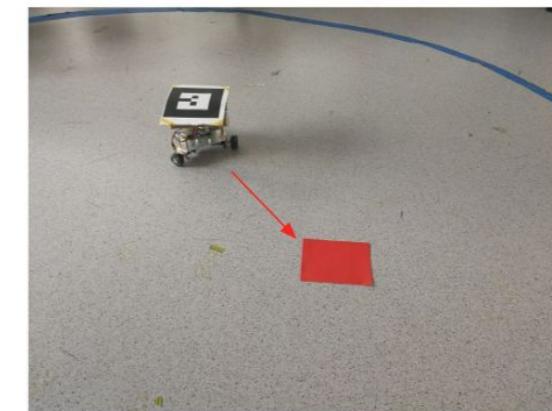
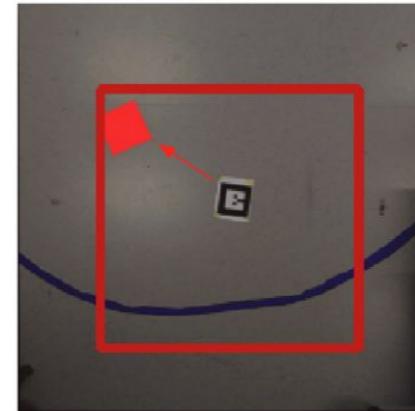
[Lesort'18, State representation learning for control: An overview. Neural Networks]

DREAM EU H2020 Project: www.robotsthatdream.eu

Continual RL in Multi-Task, Lifelong, Real Life Settings

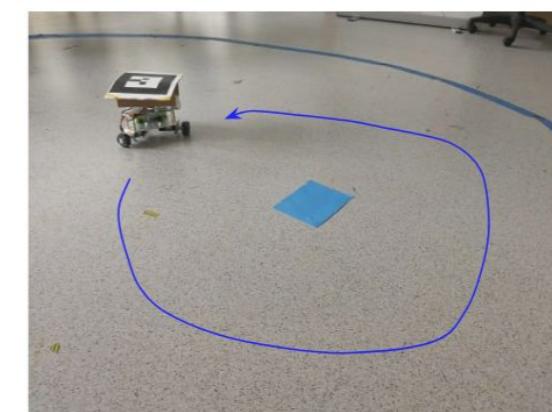
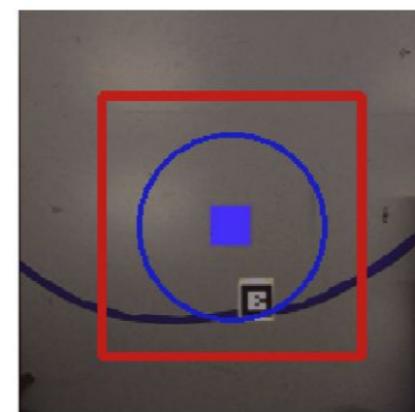


Task 1

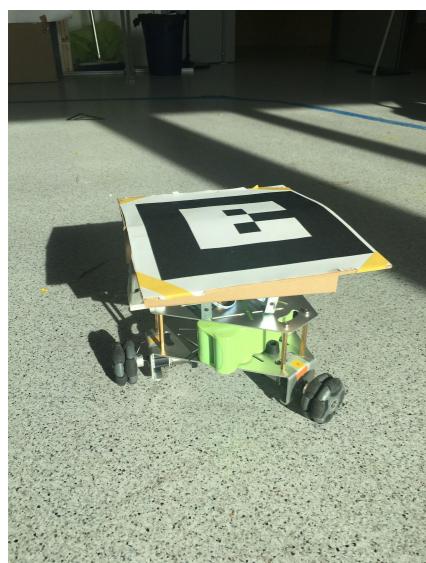


Target
Circling

Task 2



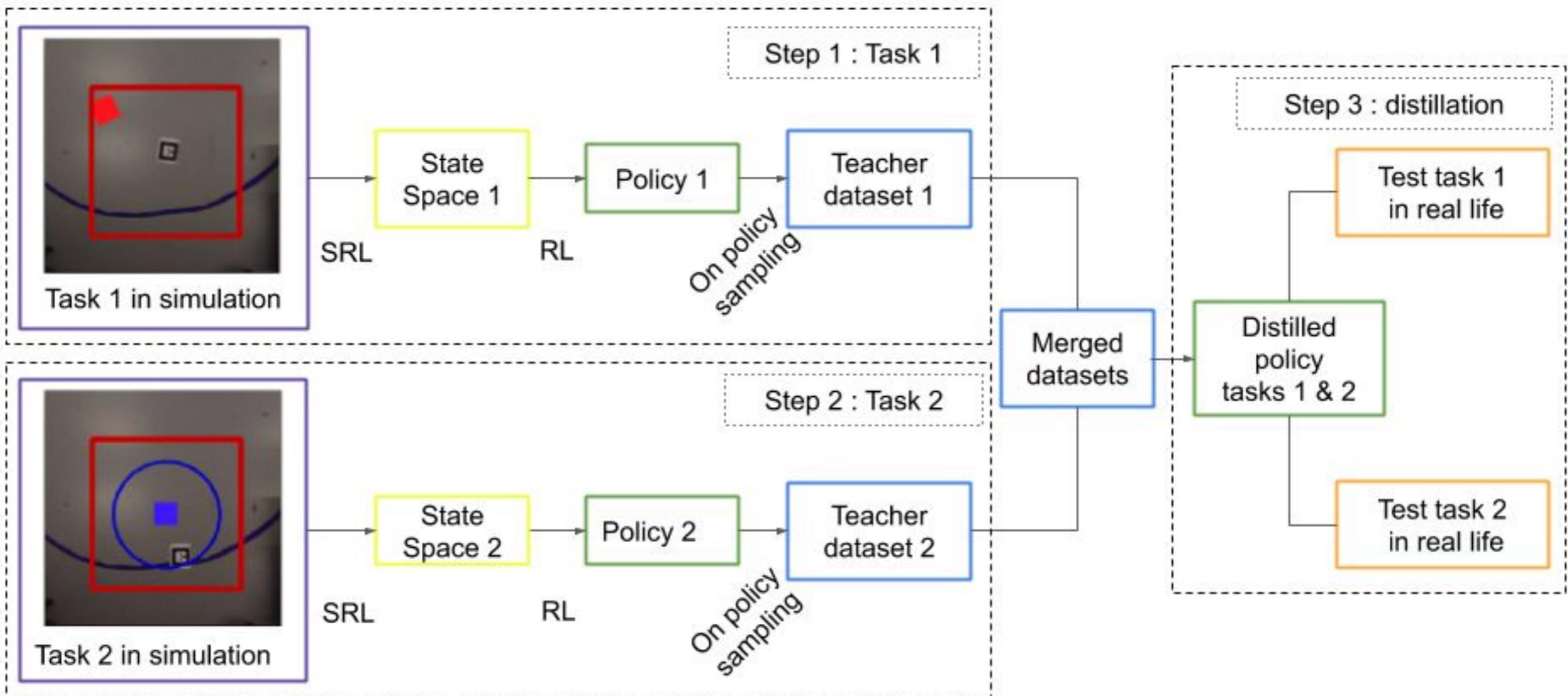
Target
Reaching



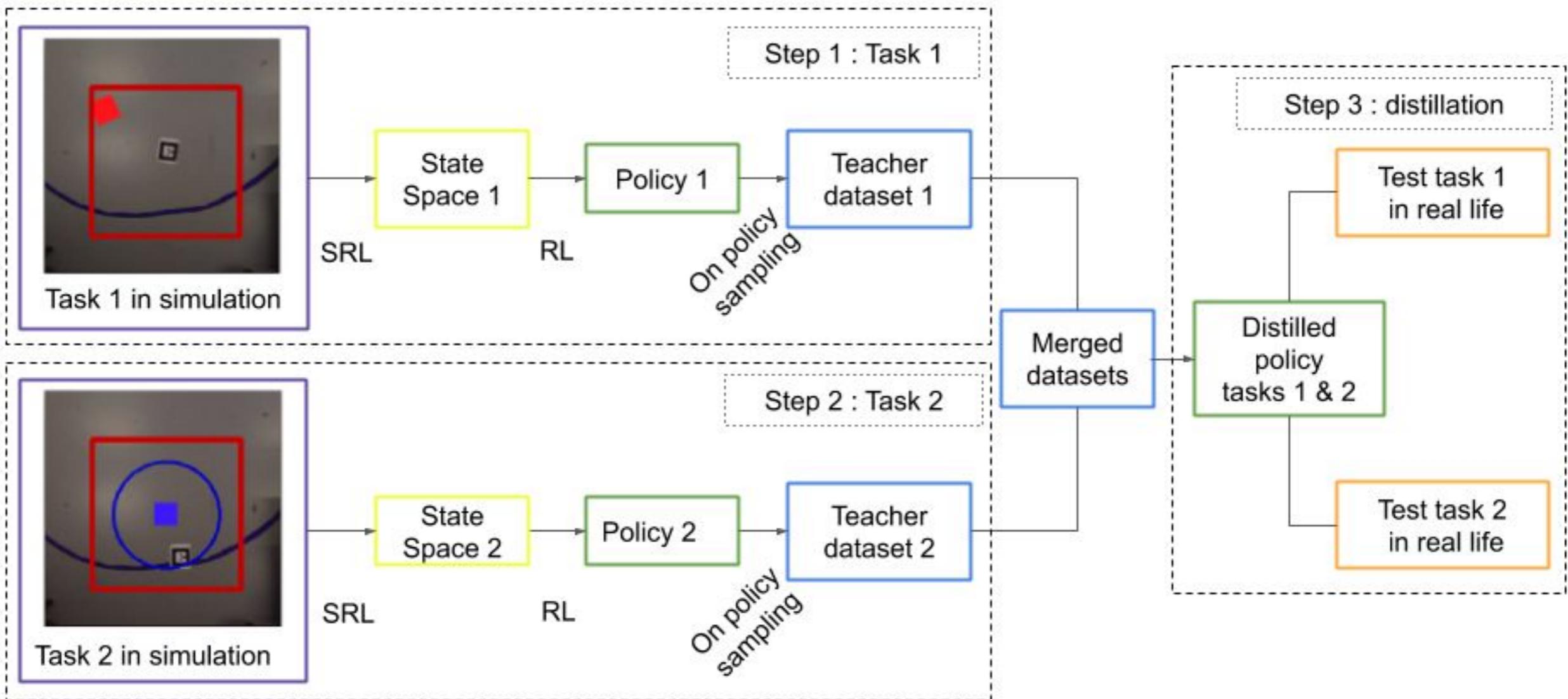
Simulation

Real Life

Continual RL in Multi-Task, Lifelong & Real Life Settings



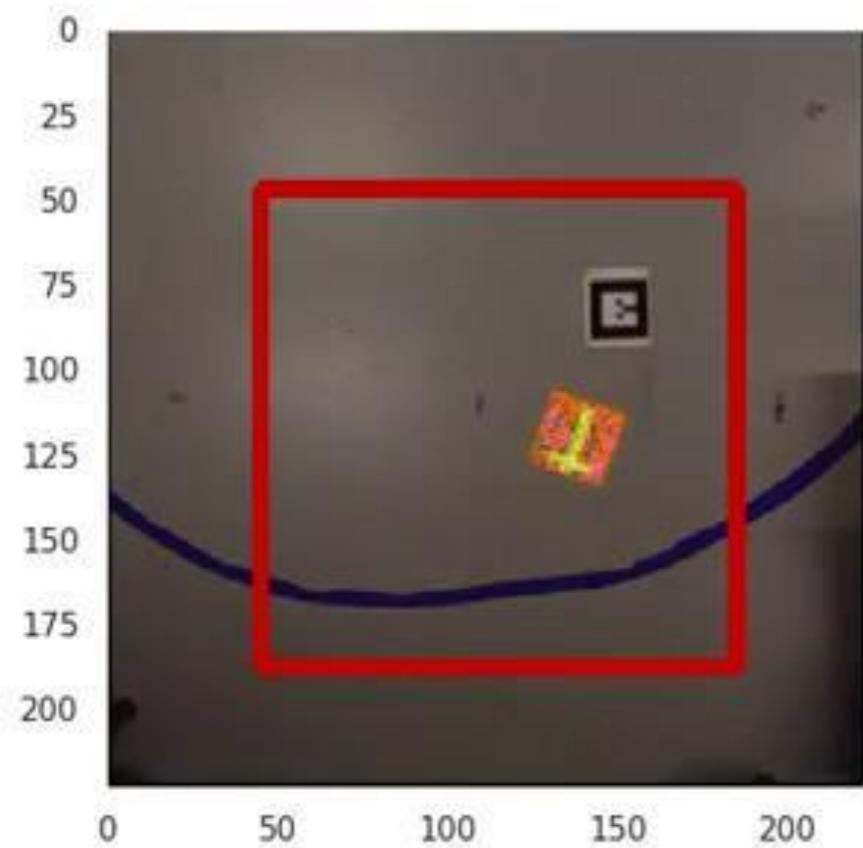
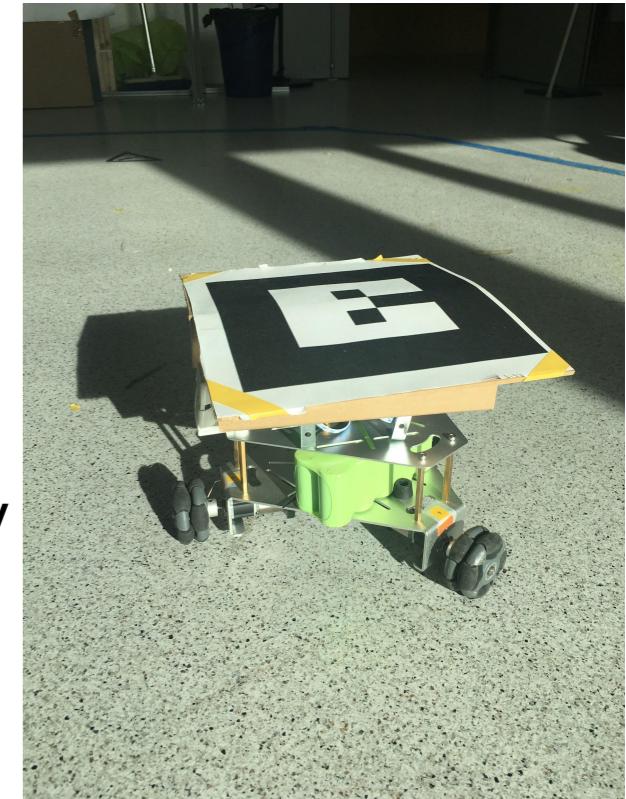
Continual RL in Multi-Task, Lifelong & Real Life Settings



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

Challenges

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



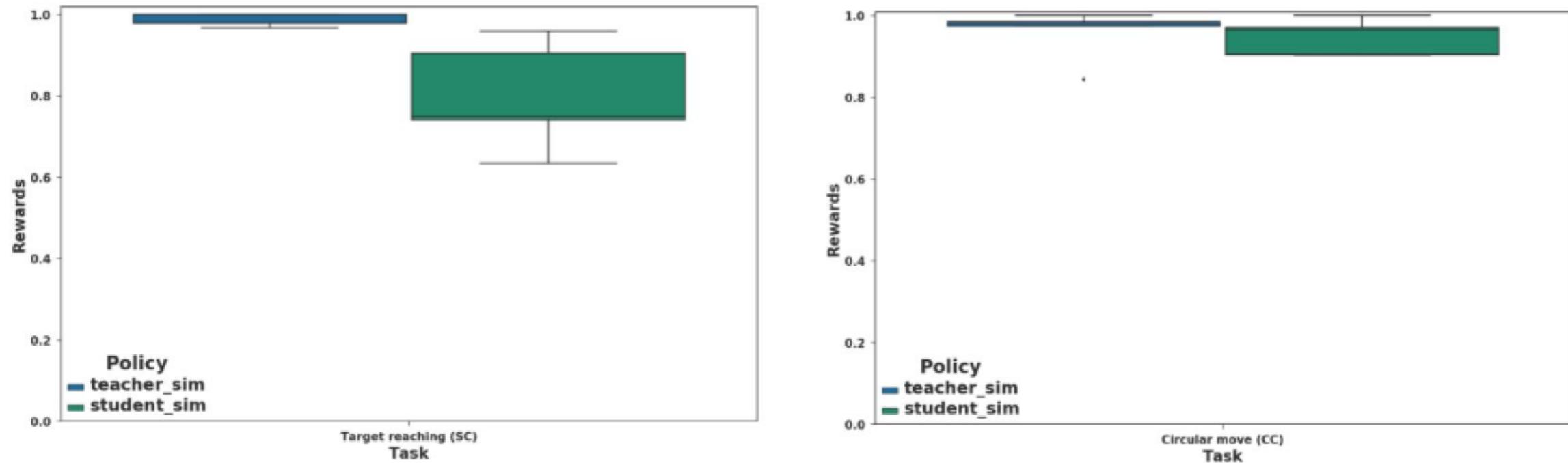
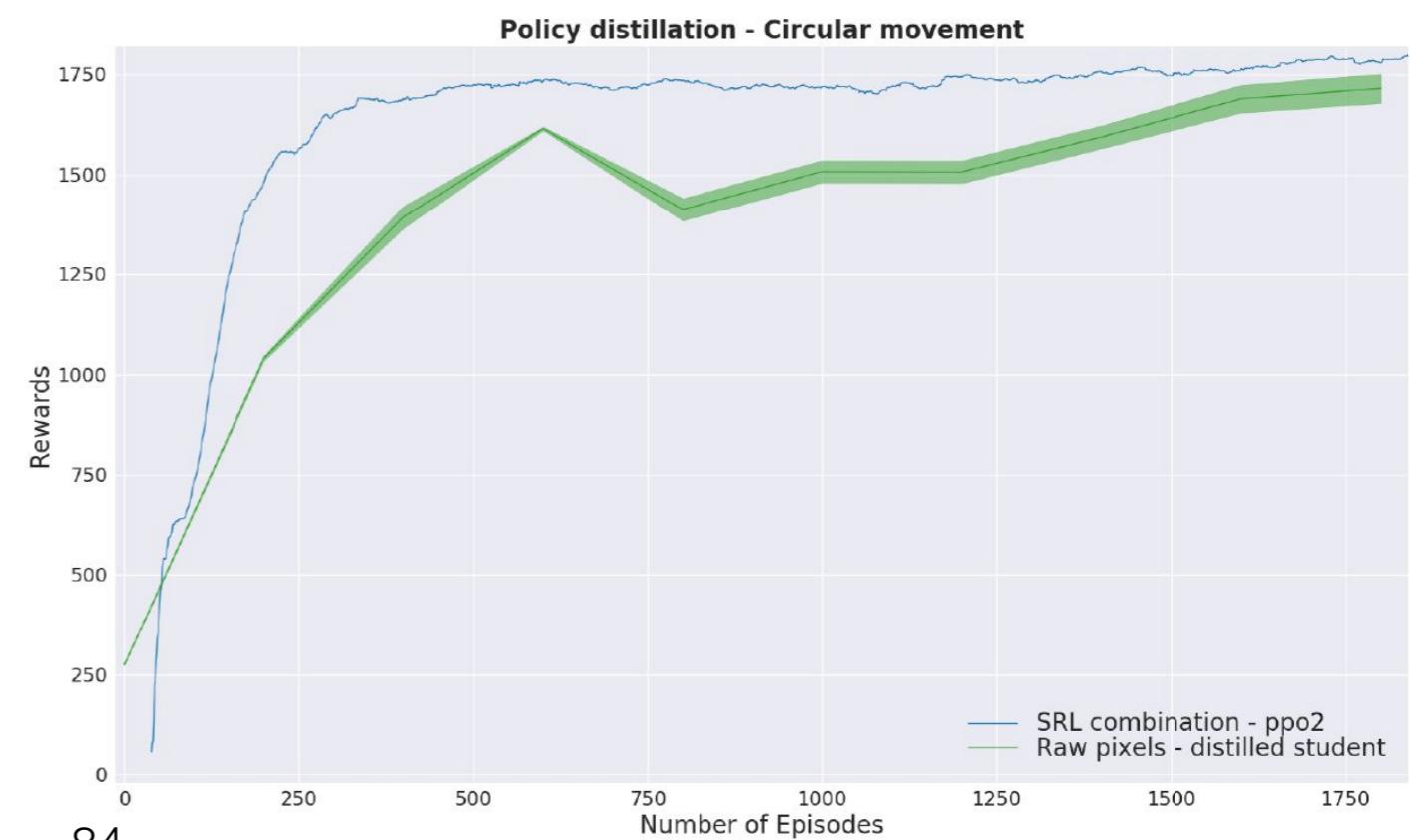
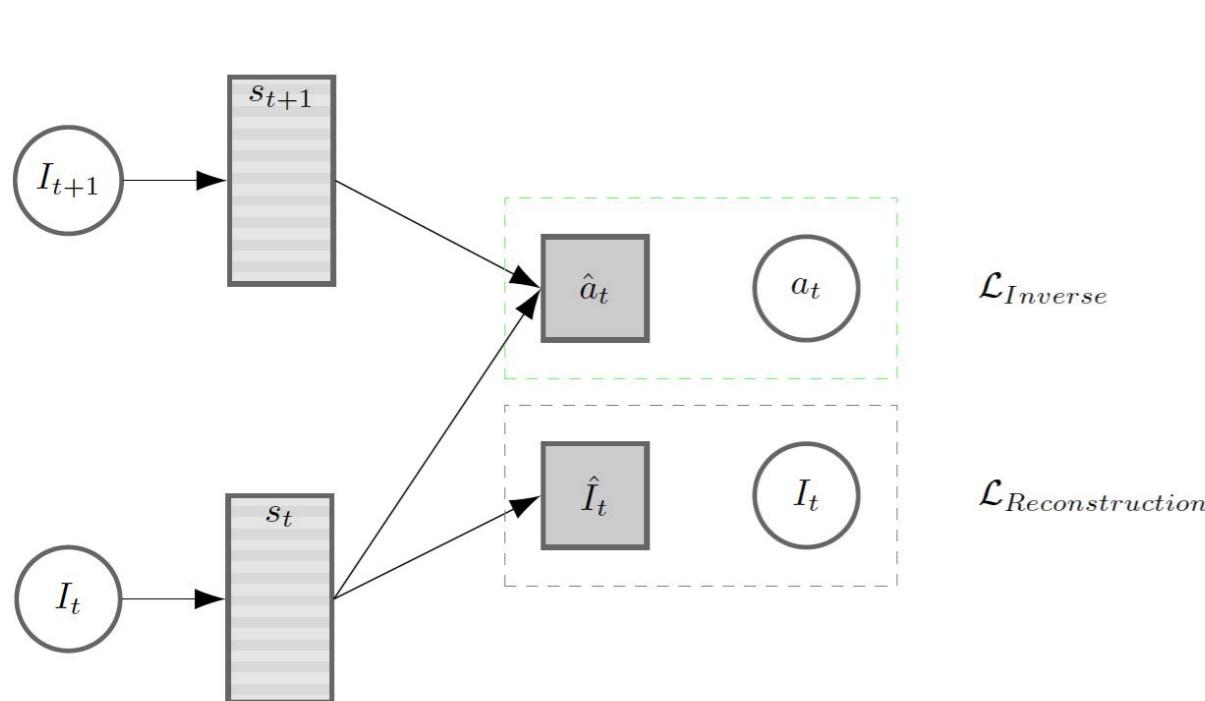
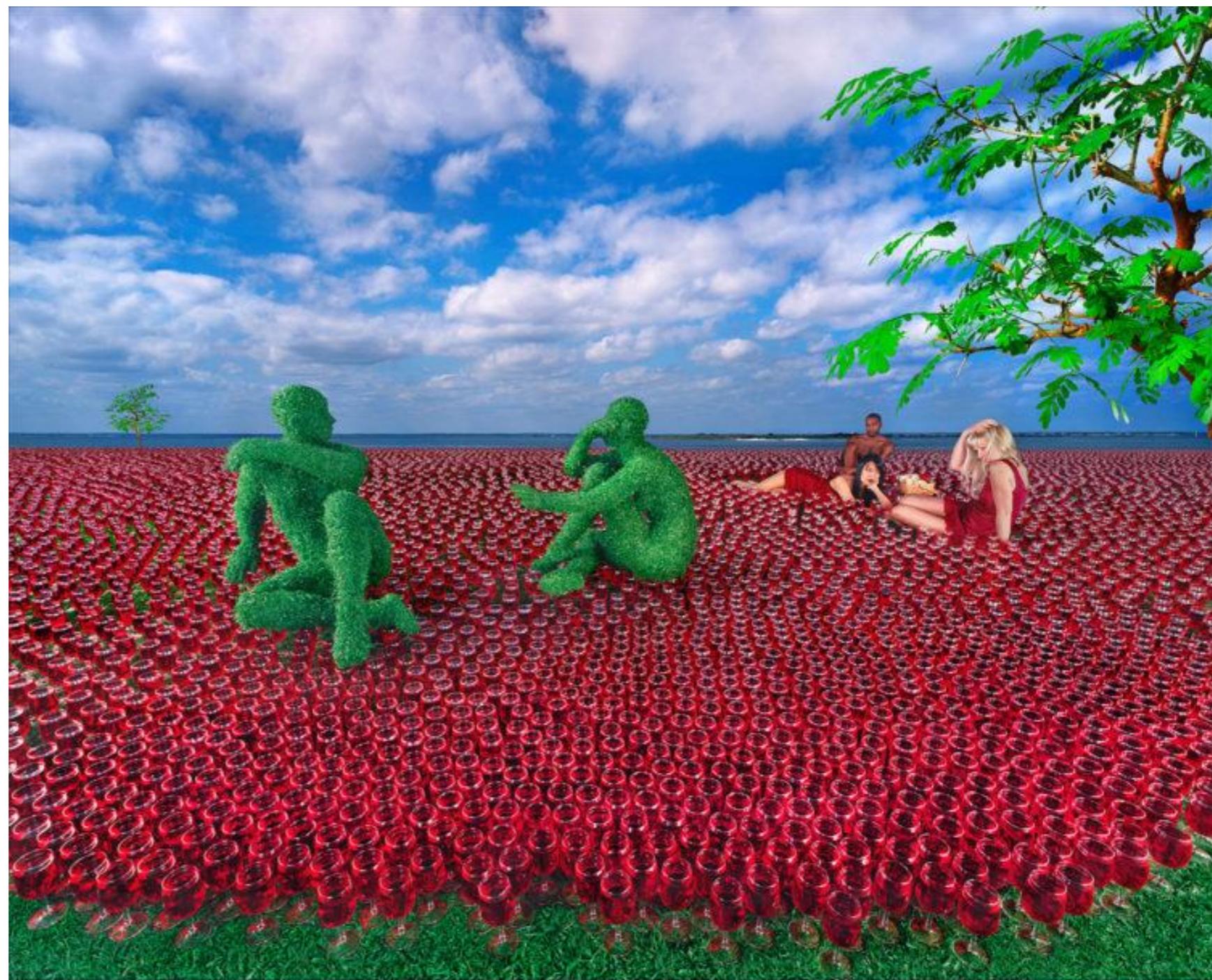


Figure 4. Comparison between performance (normalized mean reward and standard error) of policy trained on one task only to distilled student policy on the two tasks. The student policy has similar performance on both tasks. **Left:** Target Reaching (TR). **Right:** Target Circling (TC) task.



Future Perspectives



Future Perspectives

- More & More complex tasks: Rewards?
- Multimodality

IMAGES



CAPTIONS

A man is sitting on a curb with a bike.



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

Future Perspectives

Explaining forgetting, learning and memorizing in Deep Nets

arXiv:1910.10045v1 [cs.AI] 22 Oct 2019

Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI

Alejandro Barredo Arrieta^a, Natalia Díaz-Rodríguez^b, Javier Del Ser^{a,c,d}, Adrien Bennetot^{b,e,f},
Siham Tabik^g, Alberto Barbado^h, Salvador Garcia^g, Sergio Gil-Lopez^a, Daniel Molina^g,
Richard Benjamins^h, Raja Chatila^f, and Francisco Herrera^g

^aTECNALIA, 48160 Derio, Spain

^bENSTA, Institute Polytechnique Paris and INRIA Flowers Team, Palaiseau, France

^cUniversity of the Basque Country (UPV/EHU), 48013 Bilbao, Spain

^dBasque Center for Applied Mathematics (BCAM), 48009 Bilbao, Bizkaia, Spain

^eSegula Technologies, Parc d'activité de Pissaloup, Trappes, France

^fInstitut des Systèmes Intelligents et de Robotique, Sorbonne Université, France

^gDaSCI Andalusian Institute of Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain

^hTelefonica, 28050 Madrid, Spain

Abstract

In the last few years, Artificial Intelligence (AI) has achieved a notable momentum that, if harnessed appropriately, may deliver the best of expectations over many application sectors across the field. For this to occur shortly in Machine Learning, the entire community stands in front of the barrier of explainability, an inherent problem of the latest techniques brought by sub-symbolism (e.g. ensembles or Deep Neural Networks) that were not present in the last type of AI (namely, expert systems and rule based models). Paradigms underlying this problem fall within the so-called *Explainable* AI (XAI) field, which is widely acknowledged as a crucial feature for the practical deployment of AI models. The overview presented in this article examines the existing literature and contributions already done in the field of XAI, including a prospect toward what is yet to be reached. For this purpose we summarize previous efforts made to define explainability in Machine Learning, establishing a novel definition of explainable Machine Learning that covers such prior conceptual propositions with a major focus on the audience for which the explainability is sought. Departing from this definition, we propose and discuss about a taxonomy of recent contributions related to the explainability of different Machine Learning models, including those aimed at explaining Deep Learning methods for which a second dedicated taxonomy is built and examined in detail. This critical literature analysis serves as the motivating background for a series of challenges faced by XAI, such as the interesting crossroads of data fusion and explainability. Our prospects lead toward the concept of *Responsible Artificial Intelligence*, namely, a methodology for the large-scale implementation of AI methods in real organizations with fairness, model explainability and accountability at its core. Our ultimate goal is to provide newcomers to the field of XAI with a thorough taxonomy that can serve as reference material in order to stimulate future research advances, but also to encourage experts and professionals from other disciplines to embrace the benefits of AI in their activity sectors, without any prior bias for its lack of interpretability.

Keywords: Explainable Artificial Intelligence, Machine Learning, Deep Learning, Data Fusion, Interpretability, Comprehensibility, Transparency, Privacy, Fairness, Accountability, Responsible Artificial Intelligence.

Future Perspectives

Towards Explainable Neural-Symbolic Visual Reasoning

Adrien Bennetot^{1,2,3}, Jean-Luc Laurent¹, Raja Chatila² and Natalia Díaz-Rodríguez³

¹ Segula Technologies, Parc d'activité de Pissaloup - Trappes, France

² Institut des Systèmes Intelligents et de Robotique, Sorbonne Université, France

³ U2IS & INRIA FLOWERS <https://flowers.inria.fr> Team, ENSTA ParisTech, Palaiseau, France
{adrien.bennetot, jeanluc.laurent}@segula.fr, raja.chatila@sorbonne-universite.fr,
natalia.diaz@ensta-paristech.fr}

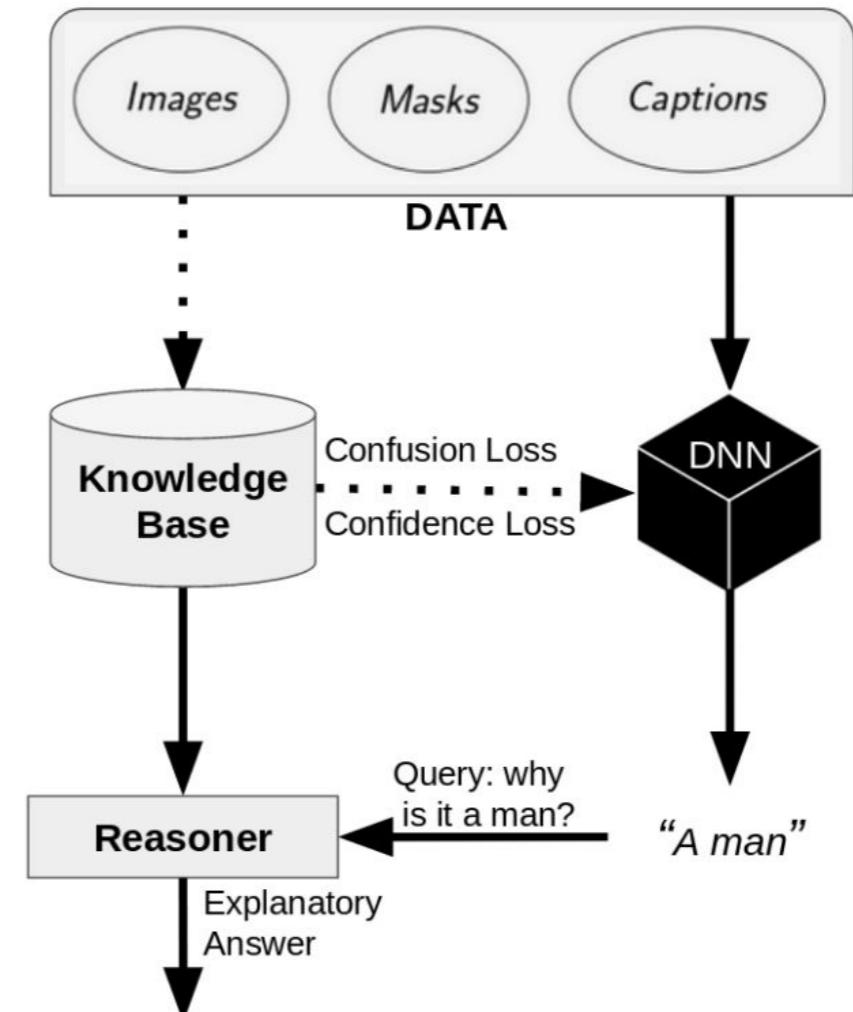
- The **black box** gives the caption output
- The **reasoner** facilitates the explanation of the output: the natural language explanation (NLE).



Model (M): A man on a bench

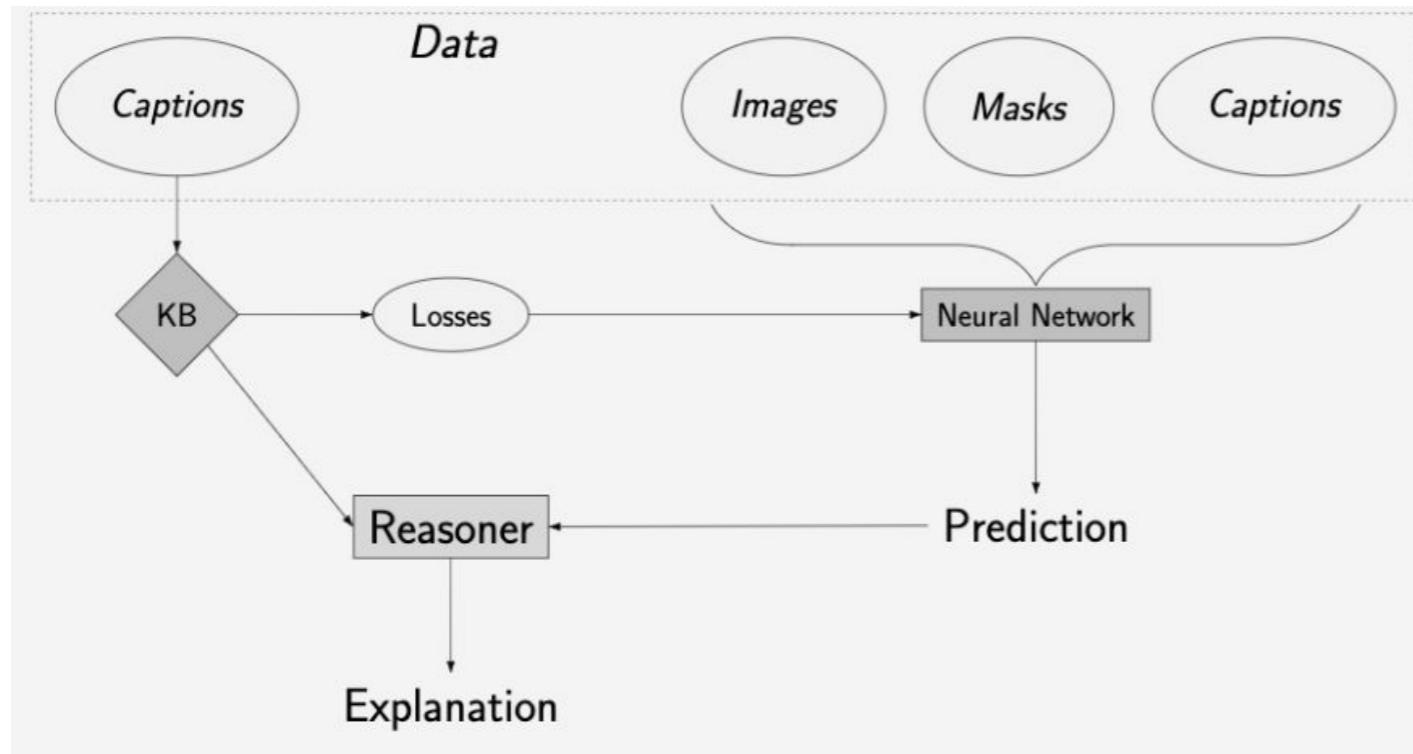
User (U): Why is it a man ?

M: I recognise a **person** and I know that a **person** can be a **man**, a **woman**, a **boy** or a **girl**. I estimate at 85% that this **person** is a **man** so I am **confident** that this **person** is a **man**



"I recognize a [person]. I know from the KB that a [person] can be a [man], a [teenager], a [boy] or a [senior]. I succeeded to determine the [person] type so I am confident: this is a [man]"

Future Perspectives



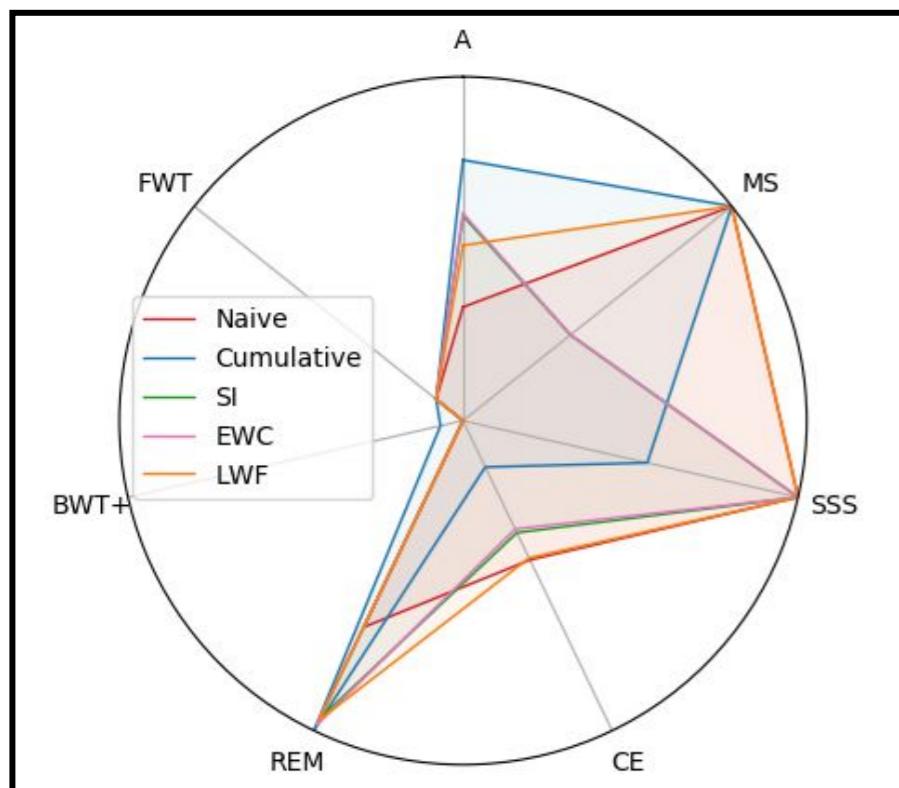
Model (M): A person on a bench

User (U): Why is it a person ?

M: I recognise at 30% a **man**,
30% a **woman** and 40% a **boy**.
I'm **confused** about the fact that
it's a **man**, a **woman** or a **boy** but
I'm 100% sure this is a **person**.

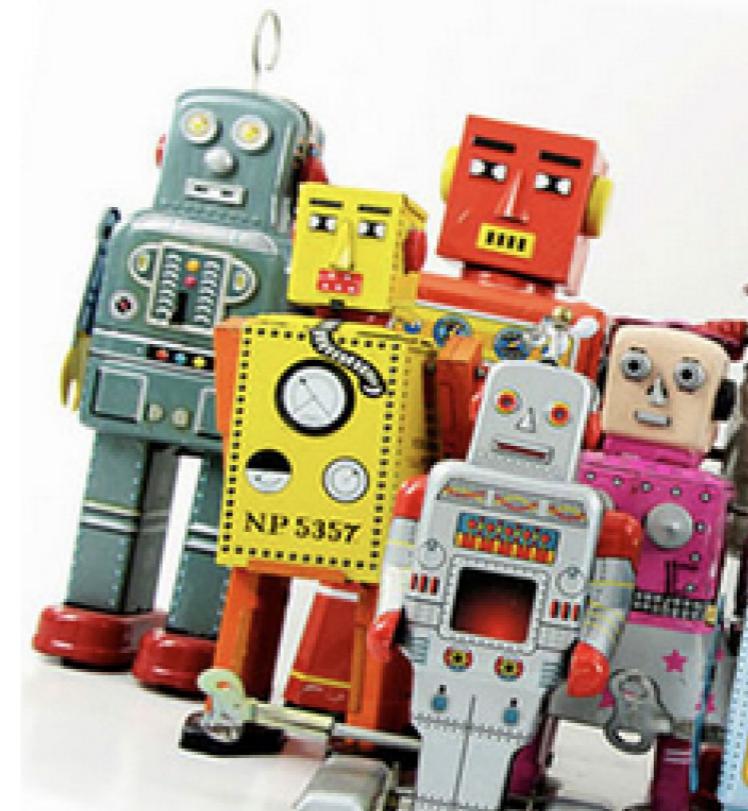
Future Perspectives (II)

- Wider, + comprehensive metrics:
 - No more benchmarking on MNIST!
 - Permutation-based datasets shouldn't be used to investigate CF [Pföhl'19]



Future Perspectives (II)

- Wider, + comprehensive metrics:
 - No more benchmarking on MNIST!
 - Permutation-based datasets shouldn't be used to investigate CF [Pföhlb'19]
- + Ablation studies (assumptions, relaxations, constraints)



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!

PAPER & POSTER: *Continual Reinforcement Learning deployed in Real-life using Policy Distillation and Sim2Real Transfer.* [Traoré et. al.'19]

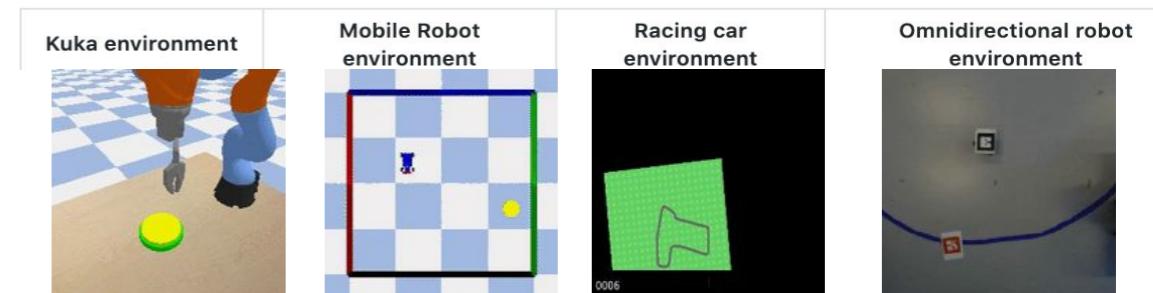
CODE: https://github.com/kalifou/robotics-rl-srl/tree/circular_movement_omnibot

S-RL Toolbox: RL & SRL for Robotics <https://github.com/araffin/robotics-rl-srl>

Connect: nataliadiaz.github.io

@NataliaDiazRodr

natalia.diaz@ensta-paris.fr



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

ContinualAI

Co-Founders:



ContinualAI Advisors:



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

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



ContinualAI

A Non-profit Research Organization and Open Community on
Continual Learning for AI

[Home](#)

[Connect](#)

[Research](#)

[Educate](#)

[Projects](#)

[News](#)

[Supporters](#)

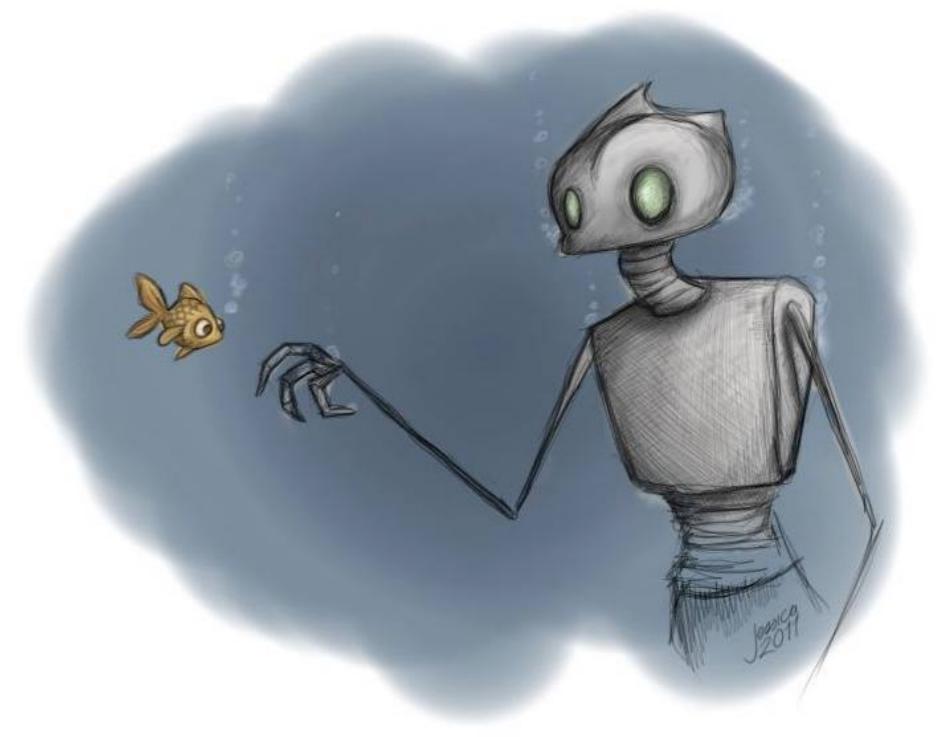
[About us](#)

[Join us!](#)



Hands-on!

1. Go to: <https://github.com/ContinualAI/colab>
2. Follow instructions in ReadMe
3. Run and experiment with the Notebook:
[Intro to Continual Learning](#)



intro_to_continual_learning.ipynb

File Edit View Insert Runtime Tools Help Unsaved changes since 11:32 AM

+ Code + Text Copy to Drive

RAM Disk Editing

Table of contents Code snippets Files X

A Gentle Introduction to Continual Learning in PyTorch

Google Colaboratory

Installing PyTorch 0.4.0

MNIST: Digits recognition with PyTorch

CL Strategies

Naive Strategy

Rehearsal Strategy

Elastic Weights Consolidation (EWC) Strategy

Plot Results

SECTION

MNIST? Let's define our custom function to permute it!

```
[ ] 1 def permute_mnist(mnist, seed):
2     """ Given the training set, permute pixels of each img the same way. """
3
4     np.random.seed(seed)
5     print("starting permutation...")
6     h = w = 28
7     perm_inds = list(range(h*w))
8     np.random.shuffle(perm_inds)
9     # print(perm_inds)
10    perm_mnist = []
11    for set in mnist:
12        num_img = set.shape[0]
13        flat_set = set.reshape(num_img, w * h)
14        perm_mnist.append(flat_set[:, perm_inds].reshape(num_img, 1, w, h))
15    print("done.")
16    return perm_mnist
```

```
[ ] 1 x_train2, x_test2 = permute_mnist([x_train, x_test], 0)
```

starting permutation...
done.

```
[ ] 1 f, axarr = plt.subplots(1,2)
2 axarr[0].imshow(x_train[1, 0], cmap="gray")
3 axarr[1].imshow(x_train2[2, 0], cmap="gray")
4 np.vectorize(lambda ax:ax.axis('off'))(axarr);
```

Two 28x28 pixel grayscale images. The left image shows a handwritten digit '4' on a black background. The right image shows the same digit '4' after being permuted by the custom function, appearing as a noisy, pixelated pattern.



+ Code + Text

Copy to Drive

RAM Disk

Editing

[Table of contents](#)[Code snippets](#)[Files X](#)

A Gentle Introduction to Continual Learning in PyTorch

Google Colaboratory

Installing PyTorch 0.4.0

MNIST: Digits recognition with PyTorch

CL Strategies

Naive Strategy

Rehearsal Strategy

Elastic Weights Consolidation (EWC) Strategy

Plot Results

+ SECTION

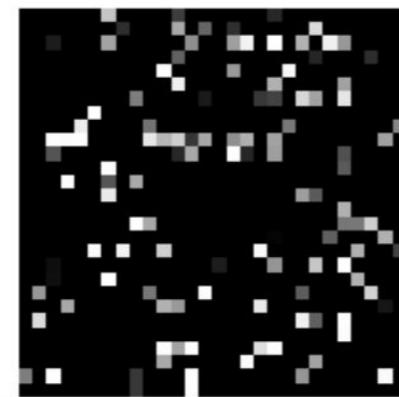
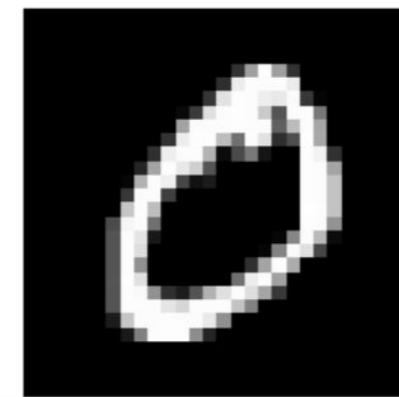
MNIST? Let's define our custom function to permute it!

```
[ ] 1 def permute_mnist(mnist, seed):
2     """ Given the training set, permute pixels of each img the same way. """
3
4     np.random.seed(seed)
5     print("starting permutation...")
6     h = w = 28
7     perm_inds = list(range(h*w))
8     np.random.shuffle(perm_inds)
9     # print(perm_inds)
10    perm_mnist = []
11    for set in mnist:
12        num_img = set.shape[0]
13        flat_set = set.reshape(num_img, w * h)
14        perm_mnist.append(flat_set[:, perm_inds].reshape(num_img, 1, w, h))
15    print("done.")
16    return perm_mnist
```

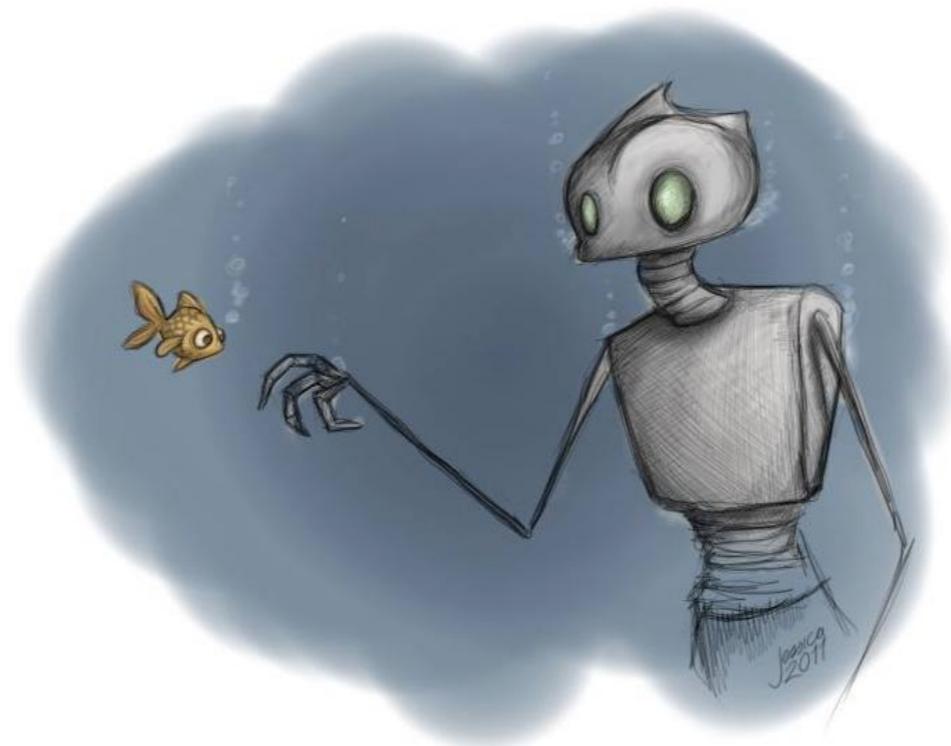
```
[ ] 1 x_train2, x_test2 = permute_mnist([x_train, x_test], 0)
```

👤 starting permutation...
done.

```
[ ] 1 f, axarr = plt.subplots(1,2)
2 axarr[0].imshow(x_train[1, 0], cmap="gray")
3 axarr[1].imshow(x_train2[2, 0], cmap="gray")
4 np.vectorize(lambda ax:ax.axis('off'))(axarr);
```



APPENDIX



Omni-robot sim2real replay

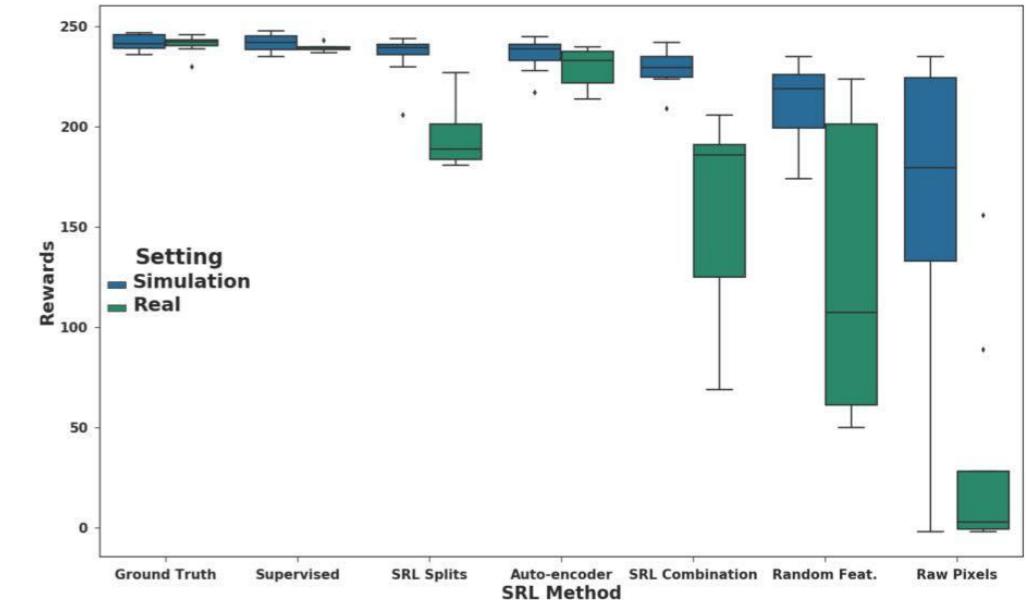


Figure 12: From simulation to real robot: Mean reward and standard deviation for policies trained in simulation (5M steps budget) and replayed in Simulated and Real Omnidot (250 steps, 8 runs).

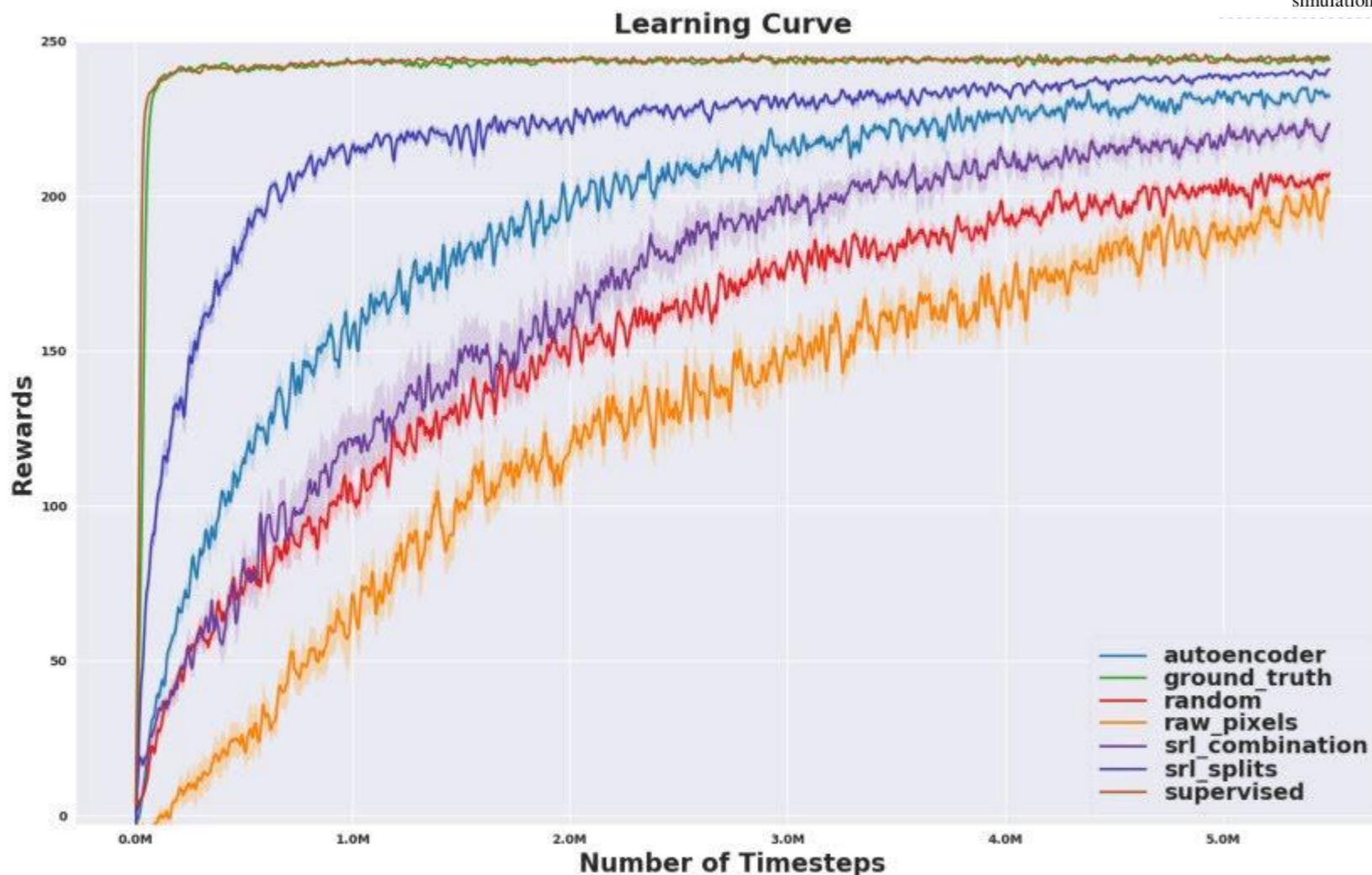


Figure 3: Performance (mean and standard error for 8 runs) for PPO algorithm for different state representations learned in Simulated OmniRobot with randomly initialized target environment.

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 \times height \times 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		class balance (%)	
			train	test	train	test
CIFAR10		$32 \times 32 \times 3$	50.000	10.000	0	0
Devanagari		$32 \times 32 \times 1$	18.000	2.000	0.3	2.7
EMNIST		$28 \times 28 \times 1$	345.035	57.918	2.0	2.0
FashionMNIST		$28 \times 28 \times 1$	60.000	10.000	0	0
Fruits 360		$100 \times 100 \times 3$	6.148	2.052	4.0	4.2
MADBase		$28 \times 28 \times 1$	60.000	10.000	0	0
MNIST		$28 \times 28 \times 1$	55.000	10.000	2.2	2.4
NotMNIST		$28 \times 28 \times 1$	529.114	18.724	~ 0	~ 0
SVHN		$32 \times 32 \times 3$	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 consists 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)”.

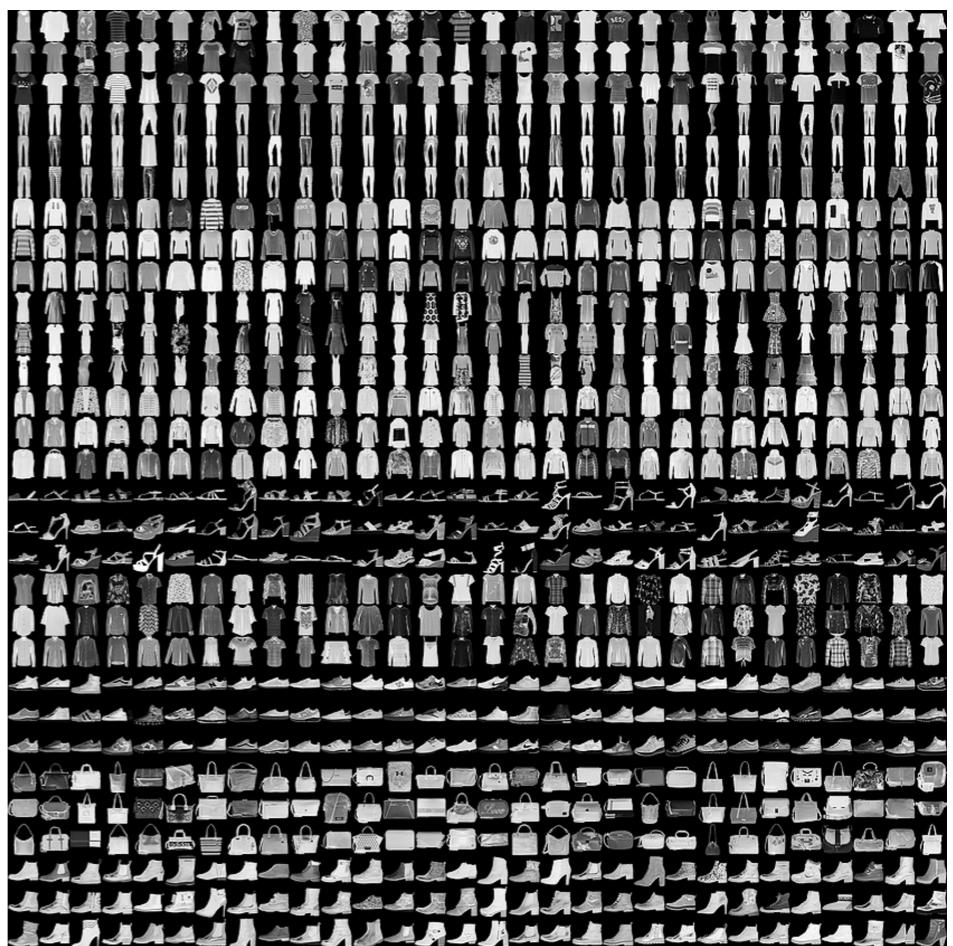
SVHN (Netzer et al., 2011) is a 10-class dataset based on photos of house numbers (0-9). We use the cropped digit format, where the number is centered in the color image.

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.

Evaluation Benchmarks

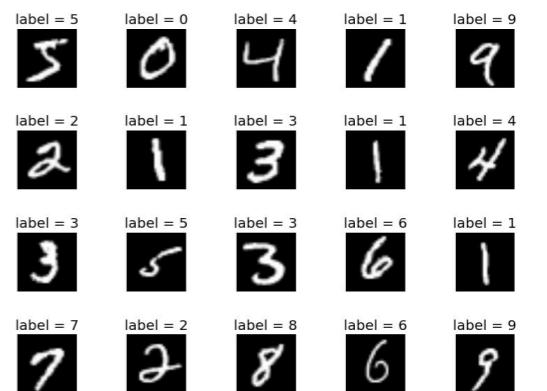


Fashion MNIST [Xiao'17]

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.

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)

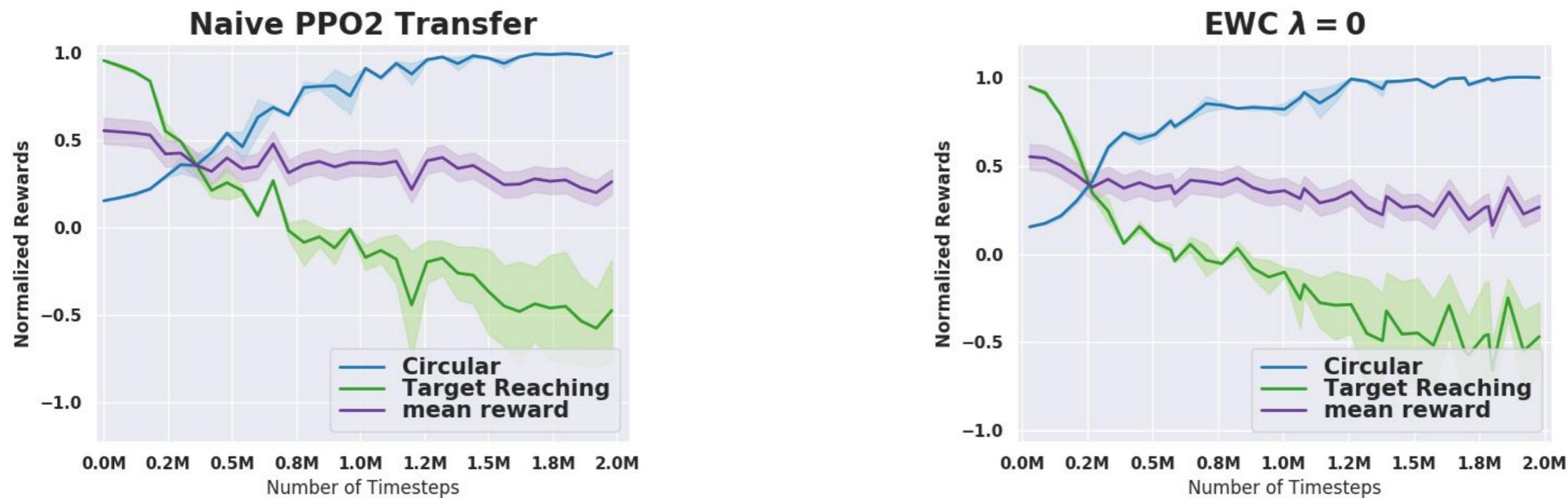


Fig 1: Implementation validation of EWC when lambda is zero (the result should be very similar to a simple continual transfer from PPO2)

Essential Tools to face CL challenges in Robotics

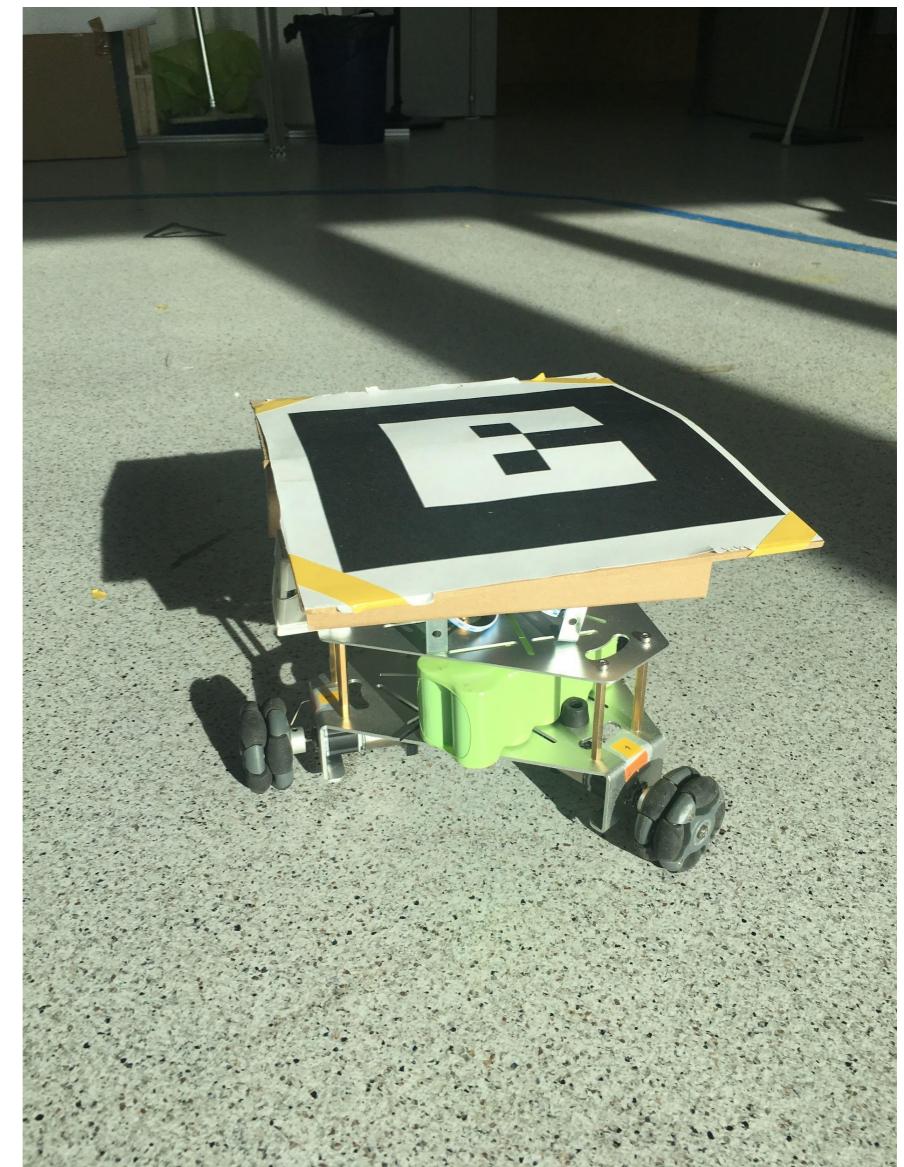
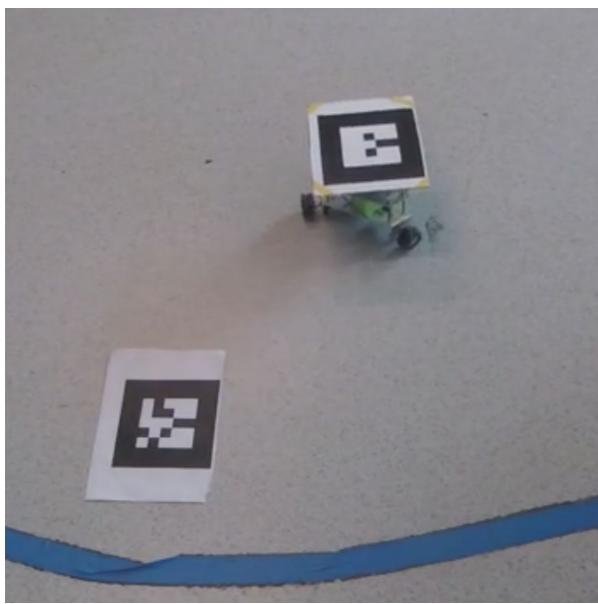
- SRL: allows to build strong representations of the world since agents should be able to understand their surroundings, and extract general concepts from sensory inputs of complex scenes. An agent better sees a chair as an object, not as a bunch of pixels together in an image.
- Helps scaling end-to-end 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.

SRL for Robotics: Challenges

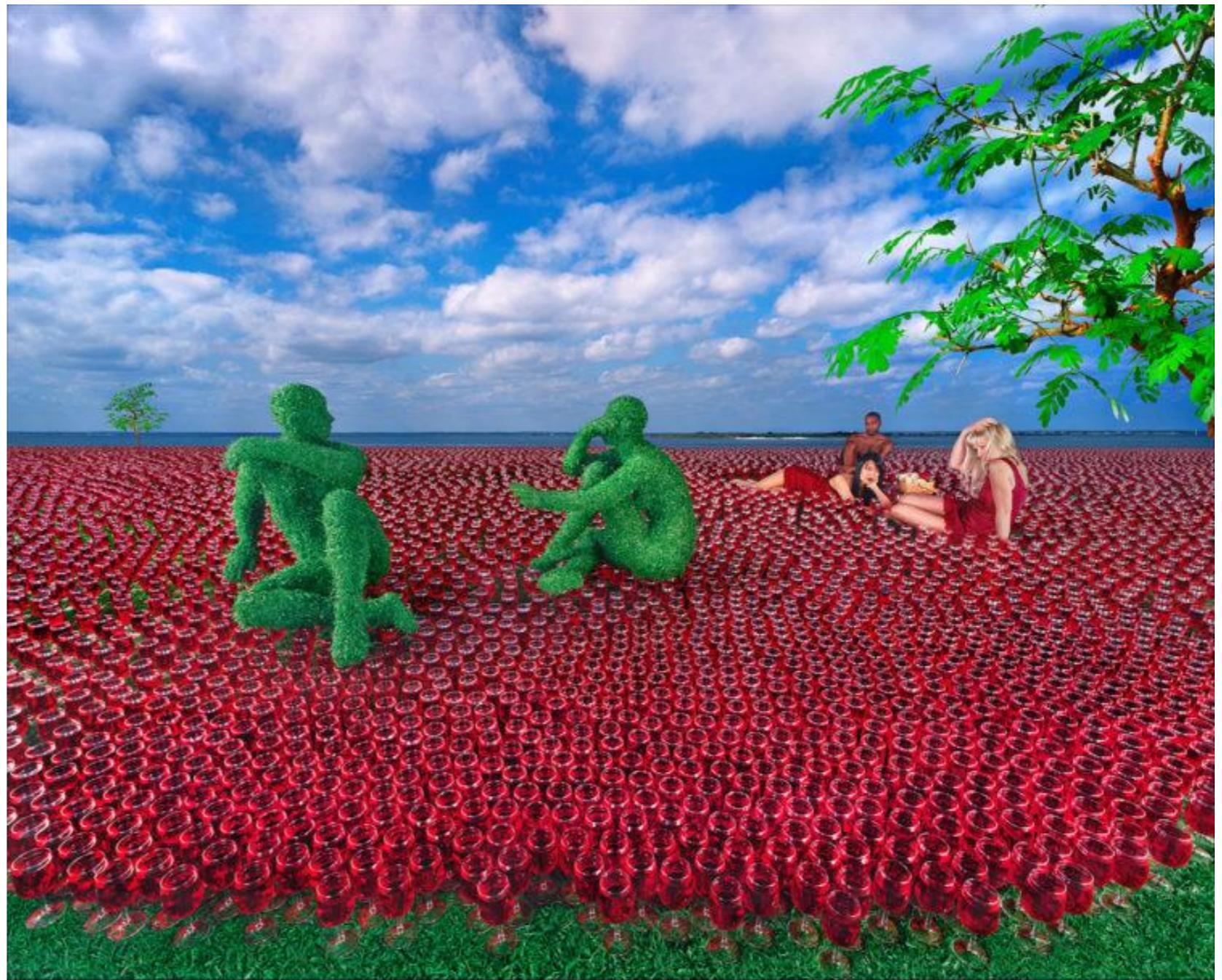
Vision goal-based tasks (mobile, visible goal to reach -vs open-ended tasks s.t. *walking*-)

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

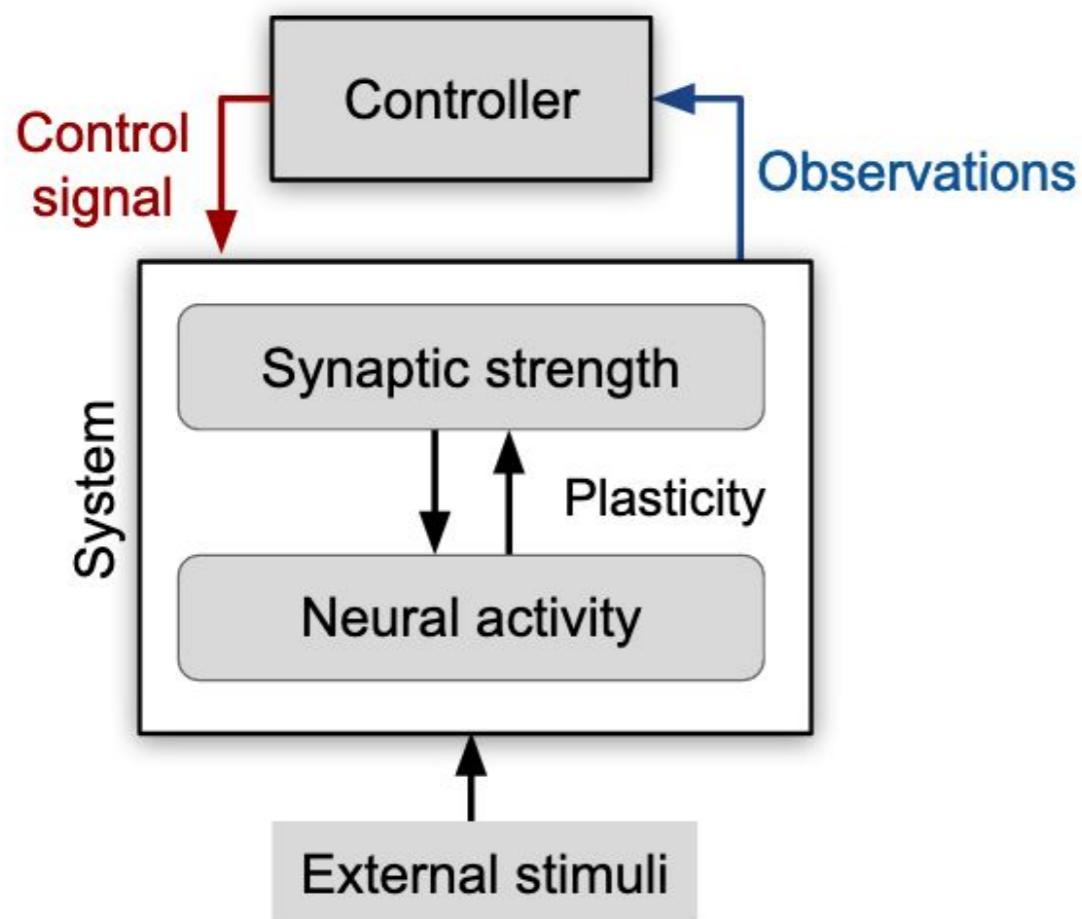
Evaluating learned representations: Environments: Omnipilot



Hybrid Approaches



a) Hebbian and Homeostatic Plasticity



b) Complementary Learning Systems (CLS) theory

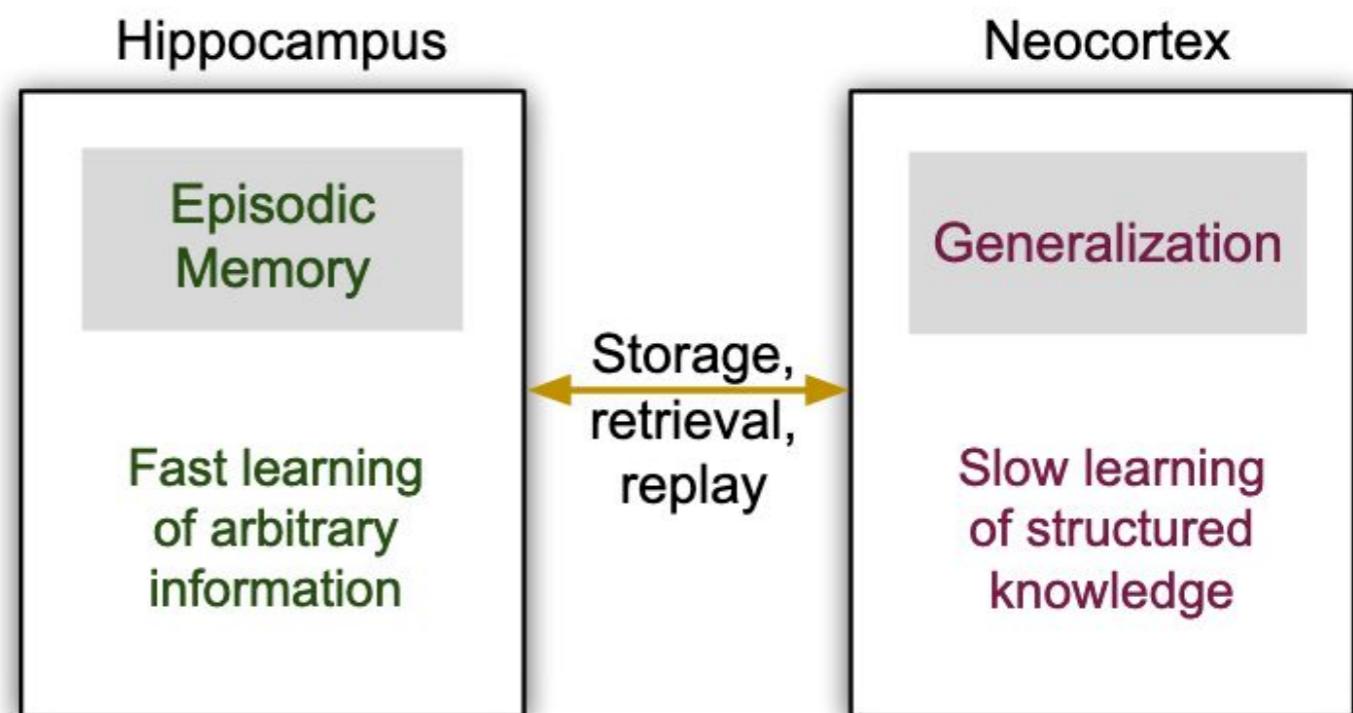


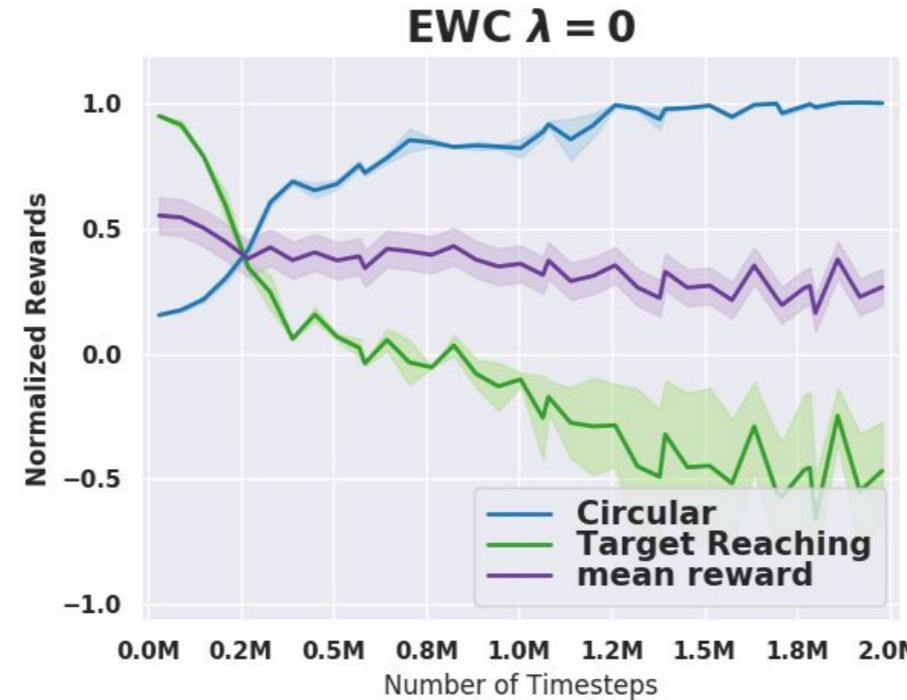
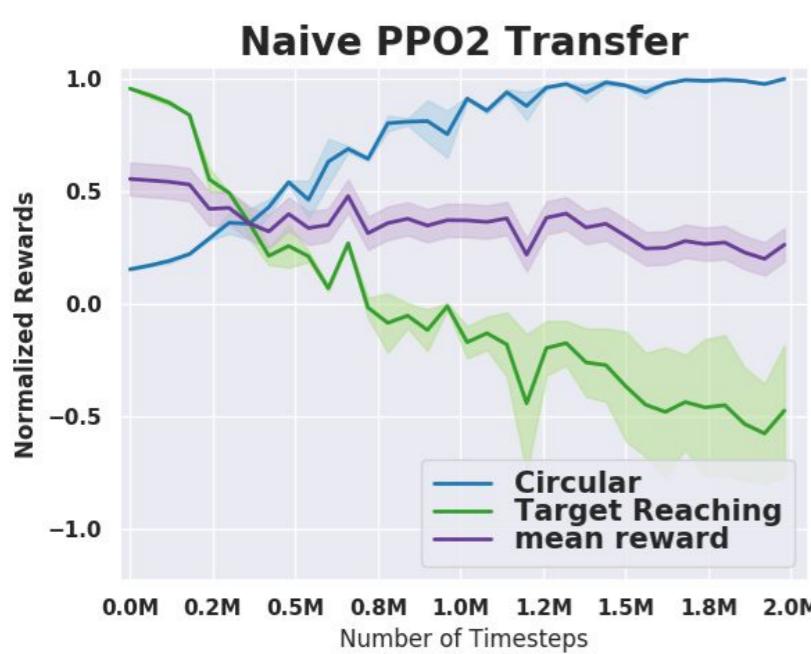
Figure 1: Schematic view of two aspects of neurosynaptic adaptation: a) Hebbian learning with homeostatic plasticity as a compensatory mechanism that uses observations to compute a feedback control signal (Adapted with permission from Zenke, Gerstner & Ganguli (2017)). b) The complementary learning systems (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.

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

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

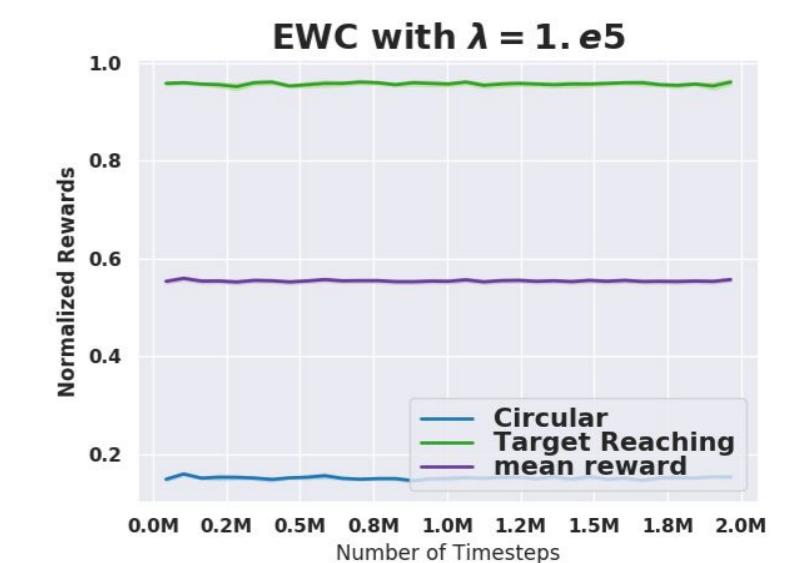
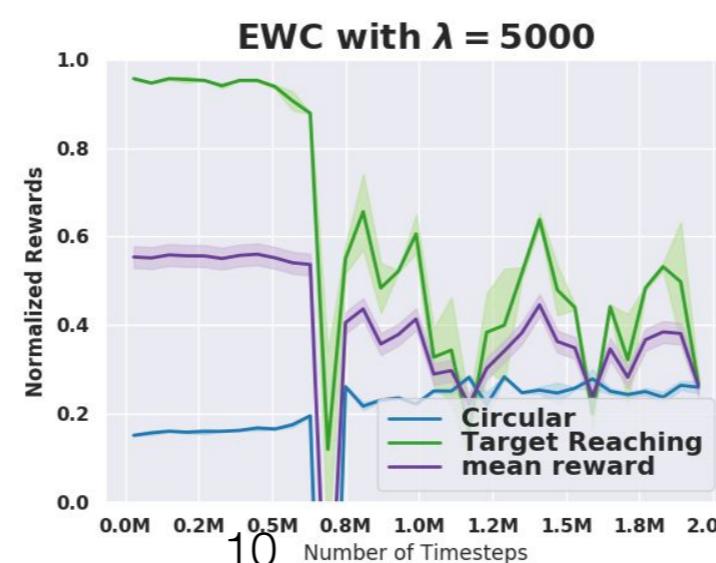
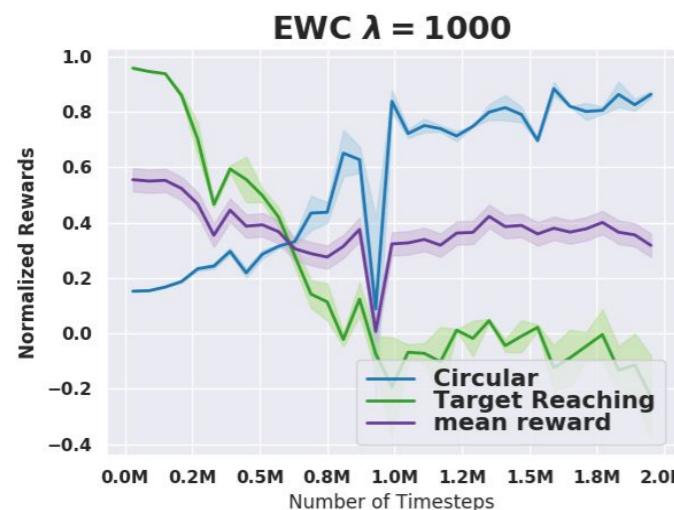
$$R_{t,circle} = 1 - \lambda(\|z_t\| - R_{circle})^2 \quad (1)$$

Without regularization:

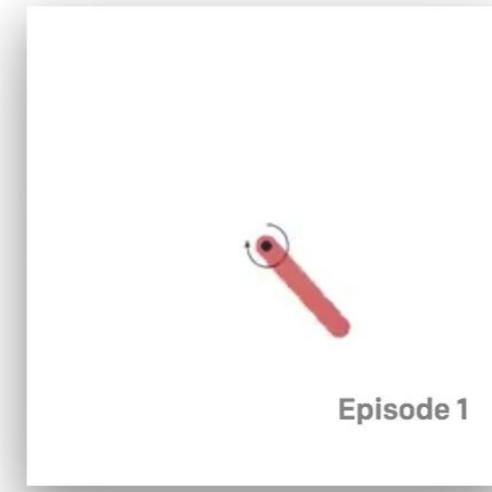
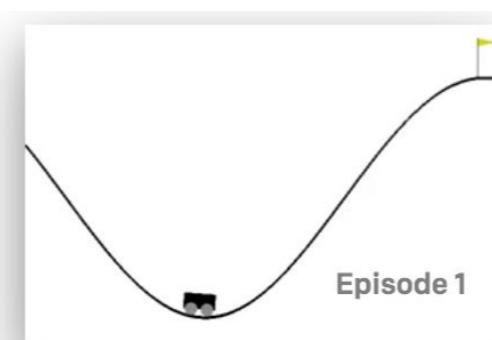


$$\lambda R_{t,circle} * R_{t,movement} + \lambda^2 R_{t,bump} \quad (3)$$

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



Evaluation Benchmarks: RL & Robotics



Ball in cup



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

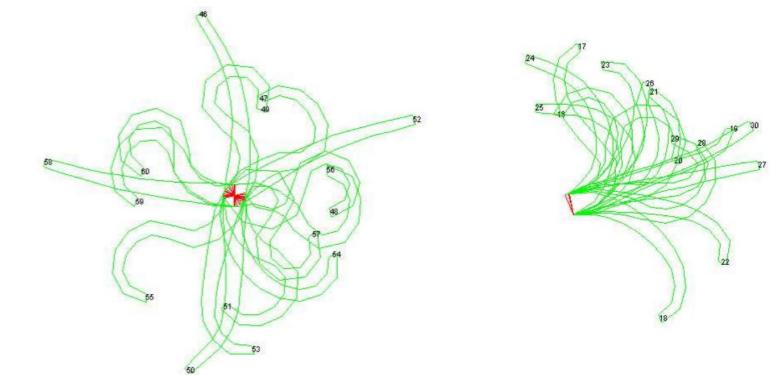


Figure 3: Examples of initial states for the rotating-base experiments (left) and the fixed-base experiments (right). Starting states also include velocities, which are not shown.

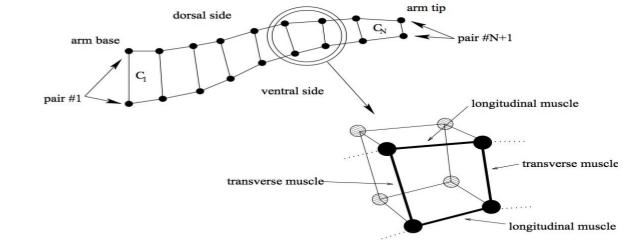
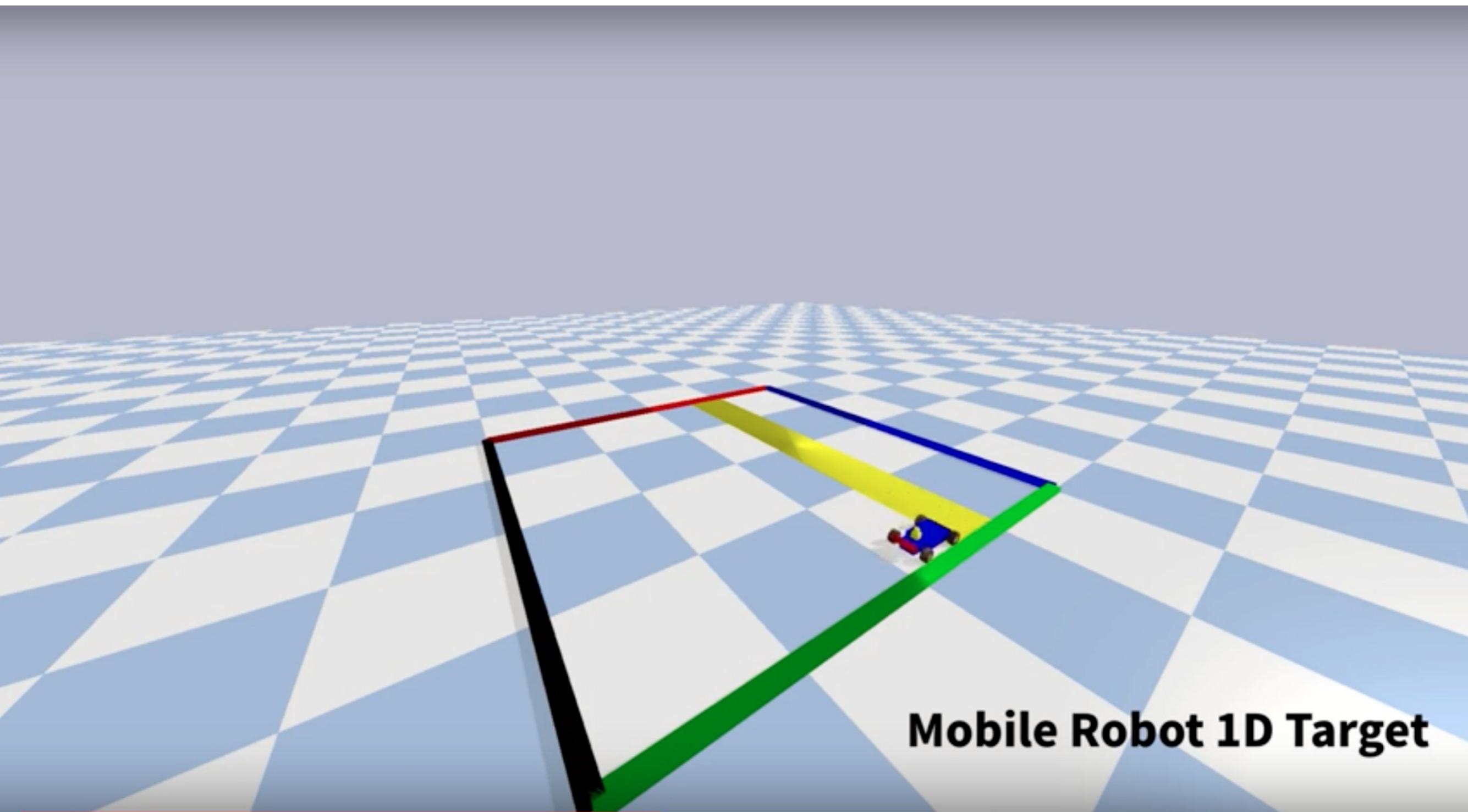


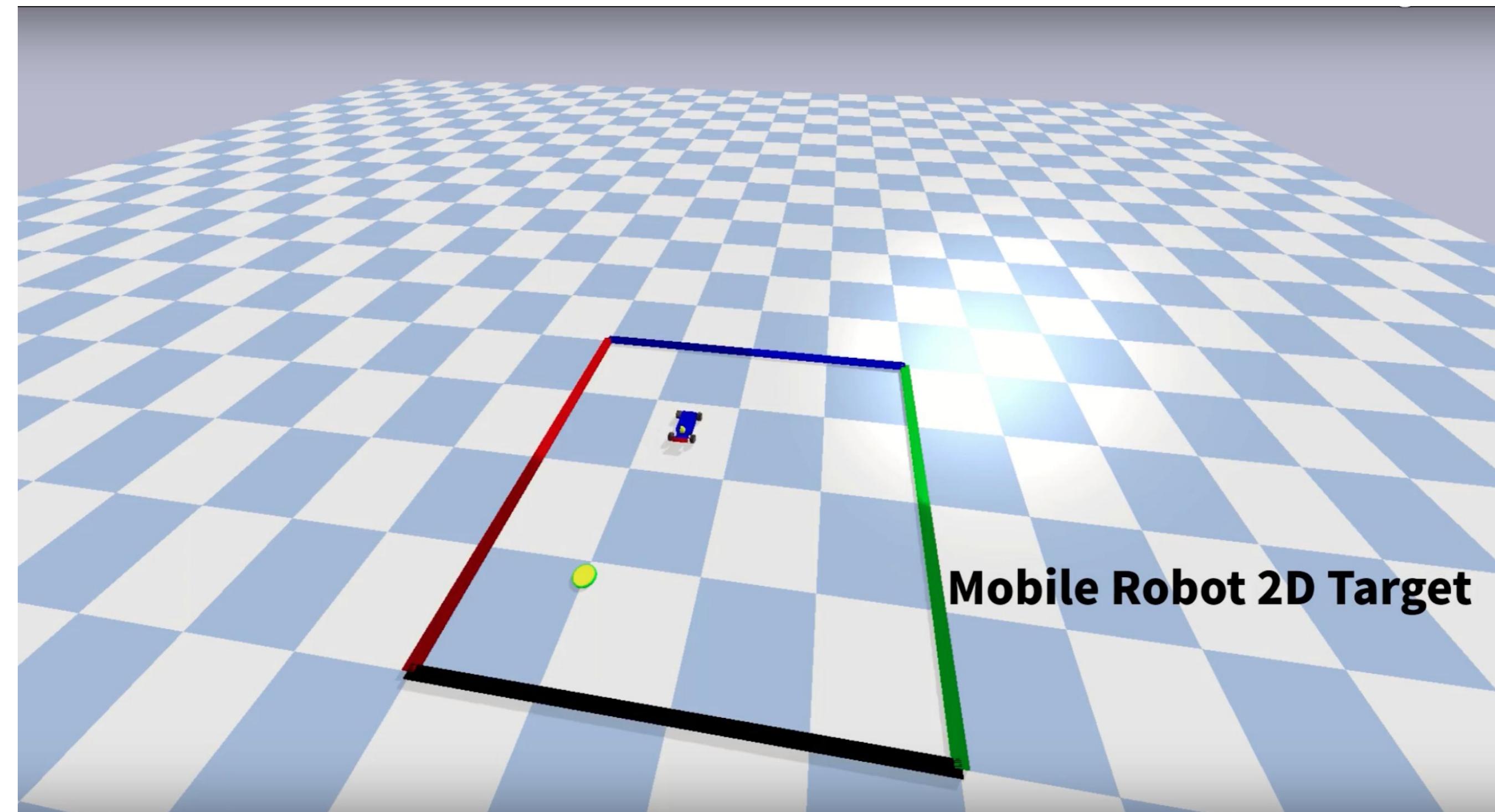
Figure 1: An N compartment simulated Octopus arm. Each constant area compartment C_i is defined by its surrounding 2 longitudinal muscles (ventral and dorsal) and 2 transverse muscles. Circles mark the $2N + 2$ point masses in which the arm's mass is distributed. In the bottom right one compartment is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

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

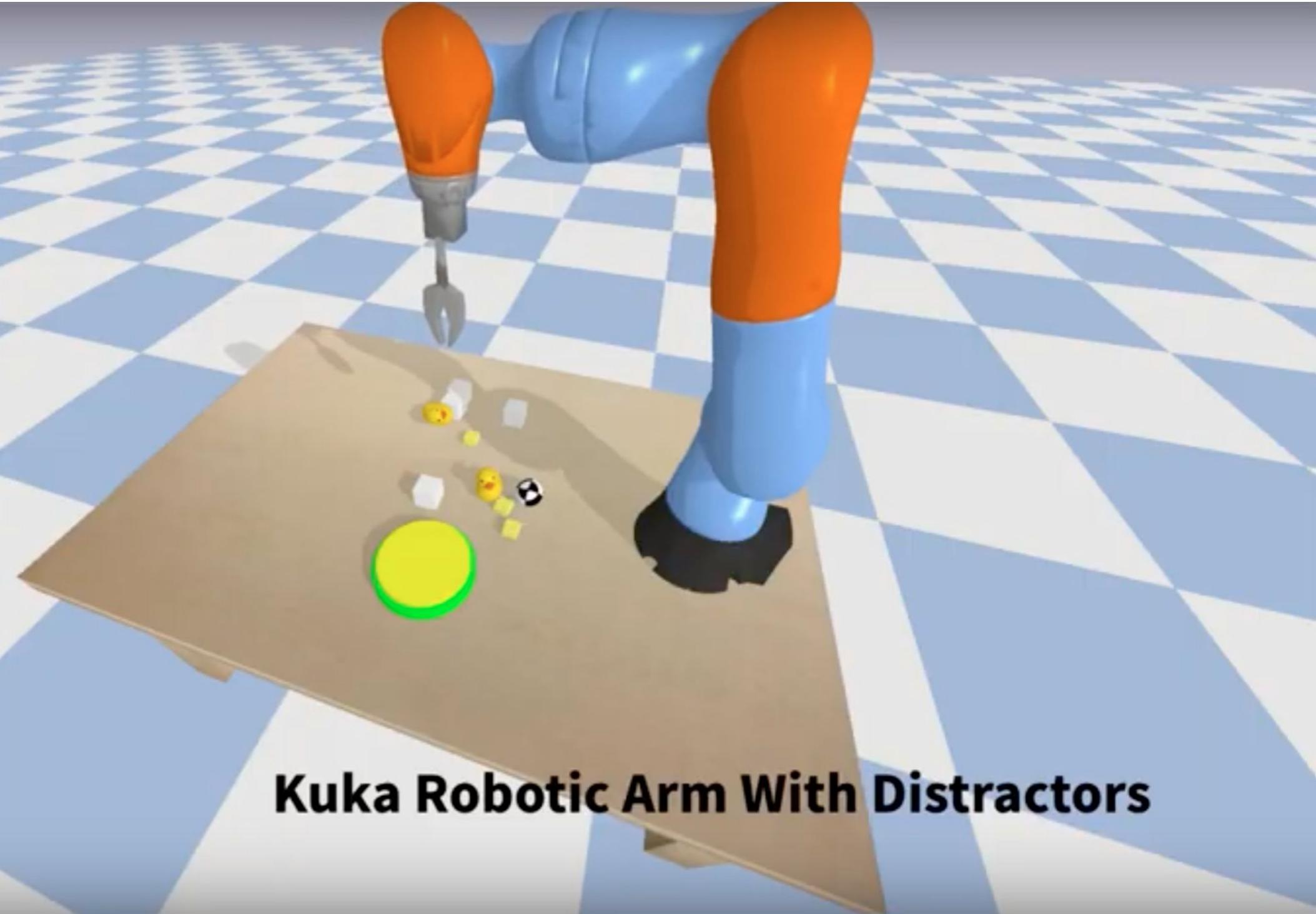
S-RL Toolbox



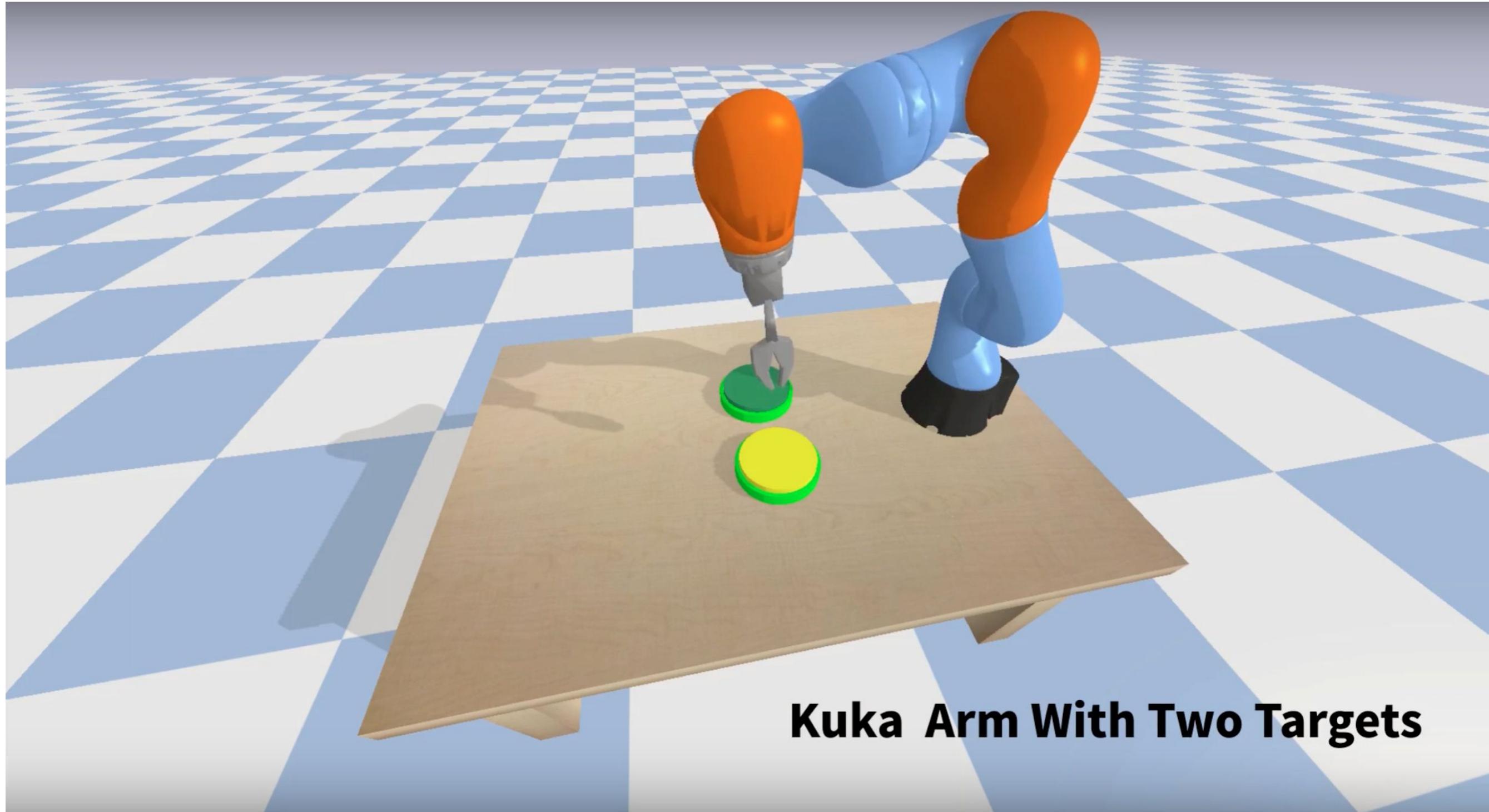
S-RL Toolbox



S-RL Toolbox

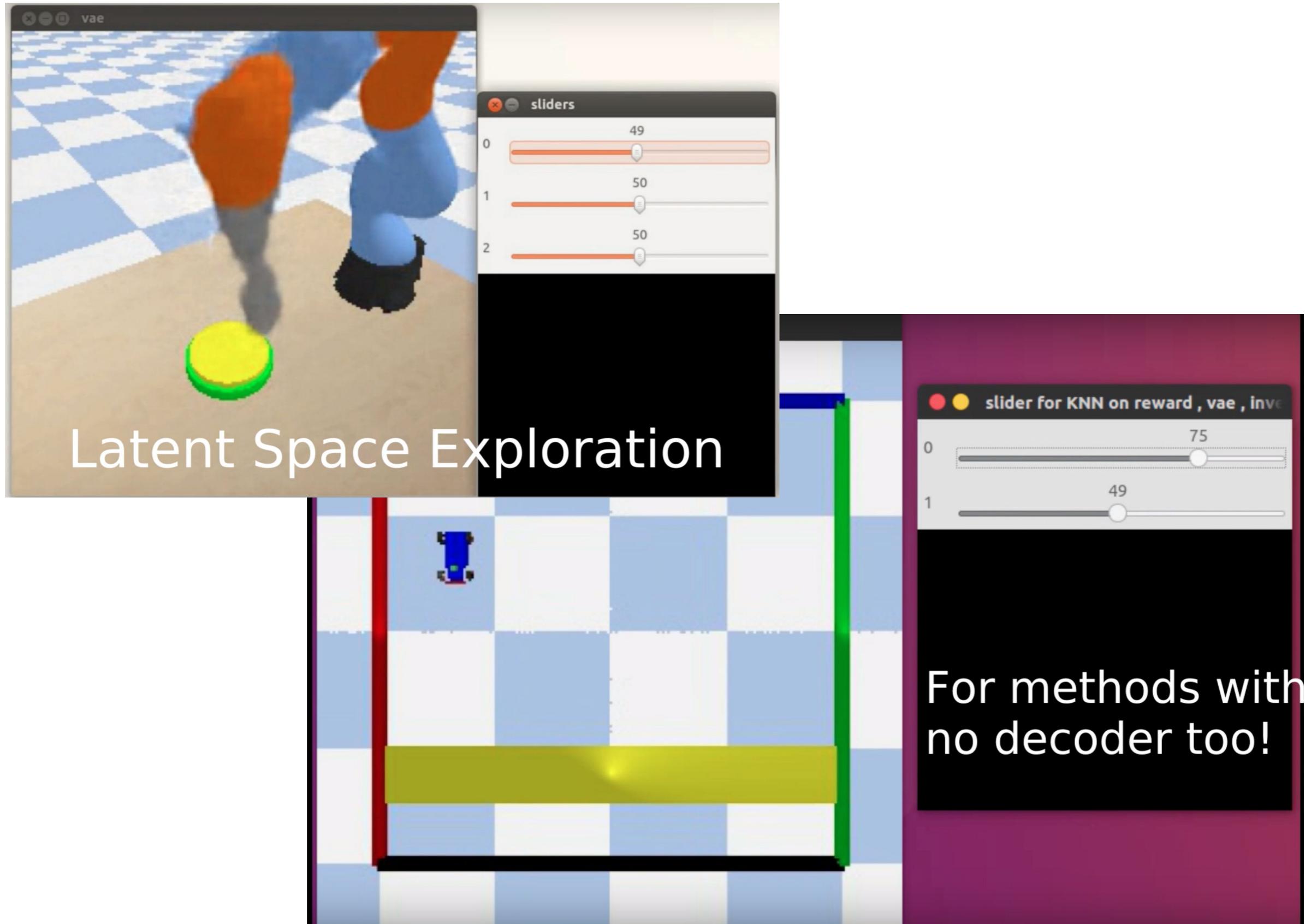


S-RL Toolbox



Kuka Arm With Two Targets

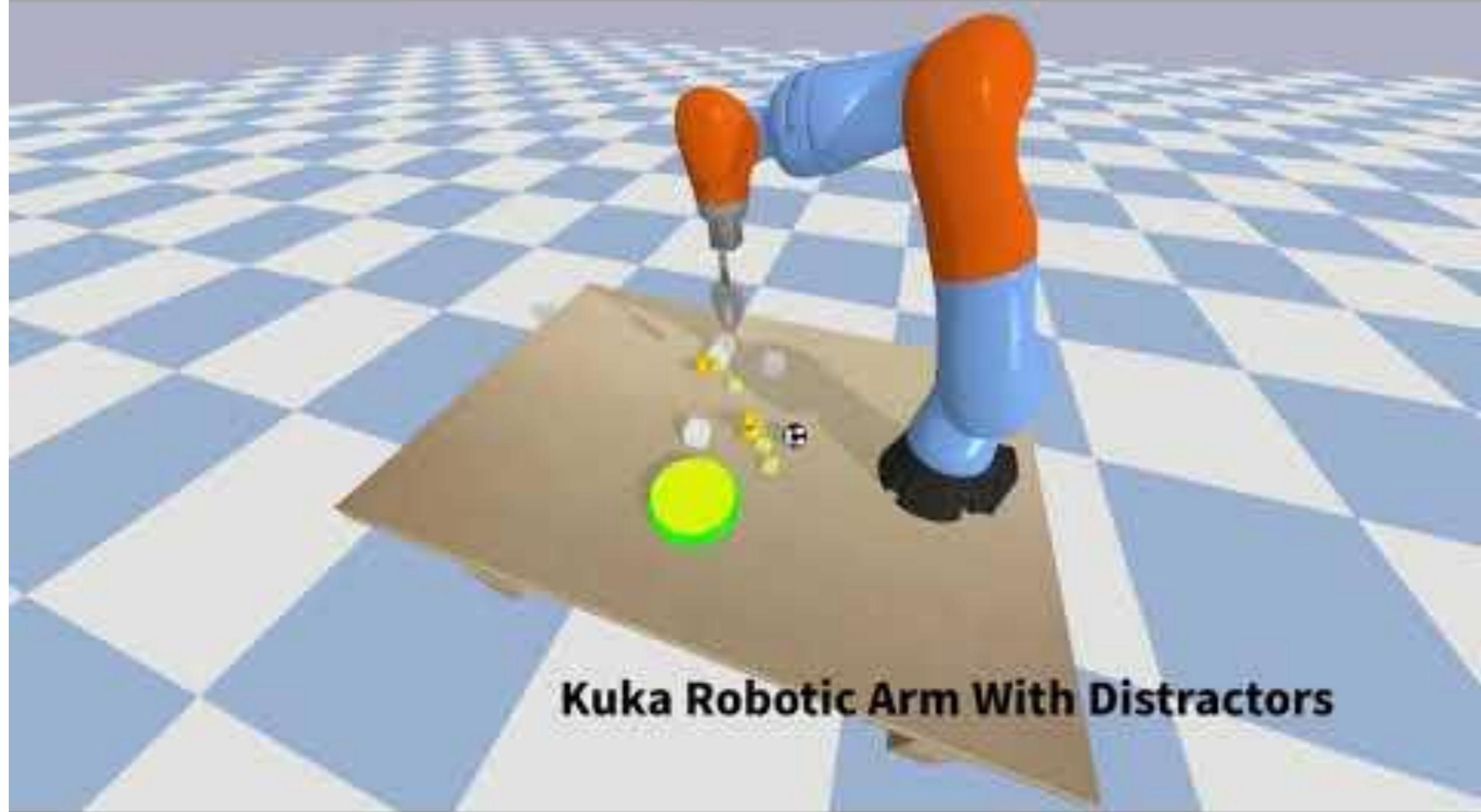
S-RL Toolbox



S-RL Toolbox



Real Baxter Robot Static Target



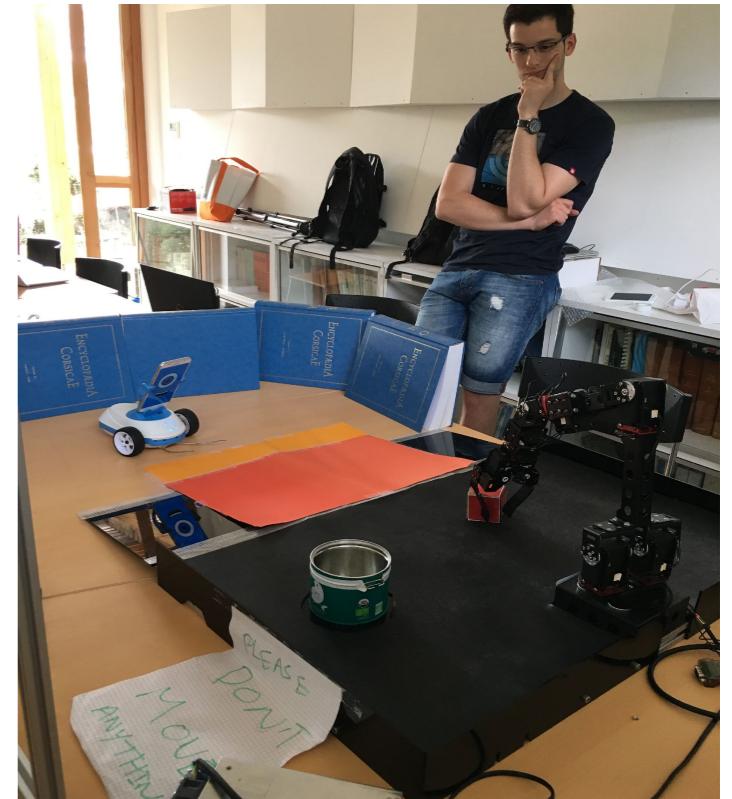
KUKA Robotic Arm With Distractors

Future Perspectives

- More & More complex tasks: Rewards?



Robobo



Intrinsically Motivated Multi-Task Reinforcement Learning

State Representation Learning Approaches

