

# Explainable Lifelong Streaming Learning

CHUKIONG LOO

[ckloo.um@um.edu.my](mailto:ckloo.um@um.edu.my)

[www.um.edu.my](http://www.um.edu.my)

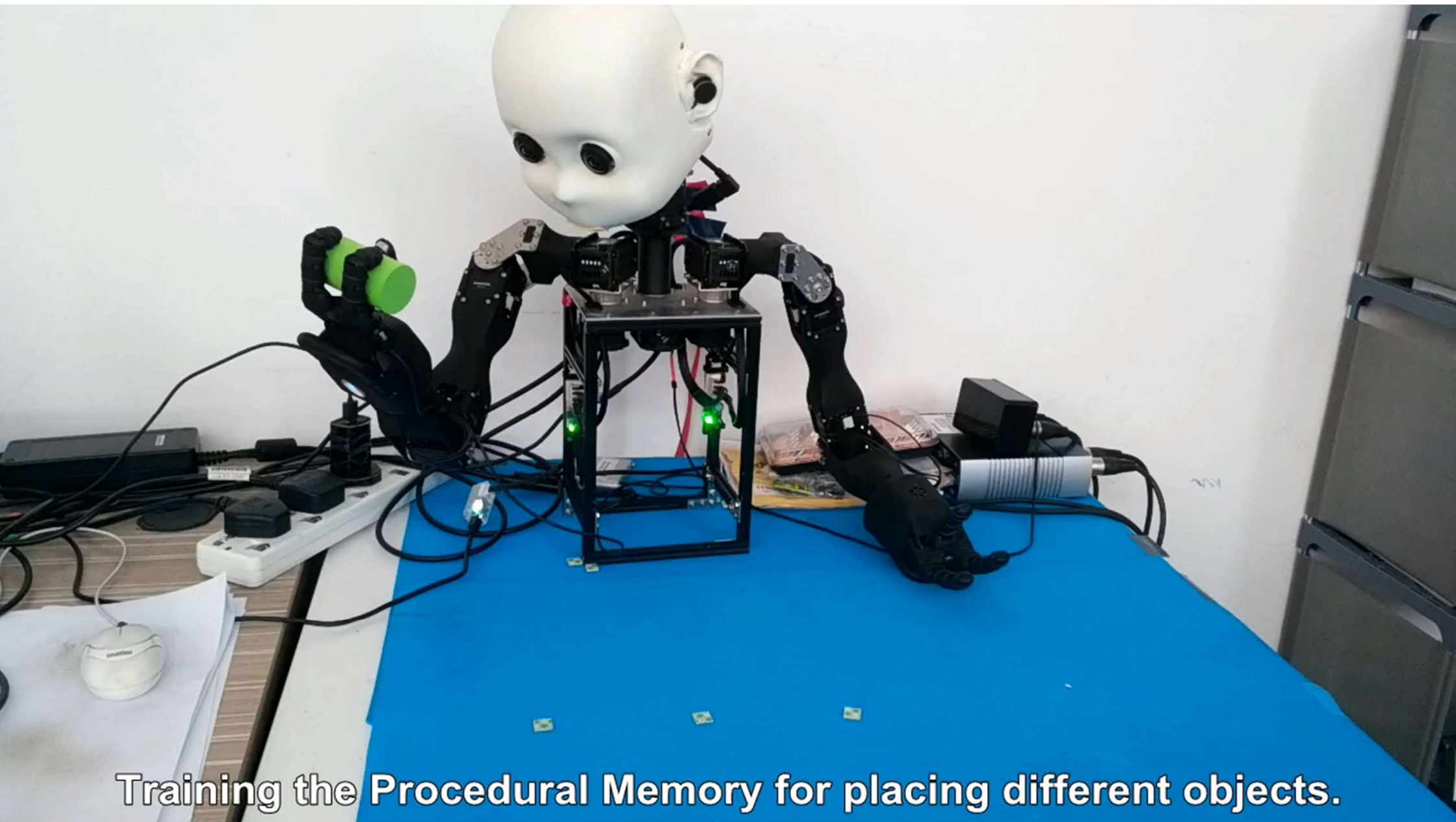
Faculty of Computer Science & Information  
Technology



**UNIVERSITI  
MALAYA**

# Outlines

- Continual Learning Overview
- Introduction to Lifelong Learning
- Lifelong Learning Setup
- Explainable Lifelong Streaming Learning
- Future Works



Training the Procedural Memory for placing different objects.





Test 2: Localize and recognize multiple objects.

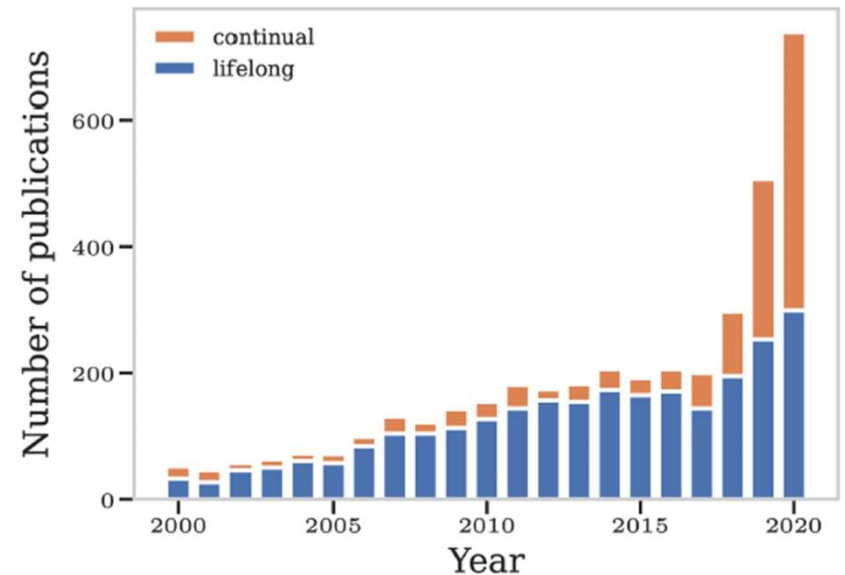


# Introduction to 'Continual Learning' literature

- 'Continual learning' vs. 'lifelong learning'
- Often used interchangeably
- Popularity of 'continual learning' more recent

We use these terms as follows:

- **Continual learning** (**narrow**) how to deal with non-stationarity in training data
- **Lifelong learning** (**broad**) an agent learning throughout its lifetime

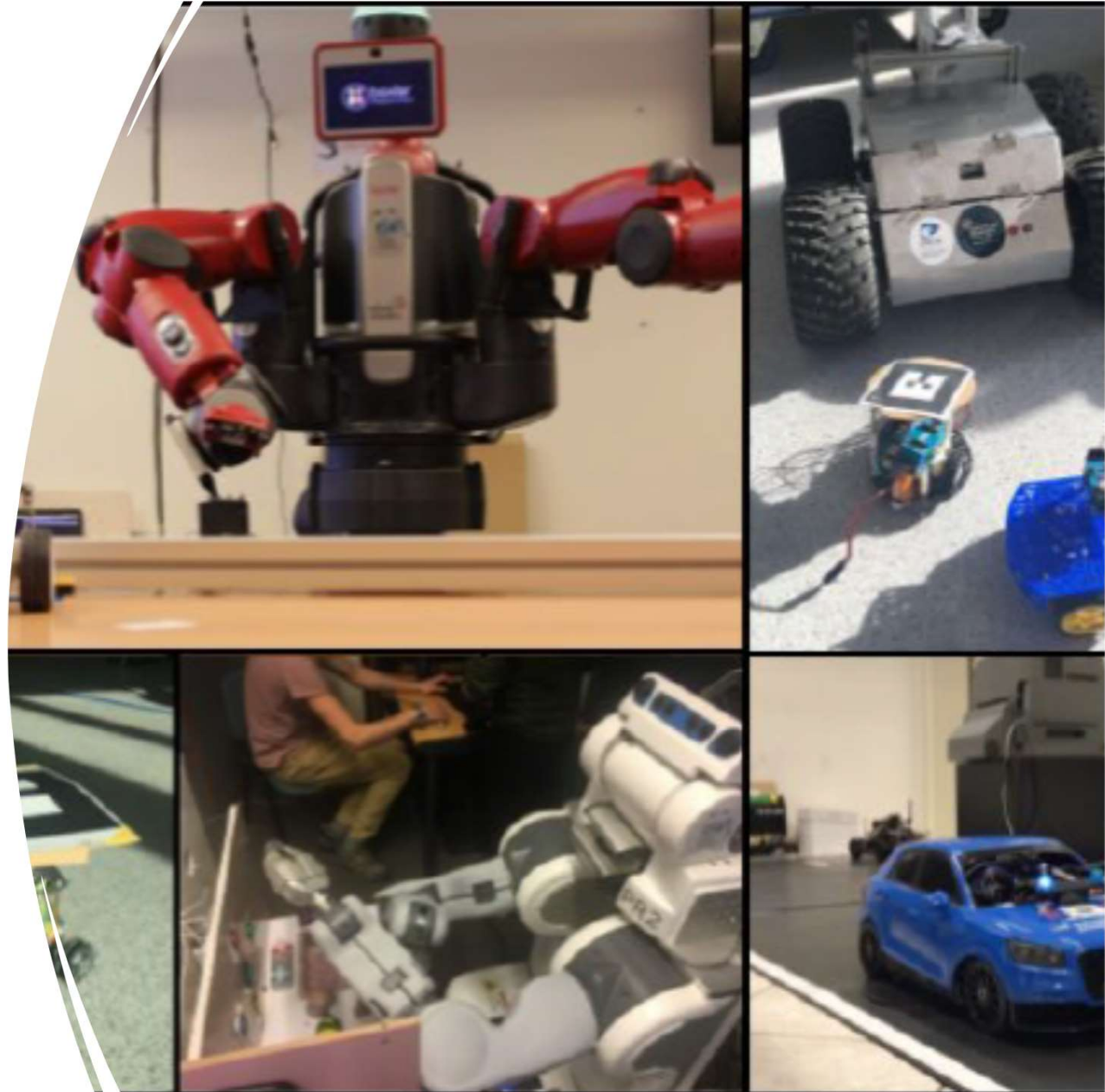


Number of machine learning publications per year, based on keyword occurrence in abstract.  
Source: [Mundt et al. \(2022, ICLR\)](#)

# Continual Learning Challenges in Practical Applications

---

- A robot acquiring new skills in different environment, adapting to new situations, learning new tasks



# Continual Learning Challenges in Practical Applications

---

- A self-driving car adapting to different environments (from a country road to a highway to a city)





# Continual Learning Challenges in Practical Applications

- Conversational agents adapting to different users, situations, tasks



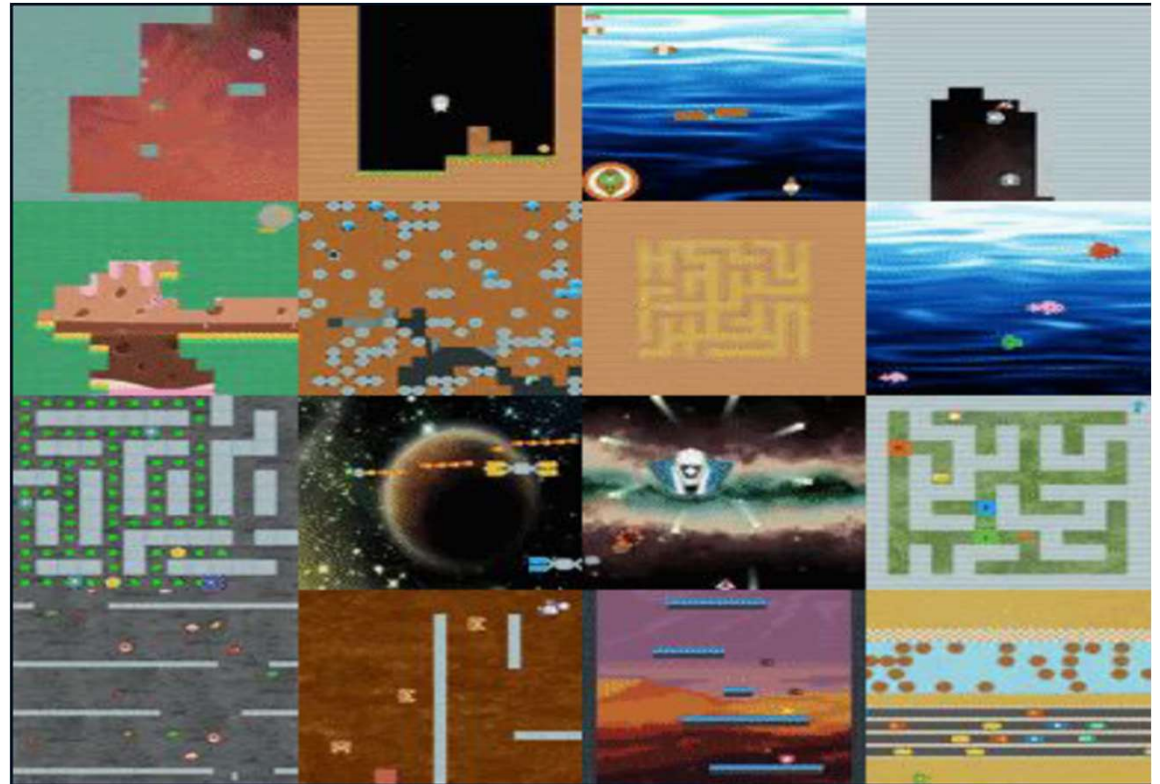
# Continual Learning Challenges in Practical Applications

- Medical applications: adapting to new patients, new hospital conditions



# Continual Learning Challenges in Practical Applications

- Multi-game environments  
(e.g. OpenAI gym)





# Static datasets: Controlled

Small scale, but (some) controlled acquisition parameters

| Image number | Object pose |                 |                | Illumination direction |                    |                     |
|--------------|-------------|-----------------|----------------|------------------------|--------------------|---------------------|
|              | Frontal     | 22.5 °<br>right | 22.5 °<br>left | Frontal                | ≈ 45 °<br>from top | ≈ 45 °<br>from side |
| 1            | x           |                 |                | x                      |                    |                     |
| 2            | x           |                 |                |                        | x                  |                     |
| 3            | x           |                 |                |                        |                    | x                   |
| 4            |             | x               |                | x                      |                    |                     |
| 5            |             | x               |                |                        | x                  |                     |
| 6            |             | x               |                |                        |                    | x                   |
| 7            |             |                 | x              | x                      |                    |                     |
| 8            |             |                 | x              |                        | x                  |                     |
| 9            |             |                 | x              |                        |                    | x                   |

Table 3: The labeling of images within each scale in the KTH-TIPS database.



Image #1



Image #2



Image #3



Image #4



Image #5



Image #6



Image #7



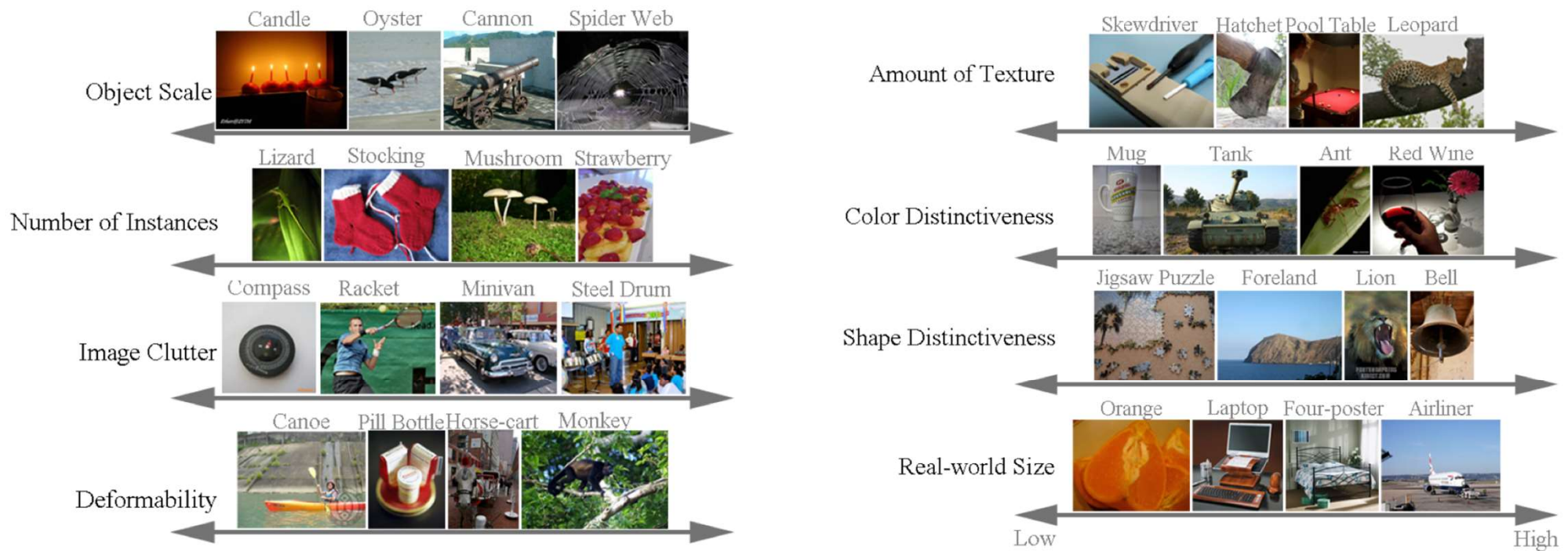
Image #8



Image #9

# Static datasets: large scale

A big focus of modern dataset has been on large scale & diversity



Russakovsky & Deng et al, "ImageNet Large Scale Visual Recognition Challenge, IJCV 2015, (challenges since 2010)

# Challenge: Data Stream



**50GB/s** streaming data.

**~30240 TB of data** after only a week.

**Impossible** to re-train the mini-spot brain from scratch and to **adapt fast**.

<https://www.bostondynamics.com/products/spot>



# Continual Learning....

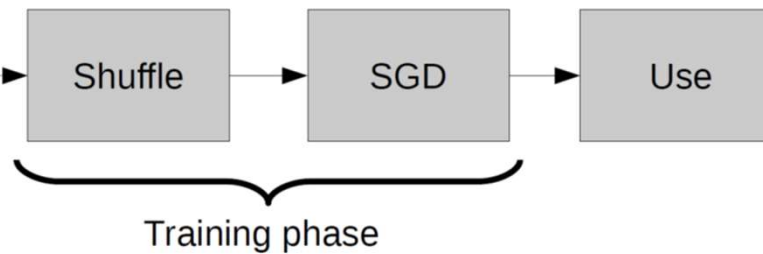
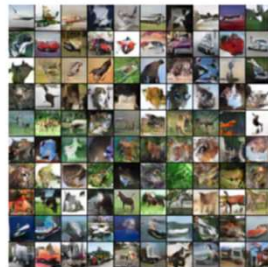
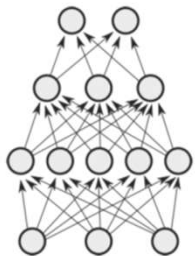
**Humans:** Continual learning, non stationary data



L Muntz, 1898  
J-H Fragonard, 1770  
S Koninck, 1643

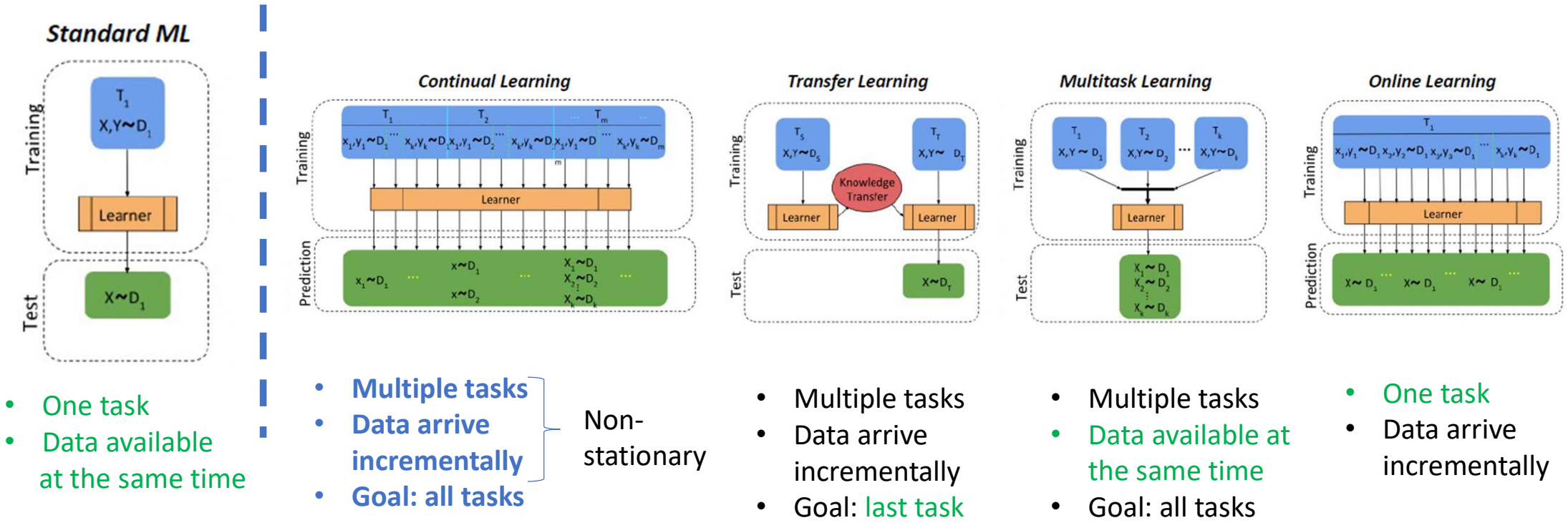
**Human-like learning**  
**Small data,  $n \rightarrow 1$**

**Machines:** Training phase, stationary data



**Supervised learning**  
**Big data,  $n \rightarrow \infty$**

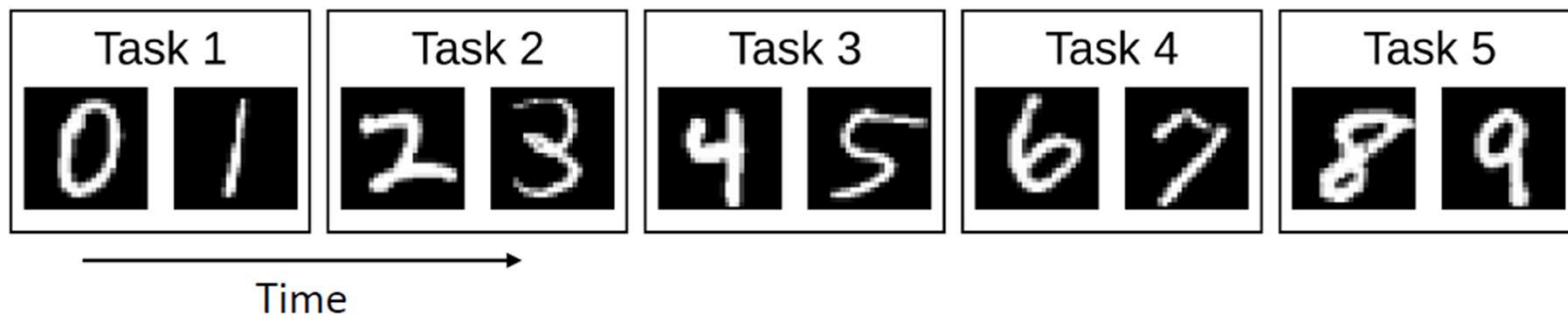
# Relation to other fields



Source: De Lange et al. (2021, TPAMI)

# The canonical continual learning example: Split MNIST 15

- MNIST dataset is split into multiple parts/episodes/tasks(\*) that must be learned sequentially
- After all tasks have been learned, the model should be good at all tasks
- Typically, **no or only a small amount of data** from past tasks can be stored



Important problem: ***catastrophic forgetting***

- When learning a new task, deep neural networks tend to rapidly forget past tasks

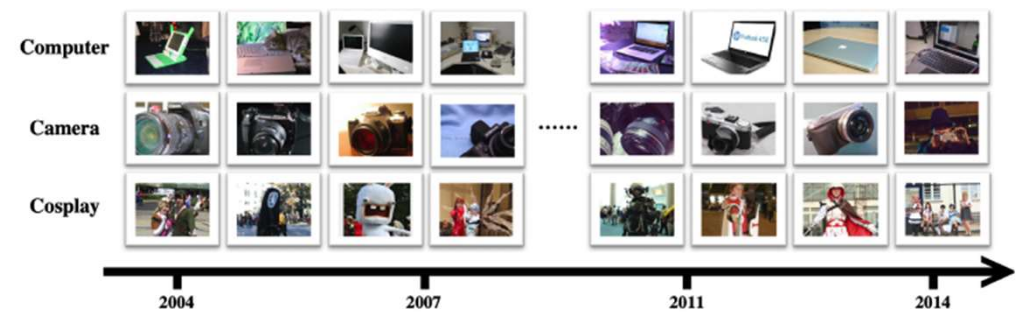


# Going beyond Split MNIST

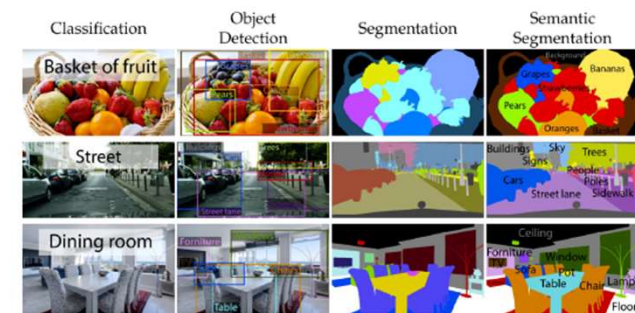
- Splitting up existing image datasets:
  - CIFAR-10
  - CIFAR-100
  - (Tiny)ImageNet
  - ....
- Datasets specific for continual learning
  - CORe-50
  - Stream-51
  - The CLEAR Benchmark
- Beyond Classification
  - Continual reinforcement learning
  - Continual object detection
  - Continual semantic segmentation
  - ....



Source: [van de Ven et al. \(2020, Nature Communications\)](#)



Source: [Lin et al. \(2021, NeurIPS Datasets and Benchmarks Track\)](#)



Source: [Toldo et al. \(2020, Technologies\)](#)



# Three Continual Learning Scenarios: intuitively 18

- Task-incremental learning (*Task-IL*)
  - Incrementally learn a set of clearly distinguishable tasks

**Important challenge:** achieve positive transfer between tasks



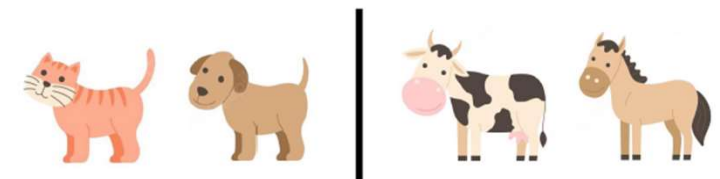
- Domain-incremental learning (*Domain-IL*)
  - Learn the same type of problem in different contexts

**Important challenge:** alleviate catastrophic forgetting



- Class-incremental learning (*Class-IL*)
  - Incrementally learn a growing number of classes

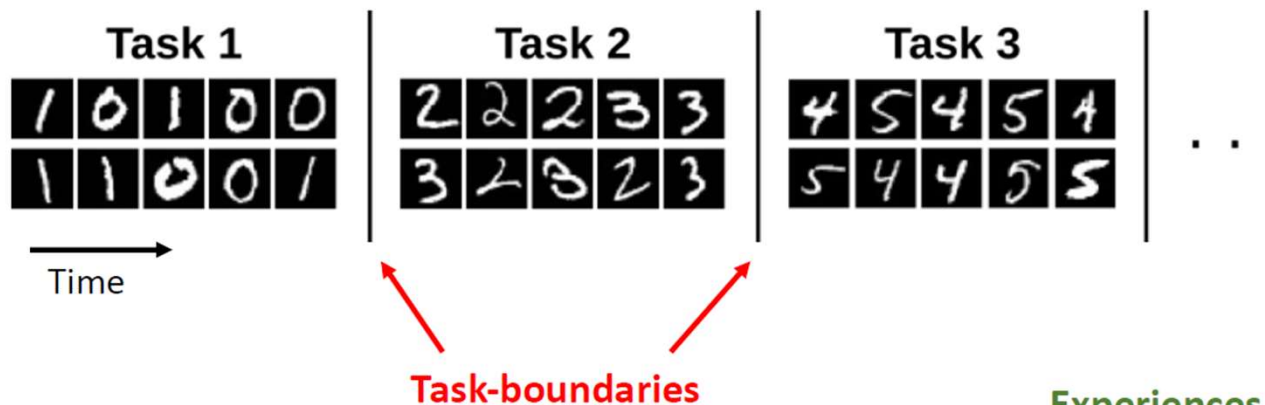
**Important challenge:** learn to discriminate between objects not observed together



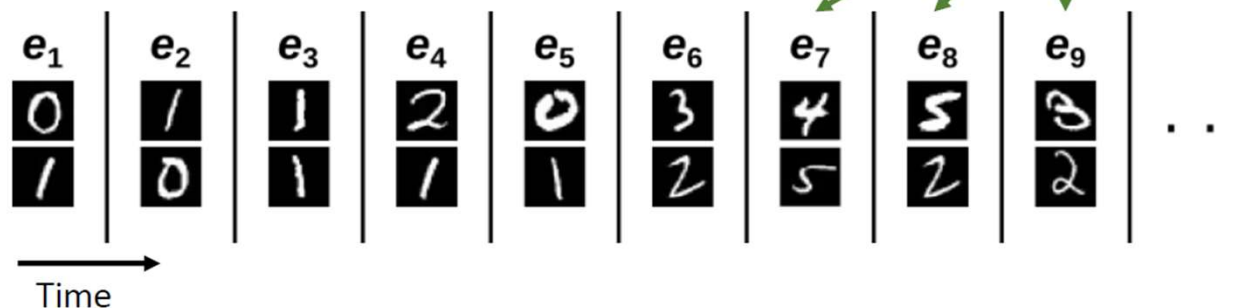
Images designed by Freepik

# Task-based vs. Task-free Continual Learning

*Task-based data stream*



*Task-free data stream*



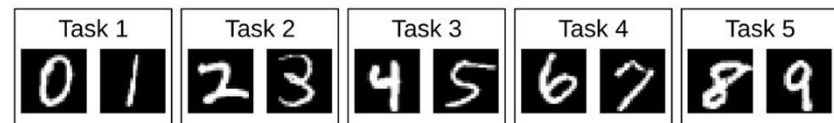


# Baselines: finetuning (lower target) & joint training (upper target)

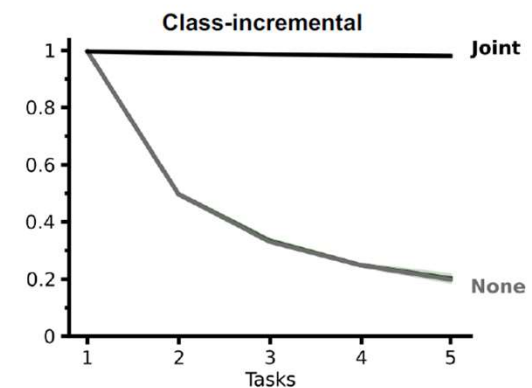
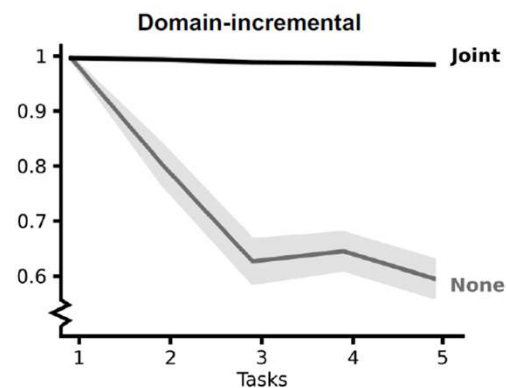
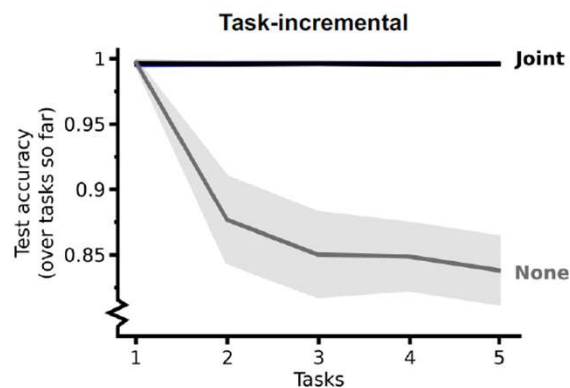
**None:** Network sequentially trained on each task in the standard way (lower target)

**Joint:** Network trained on all tasks at the same time (upper target)

Split MNIST:

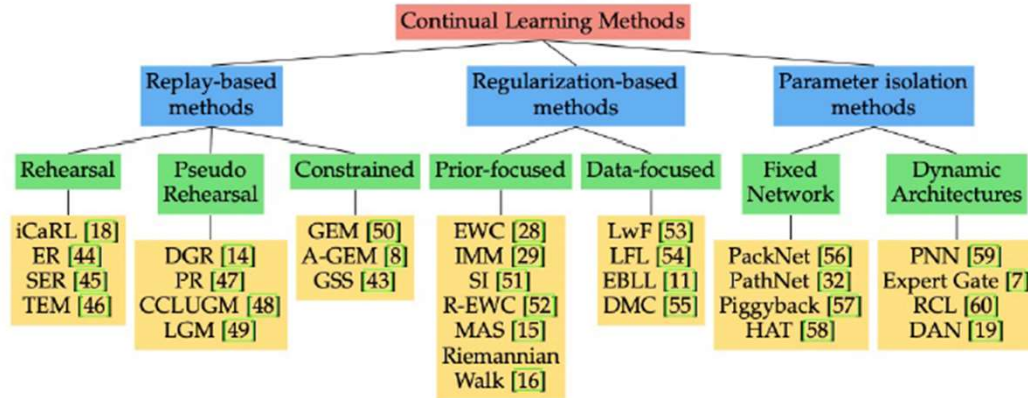


| <i>Type of choice</i> |                                           |
|-----------------------|-------------------------------------------|
| Task-incremental      | Choice between the two digits of the task |
| Domain-incremental    | Is the digit odd or even?                 |
| Class-incremental     | Choice between all ten digits             |

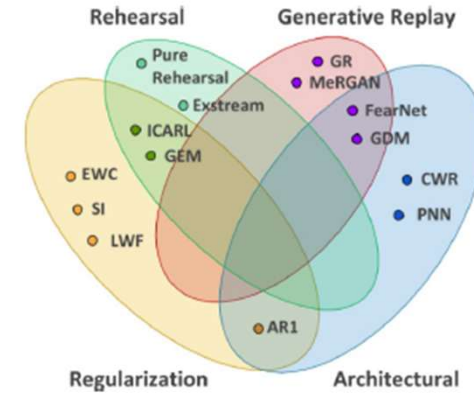


Code for these experiments: <https://github.com/GMvandeVen/continual-learning>

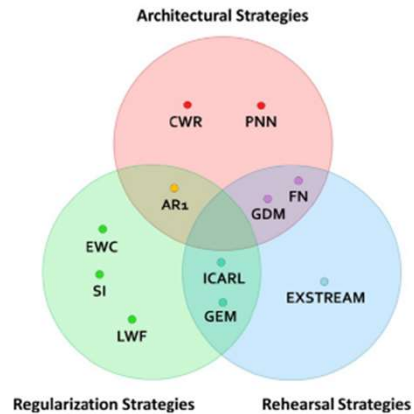
# Continual Learning Categorization



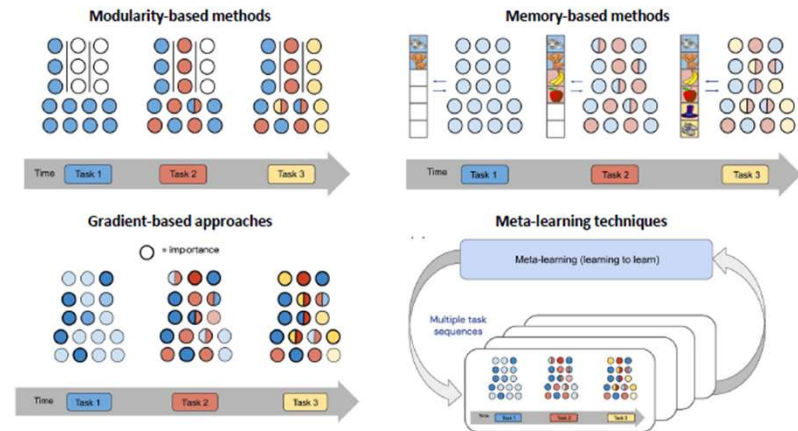
Source: [De Lange et al. \(2021, TPAMI\)](#)



Source: [Lesort et al. \(2020, Information Fusion\)](#)



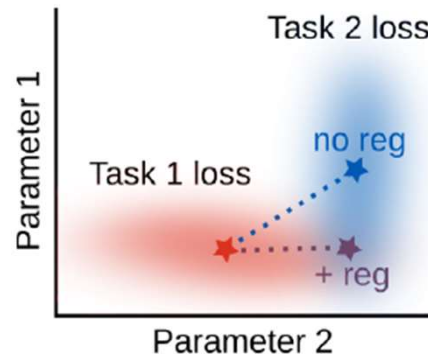
Source: [Maltoni & Lomonaco \(2019, Neural Networks\)](#)



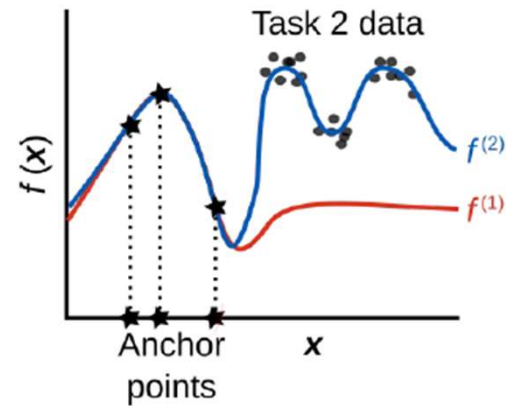
Source: [Hadsell et al. \(2020, Trends in Cognitive Sciences\)](#)

# Continual Learning Strategies

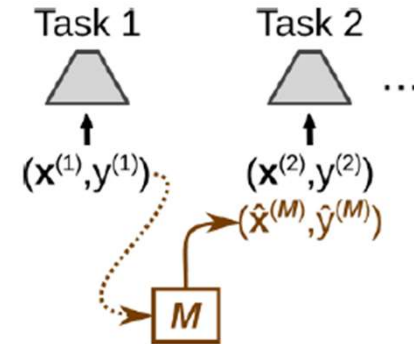
### Parameter regularization



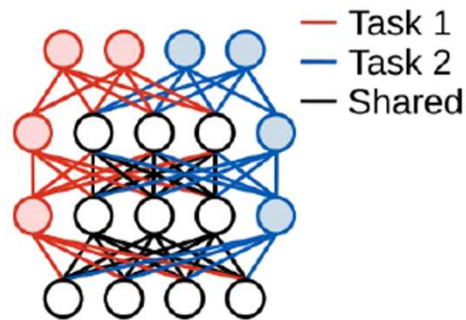
### Functional regularization



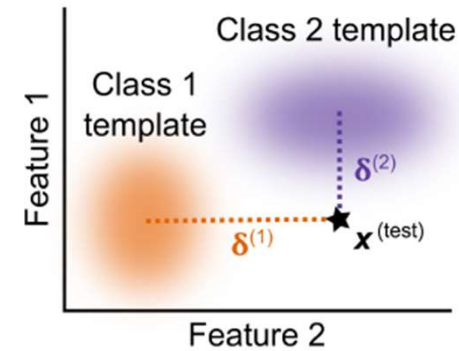
### Replay



### Context-specific components



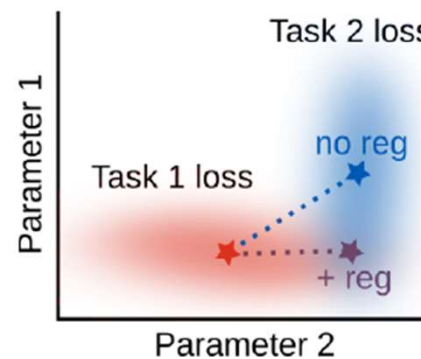
### Template-based classification



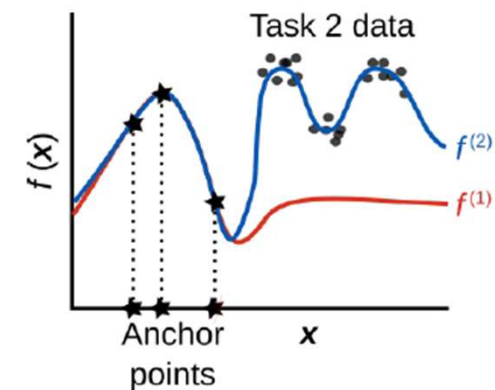
# Regularization

- In continual learning, regularization typically means adding a penalty term to the loss function to encourage the model to stay close to a previous version of itself.
- Often, the version relative to which changes are penalized is a copy of the model stored after finishing training on the last task
- Two forms of regularization:

Parameter regularization



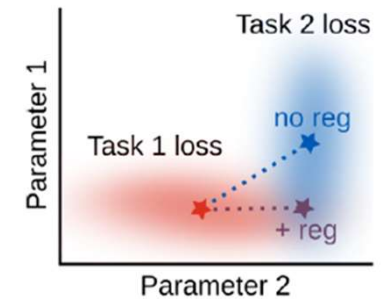
Functional regularization





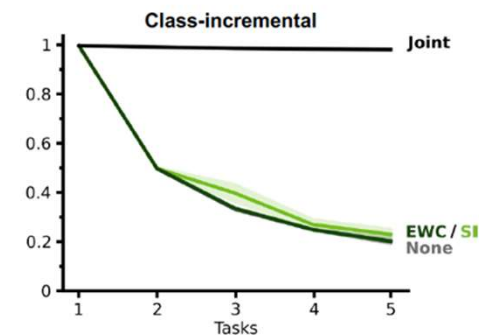
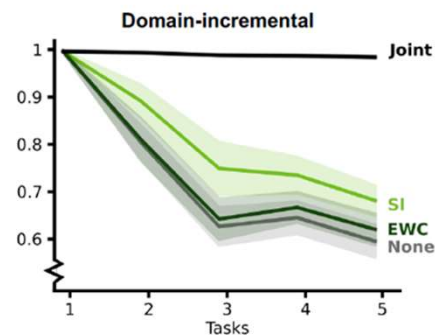
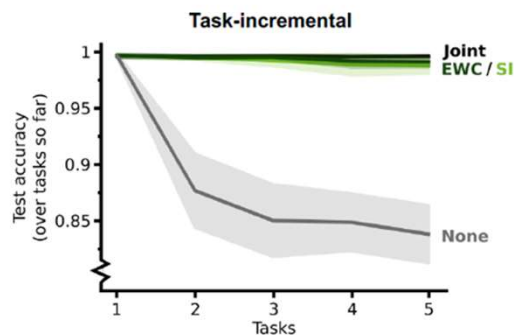
# Parameter Regularization

- Parameters important for past tasks are encouraged not to change too much when learning a new task
- Can often be interpreted as sequential approximate Bayesian inference on the network's parameters
- Representative methods:
  - Elastic Weight Consolidation [EWC] (Kirkpatrick et al., 2017 PNAS)
  - Synaptic Intelligence [SI] (Zenke et al., 2017 ICML)



$$\mathcal{L}_{\text{total}} = \mathcal{L} + \|\theta - \theta^*\|_{\Sigma}$$

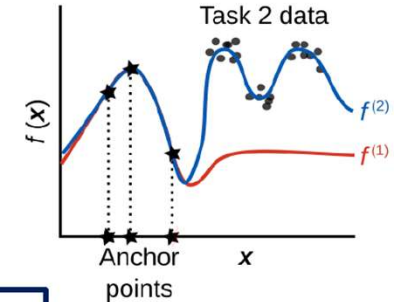
$\theta^*$ : parameters relative to which changes are penalized  
 $\Sigma$ : estimate of how important parameters are  
 $\|\cdot\|_{\Sigma}$ : weighted norm



Code for these experiments: <https://github.com/GMvandeVen/continual-learning>

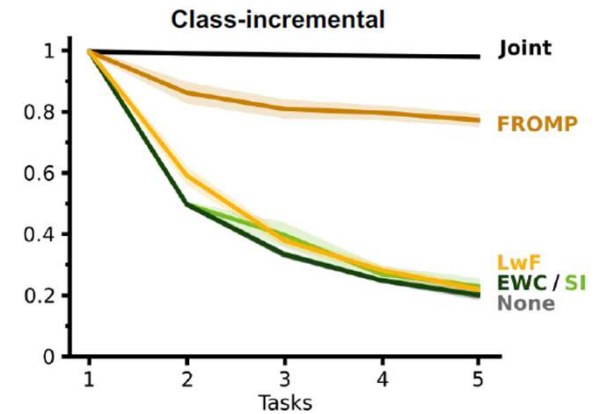
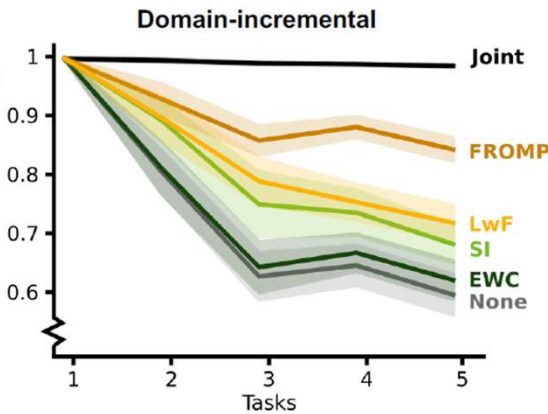
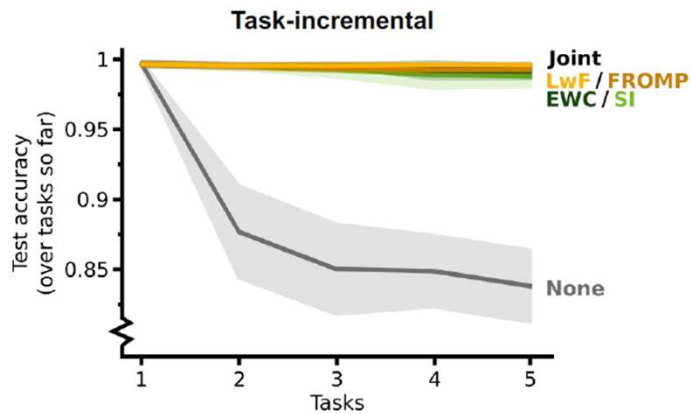
# Functional Regularization

- The input-output mapping learned previously is encouraged not to change too much at a particular set of inputs (the ‘anchor points’)
- Also referred to as knowledge distillation
- Representative methods:
  - Learning without Forgetting [LwF] (Li & Hoiem, 2017 TPAMI)
  - Functional Regularization Of Memorable Past [FROMP] (Pan et al., 2020 NeurIPS)



$$\mathcal{L}_{\text{total}} = \mathcal{L} + \langle f_{\theta}, f_{\theta^*} \rangle_{\mathcal{A}}$$

$f_{\theta^*}$ : function relative to which changes are penalized  
 $\mathcal{A}$ : set of ‘anchor points’ at which the divergence between  $f_{\theta}$  and  $f_{\theta^*}$  is measured

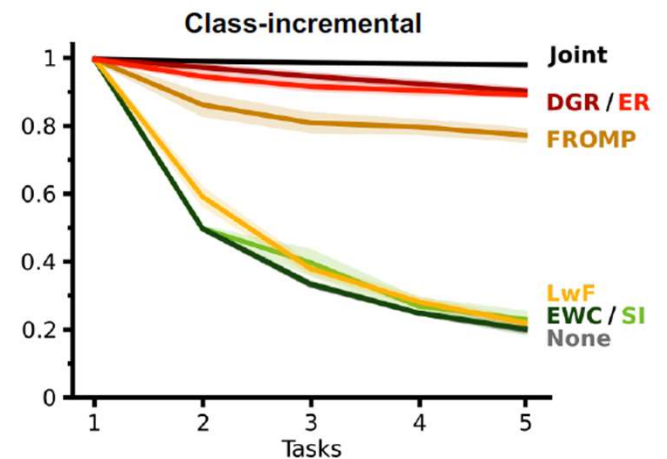
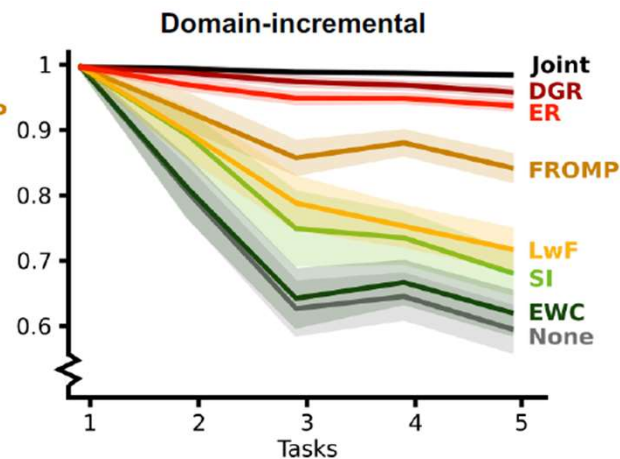
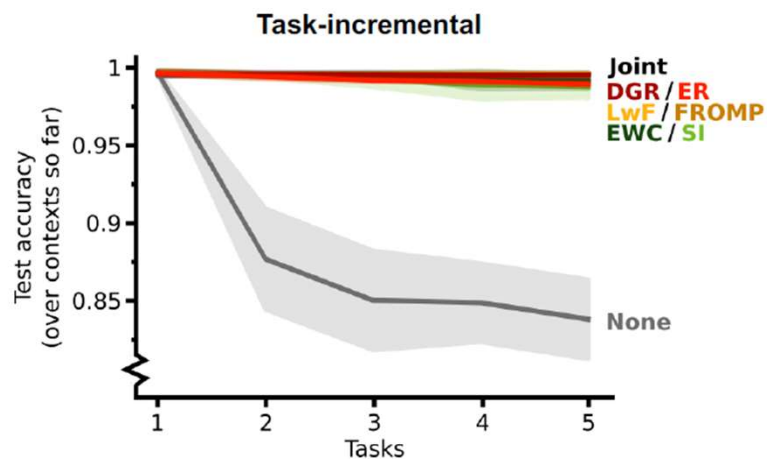
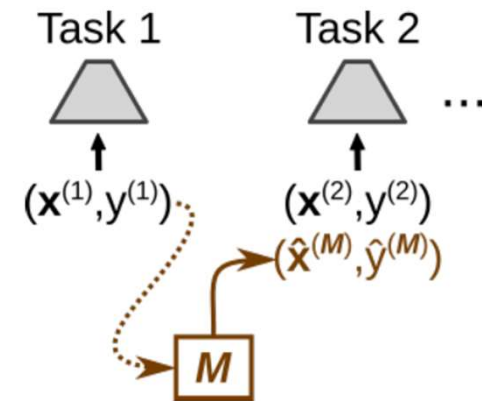


Memory buffer size (FROMP): 100 examples per class

Code for these experiments: <https://github.com/GMvandeVen/continual-learning>

# Replay

- Current training data is complemented with data representative of past observations
- The replayed data can be sampled from a memory buffer or a generative model
- Representative methods:
  - Experience Replay [ER] (Chaudhry et al., 2019 arXiv)
  - Deep Generative Replay [DGR] (Shin et al., 2017 NeurIPS)

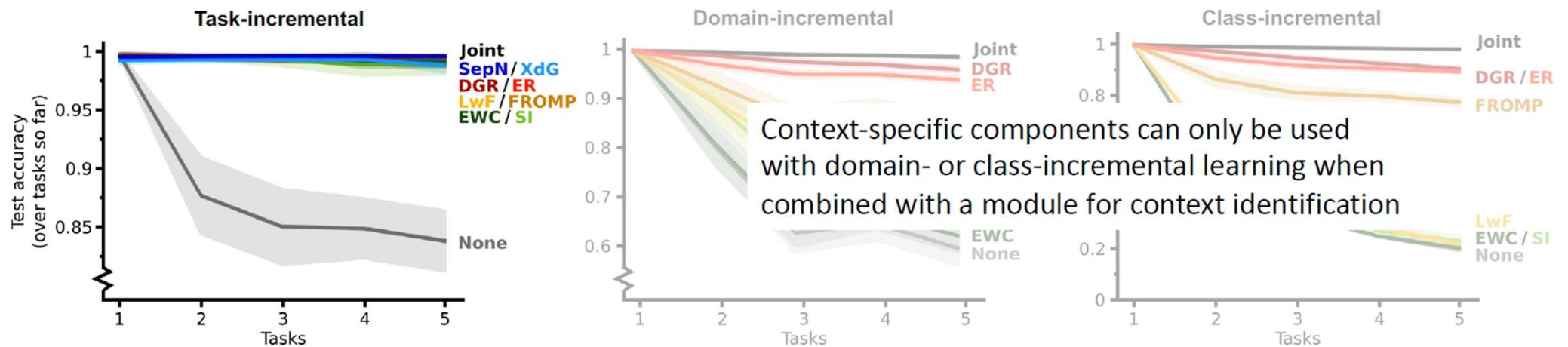
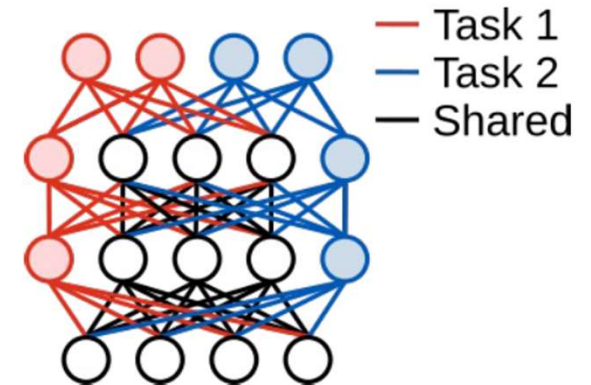


Memory buffer size (**FROMP**): 100 examples per class

Code for these experiments: <https://github.com/GMvandeVen/continual-learning>

# Context-specific components

- Parts of the network are only used for specific tasks
- Commonly used example: multi-headed output layer
- Requires knowledge of task identity at test time
  - Context-dependent Gating [XdG] (Masse et al., 2018 PNAS)
  - Separate Networks [SepN]



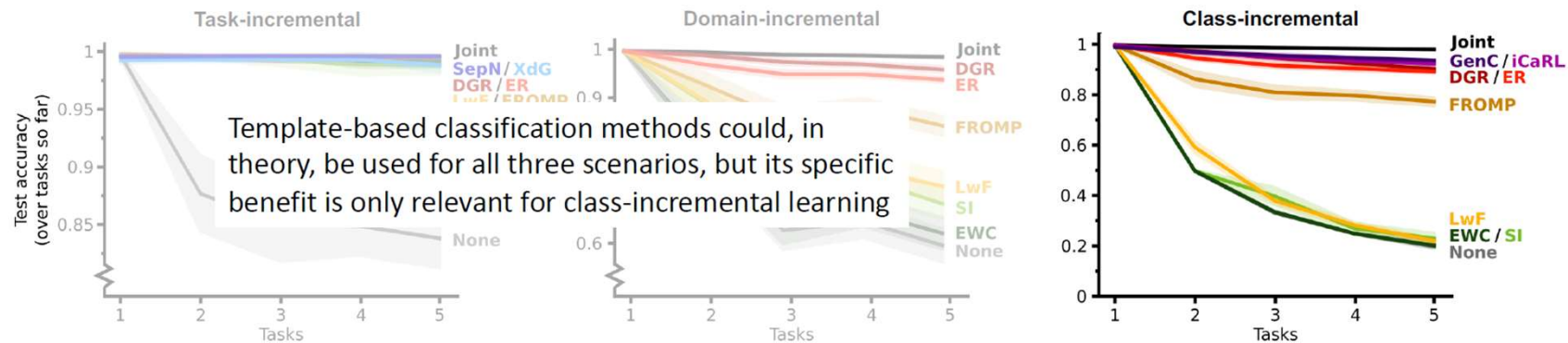
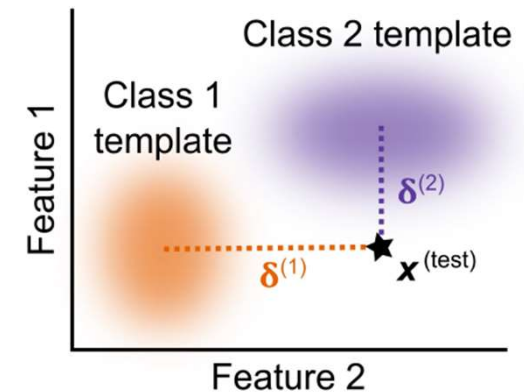
Memory buffer size (FROMP): 100 examples per class

Code for these experiments: <https://github.com/GMvandeVen/continual-learning>



# Template-based classification

- A ‘template’ is learned for each class, and classification is performed based on which template is most suitable for sample to be classified
- Examples of templates are prototypes or generative models
- Allows comparing classes ‘at test time’, rather than during training
- Representative methods
  - Incremental Classifier and Representation Learning [iCaRL] (Rebuffi et al., 2017 CVPR)
  - Generative Classifier [GenC] (van de Ven et al., 2021 CVPR-W)



Memory buffer size (FROMP): 100 examples per class

Code for these experiments: <https://github.com/GMvandeVen/continual-learning>

# Overview

| Strategy                      | Method                | Budget | GM  | Task-IL              | Domain-IL            | Class-IL             |
|-------------------------------|-----------------------|--------|-----|----------------------|----------------------|----------------------|
| Baselines                     | None – lower target   |        |     | 84.32 ( $\pm 0.99$ ) | 60.13 ( $\pm 1.66$ ) | 19.89 ( $\pm 0.02$ ) |
|                               | Joint – upper target  |        |     | 99.67 ( $\pm 0.03$ ) | 98.59 ( $\pm 0.05$ ) | 98.17 ( $\pm 0.04$ ) |
| Context-specific components   | Separate Networks     | -      | -   | 99.57 ( $\pm 0.03$ ) | -                    | -                    |
|                               | XdG                   | -      | -   | 99.10 ( $\pm 0.10$ ) | -                    | -                    |
| Parameter regularization      | EWC                   | -      | -   | 99.06 ( $\pm 0.15$ ) | 63.03 ( $\pm 1.58$ ) | 20.64 ( $\pm 0.52$ ) |
|                               | SI                    | -      | -   | 99.20 ( $\pm 0.11$ ) | 66.94 ( $\pm 1.13$ ) | 21.20 ( $\pm 0.57$ ) |
| Functional regularization     | LwF                   | -      | -   | 99.60 ( $\pm 0.03$ ) | 71.18 ( $\pm 1.42$ ) | 21.89 ( $\pm 0.32$ ) |
|                               | FROMP                 | 100    | -   | 99.12 ( $\pm 0.13$ ) | 84.86 ( $\pm 1.02$ ) | 77.38 ( $\pm 0.64$ ) |
| Replay                        | DGR                   | -      | Yes | 99.50 ( $\pm 0.03$ ) | 95.57 ( $\pm 0.30$ ) | 90.35 ( $\pm 0.24$ ) |
|                               | BI-R                  | -      | Yes | 99.61 ( $\pm 0.03$ ) | 97.26 ( $\pm 0.15$ ) | 94.41 ( $\pm 0.15$ ) |
|                               | ER                    | 100    | -   | 98.98 ( $\pm 0.07$ ) | 93.75 ( $\pm 0.24$ ) | 88.79 ( $\pm 0.20$ ) |
|                               | A-GEM                 | 100    | -   | 98.54 ( $\pm 0.10$ ) | 87.67 ( $\pm 1.33$ ) | 65.10 ( $\pm 3.64$ ) |
| Template-based classification | Generative Classifier | -      | Yes | -                    | -                    | 93.82 ( $\pm 0.06$ ) |
|                               | iCaRL                 | 100    | -   | -                    | -                    | 92.49 ( $\pm 0.12$ ) |

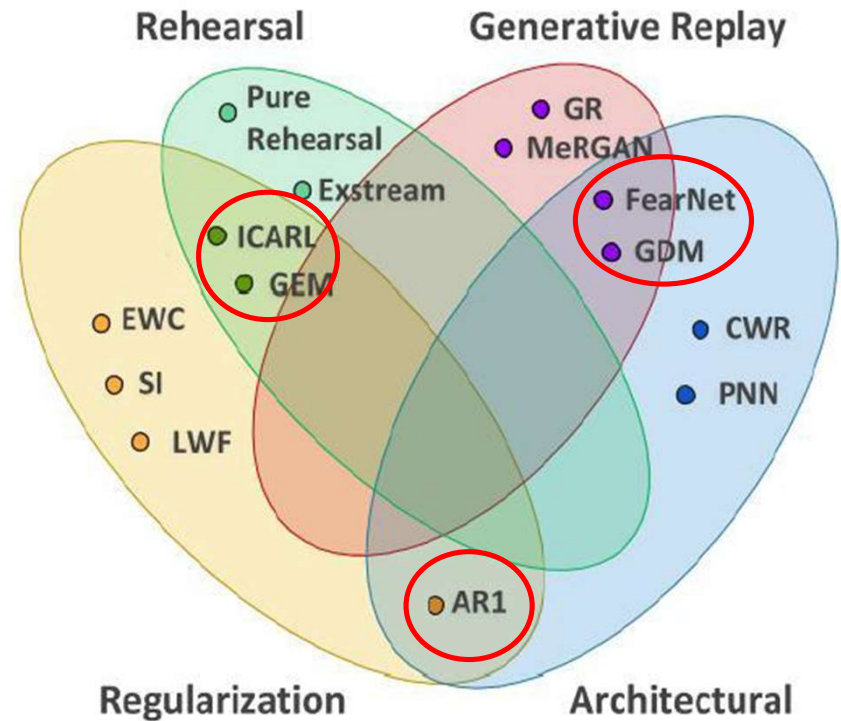
Reported is the final test accuracy (as percentage, averaged over all contexts) of all compared methods on the Split MNIST protocol, which is performed according to all three scenarios. The experiments followed the academic continual learning setting and context identity information was available during training. The column 'Budget' indicates the number of examples per class that was allowed to be stored in a memory buffer. The column 'GM' indicates whether a generative model was learned, for which additional network capacity was used. Each experiment was performed 20 times with different random seeds, reported is the mean ( $\pm$ s.e.m.) over these runs.

Source: *van de Ven et al. (2022, Nature Machine Intelligence)*

# Hybrid Models

# Why Hybrid?

- Each CL algorithm has advantages and disadvantages and works best within specific scenarios
- Such approaches are often orthogonal with respect to each other
- Biological learning systems seems to apply several approaches for learning continually
- Hybrid approaches are underexplored and may potentially find better Effectiveness-Efficiency trade-offs



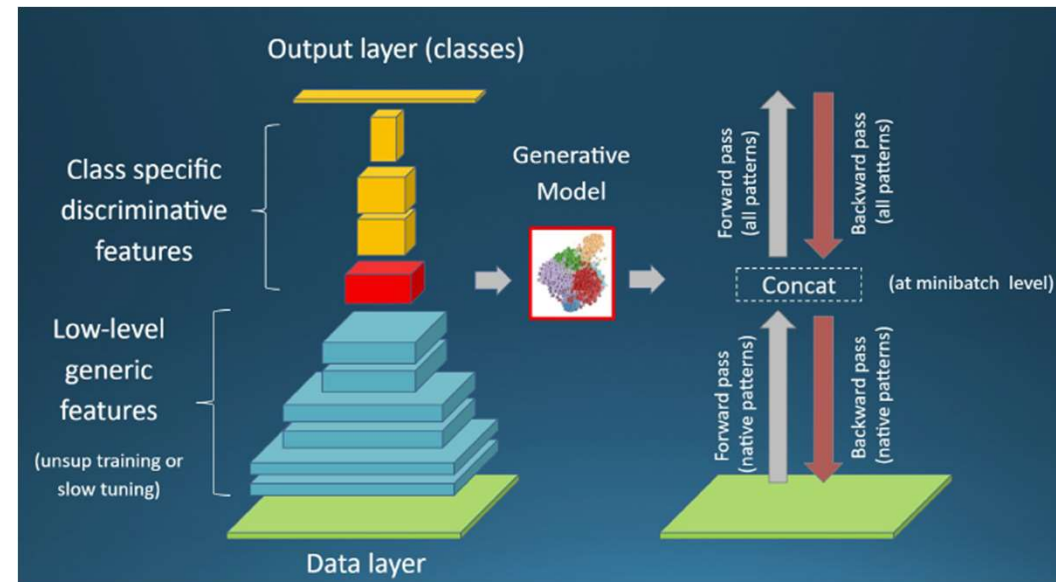


# ARI and (Negative) Generative Replay?

32

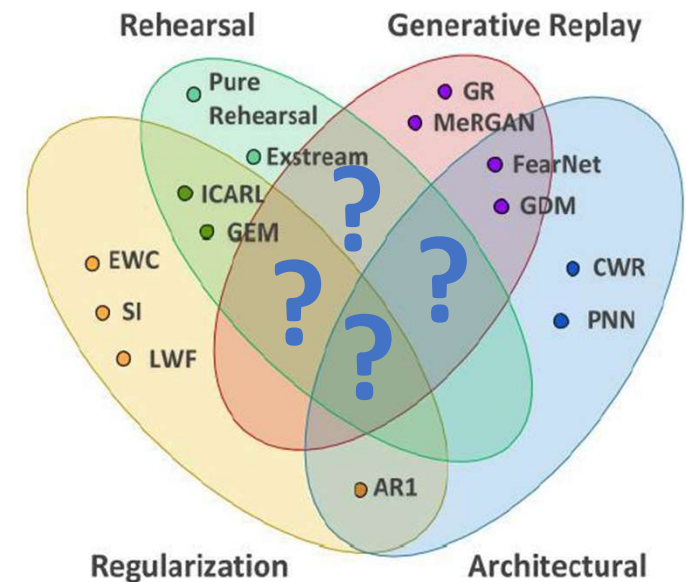
## Key Aspects

- Generative Replay is often difficult to scale (quality and diversity), what about generative latent replay?
- Sharing weights between the discriminator and the generator is possible
- Incremental training of the generator in the loop
- Negative replay: use generated patterns as negative examples only



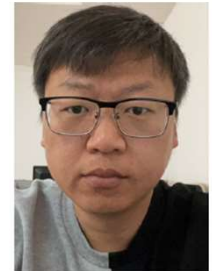
# Summary and Next Steps

- Hybrid approaches are more complex and more difficult to parametrize in general but they can provide Effectiveness-Efficiency trade-offs.
- Such approaches are still not well investigated but offer a nice path for future research explorations.
- They are often among the winning approaches in continual learning challenges
- More flexible and tunable algorithms (and possible self-adjusting hybrid approaches) may be quite interesting to investigate.



# Research Journey (2012 – 2023)

| Algorithm                                                                                                                                                                                                                            | Application                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Kernel Bayesian ART/ARTMAP/Associative Memory</li><li>• Topological Kernel Bayesian ART/ARTMAP</li><li>• Deep Kernel Bayesian ART/ARTMAP/Associative Memory</li></ul>                        | <b>Cognitive Robotics</b>                                     |
| <ul style="list-style-type: none"><li>• Kernel Bayesian ART/ARTMAP/Associative Memory</li><li>• Online Recurrent Kernel-based Extreme Learning/Reservoir Machine</li><li>• Genetic Ensemble Fuzzy Extreme Learning Machine</li></ul> | <b>Connected Health Analytics – Diagnosis &amp; Prognosis</b> |
| <ul style="list-style-type: none"><li>• Topological Kernel Bayesian ART/ARTMAP</li><li>• Biologically Inspired Vision</li><li>• Deep Kernel Bayesian ART/ARTMAP/Associative Memory</li></ul>                                         | <b>Topological 2D/SLAM/surface reconstruction</b>             |



# ART-based Triple Memory Model (2019-2022)

35

Multi-Channel Art-based Triplet Memory (2022)

**Multi-Modal ART-based Triple-Memory + Recurrent Kernel Machine**

WH Chin, N Kubota, CK Loo, (2022) An Episodic-Procedural Semantic Memory Model for Continuous Topological Sensorimotor Map Building, Cognitive Robotics and Adaptive Behavior, Interopen publisher.

Multi-Channel Art-based Dual-Memory (2020)

**Multi-Modal ART-based Dual-Memory + Recurrent Kernel Machine**

Wei Hong Chin, Chu Kiong Loo, Stefan Wermter, (2020), Multichannel Recurrent Kernel Machines for Robot Episodic-Semantic Map Building, 1st SMILES (Sensorimotor Interaction, Language and Embodiment of Symbols) workshop, ICDL 2020

Enhanced Episodic Memory ART (EEM-ART) 2019

**Multi-modal ART-based Episodic Memory**

Wei Hong Chin, Yuichiro Toda, Naoyuki Kubota, Chu Kiong Loo, Manjeevan Seera (2019) Episodic Memory Multimodal Learning for Robot Sensorimotor Map Building and Navigation. IEEE Trans. Cogn. Dev. Syst. 11(2): 210-220

Multi-Channel BART 2016

**Topological Bayesian ART + Multi-Modality**

Wei Hong Chin, Chu Kiong Loo, Manjeevan Seera, Naoyuki Kubota, Yuichiro Toda (2016) Multi-channel Bayesian Adaptive Resonance Associate Memory for on-line topological map building. Appl. Soft Comput. 38: 269-280

Topological Gaussian ART (TGA) 2016

**Topological Gaussian ART + Fuzzy Motion Planning**

Wei Hong Chin, Chu Kiong Loo, Yuichiro Toda, Naoyuki Kubota (2016) An Odometry-Free Approach for Simultaneous Localization and Online Hybrid Map Building. Frontiers Robotics AI 3: 68



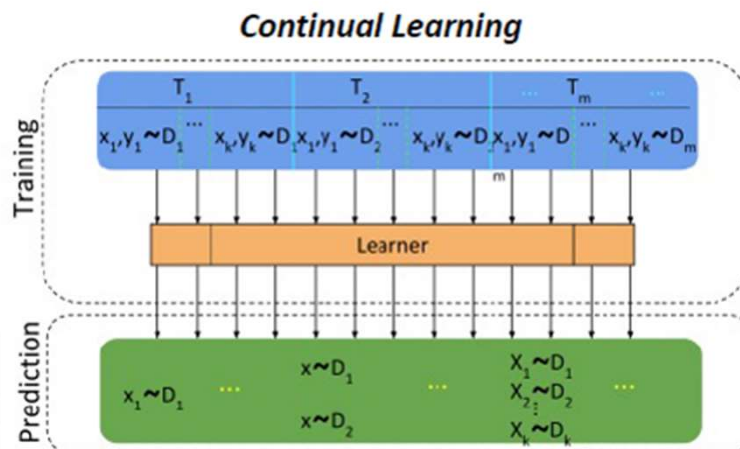
# Research Journey (2012 – 2023)

## Continual learning (narrow)

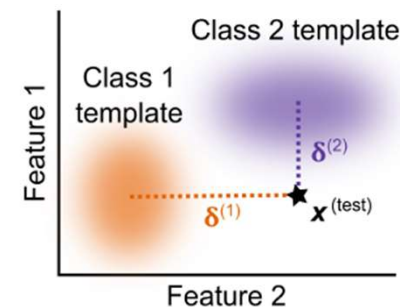
how to deal with non-stationarity in training data

## Lifelong learning - (broad)

an agent learning throughout its lifetime



## Template-based classification



| Continual / Lifelong Learning                     | Challenges                                                                                                                                                                                  | Period      | International Collaboration                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (Continual Learning)<br>Data arrive incrementally | <ul style="list-style-type: none"> <li>• Concept drift</li> <li>• Data stream learning</li> </ul>                                                                                           | 2012 – 2022 | <ul style="list-style-type: none"> <li>• <b>Tokyo Metropolitan University</b></li> <li>• <b>Osaka Metropolitan University</b></li> <li>• University of Hamburg</li> <li>• King Mongkut Institute of Technology Latkrabang</li> <li>• Dalian Maritime University</li> <li>• Hohai University</li> <li>• Murdoch University</li> </ul> |
| (Lifelong Learning)<br>Multiple tasks             | <ul style="list-style-type: none"> <li>• Concept drift</li> <li>• Data stream learning</li> <li>• <b>Catastrophic Forgetting</b></li> <li>• <b>Task/Domain/Class Incremental</b></li> </ul> | 2023 - now  |                                                                                                                                                                                                                                                                                                                                      |

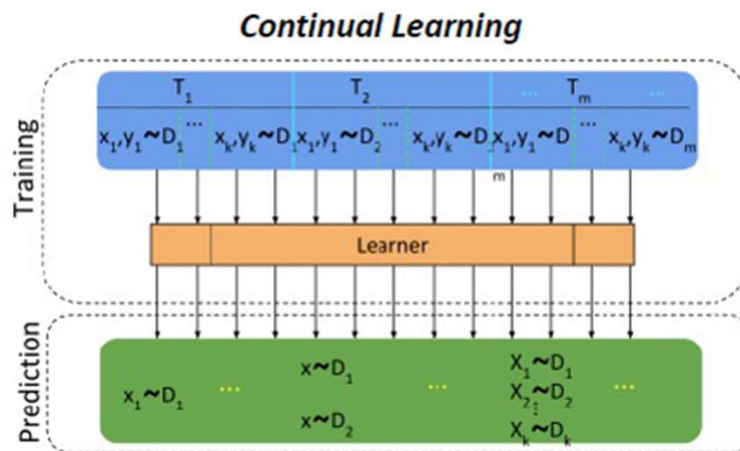
# Research Journey (2012 – 2023)

## Continual learning (narrow)

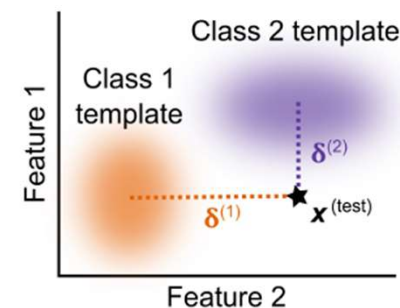
how to deal with non-stationarity in training data

## Lifelong learning - (broad)

an agent learning throughout its lifetime



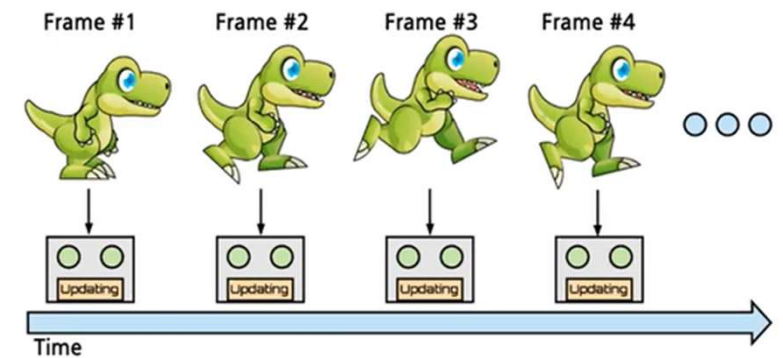
## Template-based classification



| Continual / Lifelong Learning                     | Challenges                                                                                                                                                                                  | Period      | International Collaboration                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (Continual Learning)<br>Data arrive incrementally | <ul style="list-style-type: none"> <li>• Concept drift</li> <li>• Data stream learning</li> </ul>                                                                                           | 2012 – 2022 | <ul style="list-style-type: none"> <li>• <b>Tokyo Metropolitan University</b></li> <li>• <b>Osaka Metropolitan University</b></li> <li>• University of Hamburg</li> <li>• King Mongkut Institute of Technology Latkrabang</li> <li>• Dalian Maritime University</li> <li>• Hohai University</li> <li>• Murdoch University</li> </ul> |
| (Lifelong Learning)<br>Multiple tasks             | <ul style="list-style-type: none"> <li>• Concept drift</li> <li>• Data stream learning</li> <li>• <b>Catastrophic Forgetting</b></li> <li>• <b>Task/Domain/Class Incremental</b></li> </ul> | 2023 - now  | <ul style="list-style-type: none"> <li>• <b>Tokyo Metropolitan University</b></li> <li>• <b>Osaka Metropolitan University</b></li> <li>• University of Hamburg</li> <li>• King Mongkut Institute of Technology Latkrabang</li> <li>• Dalian Maritime University</li> <li>• Hohai University</li> <li>• Murdoch University</li> </ul> |

# What is Lifelong Learning?

- The ability to update a learner one sample at a time on an evolving data stream
  - Closely resembles the evolving natural world (e.g., temporal correlations)
  - **Inputs are typically not independent and identically distributed (non-i.i.d.)**
- **Challenges**
  - Catastrophic forgetting of previous knowledge
  - Enabling backward and forward knowledge transfer



# Why Study Embedded Lifelong Learning?

- **Less compute means agents learn on-device**
- **No cloud computing in space or in internet deprived locations**
- **Customized for each user without sending personal information through the web**
- **On-device learning for AR/VR, smart toys, robots, phones, and more**



# How Could Lifelong Learning Benefit AI?

## 1. Agents could learn and adapt in real-time

- Data naturally **evolves** over time
- **Immediate inferences** about new data could be made
- On-device learning could reduce **privacy concerns**



## 2. Transfer of knowledge among similar tasks could make learning more efficient

## 3. Could result in more efficient learning algorithms



# Lifelong Learning Considerations

## 1. Learn and recall immediately

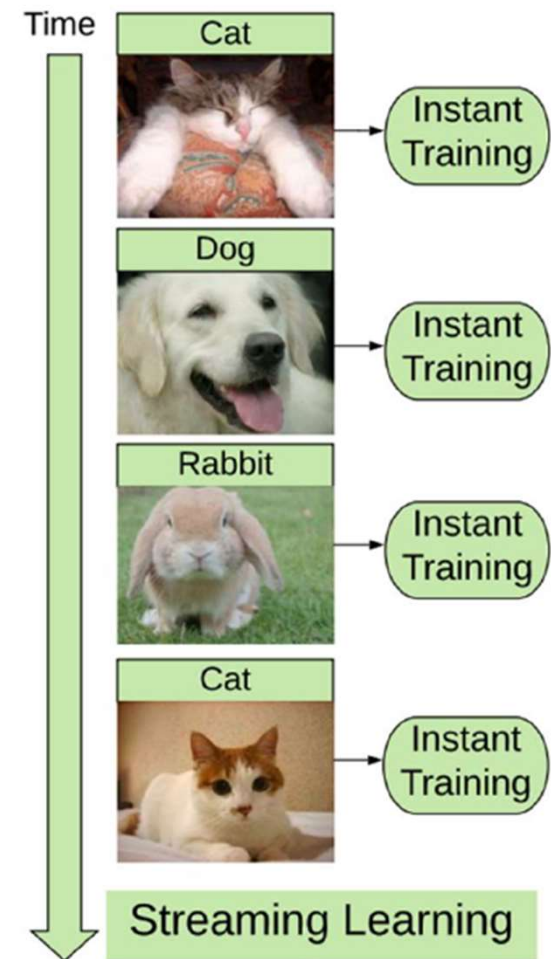
## 2. Learn data in any order without catastrophically forgetting

- Learning one class at a time induces the most forgetting
- Learning from an i.i.d. data stream induces the least forgetting
- Must test both capabilities and those in between

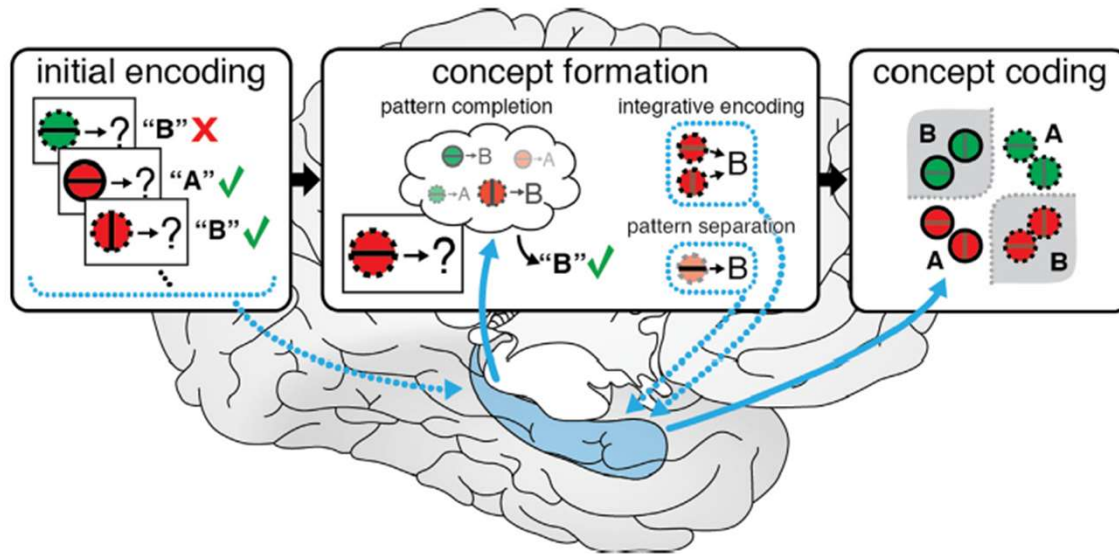
## 3. Transfer of learned representations for efficiency

## 4. Learn using limited memory and compute overhead

## 5. Scale to large-scale, high-dimensional problems



# The Episodes-to-Concepts (EpCon) Model



## Initial Encoding

- Attention Biasing

## Concept Formation

- Pattern completion
- Memory-based prediction error
  - Pattern separation
  - Pattern integration

## Concept coding

- Affected attention biasing

# Lifelong Learning Setup



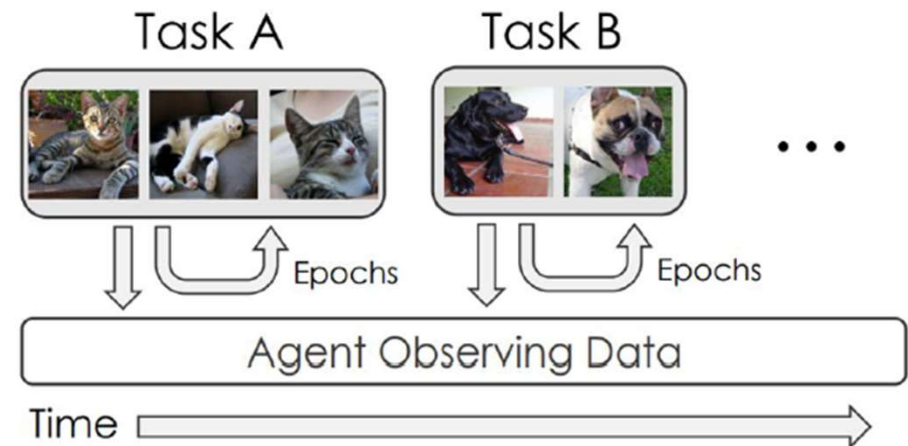
# Incremental Batch Learning

## 1. Dataset is broken into several batches (chunks)

- Includes task-based learning

## 2. At each time step, the learner...

- Receives a batch of data from one or more classes
- Loops over the batch until learned
- Is evaluated at the end of training the batch



# Incremental Batch Learning

## Advantages

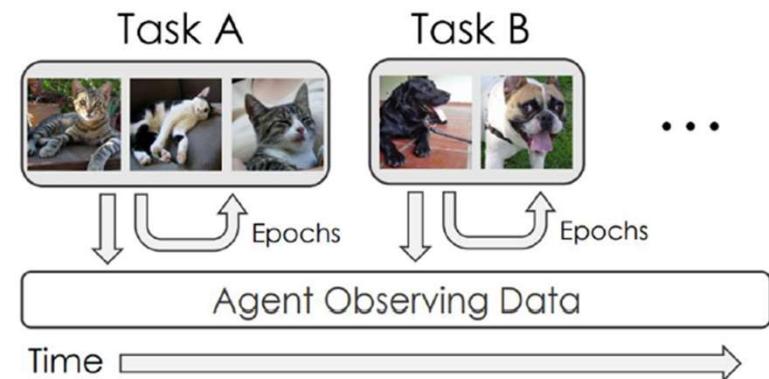
- Recently demonstrated much success
- Makes learning easier since large batches are closer to independent & identically distributed (i.i.d.)

## Caveats

- Slow (i.e., data accumulation, looping, delayed evaluation)
- Not reminiscent of how humans and animals learn

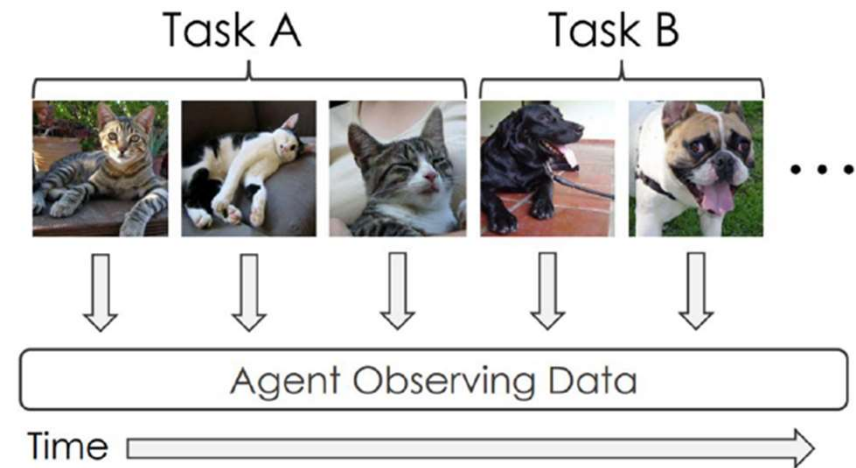
## Relevant for applications where:

- More memory and compute are available
- Immediate updates and inferences are not needed
- Batch processing could be advantageous



# Streaming Learning

1. Instances often have temporal correlations and are non-i.i.d. (videos)
2. At each time step, the learner...
  - Receives one new sample
  - Learns the sample and then is evaluated
3. The learner is only allowed **one loop** through the entire dataset



# Streaming Learning

## Advantages

- Closer to how humans/animals learn
- Better suited for real-time applications

## Caveats

- Performance may not match offline batch processing
- For supervised learning, must have labels at each time-step

## Relevant for applications where:

- Memory and compute are limited
- Immediate updates and inferences are needed



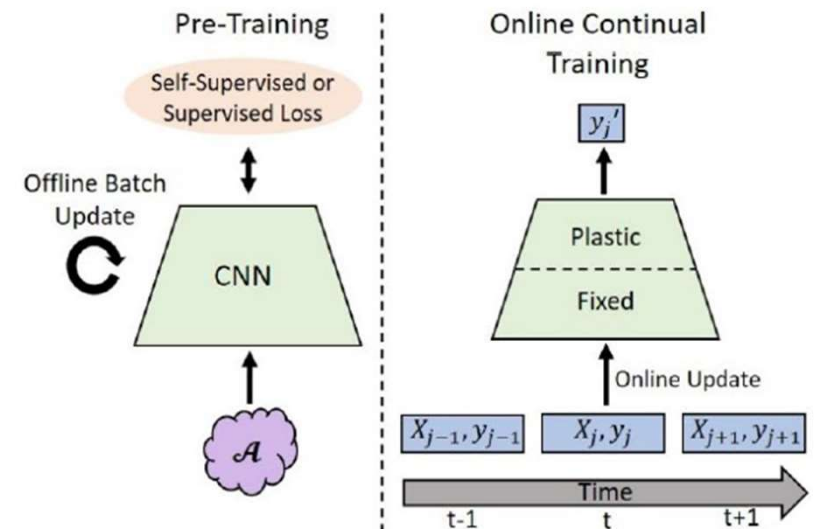


# Pre-Training Before Lifelong Learning

Many continual learning systems perform an **offline pre-training phase**

## Offline pre-training:

- Train on the first  $N$  classes / examples / first mega-batch the dataset in an offline way to initialize the deep neural network
- Then perform lifelong learning afterwards



# Lifelong Representation Learning

Every time new data is observed, update both the classifier and feature representations

Could include a pre-training phase prior to lifelong learning

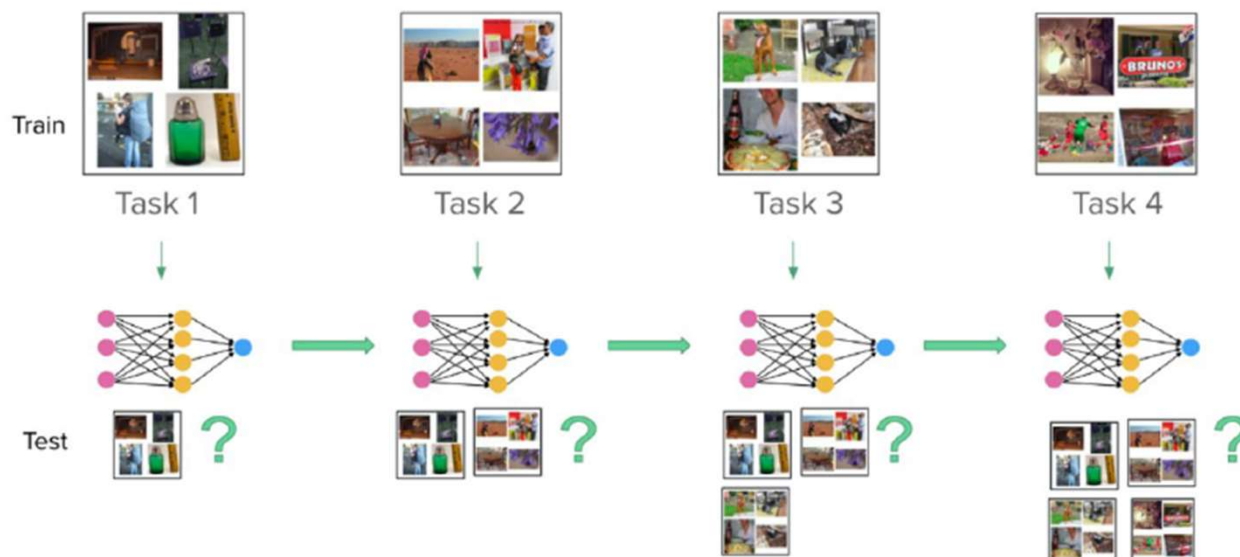


Image: <https://mila.quebec/en/article/la-maml-look-ahead-meta-learning-for-continual-learning/>

# Pre-Trained Features vs. Lifelong Representation Learning

## Pre-Training Phase (without lifelong representation learning):

- Downstream tasks are **similar** to pre-training tasks
- **Limited compute** is available for downstream tasks
- Very little downstream task data is available

## Lifelong Representation Learning:

- Downstream tasks change significantly over time
- Task-specific features are needed
- Requires more compute to update features
- Another consideration is feature update types:
  - Supervised, Unsupervised/Self-supervised

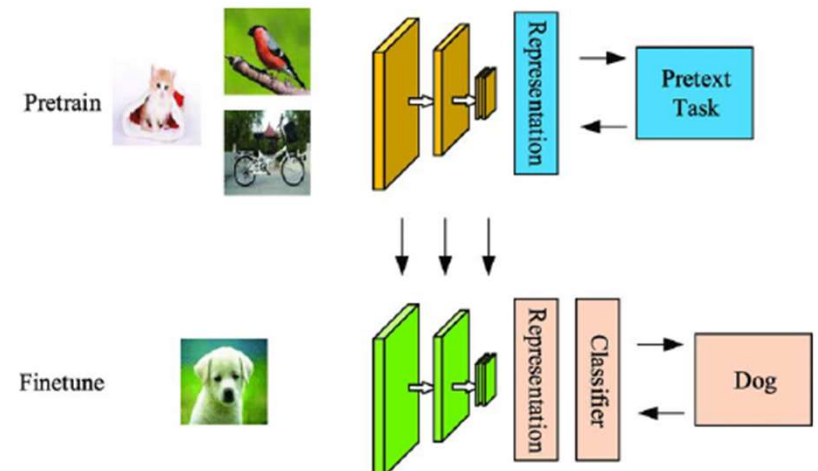
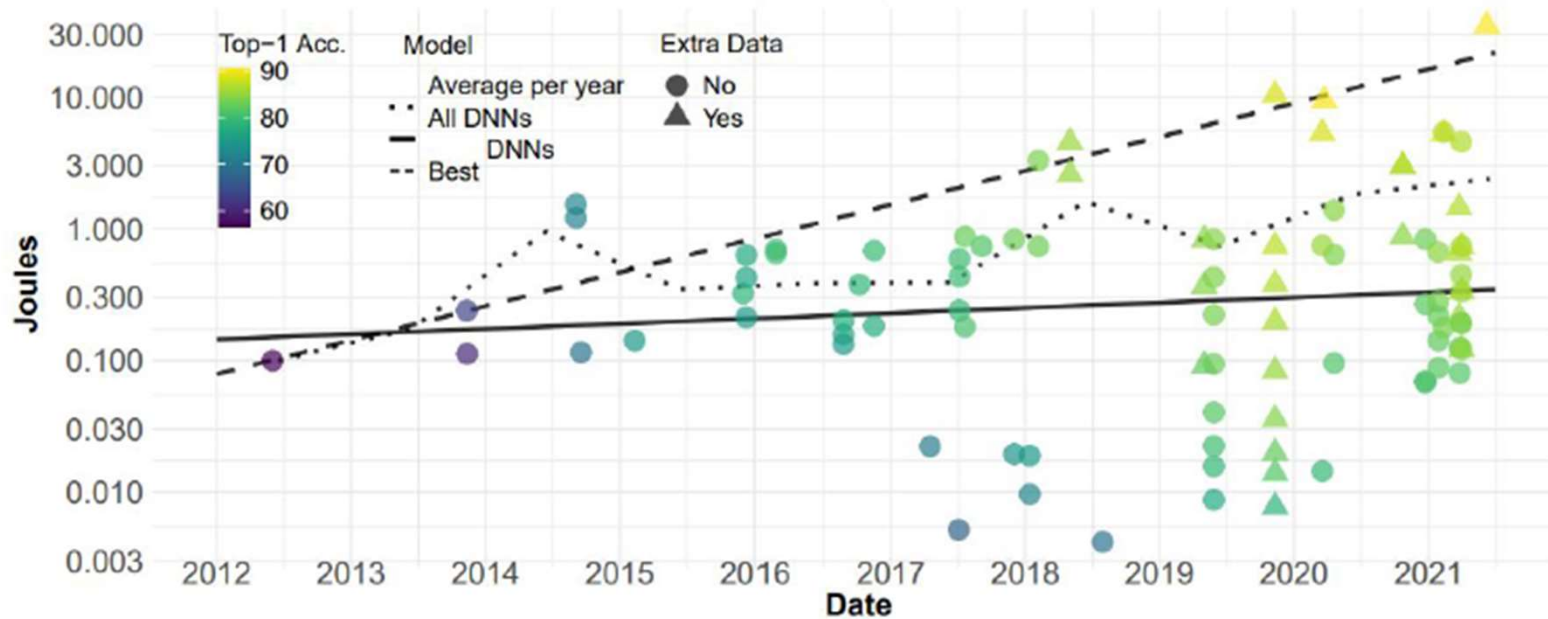


Image: <https://www.v7labs.com/blog/self-supervised-learning-guide>

# Practical Hardware Considerations

- Beyond real-time processing, successful lifelong learners could make learning algorithms **more efficient** thus reducing energy and power consumption
- Better for the **environment**



# Lifelong Learning for Embedded Devices

- **Embedded Device:** a device that is purpose built for its application
- Typically resource constrained in terms of compute, storage, and power

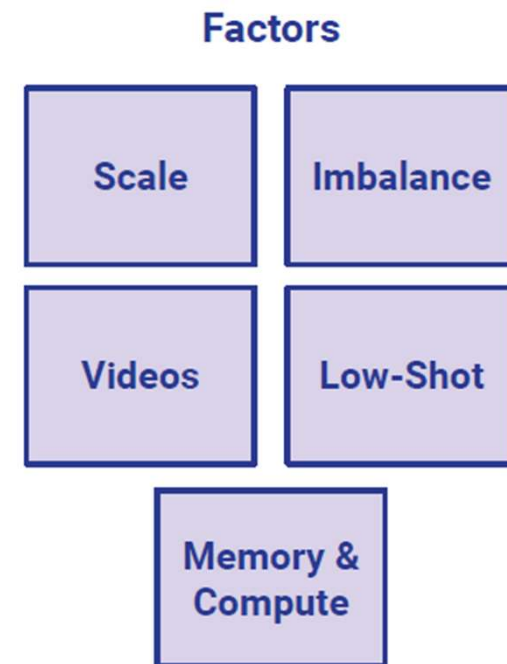




# Embedded Lifelong Learning Considerations

- Embedded lifelong learning poses unique challenges due to **real-world** and **hardware constraints**
- Limited memory and compute
- Natural world is **imbalanced/long-tailed** with need for **low-shot** learning
- **Scalability** to infinite data streams
- Natural world is **temporally correlated** (videos)

*We expect our agent to be performant regardless of these factors*



# Embedded Lifelong Learning Challenges

Applications may be **memory** or **compute** limited:

- Some lifelong learning approaches may not be applicable
- There has not been a significant amount of research on **imbalanced** and **low-shot** lifelong learning
- Lifelong learning approaches may not **scale** to millions of classes or examples

*More research is needed to apply lifelong learning mechanisms to real-world applications like embedded devices*

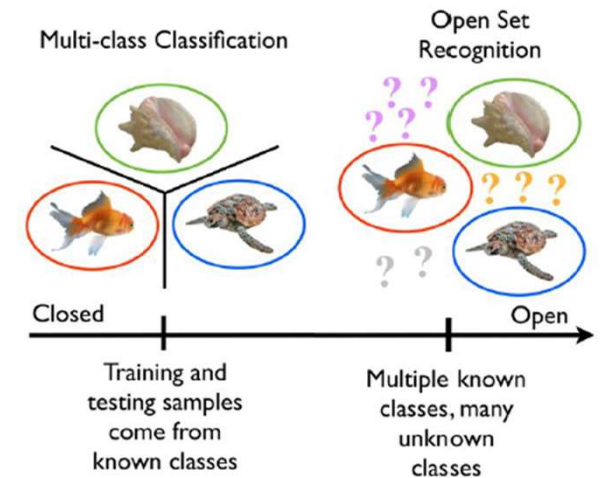


# Underexplored Capabilities for Lifelong Learning

- Better pre-trained **features**
- Better self-supervised methods and on larger/more diverse datasets
- Techniques to improve **low-shot learning**
  - Learning from very few instances can yield speed improvements
- Techniques to improve robustness to additional **domain shifts**

## Open-world learning:

- Identify unknown inputs and then incrementally learn them

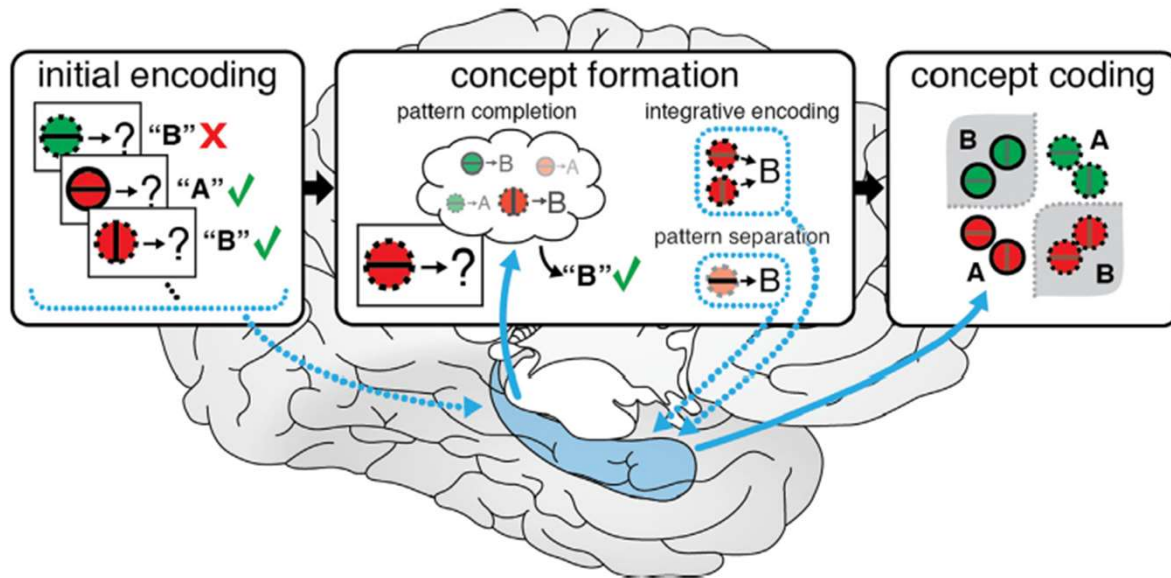


Scheirer et al., 2013

# Explainable Lifelong Streaming Learning (ExLL)

- **The Episodes-to-Concepts (EpCon) Model**
- **Empirical Data Analysis (EDA)**

# The Episodes-to-Concepts (EpCon) Model



## Initial Encoding

- Attention Biasing

## Concept Formation

- Pattern completion
- Memory-based prediction error
  - Pattern separation
  - Pattern integration

## Concept coding

- Affected attention biasing



Studies in Computational Intelligence 800

800th Volume of SCI · 800th Volume of SCI · 800th Volume of SCI · 800th Volume of SCI

Plamen P. Angelov  
Xiaowei Gu

# Empirical Approach to Machine Learning

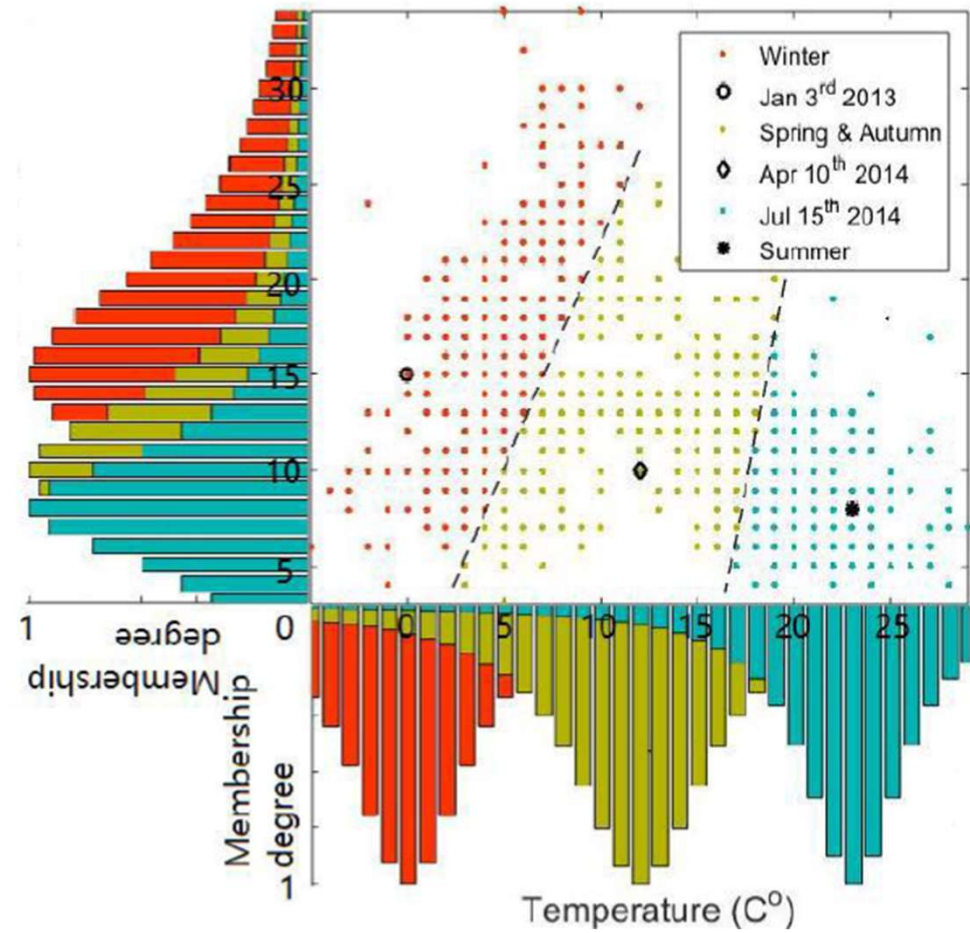
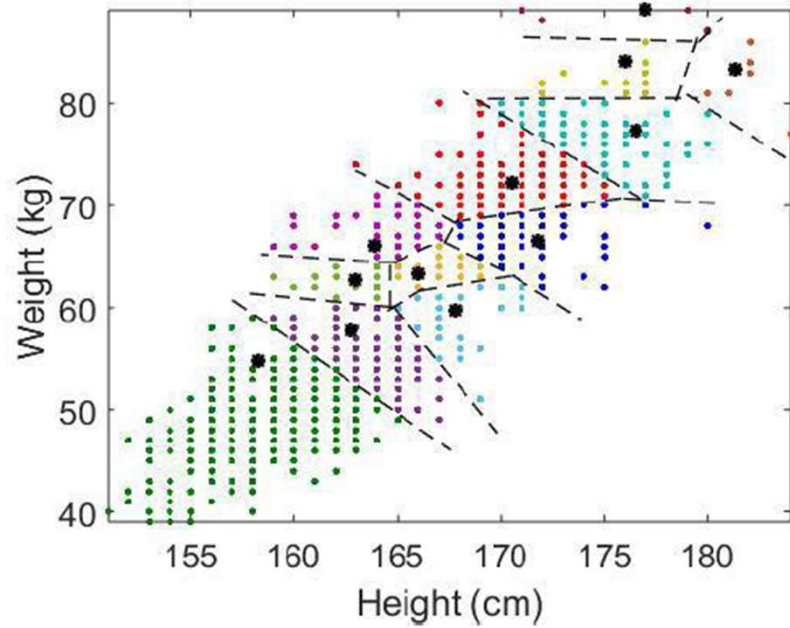
EXTRAS ONLINE

 Springer

# Empirical Data Analysis (EDA)

---

# Data Clouds



# Data Clouds vs Clusters

| Features                      | Clusters                                                 | Data Clouds                        |
|-------------------------------|----------------------------------------------------------|------------------------------------|
| Boundaries                    | Defined as hyper-ellipsoids                              | Voronoi tessellation               |
| Centre/Prototype              | Defined                                                  | Extracted post factum              |
| Distance between a data point | Centre/Mean                                              | Focal point                        |
| Membership function           | Approximation of an ideal distribution, assumed a priori | Reflect the real data distribution |

Having a data clouds, we can extract rules

**Rule<sup>i</sup> : IF ( $x \sim C^i$ ) THEN (Output<sup>i</sup>)**

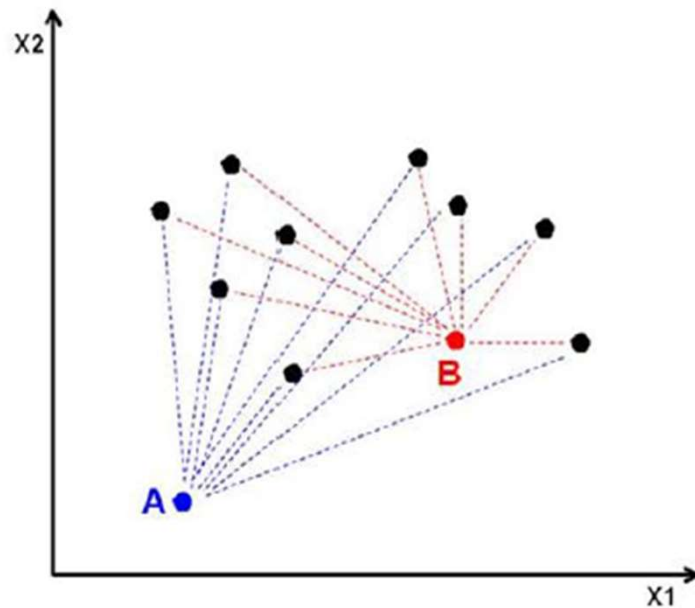
# Empirical Data Analysis (EDA)

Probability theory, statistics number of restrictive assumptions which usually do not hold in reality

- Pre defined smooth, “convenient to use” types of distribution;
- Infinite amount of observations/data points;
- Independence between data points (so called iid (Independent and Identically Distributed data))
- EDA, Entirely based on the empirical observations of discrete data points and their mutual position forming a unique pattern in the data space.
- An effective combination of the frequency and the space distance

# EDA – Cumulative Proximity

Cumulative proximity is a measure indicating the degree of closeness/similarity of a particular data point to all other existing data points:



$$\pi_k(x_i) = \sum_{j=1}^k d^2(x_i, x_j) \quad k > 1 \quad \pi_k(x_i) > 0$$



## EDA – Basic measure,

Standardized Eccentricity - represents the association of the data point with the tail of the distribution and the property of being an outlier/anomaly

$$\varepsilon_k(x_i) = k \xi_k(x_i) = \frac{2\pi_k(x_i)}{\frac{1}{k} \sum_{j=1}^k \pi_k(x_j)} \quad k > 1 \quad \pi_k(x_i) > 0$$

$\varepsilon$  is very convenient to represent well known Chebyshev inequality. It turns into a simple check if  $\varepsilon_N(\mathbf{x}) > 10$  or not for  $n=3$  because:

$$P\left(\varepsilon_N(\mathbf{x}) \leq n^2 + 1\right) \geq 1 - \frac{1}{n^2}$$

# EDA – Basic measure, D

Data density is inversely proportional to the standardized eccentricity.

$$D_K(\mathbf{x}_i) = \frac{1}{\varepsilon_K(\mathbf{x}_i)}; \quad i = 1, 2, \dots, K$$

It can be proven that for Euclidean and Mahalanobis distances D gets the form of Cauchy function:

$$D_N(\mathbf{x}_i) = \frac{1}{1 + \frac{\|\mathbf{x}_i - \boldsymbol{\mu}_N\|^2}{X_N - \boldsymbol{\mu}_N^\top \boldsymbol{\mu}_N}}$$

$$\boldsymbol{\mu}_N = \frac{N-1}{N} \boldsymbol{\mu}_{N-1} + \frac{1}{N} \mathbf{x}_N; \quad \boldsymbol{\mu}_1 = \mathbf{x}_1$$

$$X_N = \frac{N-1}{N} X_{N-1} + \frac{1}{N} \mathbf{x}_N^\top \mathbf{x}_N; \quad X_1 = \mathbf{x}_1^\top \mathbf{x}_1$$

RDE–Cauchy type, Angelov('02)

$$\begin{aligned} D(\mathbf{x}_k) &= e^{-\frac{1}{k} \sum_{i=1}^k \|\mathbf{x}_i - \mathbf{x}_k\|^2} \\ &= \frac{1}{e^{\frac{1}{k} \sum_{i=1}^k \|\mathbf{x}_i - \mathbf{x}_k\|^2}} \\ &\approx \frac{1}{1 + \frac{1}{k} \sum_{i=1}^k \|\mathbf{x}_i - \mathbf{x}_k\|^2 + \dots} \end{aligned}$$

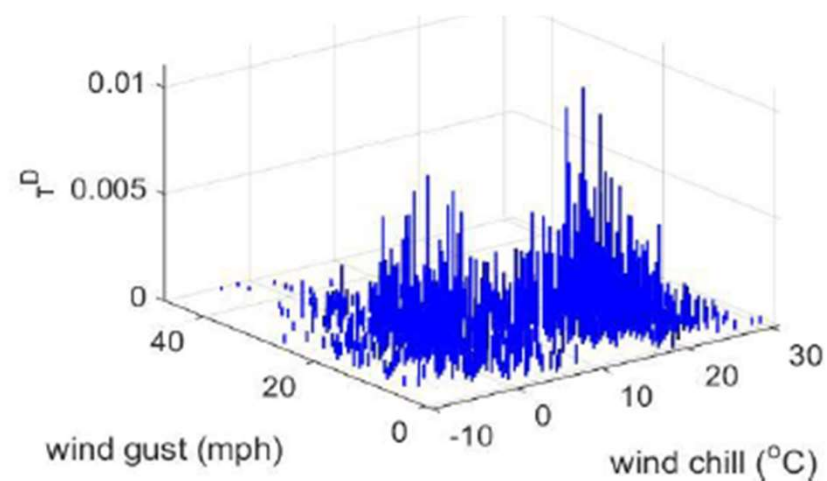
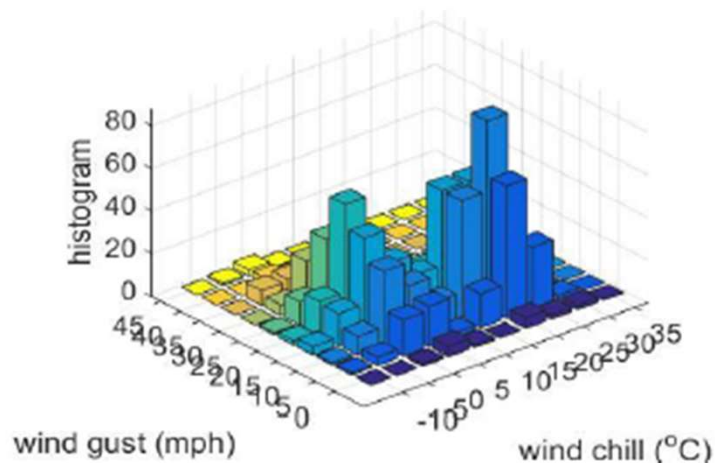
The density in the data space is a key characteristic of anomalies and model structure (focal points for local sub-models).

# EDA – Basic measure, $\tau$

Multi-modal (discrete global) **typicality**

$$\tau_N^D(\mathbf{u}_i) = \frac{f_i D_N(\mathbf{u}_i)}{\sum_{j=1}^{L_N} f_j D_N(\mathbf{u}_j)} = \frac{f_j q_N^{-1}(\mathbf{u}_i)}{\sum_{j=1}^{L_N} f_j q_N^{-1}(\mathbf{u}_j)}$$

Can be used as pdf, but is derived entirely from data with no prior assumptions



# Evolving Intelligence based on Density

Learning concept from experience (extract knowledge from data streams)

- A data sample with high density ( $D$ ) is eligible to be a focal point of a Cloud/local sub model
- A data sample that lies in an area of data space not covered by other local sub-models is also eligible to form a new local sub model
- Avoid overlap and information redundancy in forming new local sub models
- Remove old clouds and low support utility ones

# EDA + Episodes-to-Concepts (EpCon) Model

67

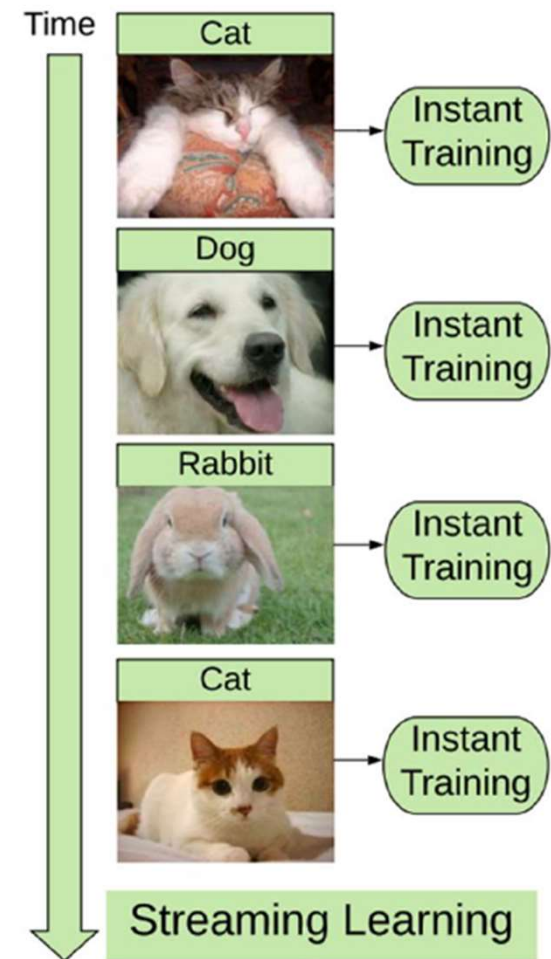
| Empirical Data Analysis                                                                                                               | Episodes-to-Concepts (EpCon) Model                                                                                                                                                                 |
|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A data sample with high density ( $D$ ) is eligible to be a focal point of a Cloud/local sub model                                    | Initial Encoding <ul style="list-style-type: none"><li>• Attention Biasing</li></ul>                                                                                                               |
| A data sample that lies in an area of data space not covered by other local sub-models is also eligible to form a new local sub model | Concept Formation <ul style="list-style-type: none"><li>• Pattern completion</li><li>• Memory-based prediction error<ul style="list-style-type: none"><li>• Pattern separation</li></ul></li></ul> |
| Avoid overlap and information redundancy in forming new local sub models                                                              | <ul style="list-style-type: none"><li>• Memory-based prediction error<ul style="list-style-type: none"><li>• Pattern integration</li></ul></li></ul>                                               |
| Remove old clouds and low support utility ones                                                                                        | <ul style="list-style-type: none"><li>• Pruning</li></ul>                                                                                                                                          |



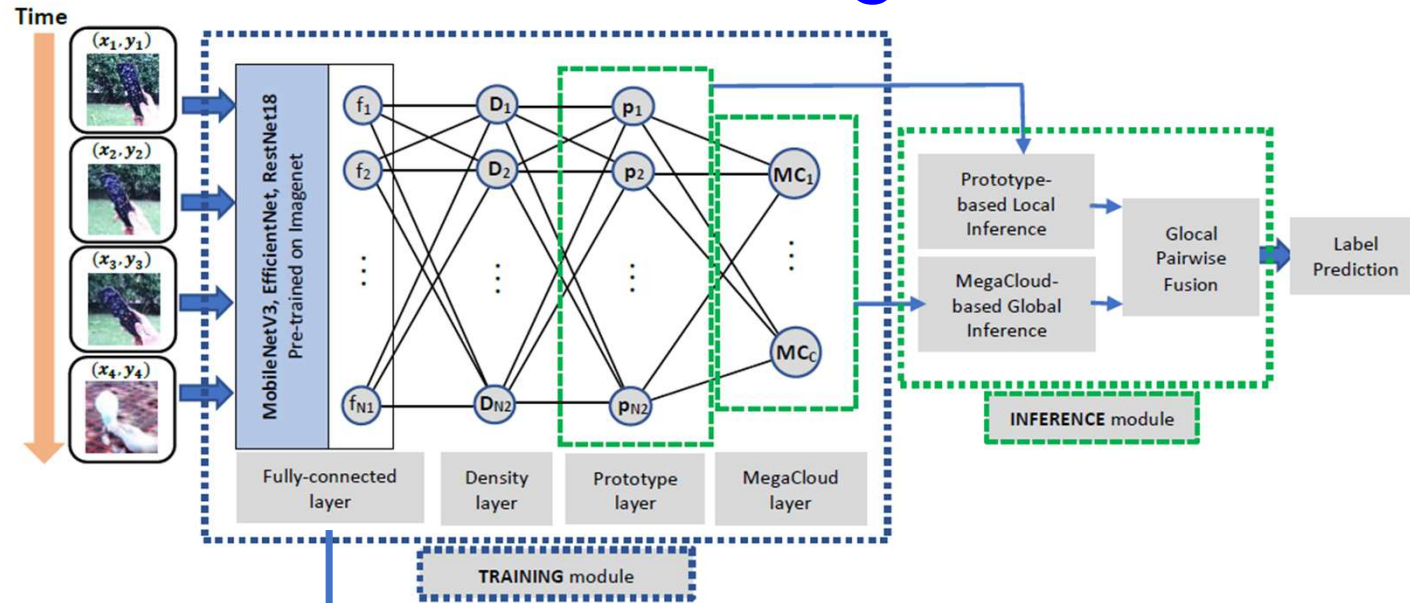
# Explainable Lifelong Streaming Learning (ExLL)

# Explainable Lifelong Learning Considerations

1. Learn and recall immediately
2. Learn data in any order without catastrophically forgetting
3. Transfer of learned representations for efficiency
4. Learn using limited memory and compute overhead
5. Scale to large-scale, high-dimensional problems
6. Explain model decisions at the intermediate and final stages of the decision-making process

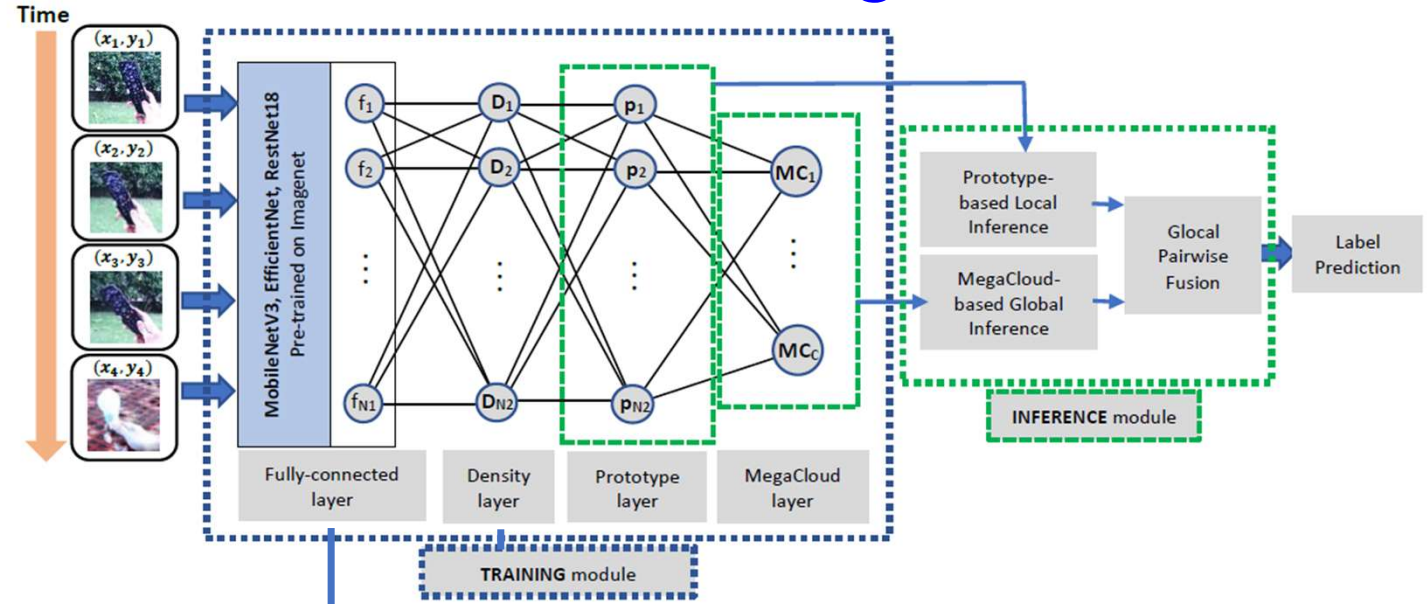


# ExLL - Training



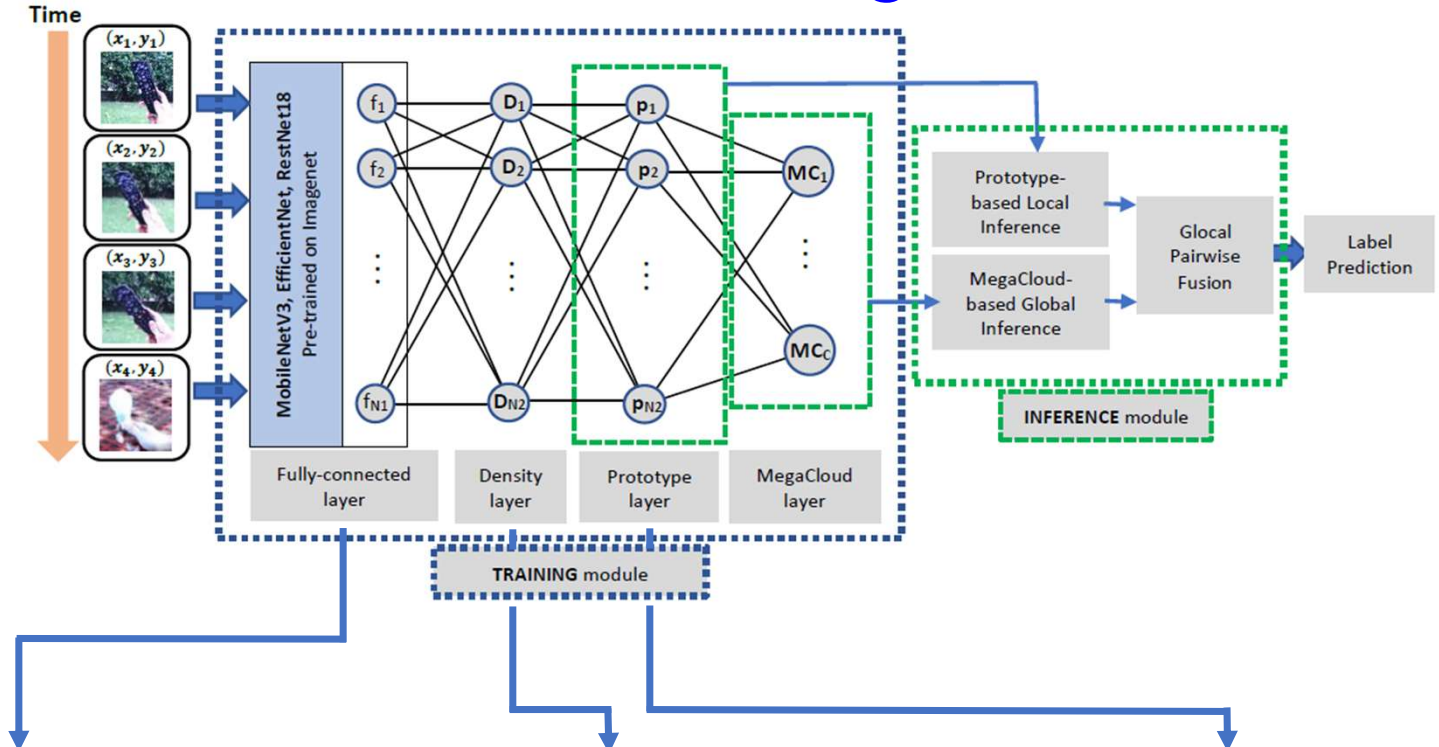
| Normalization                                                                                                           | Global Meta-parameter Update                                                                                                                                                                                                                                                                                                 | Local Meta-parameter Update                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $x_i = \frac{\tilde{x}_i}{\ \tilde{x}_i\ }$ <p>Convert to cosine dissimilarity for high-dimensional data processing</p> | $\hat{\mu}_i = \frac{i-1}{i}\hat{\mu}_{i-1} + \frac{1}{i}x_i$ $\hat{\mu}_1 = x_1$ $\hat{\sigma}_i = \frac{i-1}{i}\hat{\sigma}_{i-1} + \frac{1}{i}\ x_i\ ^2$ $\hat{\sigma}_1 = \ x_1\ ^2 = 1$ $\hat{\xi}_i = \frac{i-1}{i}\hat{\xi}_{i-1} + \frac{1}{i}(x_i - \hat{\mu}_i)(x_i - \hat{\mu}_i)^T$ $\hat{\xi}_1 = (x_1)(x_1)^T$ | $\hat{i}_k \leftarrow 1$ $g_k \leftarrow 1$ $\mu_{k,1} \leftarrow x_i$ $\sigma_{k,1} \leftarrow \ x_i\ ^2$ $E_{k,1,1} = 0$ $p_{k,1} \leftarrow x_i$ $S_{k,1} \leftarrow 1$ $r_{k,1} \leftarrow r^*$ $\hat{I}_{k,1} \leftarrow I_i$ $\mu_{k,i_k} = \frac{\hat{i}_k - 1}{\hat{i}_k}\mu_{i_k-1} + \frac{1}{\hat{i}_k}x_i$ $\sigma_{k,i_k} = \frac{\hat{i}_k - 1}{\hat{i}_k}\sigma_{i_k-1} + \frac{1}{\hat{i}_k}\ x_i\ ^2$ |

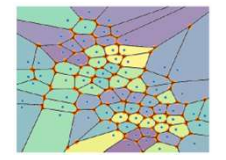
# ExLL - Training



|                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Normalization</b></p> $x_i = \frac{\tilde{x}_i}{\ \tilde{x}_i\ }$ <p>Convert to cosine dissimilarity for high-dimensional data processing</p> | <p><b>Global Meta-parameter Update</b></p> $\hat{\mu}_i = \frac{i-1}{i}\hat{\mu}_{i-1} + \frac{1}{i}x_i$ $\hat{\mu}_1 = x_1$ $\hat{\sigma}_i = \frac{i-1}{i}\hat{\sigma}_{i-1} + \frac{1}{i}\ x_i\ ^2$ $\hat{\sigma}_1 = \ x_1\ ^2 = 1$ $\hat{\xi}_i = \frac{i-1}{i}\hat{\xi}_{i-1} + \frac{1}{i}(x_i - \hat{\mu}_i)(x_i - \hat{\mu}_i)^T$ $\hat{\xi}_1 = (x_1)(x_1)^T$ | <p><b>Local Meta-parameter Update</b></p> $i_k \leftarrow 1$ $g_k \leftarrow 1$ $\mu_{k,1} \leftarrow x_i$ $\sigma_{k,1} \leftarrow \ x_i\ ^2$ $E_{k,1,1} = 0$ $\mu_{k,i_k} = \frac{i_k-1}{i_k}\mu_{i_k-1} + \frac{1}{i_k}x_i$ $\sigma_{k,i_k} = \frac{i_k-1}{i_k}\sigma_{i_k-1} + \frac{1}{i_k}\ x_i\ ^2$ | <p><b>Recursive Density Estimation</b></p> <p style="background-color: #c8e6c9; padding: 2px;">Attention Biasing</p> $D(k, x_i) = \frac{1}{1 + \ x_i - \mu_{k,i_k}\ ^2 + \sigma_{k,i_k} - \ \mu_{k,i_k}\ ^2}$ |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

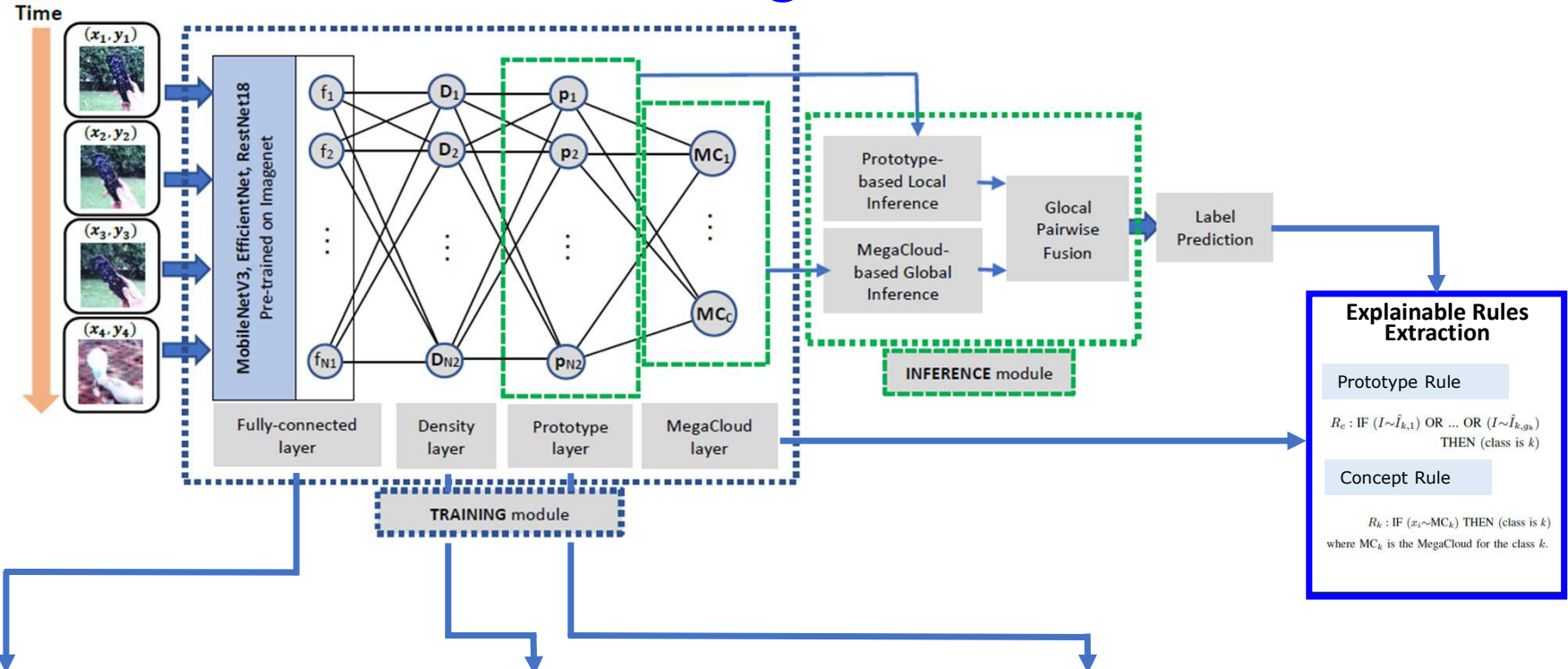
# ExLL - Training



|                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                              |                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Normalization</b></p> $x_i = \frac{\tilde{x}_i}{\ \tilde{x}_i\ }$ <p>Convert to cosine dissimilarity for high-dimensional data processing</p> | <p><b>Global Meta-parameter Update</b></p> $\hat{\mu}_i = \frac{i-1}{i}\hat{\mu}_{i-1} + \frac{1}{i}x_i$ $\hat{\mu}_1 = x_1$ $\hat{\sigma}_i = \frac{i-1}{i}\hat{\sigma}_{i-1} + \frac{1}{i}\ x_i\ ^2$ $\hat{\sigma}_1 = \ x_1\ ^2 = 1$ $\hat{\xi}_i = \frac{i-1}{i}\hat{\xi}_{i-1} + \frac{1}{i}(x_i - \hat{\mu}_i)(x_i - \hat{\mu}_i)^T$ $\hat{\xi}_1 = (x_1)(x_1)^T$ | <p><b>Local Meta-parameter Update</b></p> $i_k \leftarrow 1$ $g_k \leftarrow 1$ $\mu_{k,1} \leftarrow x_i$ $\sigma_{k,1} \leftarrow \ x_i\ ^2$ $E_{k,1,1} = 0$ $r_{k,1} \leftarrow r^*$ $\hat{I}_{k,1} \leftarrow I_i$ $p_{k,1} \leftarrow x_i$ $S_{k,1} \leftarrow 1$ $r_{k,1} \leftarrow r^*$ $\hat{I}_{k,1} \leftarrow I_i$ $\mu_{k,i_k} = \frac{i_k-1}{i_k}\mu_{i_k-1} + \frac{1}{i_k}x_i$ $\sigma_{k,i_k} = \frac{i_k-1}{i_k}\sigma_{i_k-1} + \frac{1}{i_k}\ x_i\ ^2$ | <p><b>Recursive Density Estimation</b></p> <p>Attention Biasing</p> $D(k, x_i) = \frac{1}{1 + \ x_i - \mu_{k,i_k}\ ^2 + \sigma_{k,i_k} - \ \mu_{k,i_k}\ ^2}$ | <p><b>Prototype Identification</b></p> <p>Pattern Completion</p> $b_1 = \operatorname{argmin}_{j=1, \dots, g_k} \frac{(x_i - p_j)^T (x_i - p_j)}{\xi}$ $b_2 = \operatorname{argmin}_{j=1, \dots, g_k; j \neq b_1} \frac{(x_i - p_j)^T (x_i - p_j)}{\xi}$ <p>Memory-based error Prediction</p> | <p><b>Prototype Formation</b></p> <p>Pattern separation</p> <p>IF <math>D(k, x_i) &gt; \max_{j=1, \dots, g_k} D(k, p_j)</math><br/>         OR <math>D(k, x_i) &lt; \min_{j=1, \dots, g_k} D(k, p_j)</math><br/>         THEN add a new data cloud (<math>g_k \leftarrow g_k + 1</math>)</p> <p>Pattern Integration</p> <p>Else</p> $p_{k,b_1} \leftarrow \frac{S_{k,b_1} - 1}{S_{k,b_1} - p_{k,b_1} + \frac{1}{S_{k,b_1}}} x_i$ $r_{k,b_1} \leftarrow \sqrt{\frac{r_{k,b_1}^2 + (1 - \ p_{k,b_1}\ ^2)}{2}}$ $\hat{I}_{k,b_1} \leftarrow \hat{I}_{k,b_1} + I_i$ | <p><b>Topology Formation</b></p> <p>Voronoi Tessellation</p> $E_{k,b_1, b_2} \leftarrow E_{k,b_1, b_2} + 1$ $E_{k,b_2, b_1} \leftarrow E_{k,b_2, b_1} + 1$  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



# ExLL - Training



|                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                              |                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Normalization</b></p> $x_i = \frac{\tilde{x}_i}{\ \tilde{x}_i\ }$ <p>Convert to cosine dissimilarity for high-dimensional data processing</p> | <p><b>Global Meta-parameter Update</b></p> $\hat{\mu}_i = \frac{i-1}{i}\hat{\mu}_{i-1} + \frac{1}{i}x_i$ $\hat{\mu}_1 = x_1$ $\hat{\sigma}_i = \frac{i-1}{i}\hat{\sigma}_{i-1} + \frac{1}{i}\ x_i\ ^2$ $\hat{\sigma}_1 = \ x_1\ ^2 = 1$ $\hat{\xi}_i = \frac{i-1}{i}\hat{\xi}_{i-1} + \frac{1}{i}(x_i - \hat{\mu}_i)(x_i - \hat{\mu}_i)^T$ $\hat{\xi}_1 = (x_1)(x_1)^T$ | <p><b>Local Meta-parameter Update</b></p> $i_k \leftarrow 1$ $g_k \leftarrow 1$ $\mu_{k,1} \leftarrow x_i$ $\sigma_{k,1} \leftarrow \ x_i\ ^2$ $E_{k,1,1} = 0$ $p_{k,1} \leftarrow x_i$ $S_{k,1} \leftarrow 1$ $r_{k,1} \leftarrow r^*$ $\hat{I}_{k,1} \leftarrow I_i$ $\mu_{k,i_k} = \frac{i_k-1}{i_k}\mu_{k,i_k-1} + \frac{1}{i_k}x_i$ $\sigma_{k,i_k} = \frac{i_k-1}{i_k}\sigma_{k,i_k-1} + \frac{1}{i_k}\ x_i\ ^2$ | <p><b>Recursive Density Estimation</b></p> <p>Attention Biasing</p> $D(k, x_i) = \frac{1}{1 + \ x_i - \mu_{k,i_k}\ ^2 + \sigma_{k,i_k} - \ \mu_{k,i_k}\ ^2}$ | <p><b>Prototype Identification</b></p> <p>Pattern Completion</p> $b_1 = \operatorname{argmin}_{j=1, \dots, g_k} \frac{(x_i - p_j)^T (x_i - p_j)}{\xi}$ $b_2 = \operatorname{argmin}_{j=1, \dots, g_k; j \neq b_1} \frac{(x_i - p_j)^T (x_i - p_j)}{\xi}$ <p>Memory-based error Prediction</p> | <p><b>Prototype Formation</b></p> <p>Pattern separation</p> <p>IF <math>D(k, x_i) &gt; \max_{j=1, \dots, g_k} D(k, p_j)</math><br/>         OR <math>D(k, x_i) &lt; \min_{j=1, \dots, g_k} D(k, p_j)</math><br/>         THEN add a new data cloud (<math>g_k \leftarrow g_k + 1</math>)</p> <p>Pattern Integration</p> <p>Else</p> $S_{k,b_1} \leftarrow S_{k,b_1} + 1$ $p_{k,b_1} \leftarrow \frac{S_{k,b_1} - 1}{S_{k,b_1}} p_{k,b_1} + \frac{1}{S_{k,b_1}} x_i$ $r_{k,b_1} \leftarrow \sqrt{\frac{r_{k,b_1}^2 + (1 - \ p_{k,b_1}\ ^2)}{2}}$ $\hat{I}_{k,b_1} \leftarrow \hat{I}_{k,b_1} + I_i$ | <p><b>Topology Formation</b></p> <p>Voronoi Tessellation</p> $E_{k,b_1, b_2} \leftarrow E_{k,b_1, b_2} + 1$ $E_{k,b_2, b_1} \leftarrow E_{k,b_2, b_1} + 1$ |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Explainable Rules Extraction**

Prototype Rule

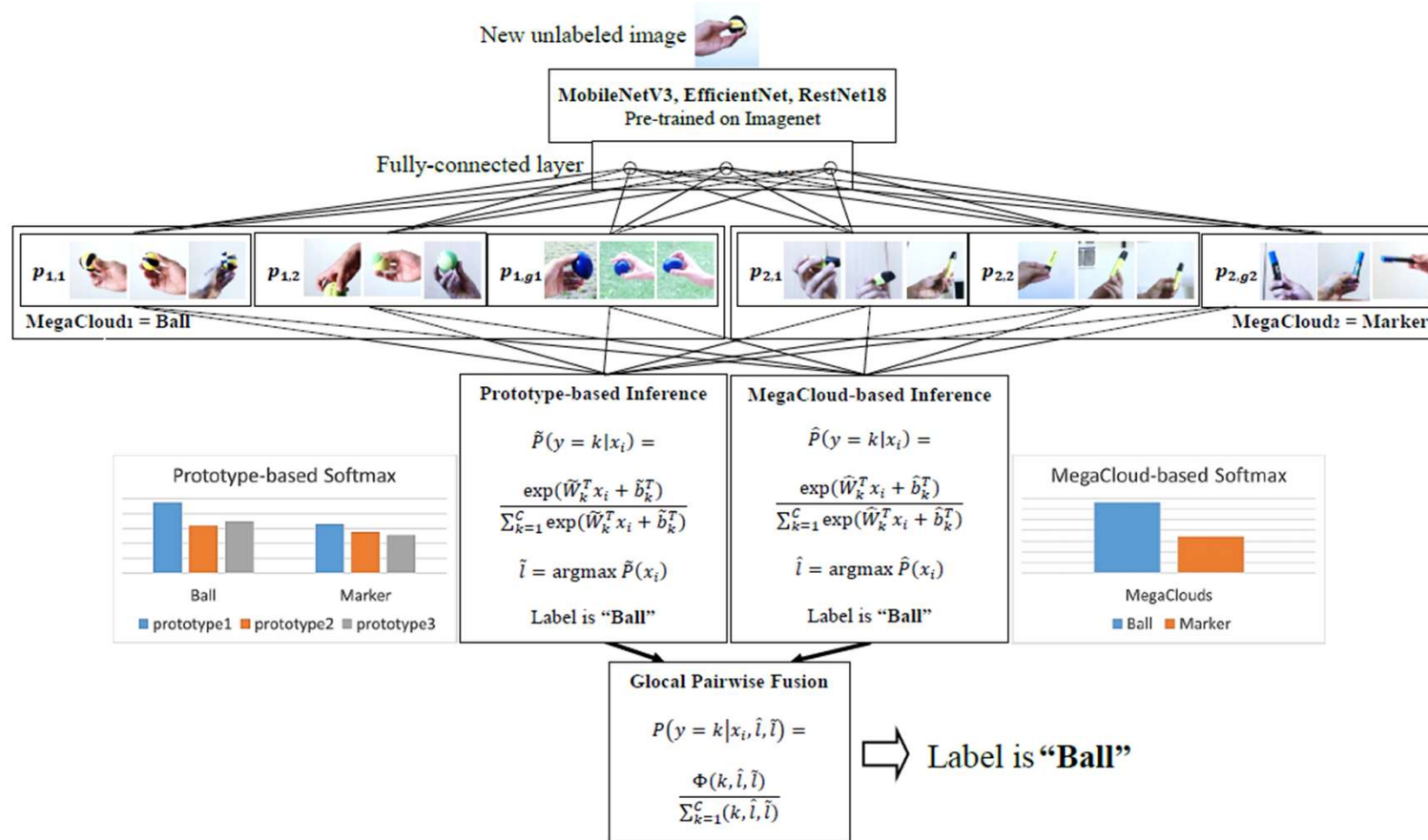
$$R_c : \text{IF } (I \sim \hat{I}_{k,1}) \text{ OR } \dots \text{ OR } (I \sim \hat{I}_{k,g_k}) \text{ THEN (class is } k)$$

Concept Rule

$$R_k : \text{IF } (x_i \sim \text{MC}_k) \text{ THEN (class is } k)$$

where  $\text{MC}_k$  is the MegaCloud for the class  $k$ .

# ExLL - Inference



Shrinkage regularization

$$\Lambda = [(1 - \epsilon)\hat{\xi} + (\epsilon)I]^{-1}$$

$$\epsilon = 1e^{-4}$$

Prototype-based Inference

$$\tilde{p}_k = \{p_{1,1}, \dots, p_{k,g_k}\}$$

$$\tilde{W}_k = \Lambda \tilde{p}_k$$

$$\tilde{b}_k = -\frac{1}{2}(\tilde{p}_k \cdot \tilde{W}_k)$$

$$\tilde{P}(y = k|x_i) = \frac{\exp(\tilde{W}_k^T x_i + \tilde{b}_k^T)}{\sum_{k=1}^C \exp(\tilde{W}_k^T x_i + \tilde{b}_k^T)}$$

$$\tilde{l} = \operatorname{argmax}_{k=1, \dots, C} \tilde{P}(y = k|x_i)$$

Megacloud-based Inference

$$\hat{W} = \Lambda \hat{\mu}$$

$$\hat{P}(y = k|x_i) = \frac{\exp(\hat{W}_k^T x_i + \hat{b}_k^T)}{\sum_{k=1}^C \exp(\hat{W}_k^T x_i + \hat{b}_k^T)}$$

$$\hat{b} = -\frac{1}{2}(\mu \cdot \hat{W})$$

$$\hat{l} = \operatorname{argmax}_{k=1, \dots, C} \hat{P}(y = k|x_i)$$

Global Pair-wise Fusion

$$P(y = k|x_i, \hat{l}, \tilde{l}) = \frac{\Phi(k, \hat{l}, \tilde{l})}{\sum_{k=1}^C \Phi(k, \hat{l}, \tilde{l})}$$

$$L = \operatorname{argmax}_{k=1, \dots, C} (P(y = k|x_i, \hat{l}, \tilde{l}))$$

## ExLL – Rule Extraction













|         |    |                                                                                   |    |                                                                                   |    |                                                                                   |        |                          |
|---------|----|-----------------------------------------------------------------------------------|----|-----------------------------------------------------------------------------------|----|-----------------------------------------------------------------------------------|--------|--------------------------|
| Rule #1 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Aqueduct" |
| Rule #2 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Aqueduct" |
| Rule #3 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Arch"     |
| Rule #4 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Arch"     |

Fig. 4: Explainable rules extracted from prototypes for the classes "Aqueduct" and "Arch" from Places-365. Each rule is made up of training images associated with the corresponding prototype.





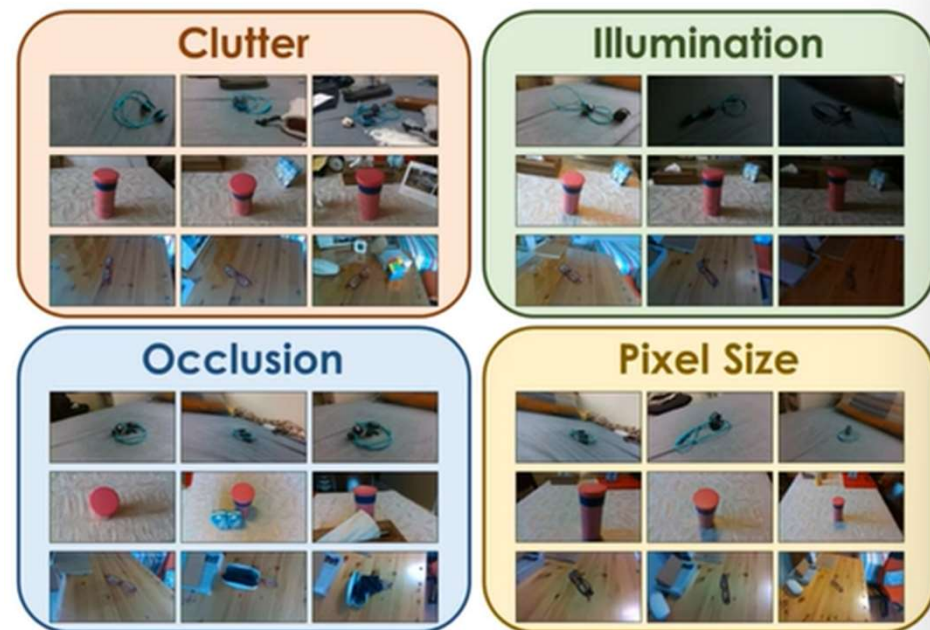
| Test Image                                                                                                                               | Hits                                                                                             | Near Hits                                                                                        | Near Misses                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <br>"Shampoo"<br>correctly predicted as<br>"Shampoo"  | <br>"Shampoo" | <br>"Shampoo" | <br>"Lotion"     |
| <br>"Shampoo"<br>Wrongly predicted as<br>"Lotion"     | <br>"Lotion"  | <br>"Lotion"  | <br>"Toothpaste" |
| <br>"Toothpaste"<br>Wrongly predicted as<br>"Shampoo" | <br>"Shampoo" | <br>"Shampoo" | <br>"Book"       |

Fig. 3: Example of "Hits", "Near Hits", and "Near Misses" for the F-SIOL-310 dataset. The test image in the top row is an example of a True Positive result, the test image in the middle row is a False Negative result, and the test image in the bottom row is a False Positive result.

# Datasets and Orderings

76

- **OpenLORIS:**
  - Videos of 40 object classes each with 1 to 9 object instances (121 instances)
  - Each object instance collected under 4 domains with 9 difficulties
  - Orderings: [Instance](#) and [Low-Shot Instances](#)
- **Places-365:**
  - Scene classification of 365 classes
  - We study the original version with over 1.8M images
  - We also study a long-tailed version
  - Orderings: [iid](#) and [class iid](#)
- **F-SIOL-310**
  - Static images of 22 household items.
  - Total 310 object instances and 620 static images
  - Using [class-iid](#) data ordering, study [5-shot](#) & [10-shot](#) learning scenarios
  - Evaluating a model's ability to learn from few training samples.





# Comparison Methods

## Online Continual Learning:

- Fine-Tune
- Multi-Class Perceptron
- Streaming One-vs-Rest (SOvR)
- Nearest Class Mean (NCM)
- Gaussian Naïve Bayes
- Streaming Linear Discriminant Analysis (SLDA)
- Replay
  - Store 2 examples or 20 examples per class

## Mobile CNNs:

- MobileNet-v3 (Small)
- MobileNet-v3 (Large)
- EfficientNet-B0
- EfficientNet-B1
- ResNet-18



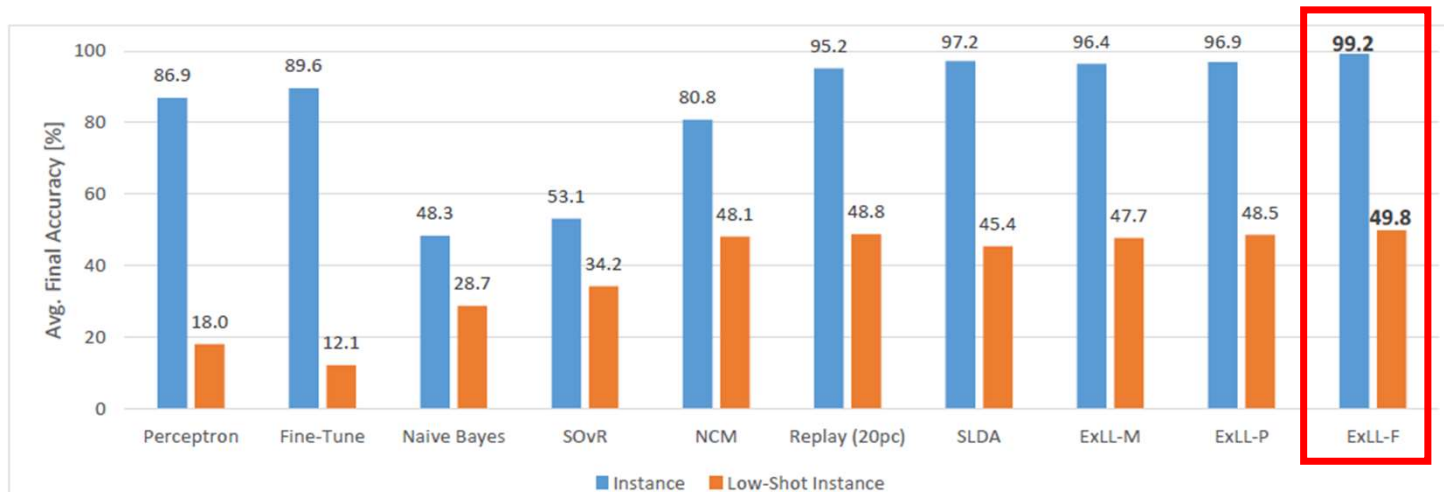
## NetScore for Lifelong Learning

$$\Omega(\mathcal{M}) = s \log\left(\frac{a(\mathcal{M})^\alpha}{p(\mathcal{M})^\beta c(\mathcal{M})^\gamma}\right)$$

- $a(\mathcal{M})$  measures the final accuracy of a model
- $p(\mathcal{M})$  is the total number of parameters
- $c(\mathcal{M})$  is the number of minutes needed for the experiment
- Scores methods across three axes: classification efficacy, memory, and computer

$s = 20$  and  $\alpha = 2$  to prioritize classification accuracy, and  $\beta = \gamma = 0.25$  to moderate the large values of  $p(\mathcal{M})$  and  $c(\mathcal{M})$  [73].

- Higher NetScores indicate better performance.



**Instance ordering** trains learners on all object instances while **low-shot instance ordering** trains learners on one object instance from each object class.

### OpenLORIS

- All models displayed lower accuracy when using low-shot instance ordering. Perceptron and Fine-Tune showed a much bigger drop, shows **poor generalization** to out-of-domain inputs
- **Naive Bayes, SOvR, and NCM** were **less accurate** than Perceptron and Fine-Tune for **the full instance ordering**, but **outperformed** them for the **low-shot condition**.
- The **ExLL models** showed the best balance between the two ordering methods, while **ExLL-F outperformed all other models** for both orderings.

## How well do methods generalize from few instances?

| Method        | MNet-S       | MNet-L       | ENet-B0      | ENet-B1      | RN-18        | Mean         |
|---------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Perceptron    | 0.793        | 0.880        | 0.935        | 0.942        | 0.796        | 0.869        |
| Fine-Tune     | 0.835        | 0.915        | 0.958        | 0.963        | 0.809        | 0.896        |
| Naive Bayes   | 0.311        | 0.526        | 0.780        | 0.787        | 0.015        | 0.483        |
| SOvR          | 0.374        | 0.477        | 0.739        | 0.723        | 0.346        | 0.531        |
| NCM           | 0.729        | 0.789        | 0.859        | 0.867        | 0.797        | 0.808        |
| Replay (20pc) | 0.921        | 0.956        | 0.977        | 0.978        | 0.929        | 0.952        |
| SLDA          | <b>0.956</b> | <b>0.982</b> | <b>0.988</b> | <b>0.988</b> | 0.950        | <b>0.972</b> |
| ExLL-M        | 0.944        | <u>0.973</u> | <u>0.982</u> | 0.982        | 0.940        | <u>0.964</u> |
| ExLL-P        | <u>0.951</u> | 0.968        | <u>0.982</u> | <u>0.983</u> | <b>0.961</b> | <u>0.969</u> |
| ExLL-F        | <b>0.987</b> | <b>0.993</b> | <b>0.996</b> | <b>0.996</b> | <b>0.988</b> | <b>0.991</b> |

Table II presents the best accuracy scores of online continual learning models across different CNN backbones when trained using [instance ordering](#).

| Method        | MNet-S       | MNet-L       | ENet-B0      | ENet-B1      | RN-18        | Mean         |
|---------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Perceptron    | 0.098        | 0.167        | 0.272        | 0.283        | 0.082        | 0.180        |
| Fine-Tune     | 0.043        | 0.066        | 0.238        | 0.232        | 0.030        | 0.121        |
| Naive Bayes   | 0.232        | 0.366        | 0.421        | 0.399        | 0.021        | 0.287        |
| SOvR          | 0.259        | 0.323        | 0.449        | 0.459        | 0.224        | 0.342        |
| NCM           | 0.442        | 0.474        | <b>0.516</b> | <b>0.514</b> | <u>0.463</u> | 0.481        |
| Replay (20pc) | 0.453        | 0.480        | <u>0.529</u> | <u>0.532</u> | 0.446        | <b>0.488</b> |
| SLDA          | 0.445        | 0.454        | 0.472        | 0.460        | 0.442        | 0.454        |
| ExLL-M        | <u>0.463</u> | <u>0.493</u> | 0.504        | 0.487        | 0.440        | <u>0.477</u> |
| ExLL-P        | <b>0.470</b> | <b>0.501</b> | 0.500        | 0.482        | <b>0.475</b> | <u>0.485</u> |
| ExLL-F        | <b>0.481</b> | <b>0.511</b> | 0.511        | 0.495        | <b>0.492</b> | <b>0.498</b> |

Table III presents the performance of the models across different CNN architectures for [low-shot instance ordering](#).

## How does CNN choice affect performance?

### OpenLORIS

- Performance is best with **EfficientNets** and worst with **ResNet-18** using [instance ordering](#)
  - EfficientNet yields better performance and requires fewer computational resources
- For low-shot instance ordering, the **EfficientNet** backbone CNNs again [outperformed the other backbone CNNs](#).

Places-365 for two data ordering methods **iid** and **class-iid**.

| Method        | IID    |        |         |         |       | Class-IID |        |         |         |       | Mean  |
|---------------|--------|--------|---------|---------|-------|-----------|--------|---------|---------|-------|-------|
|               | MNet-S | MNet-L | ENet-B0 | ENet-B1 | RN-18 | MNet-S    | MNet-L | ENet-B0 | ENet-B1 | RN-18 |       |
| Perceptron    | 0.303  | 0.344  | 0.352   | 0.340   | 0.294 | 0.004     | 0.003  | 0.012   | 0.013   | 0.005 | 0.167 |
| Fine-Tune     | 0.214  | 0.252  | 0.293   | 0.280   | 0.217 | 0.003     | 0.003  | 0.006   | 0.006   | 0.003 | 0.127 |
| Naive Bayes   | 0.028  | 0.093  | 0.250   | 0.249   | 0.003 | 0.028     | 0.093  | 0.250   | 0.249   | 0.003 | 0.124 |
| NCM           | 0.285  | 0.332  | 0.361   | 0.356   | 0.322 | 0.265     | 0.309  | 0.336   | 0.329   | 0.300 | 0.319 |
| Replay (20pc) | 0.289  | 0.323  | 0.354   | 0.348   | 0.261 | 0.251     | 0.279  | 0.297   | 0.295   | 0.235 | 0.293 |
| SLDA          | 0.362  | 0.397  | 0.412   | 0.405   | 0.362 | 0.362     | 0.397  | 0.412   | 0.405   | 0.362 | 0.387 |
| ExLL-M        | 0.375  | 0.347  | 0.392   | 0.336   | 0.312 | 0.347     | 0.362  | 0.381   | 0.367   | 0.349 | 0.356 |
| ExLL-P        | 0.354  | 0.380  | 0.381   | 0.370   | 0.343 | 0.352     | 0.378  | 0.381   | 0.373   | 0.343 | 0.365 |
| ExLL-F        | 0.444  | 0.478  | 0.488   | 0.476   | 0.440 | 0.444     | 0.473  | 0.486   | 0.479   | 0.439 | 0.464 |

Places-LT for two data ordering methods **iid** and **class-iid**, Places-LT tests how well models perform with severe **imbalance**

| Method        | IID    |        |         |         |       | Class-IID |        |         |         |       | Mean  |
|---------------|--------|--------|---------|---------|-------|-----------|--------|---------|---------|-------|-------|
|               | MNet-S | MNet-L | ENet-B0 | ENet-B1 | RN-18 | MNet-S    | MNet-L | ENet-B0 | ENet-B1 | RN-18 |       |
| Perceptron    | 0.152  | 0.185  | 0.213   | 0.206   | 0.149 | 0.017     | 0.028  | 0.071   | 0.073   | 0.015 | 0.110 |
| Fine-Tune     | 0.136  | 0.163  | 0.197   | 0.191   | 0.141 | 0.015     | 0.021  | 0.071   | 0.075   | 0.004 | 0.101 |
| Naive Bayes   | 0.015  | 0.050  | 0.199   | 0.213   | 0.100 | 0.015     | 0.050  | 0.199   | 0.213   | 0.001 | 0.105 |
| SOvR          | 0.089  | 0.149  | 0.262   | 0.245   | 0.146 | 0.089     | 0.149  | 0.262   | 0.245   | 0.146 | 0.178 |
| NCM           | 0.265  | 0.309  | 0.336   | 0.329   | 0.300 | 0.265     | 0.309  | 0.336   | 0.329   | 0.300 | 0.307 |
| Replay (20pc) | 0.239  | 0.267  | 0.290   | 0.282   | 0.223 | 0.241     | 0.268  | 0.306   | 0.295   | 0.193 | 0.260 |
| SLDA          | 0.290  | 0.318  | 0.338   | 0.328   | 0.300 | 0.290     | 0.319  | 0.338   | 0.328   | 0.300 | 0.314 |
| ExLL-M        | 0.356  | 0.392  | 0.407   | 0.400   | 0.360 | 0.311     | 0.331  | 0.338   | 0.331   | 0.345 | 0.357 |
| ExLL-P        | 0.265  | 0.297  | 0.315   | 0.305   | 0.277 | 0.265     | 0.300  | 0.314   | 0.303   | 0.273 | 0.291 |
| ExLL-F        | 0.324  | 0.349  | 0.362   | 0.351   | 0.331 | 0.325     | 0.351  | 0.359   | 0.350   | 0.330 | 0.343 |

## How robust are methods to scale and imbalance?

### Places-365 & Place-LT

- We compute the harmonic mean across orderings, emphasizing the importance of performing well on both orderings
- **Best methods: ExLL-F->ExLL-P ->ExLL-M->SLDA**
- **ExLL-M was the least affected** by dataset imbalance while prototype-based inference and pairwise fusion showed a 7.4% and 12.1% loss in performance respectively when trained with Places-LT.

F-SIOL-310 was selected to observe how the online continual learning methods perform in low-shot 82 continuous learning applications.

| Method        | 5-Shot       |              |              |              |              |              | 10-Shot      |              |              |              |              |              |
|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|               | MNet-S       | MNet-L       | ENet-B0      | ENet-B1      | RN-18        | Mean         | MNet-S       | MNet-L       | ENet-B0      | ENet-B1      | RN-18        | Mean         |
| Perceptron    | 0.181        | 0.177        | 0.406        | 0.454        | 0.049        | 0.253        | 0.158        | 0.223        | 0.354        | 0.458        | 0.051        | 0.248        |
| Fine-Tune     | 0.183        | 0.205        | 0.416        | 0.460        | 0.090        | 0.270        | 0.127        | 0.199        | 0.389        | 0.453        | 0.090        | 0.251        |
| Naive Bayes   | 0.344        | 0.554        | 0.816        | 0.828        | 0.035        | 0.515        | 0.320        | 0.537        | 0.806        | 0.854        | 0.015        | 0.506        |
| SOvR          | 0.592        | 0.666        | 0.679        | 0.693        | 0.428        | 0.611        | 0.561        | 0.702        | 0.650        | 0.752        | 0.504        | 0.633        |
| CBCL          | <u>0.853</u> | <u>0.878</u> | <u>0.886</u> | 0.838        | 0.848        | 0.860        | 0.883        | 0.906        | 0.888        | 0.892        | 0.869        | 0.887        |
| NCM           | <u>0.853</u> | 0.871        | <u>0.886</u> | <b>0.885</b> | <b>0.885</b> | <u>0.876</u> | 0.883        | 0.906        | 0.893        | 0.913        | 0.896        | 0.898        |
| Replay (20pc) | 0.541        | 0.632        | 0.594        | 0.612        | 0.624        | 0.600        | 0.625        | 0.694        | 0.714        | 0.722        | 0.731        | 0.697        |
| SLDA          | <b>0.880</b> | <b>0.899</b> | <b>0.912</b> | <b>0.903</b> | <u>0.854</u> | <b>0.889</b> | 0.924        | <b>0.948</b> | <b>0.938</b> | <b>0.936</b> | 0.910        | <u>0.931</u> |
| ExLL-M        | 0.842        | 0.873        | 0.863        | 0.847        | 0.851        | 0.855        | <u>0.926</u> | <u>0.942</u> | <b>0.938</b> | <u>0.928</u> | <b>0.948</b> | <b>0.936</b> |
| ExLL-P        | 0.827        | 0.832        | 0.799        | 0.755        | 0.803        | 0.803        | <b>0.927</b> | 0.934        | 0.897        | 0.879        | <u>0.930</u> | 0.913        |
| ExLL-F        | <b>0.889</b> | <b>0.905</b> | <b>0.887</b> | <u>0.854</u> | <b>0.885</b> | <b>0.884</b> | <u>0.961</u> | <b>0.966</b> | <b>0.952</b> | <b>0.943</b> | <b>0.968</b> | <b>0.958</b> |

## How well are methods in low-shot continual learning?

### F-SIOL-310

- For the 5-shot scenario, **ExLL-F** is slightly outperformed by SLDA (0.884 vs. 0.889 respectively).
- On the other hand, for the 10-shot scenario, **ExLL-F** significantly outperformed the next-best methods, ExLL-M and SLDA (0.958 vs. 0.936 and 0.931 respectively).



## ExLL – Prototype Analysis

| Test Image                                                                                                                             | Hits                                                                                            | Near Hits                                                                                        | Near Misses                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <br>"Shampoo"<br>correctly predicted as<br>"Shampoo"  | <br>"Shampoo" | <br>"Shampoo" | <br>"Lotion"     |
| <br>"Shampoo"<br>Wrongly predicted as<br>"Lotion"     | <br>"Lotion"  | <br>"Lotion"  | <br>"Toothpaste" |
| <br>"Toothpaste"<br>Wrongly predicted as<br>"Shampoo" | <br>"Shampoo" | <br>"Shampoo" | <br>"Book"       |

Fig. 3: Example of "Hits", "Near Hits", and "Near Misses" for the F-SIOL-310 dataset. The test image in the top row is an example of a True Positive result, the test image in the middle row is a False Negative result, and the test image in the bottom row is a False Positive result.



## ExLL – Rule Extraction













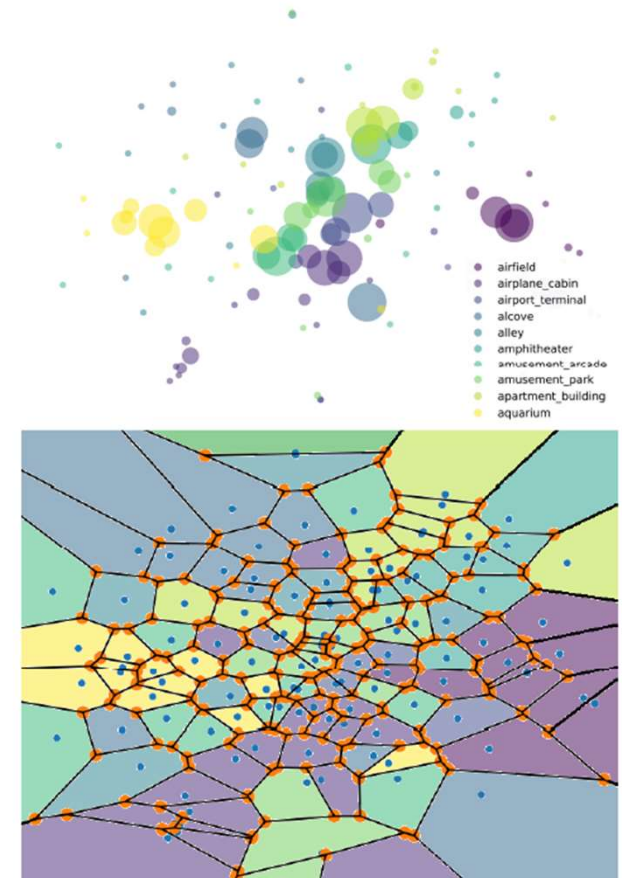
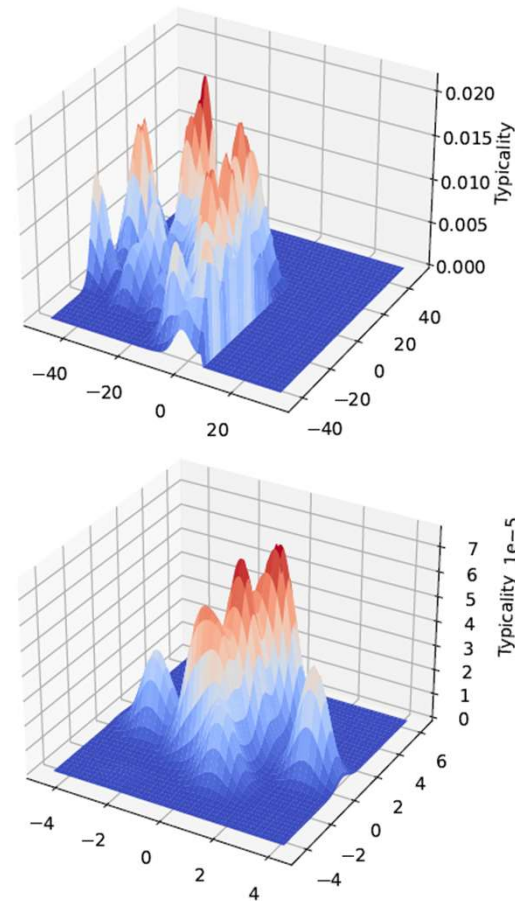
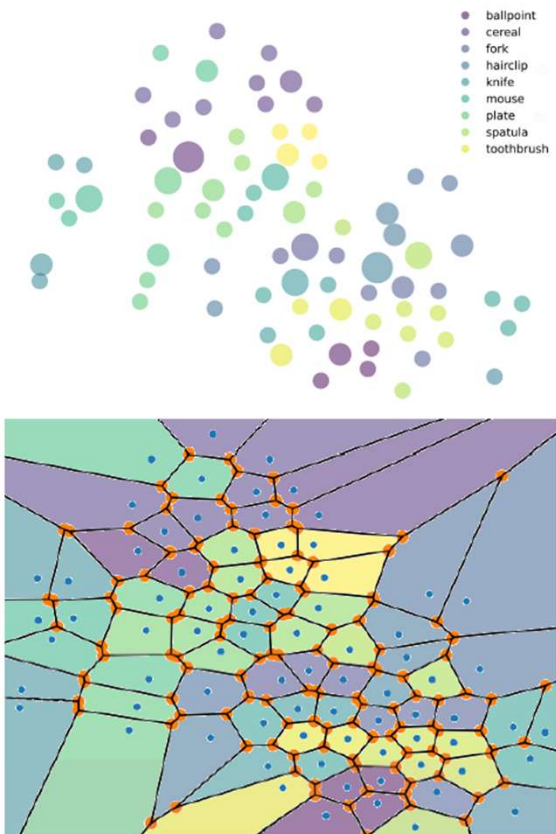
|         |    |                                                                                   |    |                                                                                   |    |                                                                                     |        |                          |
|---------|----|-----------------------------------------------------------------------------------|----|-----------------------------------------------------------------------------------|----|-------------------------------------------------------------------------------------|--------|--------------------------|
| Rule #1 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Aqueduct" |
| Rule #2 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Aqueduct" |
| Rule #3 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Arch"     |
| Rule #4 | IF |  | OR |  | OR |  | OR ... | THEN Class is "Arch"     |

Fig. 4: Explainable rules extracted from prototypes for the classes "Aqueduct" and "Arch" from Places-365. Each rule is made up of training images associated with the corresponding prototype.

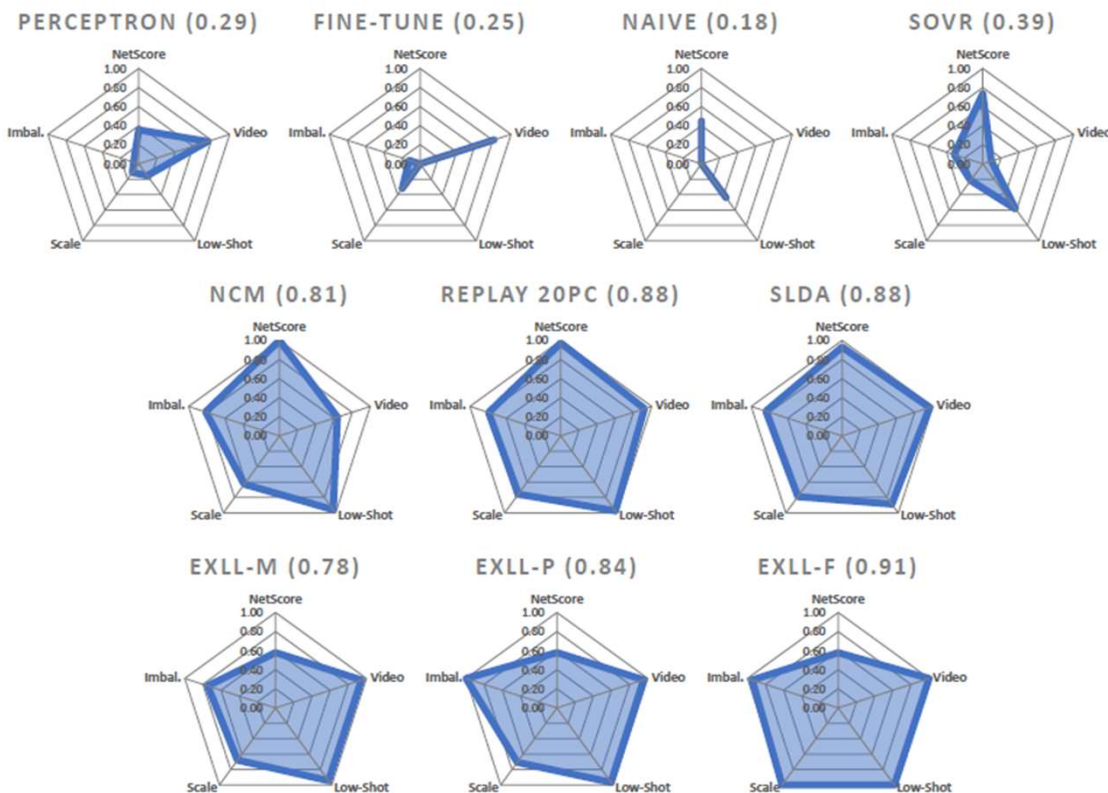
# ExLL – Topology Visualization

Visualizing the topology of learned prototypes acquired from the **FSIOL-310 dataset**.

Visualizing the topology of learned prototypes acquired from the **Places-365 dataset, for the first 10 classes**.



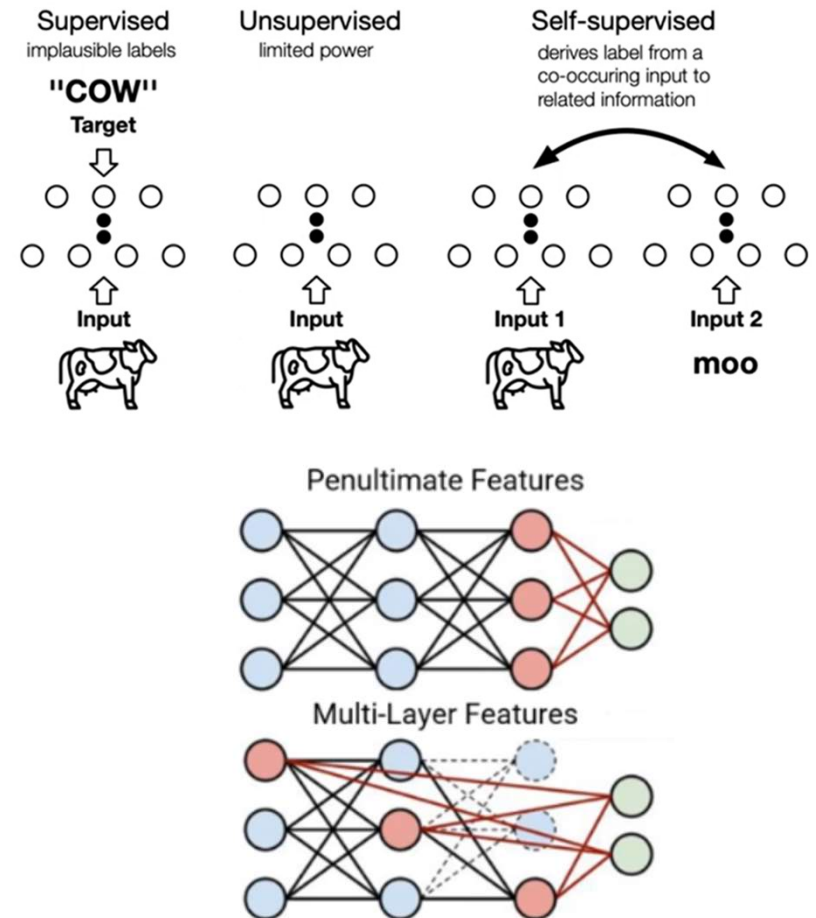
# Results Summary



- We studied the robustness of online continual learners across various axes useful for embedded learning:
  - Omega: classification efficacy, compute memory
  - Video: ability to learn from temporally correlated videos
  - Low-Shot: ability to generalize from few inputs
  - Scale: ability to scale to large-scale data
  - Imbal: ability to perform well on imbalanced data
- Main Takeaways:
  - **ExLL-F (0.91) showed the best overall performance.** Replay 20pc (0.88) and SLDA (0.88) outperformed the second-best ExLL model, ExLL-P (0.84). The worst-performing ExLL model, ExLL-M (0.78) is also outperformed by NCM (0.81).
  - ExLL is less efficient with respect to computation and memory requirements.

# Future Research Directions

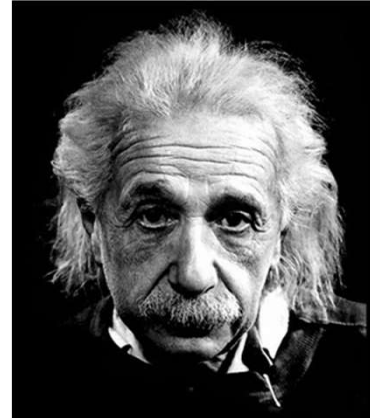
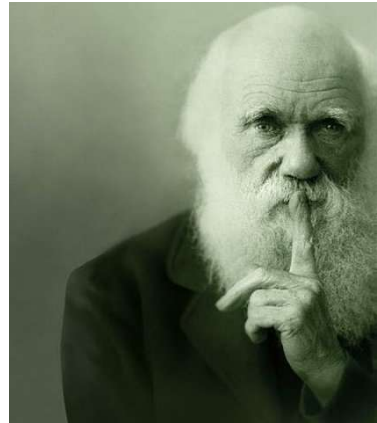
- Self-supervised pre-trained features
  - Currently do not work well for mobile CNNs
- Additional pre-training datasets
- Use of multi-layer features
- Online learners that update feature representations
  - More compute time and concept drift
- Techniques to improve low-shot learning
- Additional tasks and efficiency improvements





*“...it is not the strongest that survives; but...the one that is able best to adapt...to the changing environment....”*

*L.C. Megginson, re “On the Origin of Species”*



*“Once you stop learning, you start dying.”*

*Albert Einstein*

<https://www.izlesene.com/iz/memcn3342>

Thank you