Continual Learning Beyond Catastrophic Forgetting in Class-Incremental Scenarios

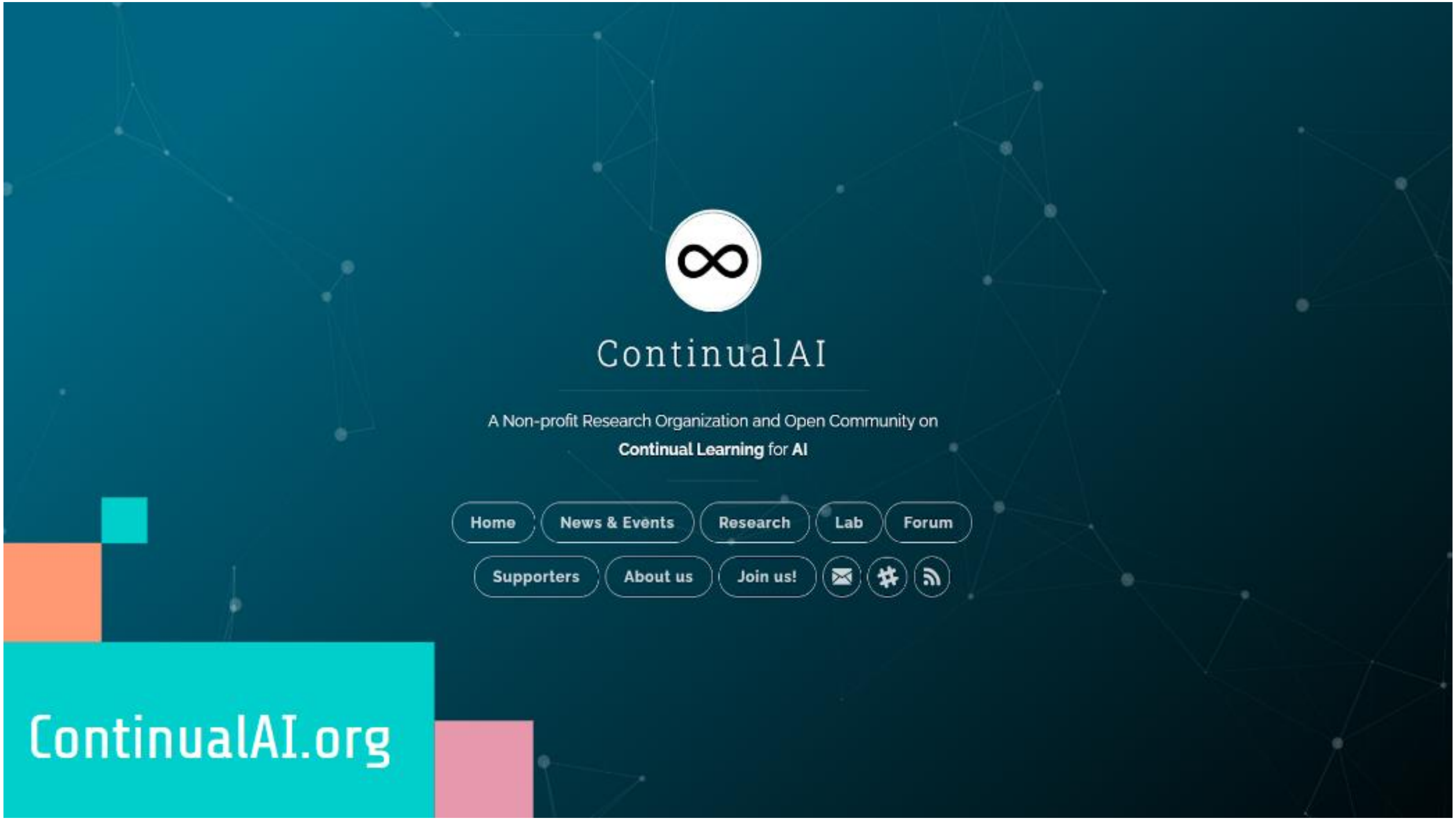
CoLLAs 2023 Tutorial

Vincenzo Lomonaco
University of Pisa, ContinualAI
vincenzo.lomonaco@unipi.it

Antonio Carta
University of Pisa
Antonio.carta@unipi.it

Pervasive AI Lab @ UniPi

ContinualAI Non-Profit



Part I

Deep Continual Learning

Fundamental concepts and current focus

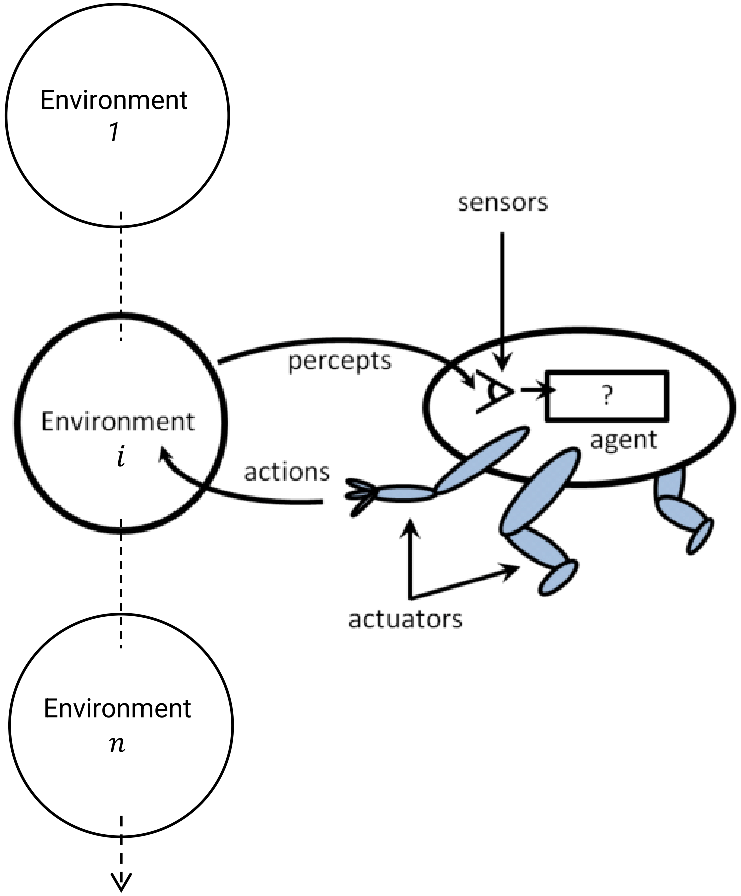
Lifelong Learning Artificial Agents

Our goals:

1. **Incremental Learning**: knowledge and skills accumulation and re-use

2. **Fast Adaptation**: adapt to ever-changing environments

AI Agent Architecture
(Russel & Norvig, 1995 - 2022)



A Long-Desired Objective

- Incremental learning with rule-based systems (**Diederich, 1987**)
- Forgetting in Neural Networks (**French, 1989**)
- Incremental learning with Kernel Machines (**Tat-Jun, 1999**)
- Continual Learning (**Ring, 1998**)
- Lifelong Learning (**Thrun, 1998**)
- Dataset Shift (**Quiñonero-Candela, 2008**)
- Never-Ending Learning (**Mitchell, 2009**)
- Concept Drift Adaptation (**Ditzler, 2015**)
- Deep Continual Learning (**Kirkpatrick, 2016**)
- Lifelong (Language) Learning (**Liu, 2018**)

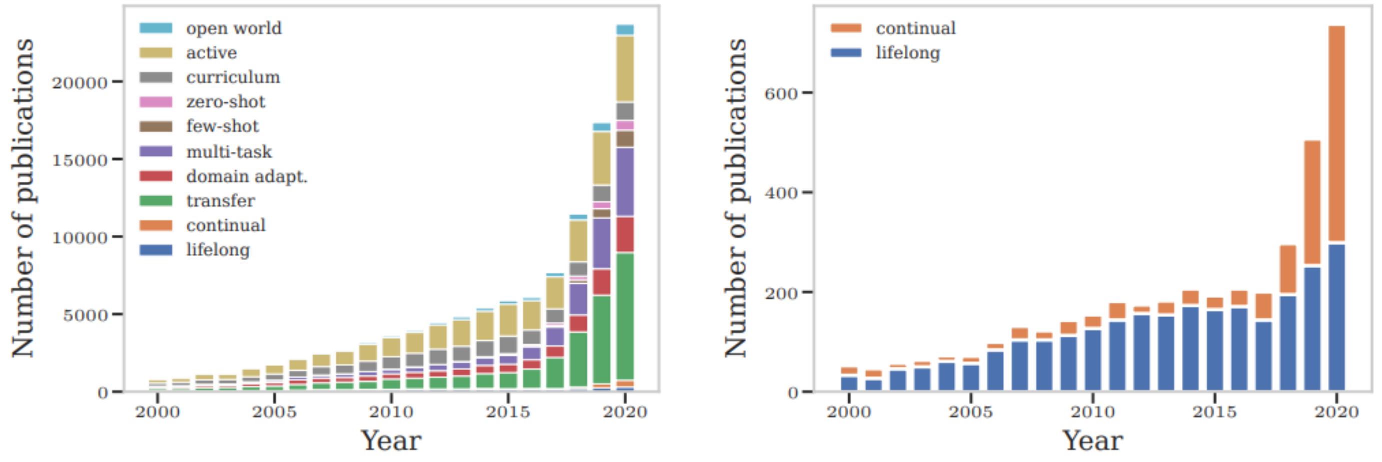


Figure 1: Per year machine learning publications. Left: cumulative amount of papers across keywords with continuous components that influence continual learning practice, see Section 2. Right: increasing use of "continual" machine learning, demonstrating a shift in use of terminology with respect to the preceding emphasis on the term "lifelong". Data queried using the Microsoft Academic Graph utilities (Sinha et al., 2015) based on keyword occurrence in the abstract.

CLEVA-Compass: A Continual Learning EValuation Assessment Compass to Promote Research Transparency and Comparability, Martin Mundt et al. preprint, 2021

Dealing with Non-Stationary Environments

"The world is changing and we must change with it" - Ragnar Lothbrok

What is Concept Drift (CD)?

What it is:

- A change in the real world
- Affects the input/output distribution
- Disrupt the model's predictions

What it's not:

- It's not noise
- It's not outliers

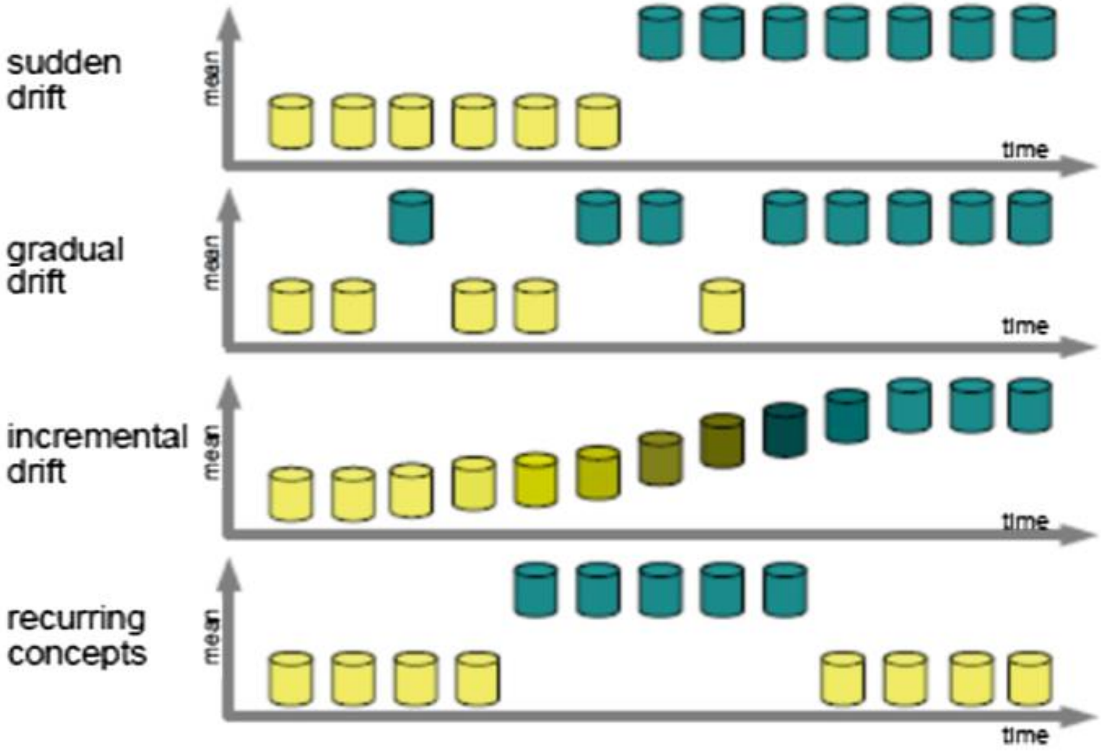


Fig. 3. Types of concept drift

Figure: V. Lemaire et al. «A Survey on Supervised Classification on Data Streams"

CD – A Probabilistic Definition

- Given an input $x_1, x_2, \dots, x_t$ of class y we can apply bayes theorem:

$$p(y|x_t) = \frac{p(y)p(x_t|y)}{p(x_t)}$$

- $p(y)$ is the prior for the output class (concept)
- $p(x_t|y)$ the conditional probability
- Why do we care?
  - Different causes for changes in each term
  - Different consequences (do we need to retrain our model?)

Dataset Shift Nomenclature

Notation:

- x covariates/input features
- y class/target variable
- p(y, x) joint distribution
- sometimes the x→ y relationship is referred with the generic term “concept“

The nomenclature is based on **causal assumptions**:

- x→y problems: class label is causally determined by input. Example: credit card fraud detection
- y→x problems: class label determines input. Example: medical diagnosis

Moreno-Torres, Jose G., Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. “A Unifying View on Dataset Shift in Classification.” Pattern Recognition 45, no. 1 (January 2012): 521–30. https://doi.org/10.1016/j.patcog.2011.06.019.

Dataset Shift Nomenclature

Dataset Shift: $p_{tra}(x, y) \neq p_{tst}(x, y)$

- Informally: any change in the distribution is a shift

Covariate shift: happens in X→Y problems when

- $p_{tra}(y|x) = p_{tst}(y|x)$ and $p_{tra}(x) \neq p_{tst}(x)$
- informally: the input distribution changes, the input->output relationship does not

Prior probability shift: happens in Y→X problems when

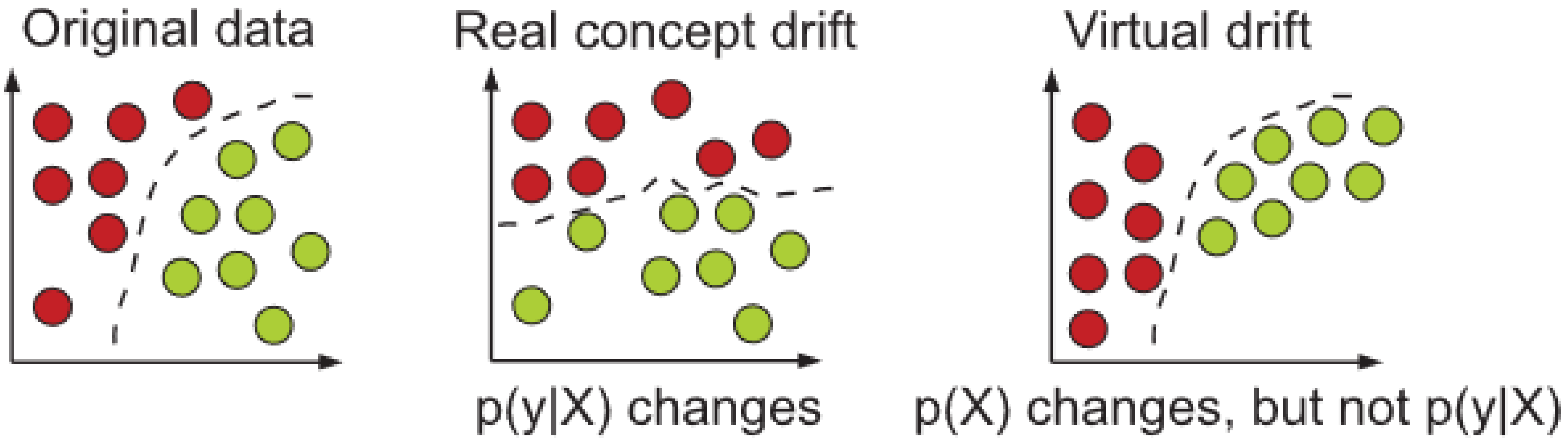
- $p_{tra}(x|y) = p_{tst}(x|y)$ and $p_{tra}(y) \neq p_{tst}(y)$
- Informally: output->input relationship is the same but the probability of each class is changed

Concept shift:

- $p_{tra}(y|x) \neq p_{tst}(y|x)$ and $p_{tra}(x) = p_{tst}(x)$ in X→Y problems.
- $p_{tra}(x|y) \neq p_{tst}(x|y)$ and $p_{tra}(y) = p_{tst}(y)$ in Y→X problems.
- Informally: the «concept» (i.e. the class)

Moreno-Torres, Jose G., Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. "A Unifying View on Dataset Shift in Classification." Pattern Recognition 45, no. 1 (January 2012): 521–30. https://doi.org/10.1016/j.patcog.2011.06.019.
Dataset Shift in Machine Learning, J. Qui˜nonero-Candela et al. MIT Press, 2008.

Real vs Virtual Drift



Gama, João, et al. A survey on concept drift adaptation. ACM computing surveys (CSUR) 46.4 (2014): 1-37.

Causes of Shifts

Sampling bias:

- The world is fixed but we only see a part of it
- The «visible part» changes over time, causing a shift
- We will also call it virtual drift
- Examples: bias in polls, limited observability of environments, change of domain…

Non-stationary environments:

- The world is continuously changing
- We will also call it real drift
- Examples: weather, financial markets, …

Deep Continual Learning has been mostly focus on "virtual drifts" and with **knowledge accumulation rather than adaptation**.

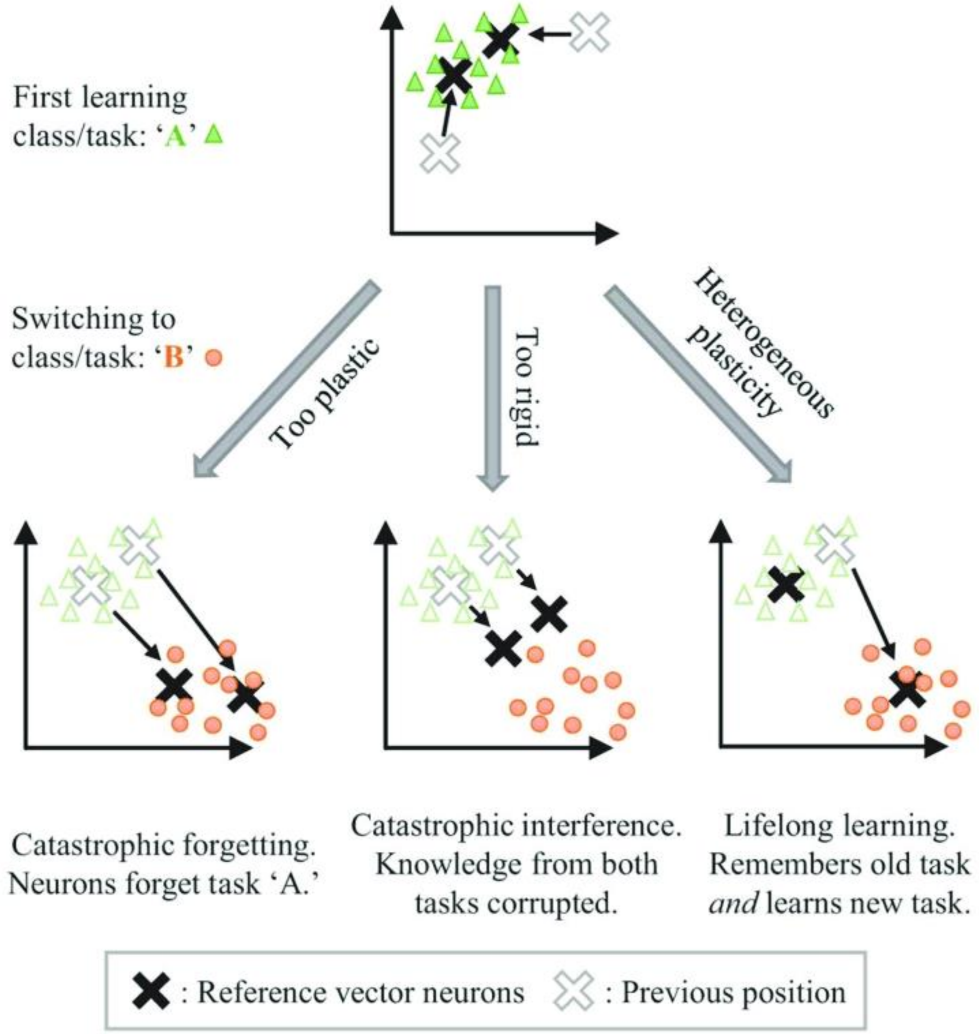
The Stability-Plasticity Dilemma

Stability-Plasticity Dilemma:

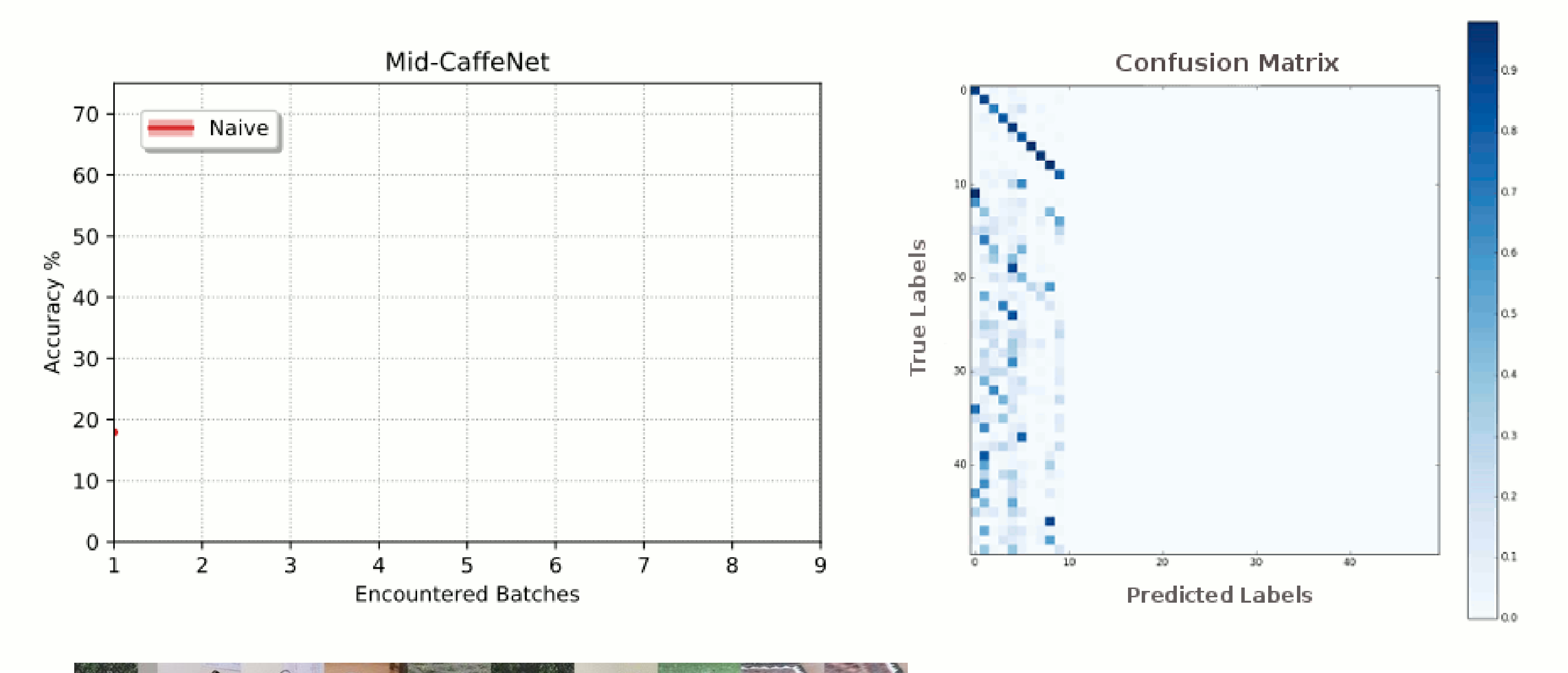
- Remember past concepts
- Learn new concepts
- Generalize

First Problem in Deep Learning:

- Catastrophic Forgetting



Catastrophic Forgetting



- *A set of new objects (classes) each day*
- *10 the first day, 5 the following*

CORe50: a new Dataset and Benchmark for Continuous Object Recognition, V. Lomonaco & D. Maltoni. Conference on Robot Learning (CoRL), 2017.

Deep Continual Learning

Definition, Objectives, Desiderata

Continual Learning Objectives and Desiderata

In continual learning (CL) data arrives in a streaming fashion as a (possibly infinite) sequence of learning experiences $S = e_1, \ldots, e_n$. For a supervised classification problem, each experience e_i consists of a batch of samples $\mathcal{D}^i$, where each sample is a tuple $\langle x_k^i, y_k^i \rangle$ of input and target, respectively, and the labels y_k^i are from the set $\mathcal{Y}^i$, which is a subset of the entire universe of classes $\mathcal{Y}$. Usually $\mathcal{D}^i$ is split into a separate train set $\mathcal{D}^i_{train}$ and test set $\mathcal{D}^i_{test}$.

A continual learning algorithm $\mathcal{A}^{CL}$ is a function with the following signature:

$$\mathcal{A}^{CL} : \langle f_{i-1}^{CL}, \mathcal{D}^i_{train}, \mathcal{M}_{i-1}, t_i \rangle \rightarrow \langle f_i^{CL}, \mathcal{M}_i \rangle \qquad (1)$$

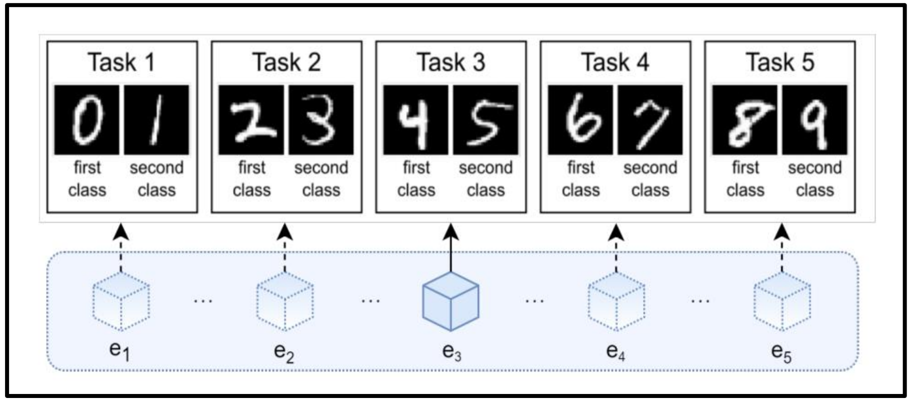
where f_i^{CL} is the model learned after training on experience e_i, $\mathcal{M}_i$ a buffer of past knowledge, such as previous samples or activations, stored from the previous experiences and usually of fixed size. The term t_i is a task label which may be used to identify the correct data distribution.

The objective of a CL algorithm is to minimize the loss $\mathcal{L}_S$ over the entire stream of data S:

$$\mathcal{L}_S(f_n^{CL}, n) = \frac{1}{\sum_{i=1}^{n} |\mathcal{D}^i_{test}|} \sum_{i=1}^{n} \mathcal{L}_{exp}(f_n^{CL}, \mathcal{D}^i_{test}) \qquad (2)$$

$$\mathcal{L}_{exp}(f_n^{CL}, \mathcal{D}^i_{test}) = \sum_{j=1}^{|\mathcal{D}^i_{test}|} \mathcal{L}(f_n^{CL}(\boldsymbol{x}_j^i), y_j^i), \qquad (3)$$

where the loss $\mathcal{L}(f_n^{CL}(x), y)$ is computed on a single sample $\langle x, y \rangle$, such as cross-entropy in classification problems.



Ex-Model: Continual Learning from a Stream of Trained Models, Carta et al, 2021.

Continual Learning Objectives and Desiderata

In continual learning (CL) data arrives in a streaming fashion as a (possibly infinite) sequence of learning experiences $S = e_1, \ldots, e_n$. For a supervised classification problem, each experience e_i consists of a batch of samples $\mathcal{D}^i$, where each sample is a tuple $\langle x_k^i, y_k^i \rangle$ of input and target, respectively, and the labels y_k^i are from the set $\mathcal{Y}^i$, which is a subset of the entire universe of classes $\mathcal{Y}$. Usually $\mathcal{D}^i$ is split into a separate train set $\mathcal{D}^i_{train}$ and test set $\mathcal{D}^i_{test}$.

A continual learning algorithm $\mathcal{A}^{CL}$ is a function with the following signature:

$$\mathcal{A}^{CL} : \langle f_{i-1}^{CL}, \mathcal{D}^i_{train}, \mathcal{M}_{i-1}, t_i \rangle \rightarrow \langle f_i^{CL}, \mathcal{M}_i \rangle \qquad (1)$$

where f_i^{CL} is the model learned after training on experience e_i, $\mathcal{M}_i$ a buffer of past knowledge, such as previous samples or activations, stored from the previous experiences and usually of fixed size. The term t_i is a task label which may be used to identify the correct data distribution.

The objective of a CL algorithm is to minimize the loss $\mathcal{L}_S$ over the entire stream of data S:

$$\mathcal{L}_S(f_n^{CL}, n) = \frac{1}{\sum_{i=1}^{n} |\mathcal{D}^i_{test}|} \sum_{i=1}^{n} \mathcal{L}_{exp}(f_n^{CL}, \mathcal{D}^i_{test}) \qquad (2)$$

$$\mathcal{L}_{exp}(f_n^{CL}, \mathcal{D}^i_{test}) = \sum_{j=1}^{|\mathcal{D}^i_{test}|} \mathcal{L}(f_n^{CL}(\boldsymbol{x}_j^i), y_j^i), \qquad (3)$$

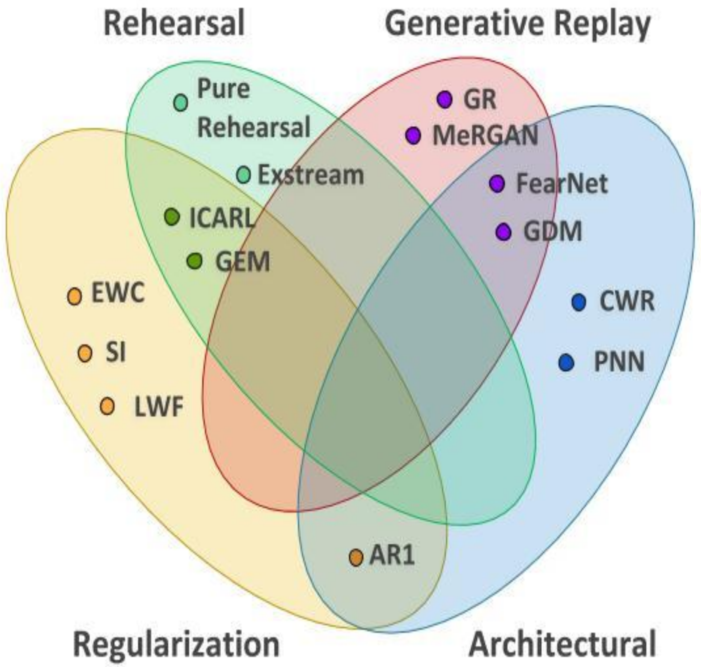
where the loss $\mathcal{L}(f_n^{CL}(\boldsymbol{x}), y)$ is computed on a single sample $\langle \boldsymbol{x}, y \rangle$, such as cross-entropy in classification problems.

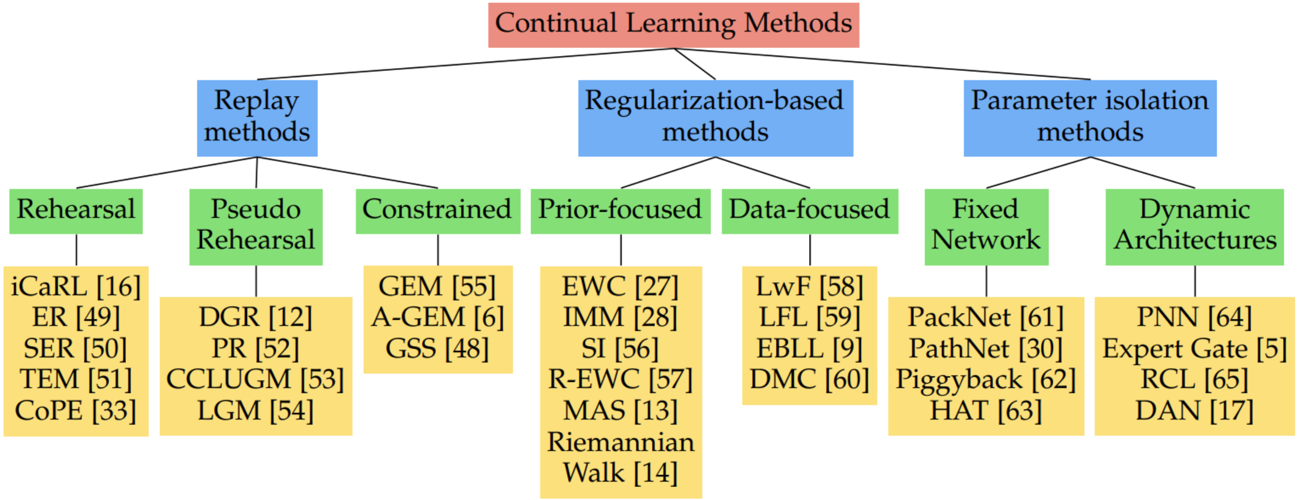
Desiderata

- Replay-Free Continual Learning
- Memory and Computationally Bounded
- Task-free Continual Learning
- Online Continual Learning

Continual Learning Approaches

Continual Learning Approaches





Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges, Lesort et al. Information Fusion, 2020.
A continual learning survey: Defying forgetting in classification tasks. De Lange et al, TPAMI 2021.

Experience Replay

A basic approach

1. Sample randomly from the current experience data

2. Fill your fixed Random Memory (RM)

3. Replace examples randomly to maintain an approximate equal number of examples for experience

Algorithm 1 Pseudocode explaining how the external memory RM is populated across the training batches. Note that the amount h of patterns to add progressively decreases to maintain a nearly balanced contribution from the different training batches, but no constraints are enforced to achieve a class-balancing.

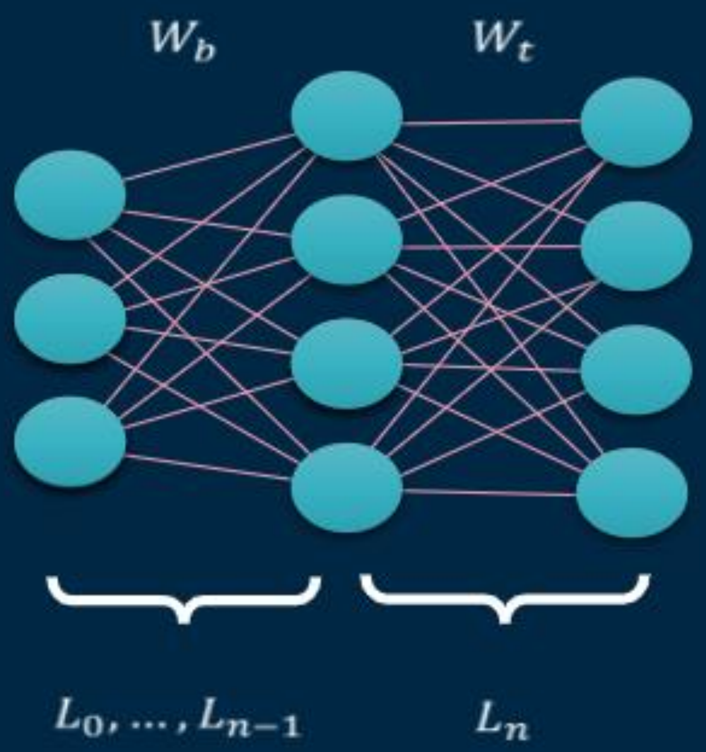
```
RM = ∅
RM_size = number of patterns to be stored in RM
for each training batch B_i:
    train the model on shuffled B_i ∪ RM
    h = RM_size / i
    R_add = random sampling h patterns from B_i
    R_replace = { ∅                                    if i == 1
                { random sample h patterns from RM     otherwise
    RM = (RM − R_replace) ∪ R_add
```

$$h = \frac{RM_{size}}{i}$$

$$R_{replace} = \begin{cases} \varnothing & \text{if } i == 1 \\ \text{random sample } h \text{ patterns from } RM & \text{otherwise} \end{cases}$$

$$RM = (RM - R_{replace}) \cup R_{add}$$

Elastic Weights Consolidation



$$\tilde{L}_\mu = L_\mu + \lambda \sum_k \Omega_k^\mu \left(\bar{\theta}_k - \theta_k\right)^2$$

$$\underbrace{w_k^v = \int \left(\frac{\partial}{\partial \theta_k} \log f(x;\theta_k)\right)^2 f(x;\theta_k)\, dx}_{\text{Fisher Information}}$$

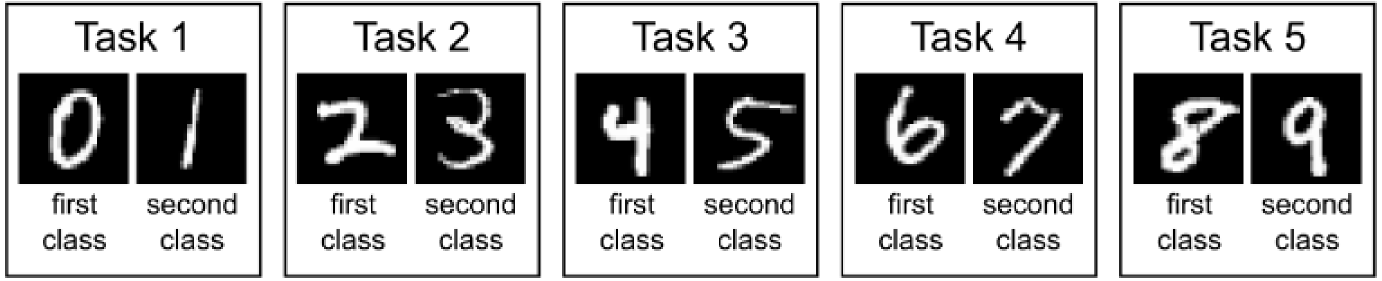
Overcoming catastrophic forgetting in neural networks, Kirkpatrick et al. PNAS, 2017.

Continual Learning Benchmarks

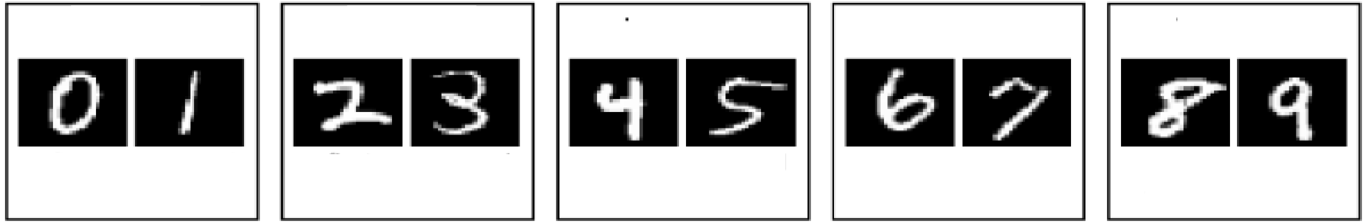
Datasets, Scenarios and Evaluation metrics

Continual Learning Scenarios

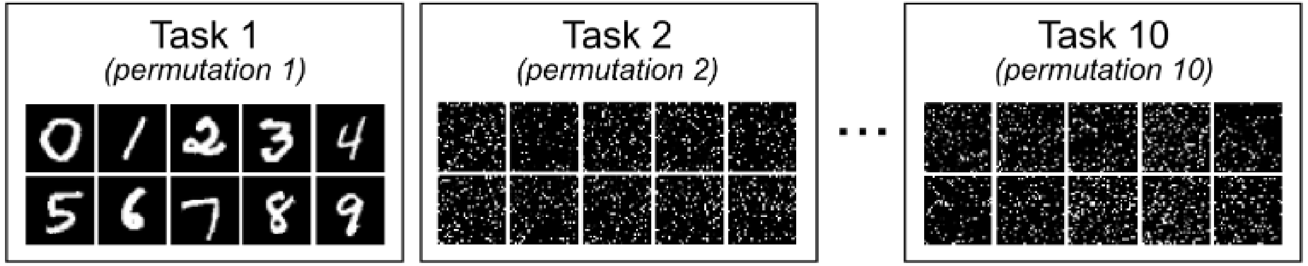
1. **Task-Incremental**: every experience is a different task.



1. **Class-Incremental**: every experience contains examples of different classes of a unique classification problem.



1. **Domain-Incremental**: every experience contains examples (from a different domain) of the same classes.



Three scenarios for continual learning, Van de Ven, 2019

Continual Learning Benchmarks

Benchmark
Split MNIST/Fashion MNIST
Rotation MNIST
Permutation MNIST
iCIFAR10/100
SVHN
CUB200
CORe50
iCubWorld28
iCubWorld-Transformation
LSUN
ImageNet
Omniglot
Pascal VOC
Atari
RNN CL benchmark
CRLMaze (based on VizDoom)
DeepMind Lab

Current Focus

- Class-Inc / Multi-Task (Often with Task Supervised Signals)
- I.I.D by Parts
- Few Big Tasks
- Unrealistic / Toy Datasets
- Mostly Supervised
- Accuracy

Recent Growing Trend

- Single-Incremental-Task
- High-Dimensional Data Streams (highly non-i.i.d.)
- Natural / Realistic Datasets
- Mostly Unsupervised
- Scalability and Efficiency

Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges, Lesort et al. Information Fusion, 2020.

Continual Learning Evaluation

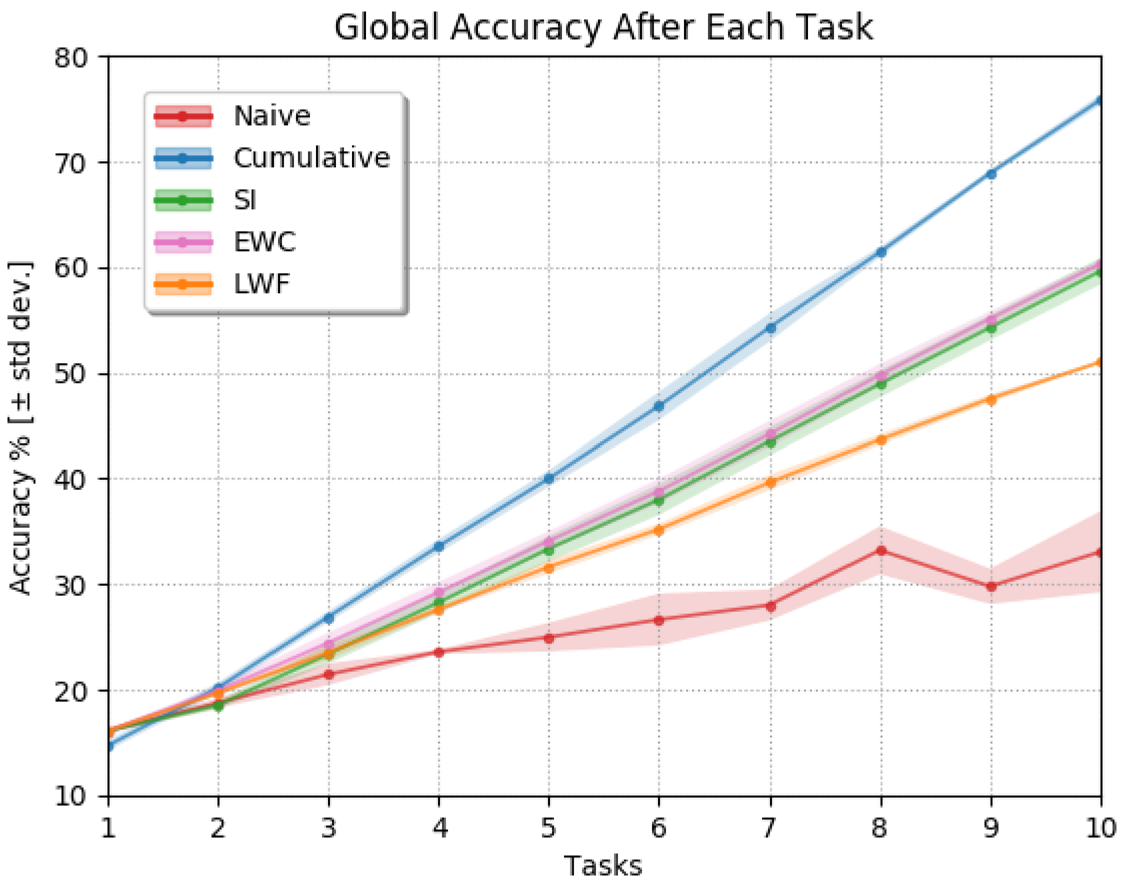


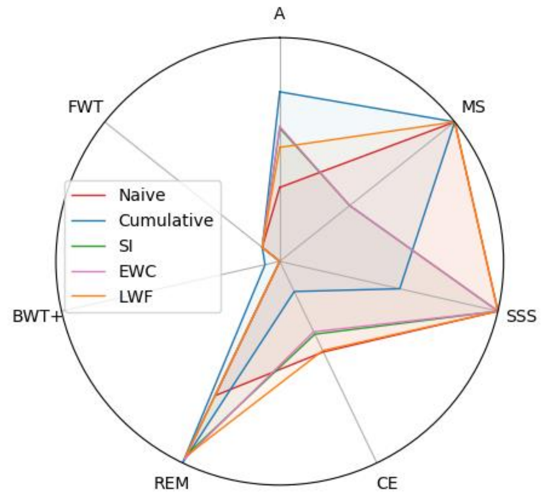
Table 2: Elements in R accounted to compute the Accuracy (white and cyan elements), BWT (in cyan), and FWT (in light gray) criteria. $R^* = R_{ii}$, Tr_i = training, Te_i= test tasks.

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

$$\textbf{Average Accuracy:}\ \ \mathrm{ACC} = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}$$

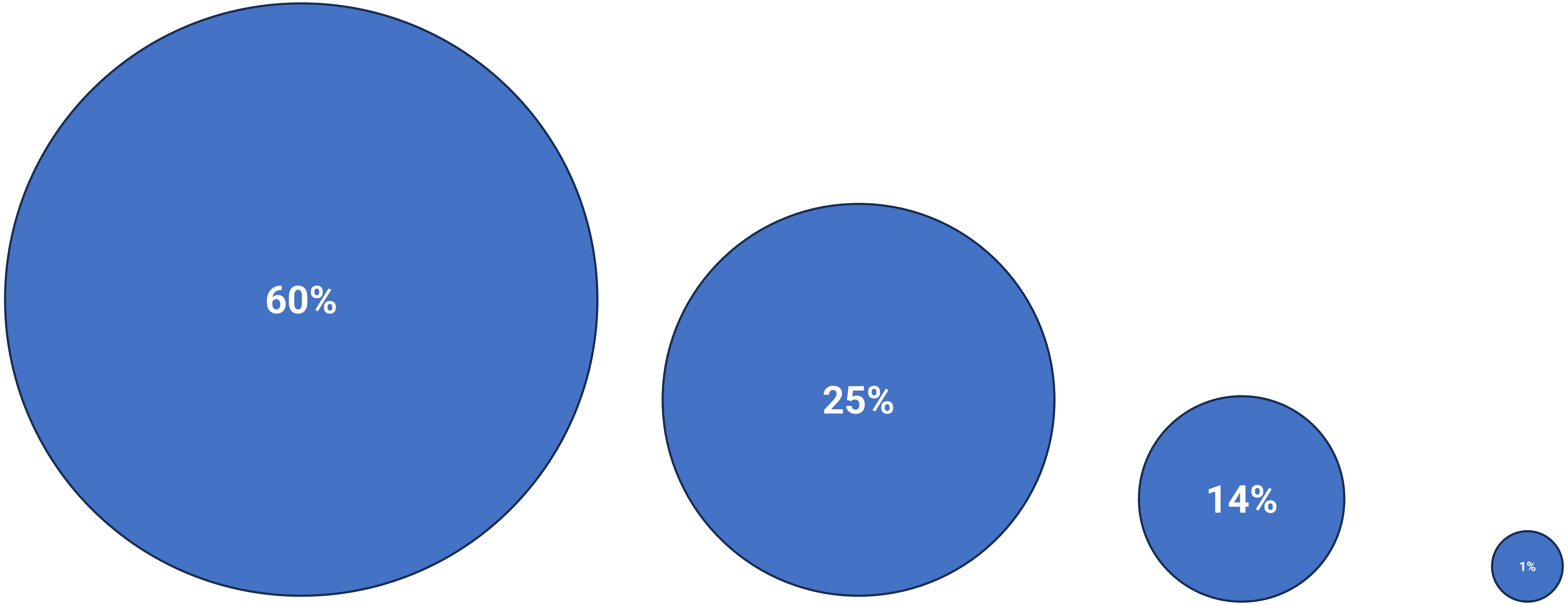
$$\textbf{Backward Transfer:}\ \ \mathrm{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i}$$

$$\textbf{Forward Transfer:}\ \ \mathrm{FWT} = \frac{1}{T-1}\sum_{i=2}^{T} R_{i-1,i} - \bar{b}_i.$$



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

Significant focus on CIL…



2230 hits
Class-Incremental

934 hits
Task-Incremental

543 hits
Domain-Incremental

?
Other

* Hits computed by keywords search (e.g., «class-incremental» and «continual learning» on google scholar on the 13-08-2023)

Is Class-Incremental Enough For Continual Learning?

Short answer: No

«Is Class-Incremental Enough For Continual Learning?», Cossu et al, Frontiers in CS, 2021

Why CIL? Pros & Cons

***Pros**:*

1. it's easy to setup
2. Any static classification benchmark can be converted in a CIL benchmark
3. It exacerbate catastrophic forgetting problem (often assumed to be the most difficult scenario but… *it depends on the evaluation metrics chosen*)

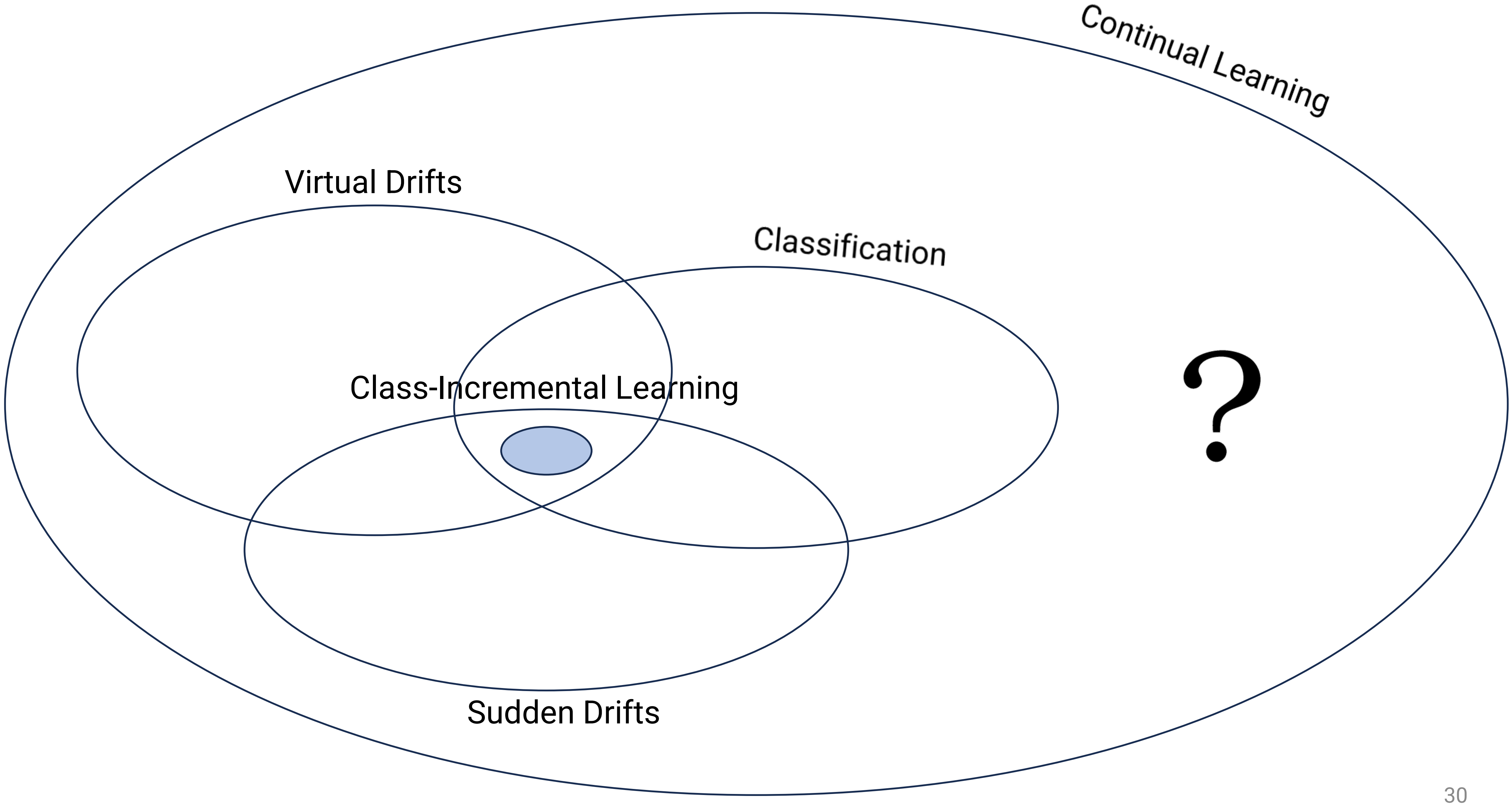
***Cons**:*

1. It is a peculiar / specific problem
2. Quite unrealistic setting for many applications

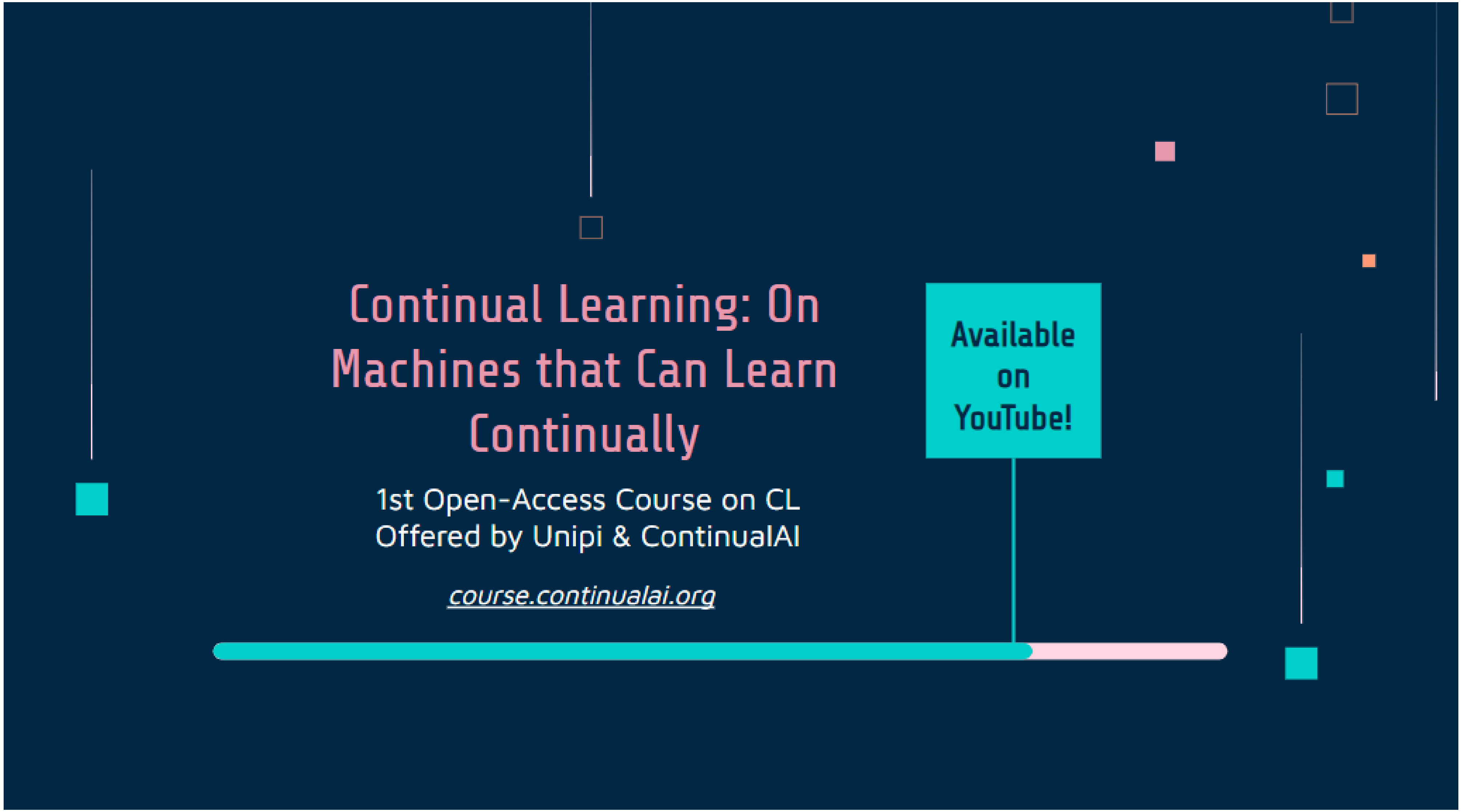
Definition 1. Class-Incremental Learning *aims to learn from an evolutive stream with incoming new classes [32]. Assume there is a sequence of B training tasks*[1] $\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^B\}$ *without overlapping classes, where* $\mathcal{D}^b = \{(\mathbf{x}_i^b, y_i^b)\}_{i=1}^{n_b}$ *is the b-th incremental step with n_b training instances.* $\mathbf{x}_i^b \in \mathbb{R}^D$ *is an instance of class $y_i \in Y_b$, Y_b is the label space of task b, where $Y_b \cap Y_{b'} = \varnothing$ for $b \neq b'$. We can only access data from $\mathcal{D}^b$ when training task b. The ultimate goal of CIL is to continually build a classification model for all classes. In other words, the model should not only acquire the knowledge from the current task $\mathcal{D}^b$ but also preserve the knowledge from former tasks. After each task, the trained model is evaluated over all seen classes $\mathcal{Y}_b = Y_1 \cup \cdots Y_b$.*

Deep Class-Incremental Learning: A Survey, Zhou et al. 2015,
iCaRL: Incremental Classifier and Representation Learning", Rebuffi, 2017

…a Tiny Portion of CL!

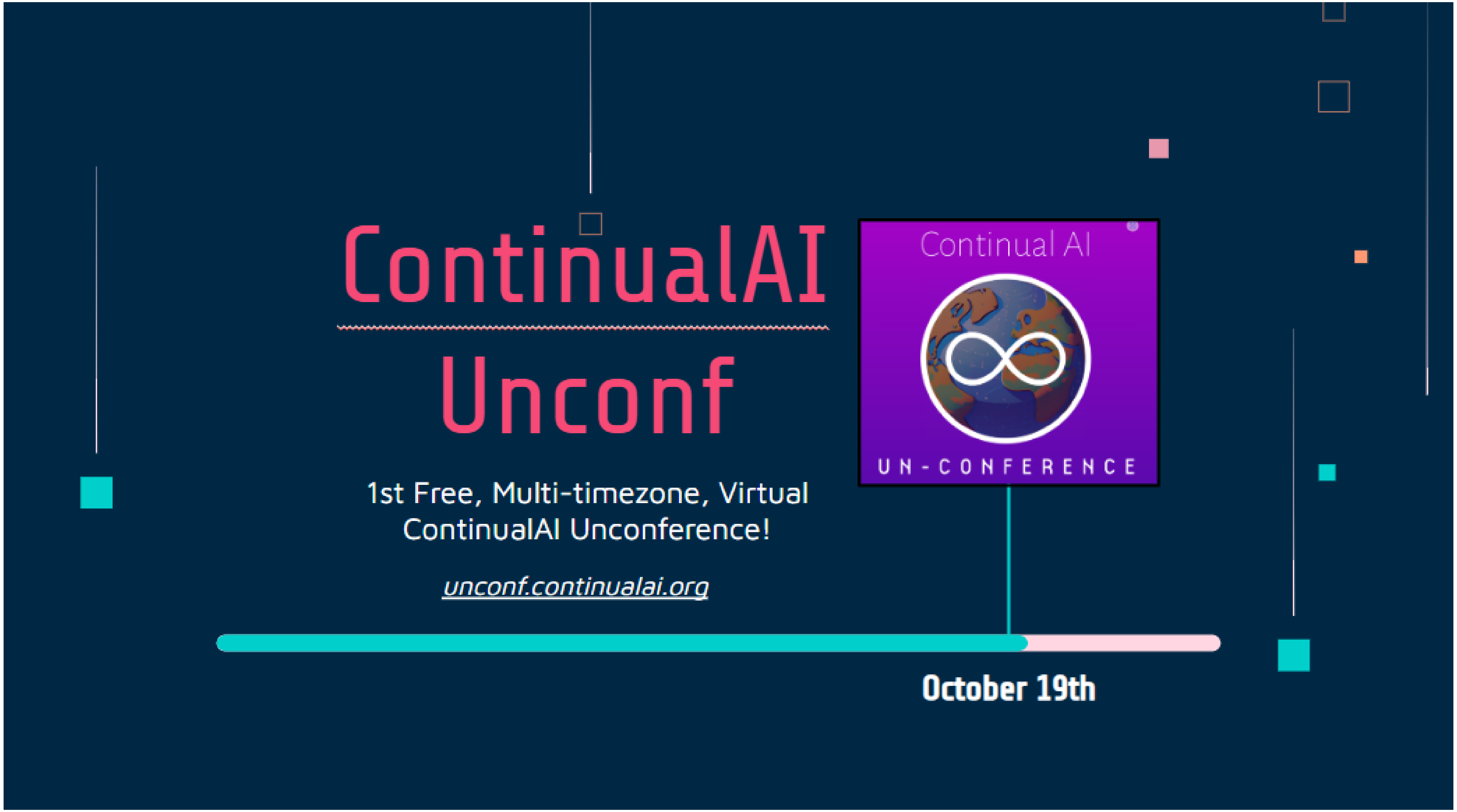


More on the basics…



https://course.continualai.org

ContinualAI Unconf: *completely free and virtual!*



https://unconf.continualai.org

Part 2 – Beyond CIL

Real World Streams

Metrics and Evaluation

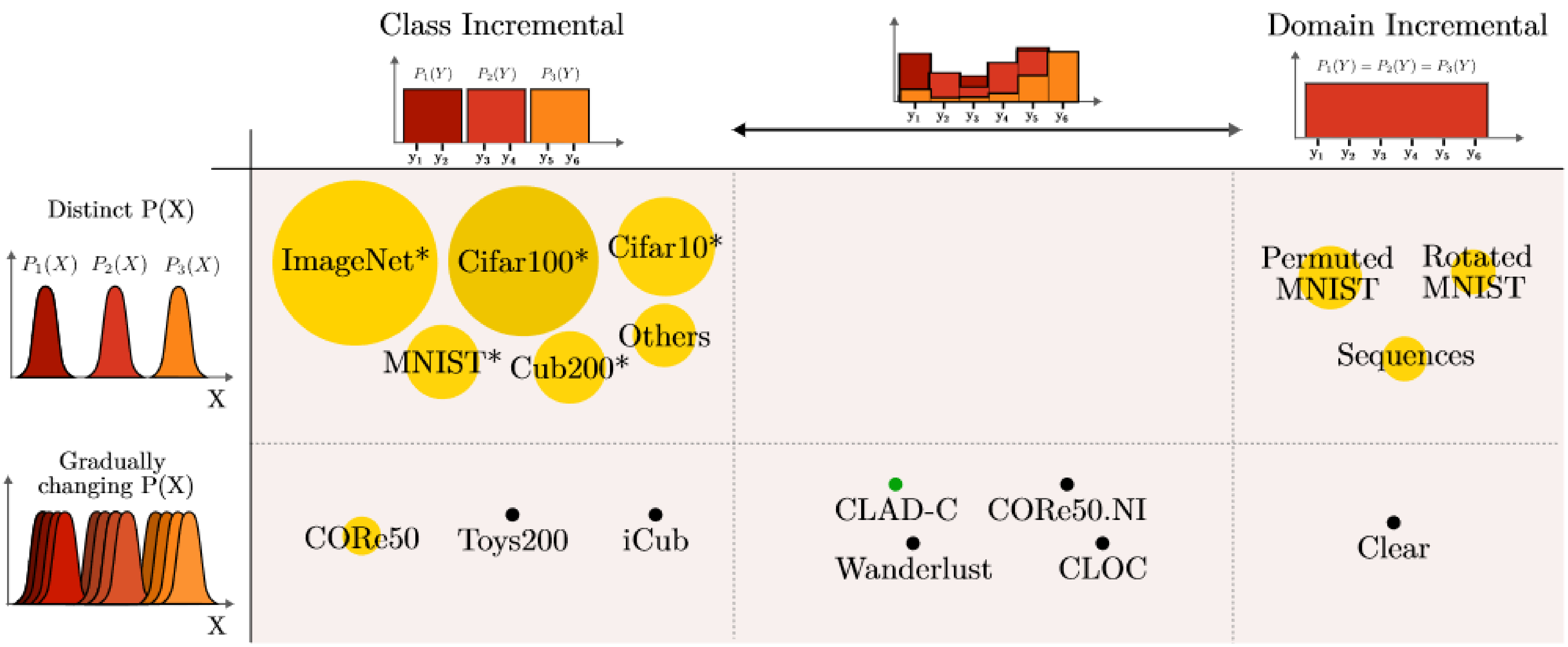
Distributed Continual Learning

Towards Realistic Streams

Existing benchmarks with natural streams and controllable simulators

- Benchmarks desiderata
- Real drifts and streaming data
- Simulators and synthetic generators

Classic CL Benchmarks



Eli Verwimp et al. 2022. “CLAD: A Realistic Continual Learning Benchmark for Autonomous Driving.”

Properties of Real World Streams

Real World Streams

- Gradual and sharp drifts
- New domains and classes appear over time
- Repetitions of old domain and classes
- Imbalanced distributions
- Real drift changes the objective function
- Temporal consistency (e.g. video frames)

CIL (as used in popular benchmarks)

- Sharp drifts
- New classes
- No repetitions
- Balanced data
- Virtual drift
- No temporal consistency

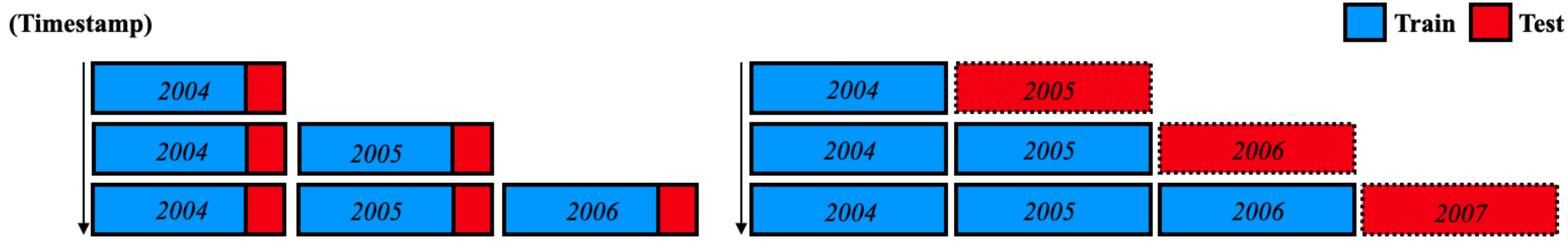
Consequences of Realistic Streams

- **Gradual drifts**: methods can’t easily freeze old components, task/domain inference is more difficult.
- **New domains**: new classes implicitly provide labels, domains don’t.
- **Repetitions**: methods can’t easily freeze old components.
- **Imbalance**: reservoir sampling mimics the unbalance in the stream.
- **Real drift**: Replay data may be incorrect.

Evaluation with Real vs Virtual Drifts

Example: Data ordered by class (0,0,0,0,0,1,1,1,1…)
- Persistent classifier (predict previous class) is optimal
- The model can (and should) exploit temporal consistency!

- **Virtual drift**
  - sampling bias
  - Evaluation on a static test set
  - a.k.a. most of the CL research
- **Real drift**
  - Concept drift. Example: politician roles and affiliations to political party
  - Evaluation on the next data (e.g. **prequential evaluation**)
  - Not a lot of research in CL right now



IID Protocol: *Train today, test on today*

Streaming Protocol: *Train today, test on tomorrow*

Image from CLEAR paper

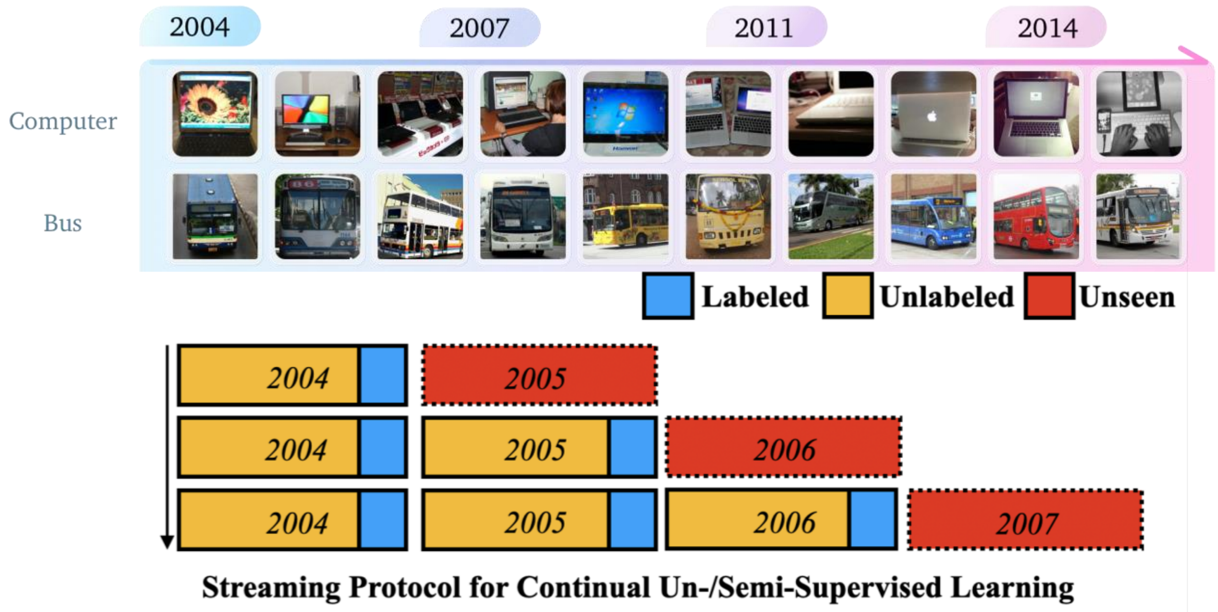
Real Drift - CLEAR / Wild-Time

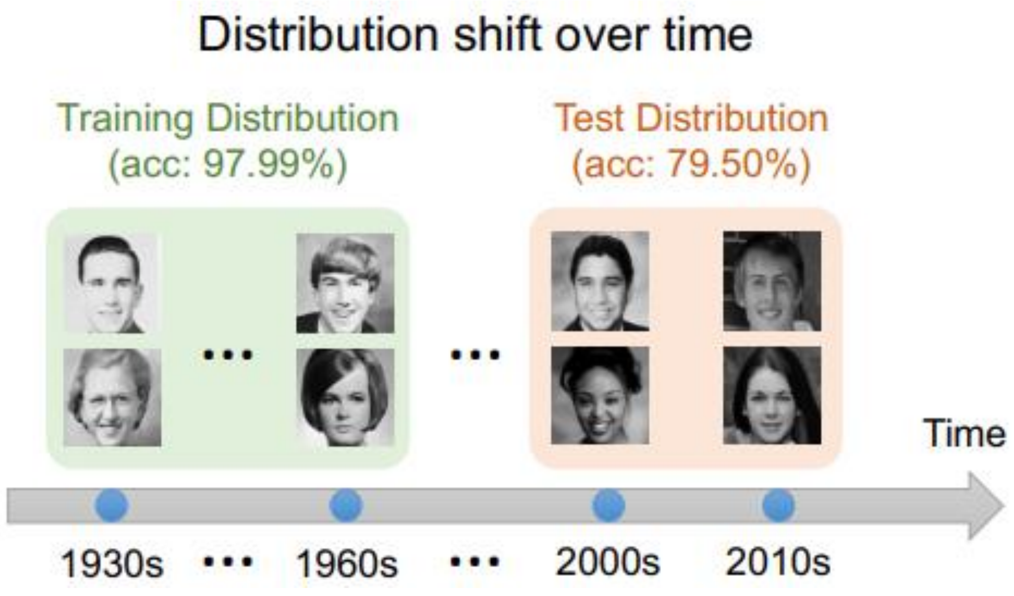
CLEAR

- real-world images with smooth temporal evolution
- Large unlabeled dataset (~7.8M images)
- Prequential evaluation
- Scenario: domain-incremental and semi-supervised

Wild-Time

- 5 datasets with temporal distribution drifts (real drift)
- Temporal metadata
- **Eval-Fix**: evaluation on static test data
- **Eval-Stream**: evaluate on the next K timestamps

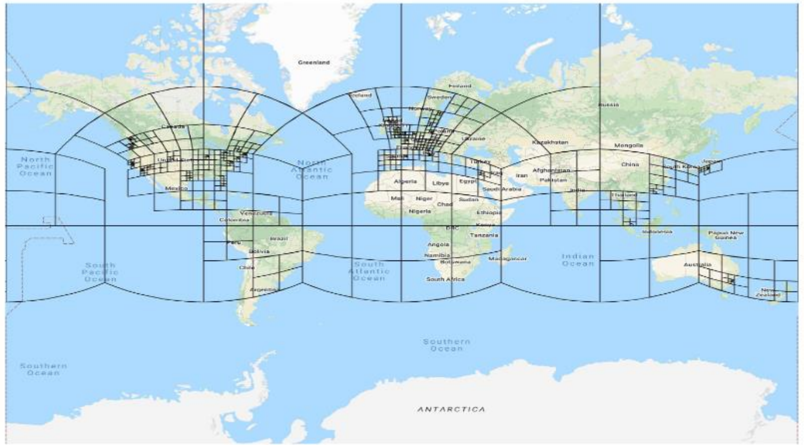




Z. Lin et al. *"The CLEAR Benchmark: Continual LEArning on Real-World Imagery" 2021*
Yao, Huaxiu et al. "Wild-Time: A Benchmark of in-the-Wild Distribution Shift over Time." NeurIPS 2022

Real Drift - CLOC – Continual Localization

- Images with geolocalization and timestamps
  - 9 years of data
  - 39M images
  - 2M for offline preprocessing
  - 712 classes (localization regions)



(a) S2 Cells in our dataset

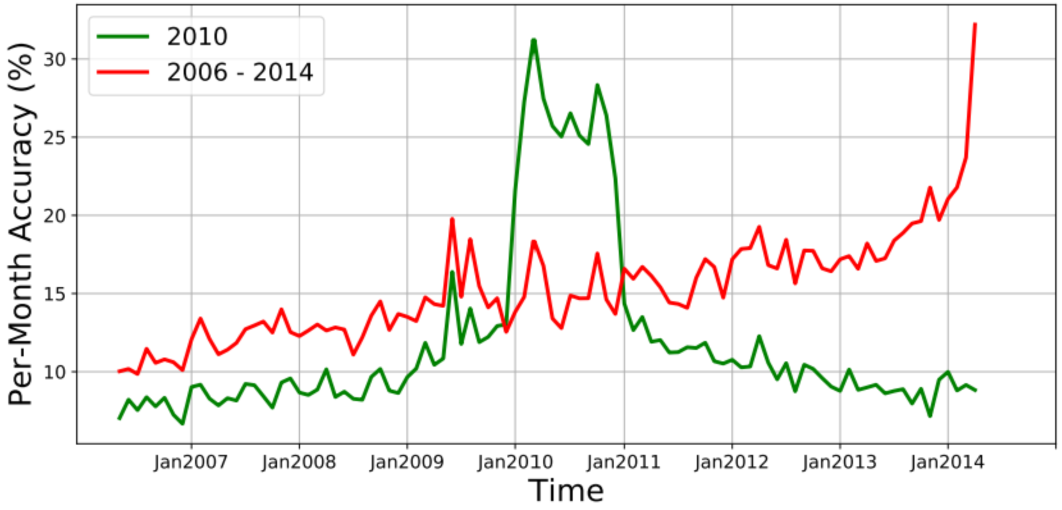


Figure 2. **Distribution shift in CLOC.** We train two supervised models, one using data from the entire temporal range and the other only on data from the year 2010. We evaluate both models on the full temporal range using the validation set (not seen during training). Due to non-stationarity in the data, the performance of the 2010 model drops sharply on data from other times.

Z. Cai et al. "Online Continual Learning with Natural Distribution Shifts: An Empirical Study with Visual Data." ICCV '21

Temporal Coherence - CoRE50

- **Temporally coherent** streams
- Domain-incremental, class-incremental, and repetitions
- **CL on-the-edge application**:
  - Given a pretrained model
  - Take a short video of a new object
  - Finetune the model

Continuous Object Recognition

- 50 classes
- Short videos of object manipulation with different background
- Temporal coherence from videos

Many scenarios: batch, online, with repetitions.

Lomonaco V. and Maltoni D. CORe50: a New Dataset and Benchmark for Continuous Object Recognition. CoRL2017.

Simulators and Synthetic Data

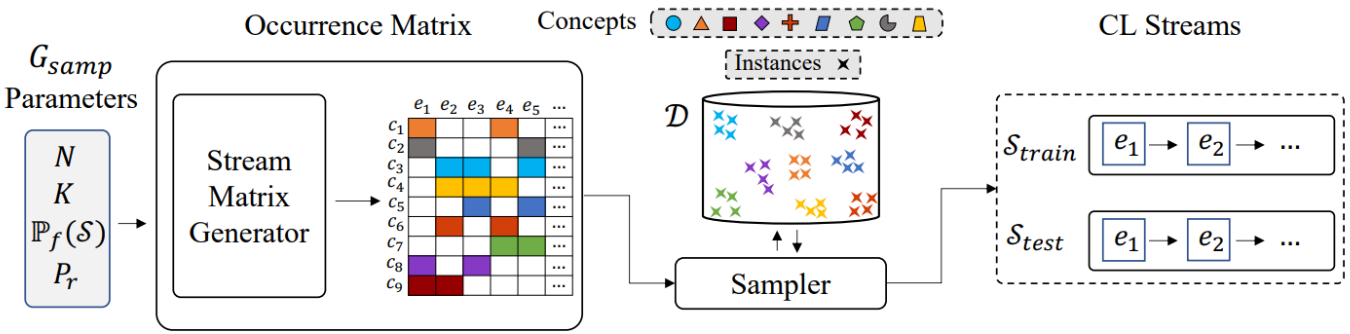
Driving simulation

- **Parameters**:
  - new classes
  - weather
  - illumination changes
- **Temporal consistency**

CIR Synthetic Generator

Poster session today!

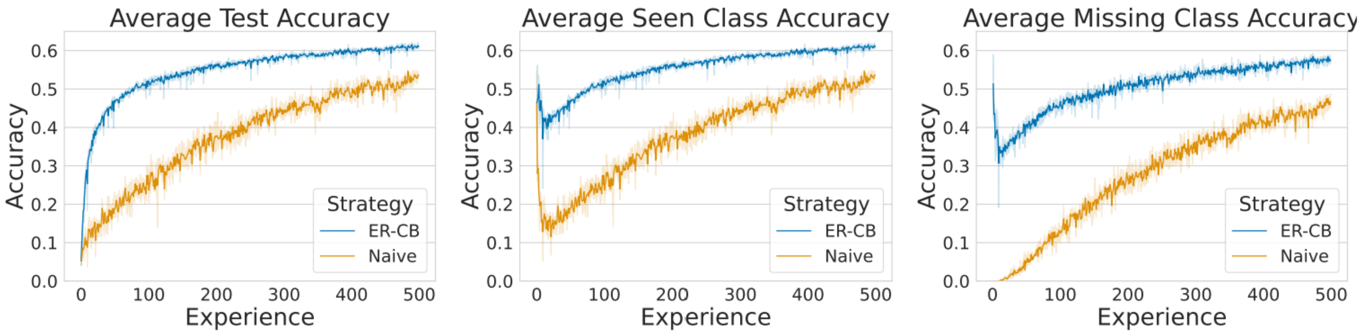
- Start from a static dataset (e.g. CIFAR100)
- Define distribution parameters: stream length, class balancing, repetitions, …
- Sample stream with the desired probability
- You can tweak the difficulty of the benchmark and check how different methods perform under different conditions



T. Hess et al. "A Procedural World Generation Framework for Systematic Evaluation of Continual Learning." 2021
H. Hemati et al. "Class-Incremental Learning with Repetition." CoLLAs '23

CIR – Results

Naive finetuning approaches replay for long streams with repetitions



Missing class accuracy improves over time, even for naive finetuning

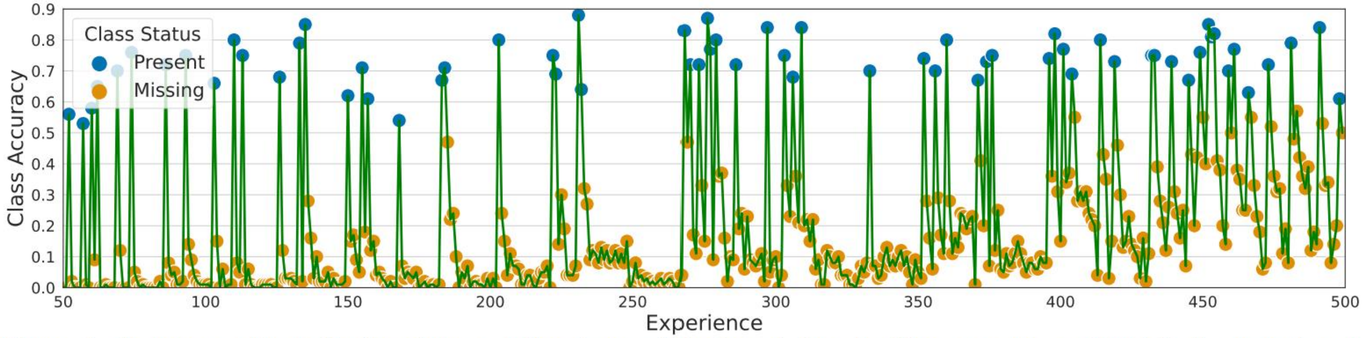


Figure 6: Accuracy of a particular class over the stream. The target class is either present or absent in the experiences indicated by the blue and orange points, respectively.

In unbalanced streams, class-balanced buffers and reservoir sampling are not effective

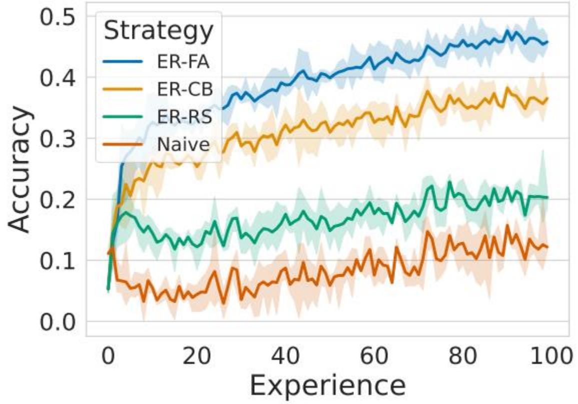


Figure 10: Accuracy of Infrequent Classes.

H. Hemati et al. "Class-Incremental Learning with Repetition." CoLLAs '23

Recap

- **Benchmarks desiderata**: gradual drifts, new domains and classes, repetitions, temporal coherence, real drift
- **Real drifts**: Wild-Time, CLEAR, CLOC. Prequential evaluation for real drifts
- **Streaming data**: CoRE50 (and many others)
- **Simulators and synthetic generators**: allow to control drift and evaluate over many different configurations

Metrics and Evaluation in Online CL

Metrics for online continual learning: cumulative accuracy, continual stability, linear probing

Results in online continual learning

Continual hyperparameter selection and robustness

Online CL (OCL)

Mandatory:

- **Online**: data arrives in *small mini-batches (possibly in a real-time stream)*. Strong constraints on memory and computational budget
- **Anytime inference**: ability to predict at every time, even during a drift.

Desiderata:

- **Task-Agnostic**: task labels are not available
- **Boundary-agnostic**: does not need knowledge about drifts (a.k.a. task-free)
  - Many OCL methods are NOT boundary-agnostic
  - CIL settings provide trivial boundaries (class labels)

Red diamonds = task boundaries

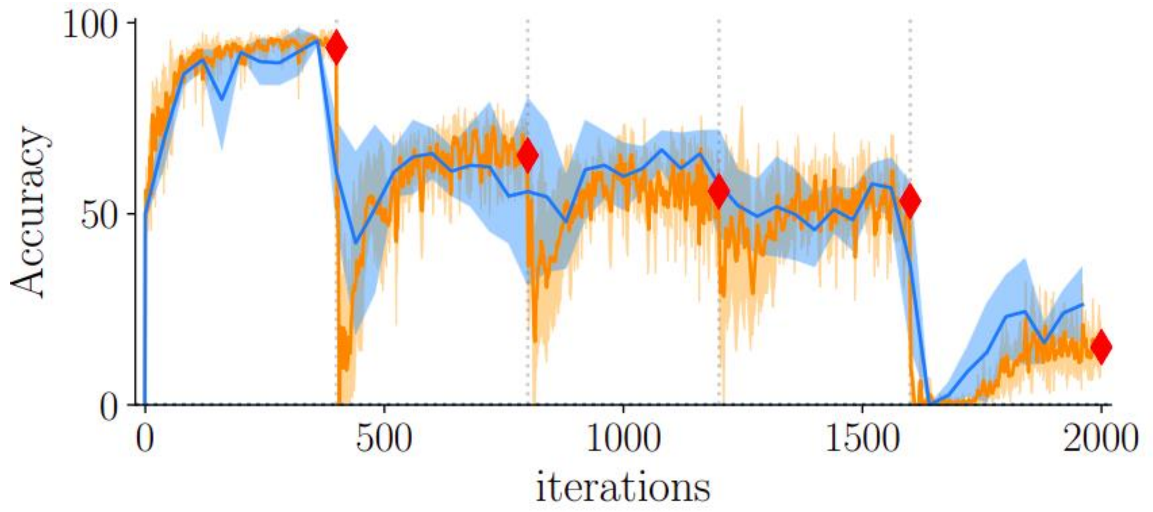


Image from Mathias Delange et. al, Continual Evaluation for Lifelong Learning: Identifying the stability gap, ICLR 2023

Replay-based Online CL

Online Continual Learning

```
# processing one example at a time
for (x_new, y_new) in train_stream:

    # performing k passes on the same data
    for k in train_passes:
        # optionally augment data
        x_new, y_new = augment(x_new, y_new)
        # extracting examples from the external memory
        x_mem, y_mem = augment(sample(memory))
        # perform an optimization step
        compute_loss_and_backprop(x_new, y_new, x_mem, y_mem)
        weights_udpate()

    # update the external memory
    update(memory, x_new, y_new)
    # eventually evaluate, inference only
    evaluation()
```

Sampling

RAR: Adversarial augumentations
MIR: Find interfered examples

Weights Update

A-GEM: uses memory only for gradient projection

Notes

GDumb cannot really considered an *Online Strategy* due to its latency in Infererence, but rather as a *Baseline.*

Loss

DER++: Logits Replay
ER-ACE: Bias Mitigation
SCR: Contrastive
ER + LwF: Distillation

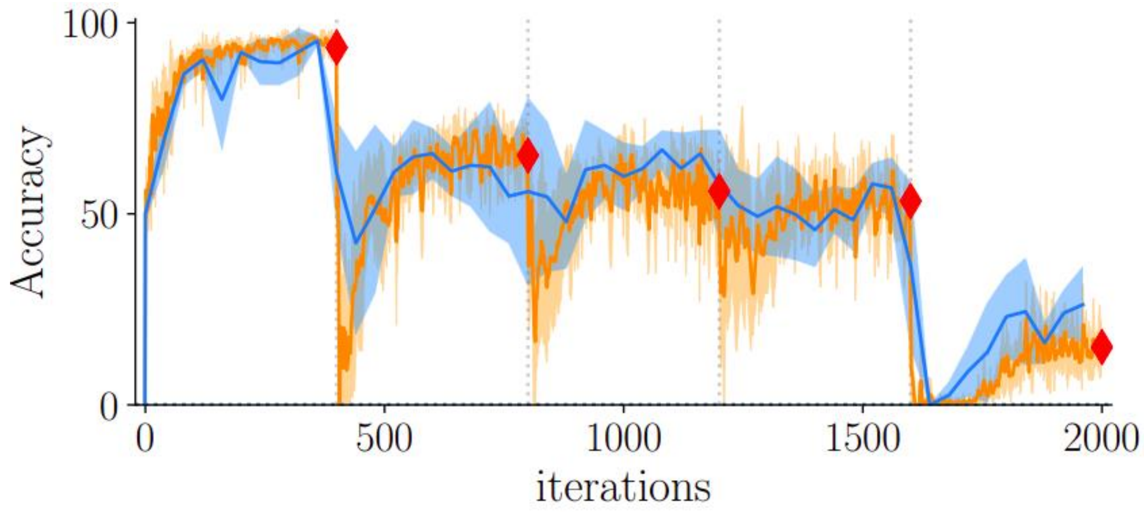
Classifier

Linear Classifier
SCR (NCM at Inference Time)

Online CL – Desiderata

- **Knowledge Accumulation**: the model should improve over time
  - At any point in time
  - High average accuracy but also fast adaptation
- **Continual Stability**: the model should not forget previous knowledge
  - At any point in time
  - We often assume virtual drifts when measuring stability
- **Representation Quality**: the latent representations should improve over time
  - A weaker form of knowledge accumulation/stability
  - Can be evaluated on out-of-distribution data or self-supervised models

Red diamonds = task boundaries

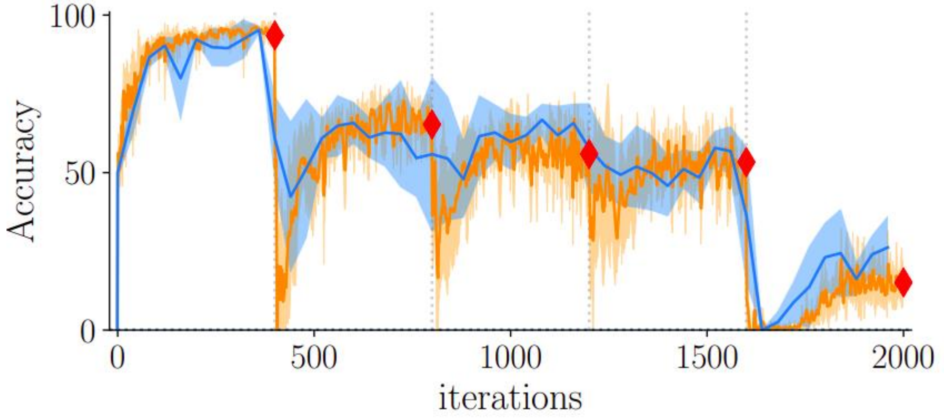


A. Soutif et al. "A Comprehensive Empirical Evaluation on Online Continual Learning" 2023

Knowledge Accumulation

Red diamonds = task boundaries

- **Average Anytime Accuracy**: accuracy along the entire curve.
- Do not confuse with
  - Avg accuracy at the end of training (final diamond)
  - Avg at task boundaries (avg of diamonds)



Notation:

- f_i model at time i
- E_i experience i
- $A(E_i, f_i)$ accuracy of model f_i for experience E_i

$$\mathrm{AAA}_t = \boxed{\frac{1}{t}} \sum_{j=1}^{t} \boxed{\frac{1}{k} \sum_{i=1}^{k} \mathbf{A}(E_i, f_j)}$$

Average along the training curve

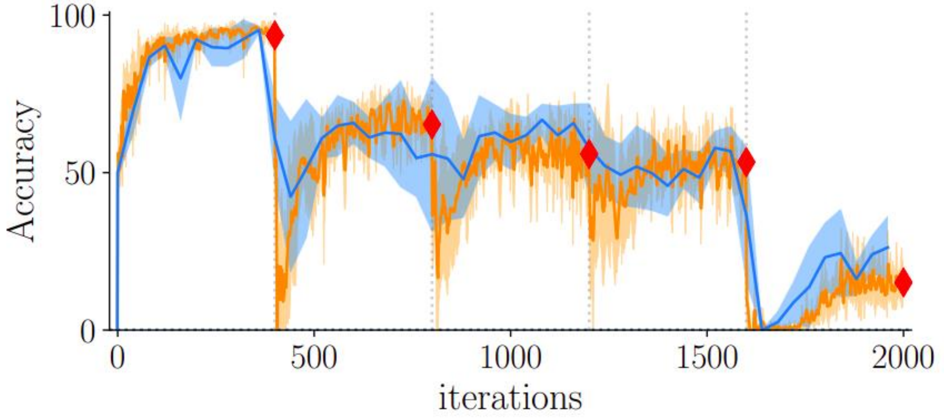
Average accuracy of data seen up to now

Cumulative Accuracy and Forgetting

- In class-incremental settings, the drop in accuracy comes from
    1. Forgetting
    2. Harder task because we have more classes
- **Cumulative Accuracy** isolates (1) by using only the logits of units seen up to training on the evaluation data (mask newer units).

$$b_k^t = \frac{1}{|E_\Sigma^k|} \sum_{x,y \in E_\Sigma^k} 1_y(\arg\max_{c \in C_\Sigma^k} f^t(x)_c)$$

Red diamonds = task boundaries



$$\mathrm{AAA}_t = \boxed{\frac{1}{t}} \sum_{j=1}^{t} \boxed{\frac{1}{k} \sum_{i=1}^{k} \mathbf{A}(E_i, f_j)}$$

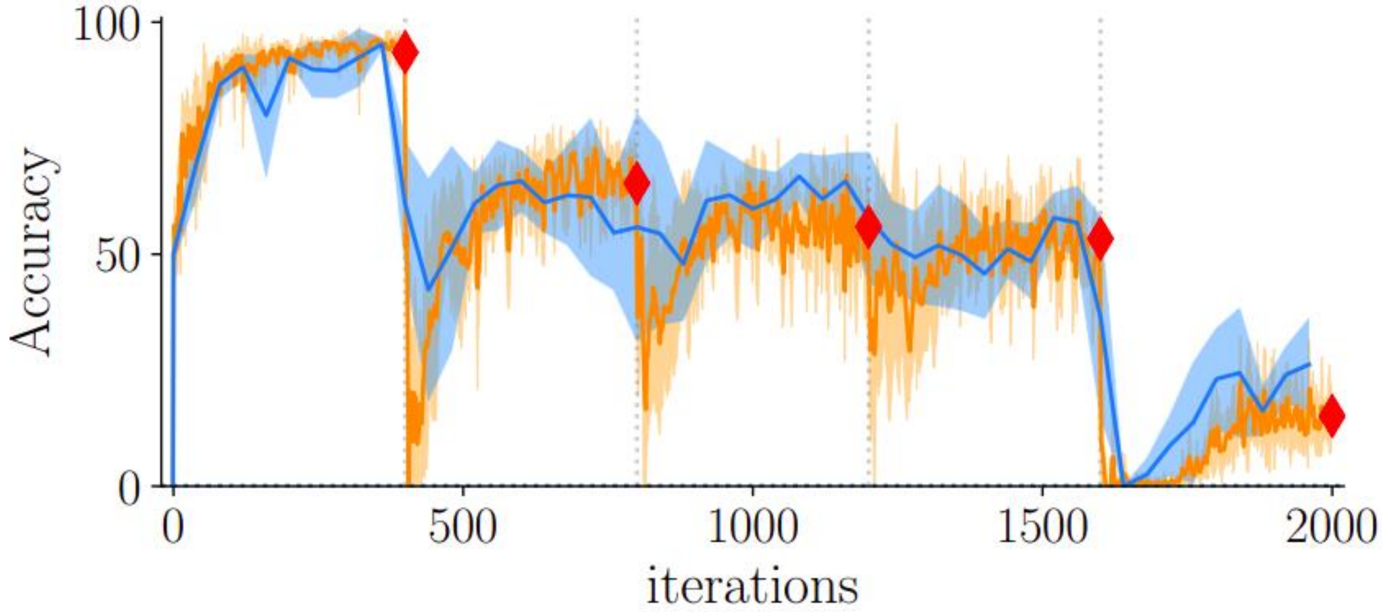
Average along the training curve

Average accuracy of data seen up to now

A. Soutif–Cormerais et al. *"On the Importance of Cross-Task Features for Class-Incremental Learning,"*

Continual Stability

- Observe the behavior of the accuracy during training (curve from one diamond to the next)
- CL methods forget and re-learn old experiences during training
- This phenomenon is masked with the typical metrics measured only at boundaries (red diamonds)



[1] Mathias Delange et. al, Continual Evaluation for Lifelong Learning: Identifying the stability gap, ICLR 2023
[2] Lucas Caccia et. al, New Insights on Reducing Abrupt Representation Change in Online Continual Learning, ICLR 2022

Stability Metrics

Worst-Case ACC: trade-off between the accuracy on iteration t of current task T_k and the worst-case metric min-ACCT_k for previous tasks

$$\textbf{min-ACC}_{T_k} = \frac{1}{k-1}\sum_{i}^{k-1} \min_{n} \mathbf{A}(E_i, f_n), \ \forall |T_{i-1}| < n \leq t$$

$$\textbf{WC-ACC}_t = \frac{1}{k}\mathbf{A}(E_k, f_t) + (1 - \frac{1}{k})\ \textbf{min-ACC}_{T_k}$$

$$\textbf{WC-ACC}_{|T_k|} \leq \textbf{ACC}_{T_k}$$

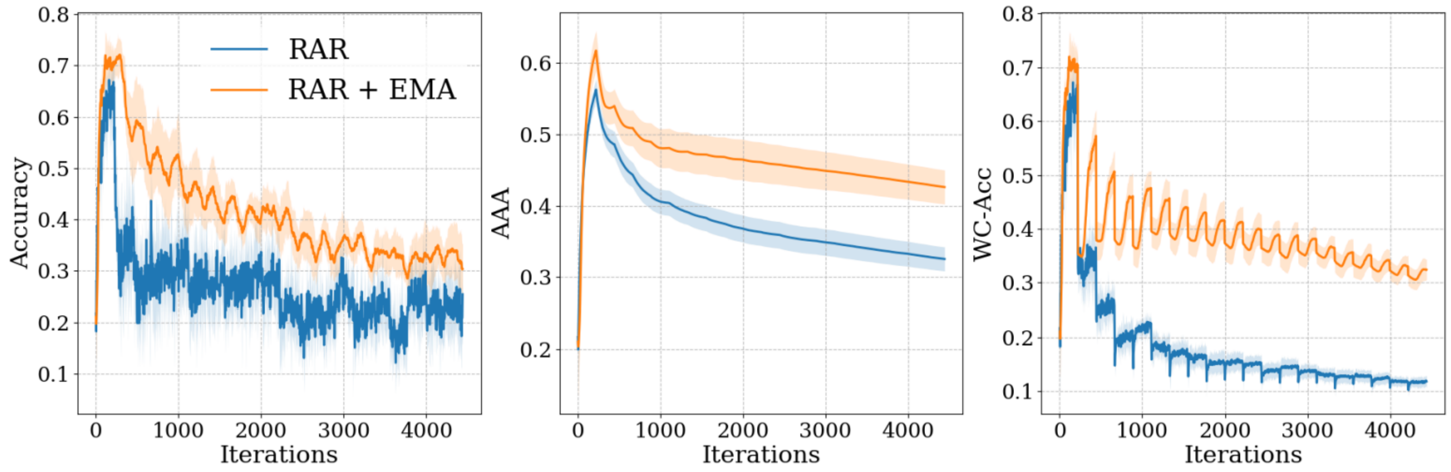


Figure 3: Split-Cifar100, validation accuracy on task 1 data (Left), Average Anytime Accuracy AAA_t (Middle) and WC-ACC (Right), for RAR and its EMA augmented version, using 2000 memory. Mean and standard deviation are computed over 6 runs.

Mathias Delange et. al, Continual Evaluation for Lifelong Learning: Identifying the stability gap, ICLR 2023
Figure from A. Soutif et al. «Improving Online Continual Learning Performance and Stability with Temporal Ensembles" CoLLAs '23

Knowledge Accumulation and Linear Probing

- Forgetting may result from a misaligned classifier
- Easy to fix (e.g. finetune only the linear classifier on replay buffer) if the representations are good
- **Linear probing** measures the quality of the representation
  - Train linear classifier with the current feature extractor using replay data
  - Evaluate the accuracy of the classifier
- useful for continual self-supervised models and continual pretraining

OCL Results

Method	Split-Cifar100 (20 Tasks)				Split-TinyImagenet (20 Tasks)			
	Acc ↑	AAA^{val} ↑	WC-Acc^{val} ↑	Probed Acc ↑	Acc ↑	AAA^{val} ↑	WC-Acc^{val} ↑	Probed Acc ↑
$i.i.d$	35.3 ± 1.5	-	-	45.8 ± 0.6	26.5 ± 0.6	-	-	34.3 ± 0.5
GDumb	18.5 ± 0.5	-	-	-	13.1 ± 0.4	-	-	-
AGEM	3.1 ± 0.2	10.4 ± 0.6	2.9 ± 0.3	18.7 ± 0.8	2.6 ± 0.2	7.3 ± 0.5	2.6 ± 0.2	23.3 ± 0.6
ER	28.2 ± 1.2	36.6 ± 2.0	12.5 ± 0.6	**44.9** ± 0.9	21.2 ± 0.6	33.9 ± 1.7	15.2 ± 0.5	**35.6** ± 0.6
ER + LwF	**30.4** ± 0.8	39.2 ± 2.0	15.3 ± 0.9	44.4 ± 0.8	22.7 ± 1.1	34.4 ± 2.4	**17.0** ± 0.7	33.8 ± 0.9
MIR	29.4 ± 1.9	33.1 ± 3.2	11.6 ± 1.6	43.4 ± 0.7	21.3 ± 0.8	31.0 ± 1.8	15.2 ± 0.5	33.0 ± 0.4
ER-ACE	29.9 ± 0.6	38.5 ± 1.8	14.9 ± 0.9	42.4 ± 0.6	**23.6** ± 0.7	**35.0** ± 1.5	16.8 ± 0.7	34.2 ± 0.3
DER++	29.3 ± 0.9	37.5 ± 2.5	13.4 ± 0.7	44.0 ± 0.8	22.9 ± 0.5	34.2 ± 4.0	16.3 ± 0.3	31.5 ± 0.9
RAR	28.2 ± 1.4	38.2 ± 1.6	14.9 ± 0.7	42.3 ± 0.9	15.7 ± 0.9	27.8 ± 2.8	10.1 ± 0.9	29.8 ± 0.9
SCR	28.3 ± 0.8	**42.1** ± 2.1	**20.3** ± 0.4	37.0 ± 0.3	16.9 ± 0.4	30.7 ± 1.5	12.3 ± 0.5	22.5 ± 0.4

Table 2: Last step results on Split-Cifar100 (20 Tasks) with 2000 memory (Left) and for Split-TinyImagenet (20 Tasks) with 4000 memory (Right). For each metric, we report the average and standard deviation over 5 seeds

A. Soutif et al. "A Comprehensive Empirical Evaluation on Online Continual Learning" 2023

OCL Results

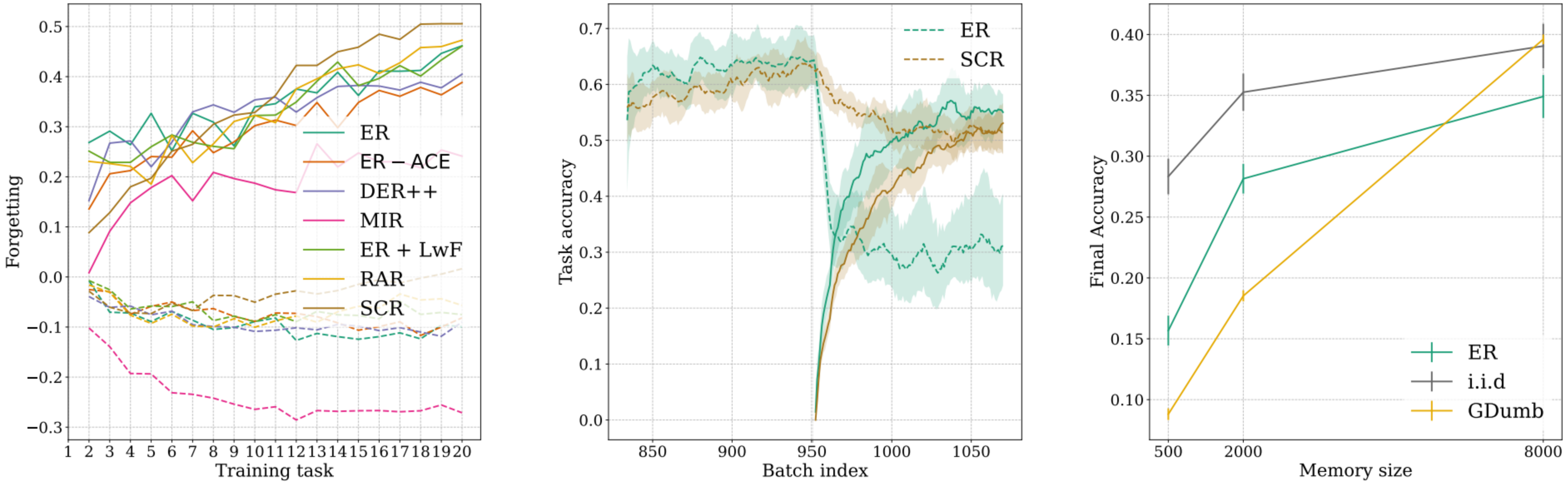


Figure 3: **Left:** Forgetting (full lines), and Cumulative Forgetting (dotted lines) on Split-Cifar100 with 2000 memory; **Middle:** Illustration of the difference in stability between ER and SCR on Split-Cifar100 (20 tasks), using 2000 memory. We place ourselves at the task shift between task 4 and 5 and display the accuracy on previous task data (dotted lines) as well as the accuracy on current task data (full lines).; **Right:** Final performance of ER, i.i.d. reference method, and GDumb baseline for 3 different memory sizes on Split-Cifar100

A. Soutif et al. "A Comprehensive Empirical Evaluation on Online Continual Learning" 2023

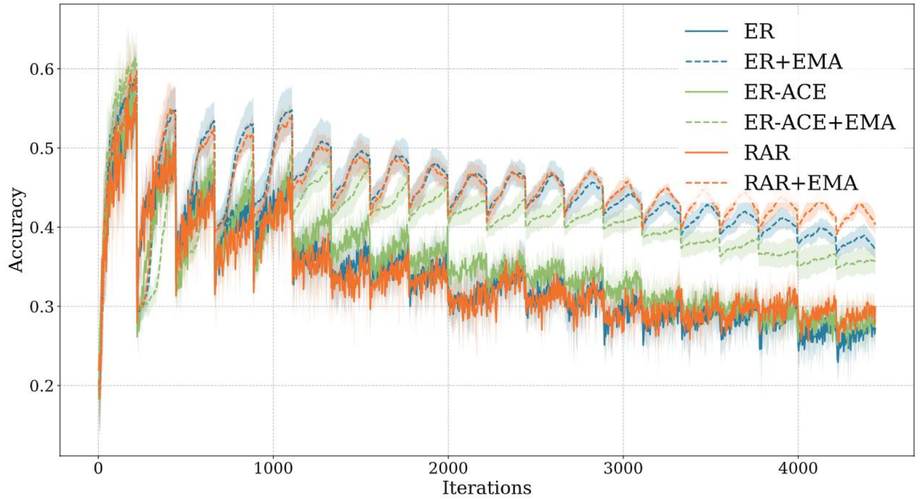
Continual Stability - Temporal Ensembles

Exponential Moving Average of the weights (EMA) mitigates the stability gap

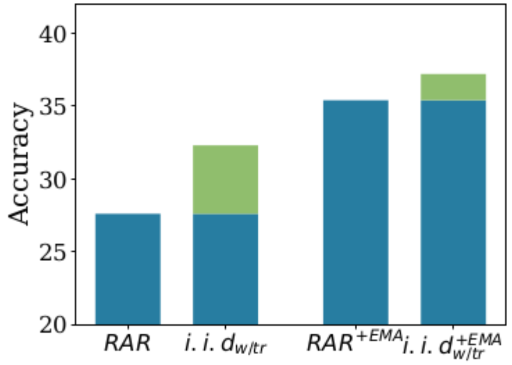
- Separate training and evaluation model
- Fixes stability only at evaluation time. Training model is still unstable
- Cheap and online method

Open question: how to fix stability gap during training.

$$\theta^t_{ema} = \lambda\theta^{t-1}_{ema} + (1-\lambda)\theta^t,$$



Check the poster for more details!



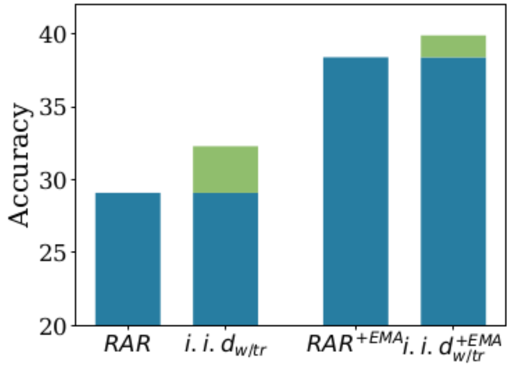


Figure 5: Comparison of previous state-of-the-art method in online continual learning RAR against the reference method $i.i.d_{w/tr}$ on Split-Cifar100 (Top) using 2000 memory and Split-Minimnet using 10000 memory (Bottom). The performance gap is indicated in green, and is greatly reduced by the use of EMA.

A. Soutif et al. «Improving Online Continual Learning Performance and Stability with Temporal Ensembles" CoLLAs '23

Continual Hyperparameter Selection

- Most researchers perform a full hyperparameter selection on the entire validation stream.
  - It's not a CL method and it's suboptimal because optimal parameters may vary over time
- Some methods are quite sensitive to hyperparameters (e.g. EWC)

Existing methods:

- [1] finds optimal stability-plasticity tradeoff at each step. Assumes that a single hyperparameter controls the tradeoff monotonically (e.g. regularization strength)
- [2] uses reinforcement learning to find optimal parameters. Online RL (bandit)
- [3] uses only the first part of the validation stream

Algorithm 1. Continual Hyperparameter Selection Framework

input $\mathcal{H}$ hyperparameter set, $\alpha \in [0,1]$ decaying factor, $p \in [0,1]$ accuracy drop margin, D^{t+1} new task data, Ψ coarse learning rate grid
require θ^t previous task model parameters
require CLM continual learning method

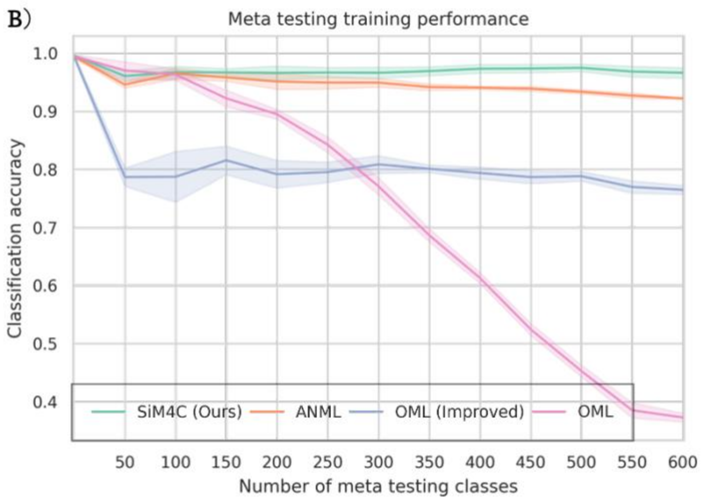
`//Maximal Plasticity Search`
$A^* = 0$
for $\eta \in \Psi$ **do**
  $A \leftarrow \text{Finetune}(D^{t+1}, \eta; \theta^t)$ ▷ Finetuning accuracy
  if $A > A^*$ **then**
    $A^*, \eta^* \leftarrow A, \eta$ ▷ Update best values
`//Stability Decay`
do
  $A \leftarrow CLM(D^{t+1}, \eta^*; \theta^t)$
  if $A < (1-p)A^*$ **then**
    $\mathcal{H} \leftarrow \alpha \cdot \mathcal{H}$ ▷ Hyperparameter decay
while $A < (1-p)A^*$

[1] M. De Lange et al. "A Continual Learning Survey: Defying Forgetting in Classification Tasks." TPAMI 2022
[2] Y. Liu et al. "Online Hyperparameter Optimization for Class-Incremental Learning." AAAI '23
[3] A. Chaudhry et al. "Efficient Lifelong Learning with A-GEM." 2019

Robust CL Methods

- **Alternative to Continual Hyperparameter Selection**: design robust models!
- **Example: SiM4C**
  - Use a single inner update step
  - Use exact gradient instead of first-order approximation
- **Results**:
  - Higher accuracy
  - No need for additional hyperparameter selection
  - Easy to plug into existing methods
  - Works in continual-meta and meta-continual learning

Omniglot



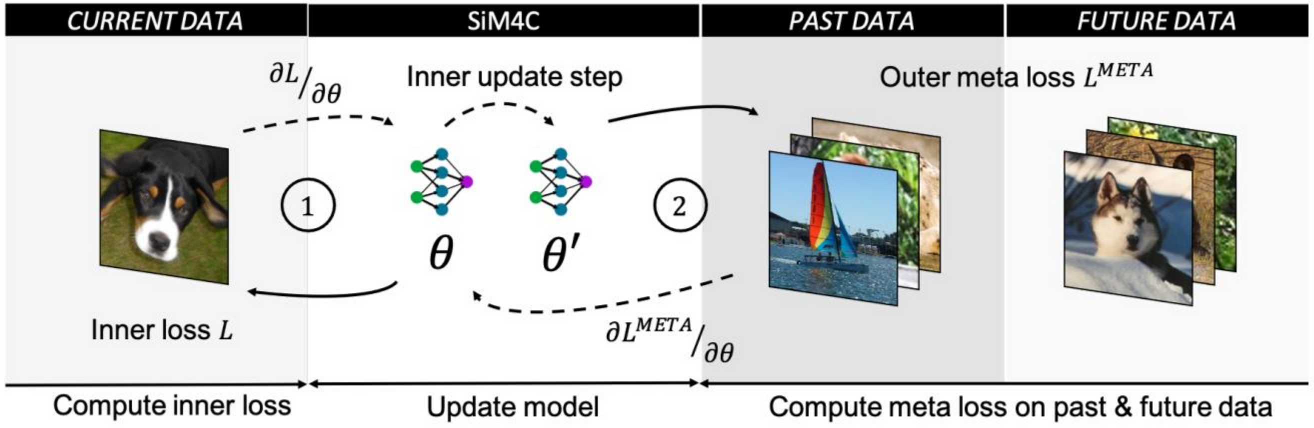


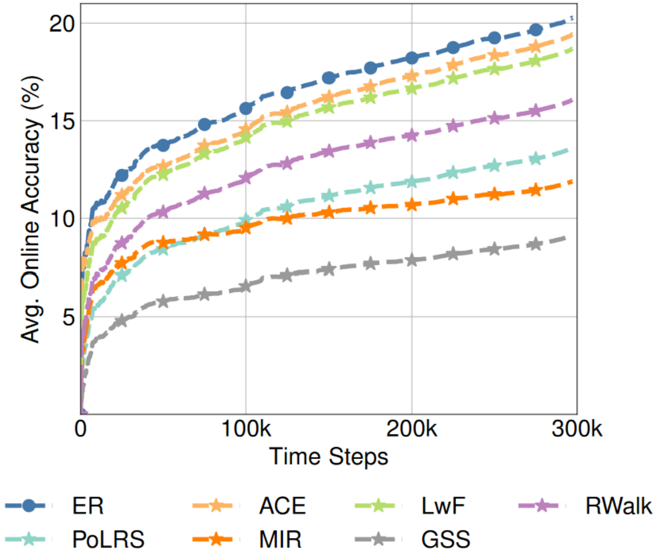
Figure 1. Schematic depiction of SiM4C, after a single inner optimization step the proposed meta-objective optimizes for forward and backward transfer by utilizing seen *past data* from previous tasks and unseen *future data* of the current task.

E. Cetin et al. «A Simple Recipe to Meta-Learn Forward and Backward Transfer ", ICCV '23

Real-Time / Infinite Memory / Finite Compute

- **Memory is cheap, compute is expensive**
  - CL methods are designed for finite memory usage. Often unrealistic
  - The "privacy argument" is not very strong, because trained models can leak data
- **Alternative: real-time, infinite memory, bounded computational cost**
  - Real-time constraints. Methods need to skip data if they are not fast enough
- **Results: Experience Replay outperforms CL methods**

CL Strategy	Method($\mathcal{A}$)	$\mathcal{C}_S(\mathcal{A})$	Delay
Experience Replay	ER [11]	1	0
Regularizations	ACE [6]	1	0
	LwF [28]	4/3	1/3
	RWalk [8]	2	1
LR Scheduler	PoLRS [7]	3	2
Sampling Strategies	MIR [3]	5/2	3/2
	GSS [4]	6*	5



Y. Ghunaim et al. "Real-Time Evaluation in Online Continual Learning: A New Hope." CVPR '23
A. Prabhu et al. "Computationally Budgeted Continual Learning: What Does Matter?" CVPR '23

Conclusion

Unsolved CL questions:

- Continual stability
- Robustness to stream parameters
- Continual hyperparameter selection (and robustness)
- Compute-bounded continual learning

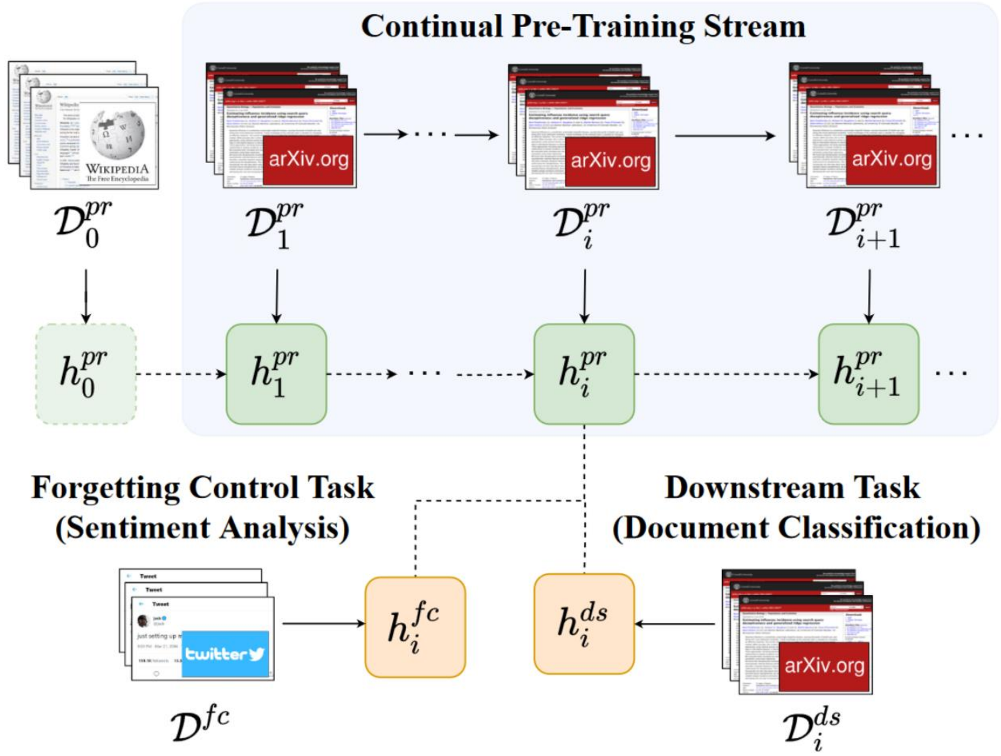
Beyond Single CL Agents

Continual pretraining

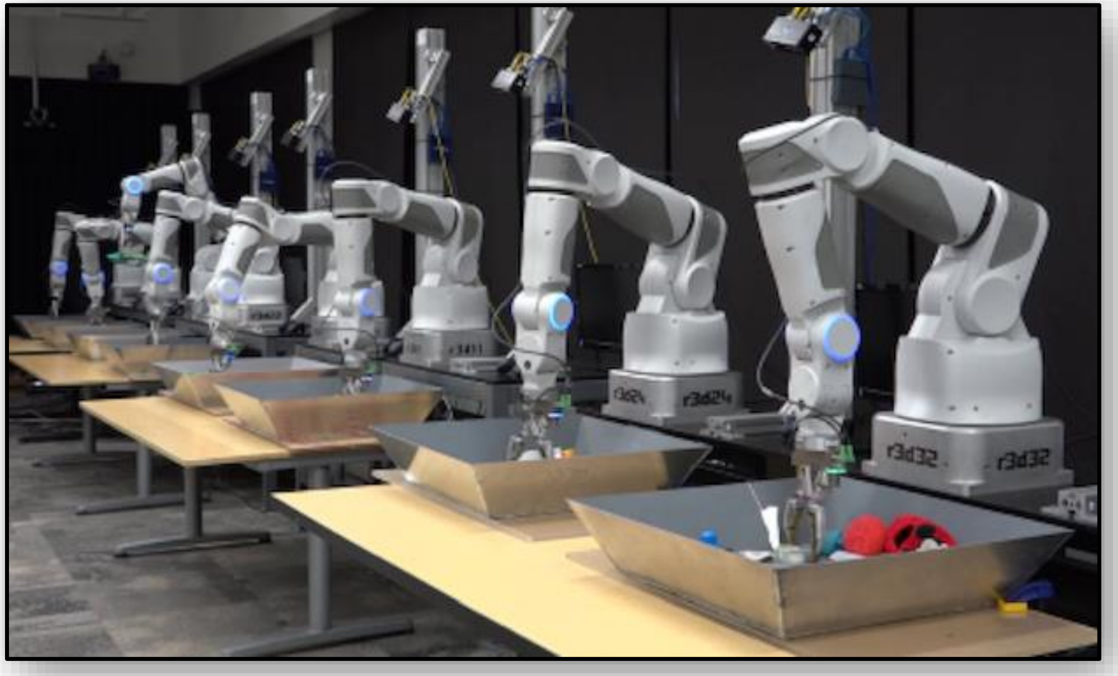
Distributed continual agents

Two Perspectives

Continual Pretraining of Large Models

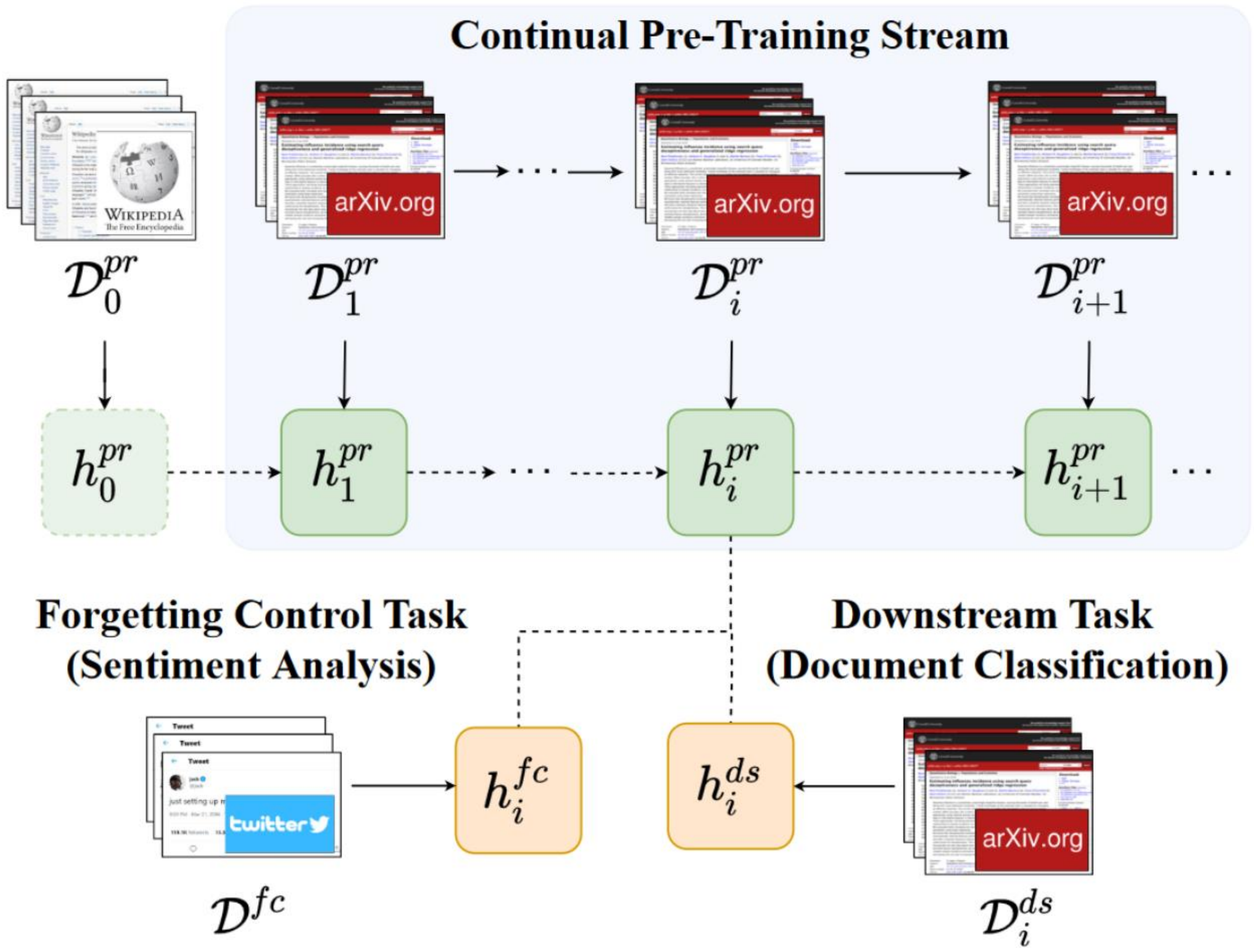


Asynchronous and Independent Continual Learning Agents



Continual Pretraining

- **Continual Pretraining** is the problem of efficiently updating a large pretrained model
- **Forgetting Control Task**: we don't want to forget general knowledge
- **Downstream Task**: we want to improve on domain-specific tasks



Cossu, A., et al. "Continual Pre-Training Mitigates Forgetting in Language and Vision." 2022.

Pretraining Results

Evaluation on the Forgetting Control Task

Table 2: Accuracy on the entire dataset of `sentiment analysis` with RoBERTa model. Continual pre-training has been performed sequentially over each experience of `scientific abstracts`. Base refers to the model pre-trained on Wikipedia, while NT refers to the model with vocabulary expansion.

RoBERTa			**Accuracy**					**1-epoch Accuracy**		
Base			93.40					92.40		
Exp.	e1	e2	e3	e4	e5	e1	e2	e3	e4	e5
Pretr	93.40	93.15	93.35	93.20	92.90	92.40	91.80	92.30	91.85	92.20
Pretr. NT	93.75	93.70	93.75	93.60	94.10	91.75	91.15	92.00	92.30	92.45

fast adaptation

Forgetting is limited even with finetuning. Dynamic vocabulary expansion (NT) slightly improves the performance.

Self-supervised pretraining is more robust than supervised methods (result for vision in the paper)

Cossu, A., et al. "Continual Pre-Training Mitigates Forgetting in Language and Vision." 2022.

Self-Supervised CL

- Distillation loss maps old representations in a new projected space
- SSL tricks such as heavy augmentations and SSL losses
- Linear probing evaluation

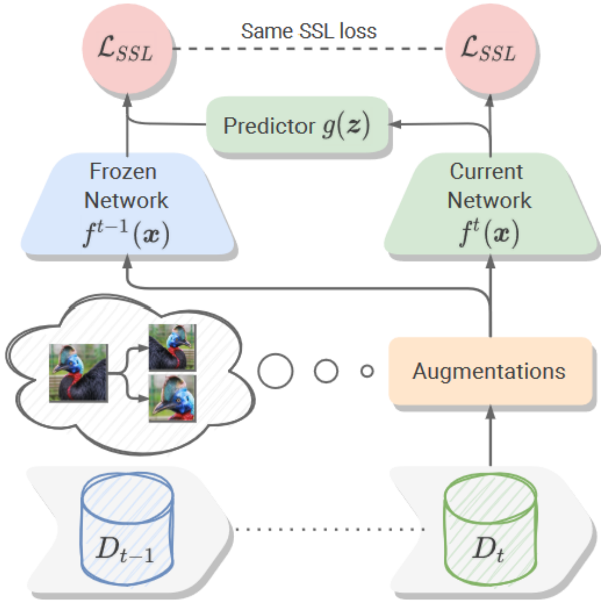


Figure 2. Overview of the CaSSLe framework.

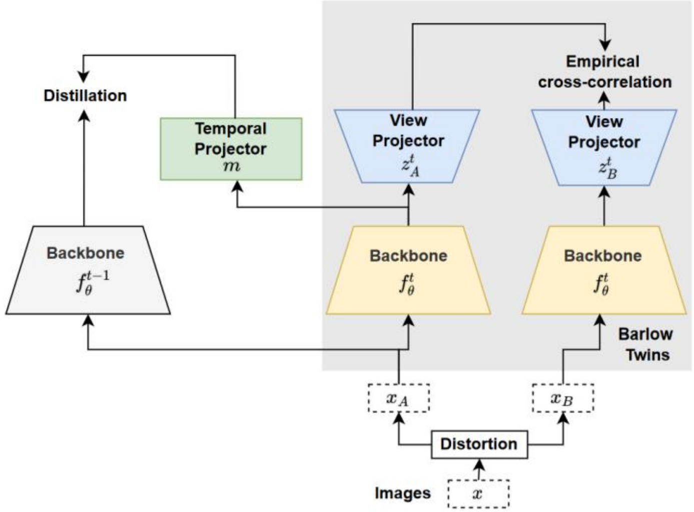


Figure 1. Self-supervised continual learning with Projected Functional Regularization. Instead of performing feature distillation directly between the previous task backbone and the new one, we use a *learned temporal projection* between the two feature spaces.

A. Gomez-Villa et al. *"Continually Learning Self-Supervised Representations With Projected Functional Regularization," CLVISION '22*
E. Fini et al. *"Self-Supervised Models Are Continual Learners." CVPR '22*

Continual Federated Learning

- **FL** methods fail in simple heterogeneous settings.
- **Local forgetting** happens in heterogeneous FL if the local models are not aggregated often enough, resulting in a local drift and forgetting of the global knowledge.

Open question: can continual learning improve federated learning in heterogeneous settings?

- **[1]** proposes **WSM loss**, a weighted cross-entropy to mitigate this problem

Not my paper, But this is also a CoLLas paper

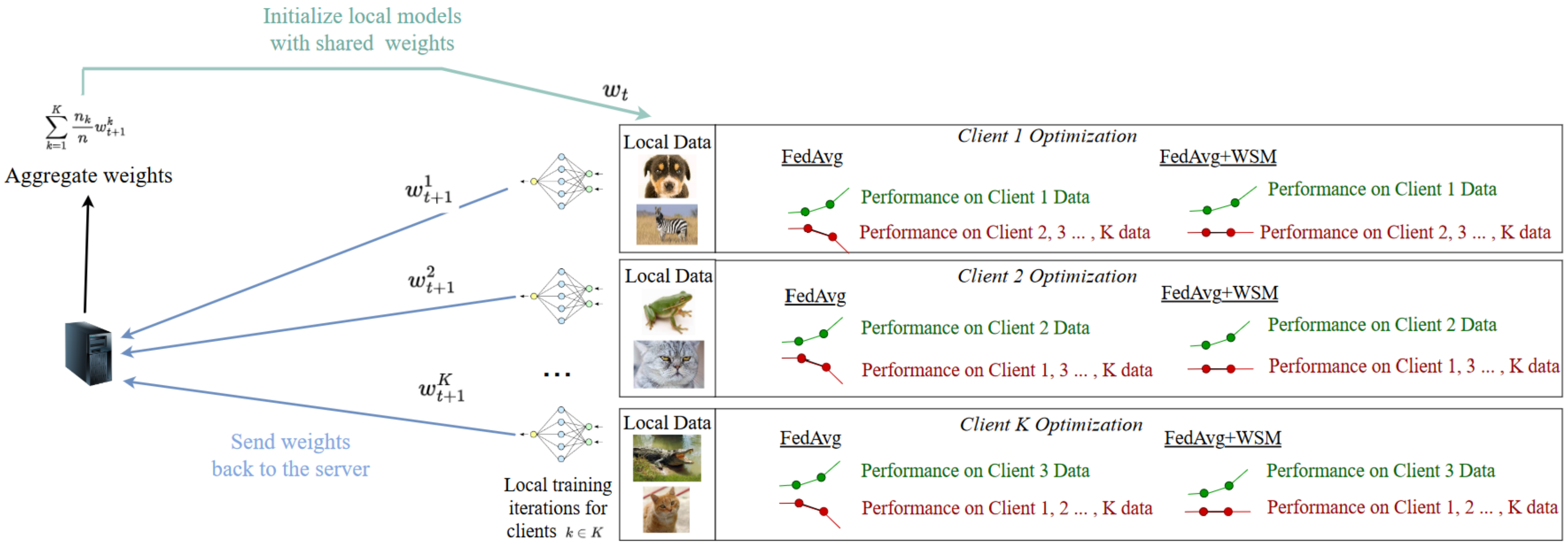


Figure 1: *Illustration of catastrophic forgetting within client rounds.* A global model with knowledge of all classes is sent to all clients participating in a given FL round. Local training increases the client model performance on the client's local distribution but tends to simultaneously decrease performance with respect other clients distributions which leads to poor aggregation and overall model performance.

[1] G. Legate et al. "Re-Weighted Softmax Cross-Entropy to Control Forgetting in Federated Learning." CoLLAs '23

FedWelt

- **Objectives**:
  - Minimize communication
  - Exploit task similarity
  - Avoid task interference
- **Modularized task-based model**:
  - Global parameters
  - Local base parameters
  - Task-adaptive parameters

Local client model:

$$\boldsymbol{\theta}_c^{(t)} = \mathbf{B}_c^{(t)} \odot \mathbf{m}_c^{(t)} + \mathbf{A}_c^{(t)} + \sum_{i \in \mathcal{C}_{\backslash c}} \sum_{j<|t|} \alpha_{i,j}^{(t)} \mathbf{A}_i^{(j)}$$

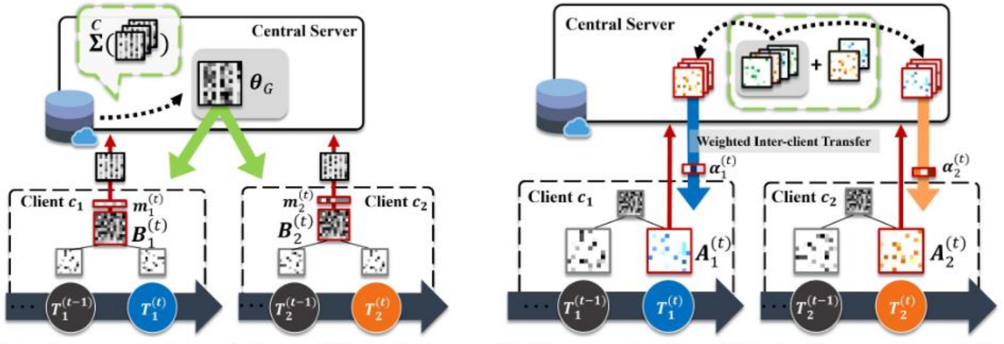


Figure 3. Updates of *FedWeIT*. **(a)** A client sends sparsified federated parameter $\boldsymbol{B}_c \odot \boldsymbol{m}_c^{(t)}$. After that, the server redistributes aggregated parameters to the clients. **(b)** The knowledge base stores previous tasks-adaptive parameters of clients, and each client selectively utilizes them with an attention mask.

J. Yoon et al. *"Federated Continual Learning with Weighted Inter-Client Transfer." ICML '21*

Ex-Model CL

- **Model aggregation is the critical missing component in heterogeneous FL!**
  - We know how to train the local model (continual learning)
  - We know how to aggregate homogeneous models as long as the aggregation is frequent enough (homogeneous federated learning)
- **If we can aggregate independent models (Ex-Model CL)**
  - we can train on multiple tasks in parallel
  - Without frequent synchnonous aggregations
  - Allows decentralized training
  - related to model patching [1]

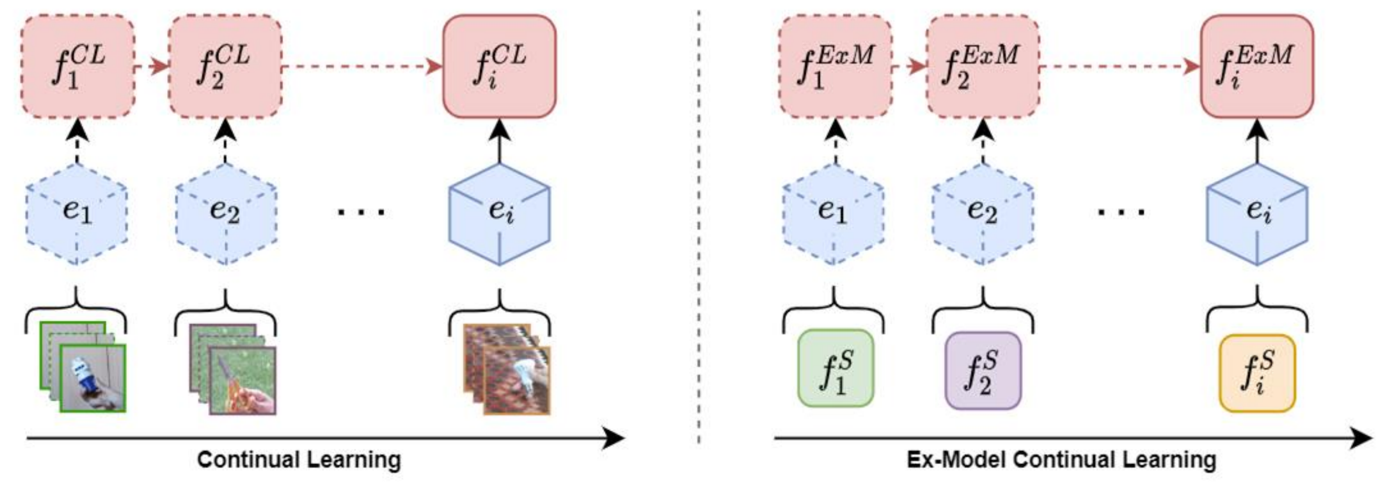


Figure 1. Ex-model Continual Learning. The stream is composed of expert models, without access to the original data.

A. Carta. "Ex-Model: Continual Learning From a Stream of Trained Models," CLVISION '22

[1] Raffel, Colin. "Building Machine Learning Models Like Open Source Software." Communications of the ACM 2023

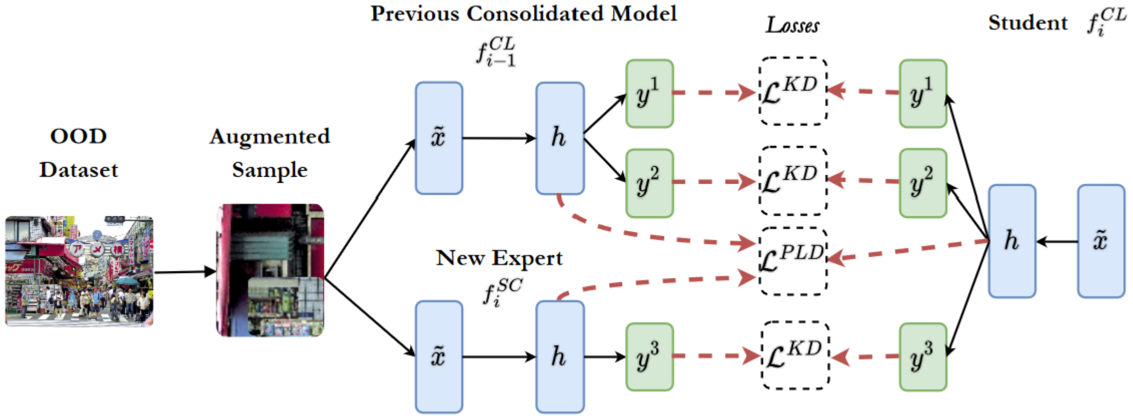
Data-Agnostic Consolidation (DAC)

Split learning into:

- *Adaptation*: learn new task
- *Consolidation*: aggregate models

Model consolidation with data-free knowledge distillation (DAC)

- Double Knowledge Distillation
  - Teachers: Previous CL model and New model
  - On the output
  - On the latent activations (Projected)
- Task-incremental method
- Surprisingly, **indipendent adaptation + sequential consolidation seems better than sequential adaptation** (i.e. what most CL methods are doing)



(a) Task-incremental SplitCIFAR100 after task 5 and 10. Baselines denoted by † are taken from (Masana et al., 2022)

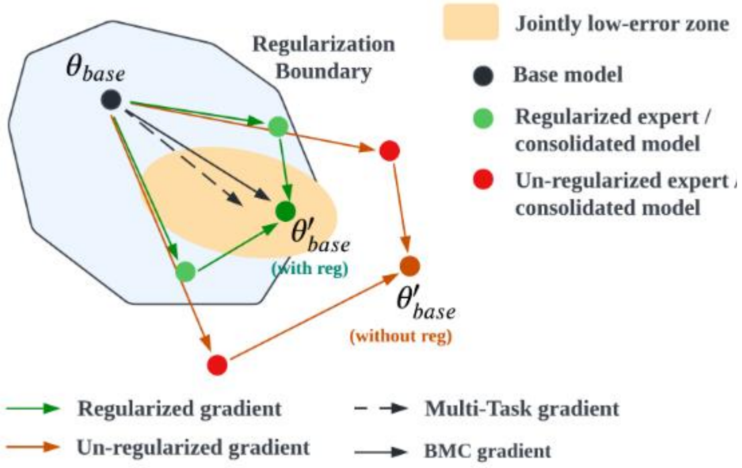
		SplitCIFAR100	
	DCL	5 Tasks	10 Tasks
Naive†	RF	49.8	38.3
EWC†	RF	60.2	56.7
PathInt†	RF	57.3	53.1
MAS†	RF	61.8	58.6
RWalk†	RF	56.3	49.3
LwM†	RF	76.2	70.4
LwF†	RF	76.7	76.6
DMC†	✓	72.3	66.7
DAC($\lambda = 0$)	✓	77.6±1.7	77.5±0.6
DAC	✓	81.4±1.6	80.5±0.8

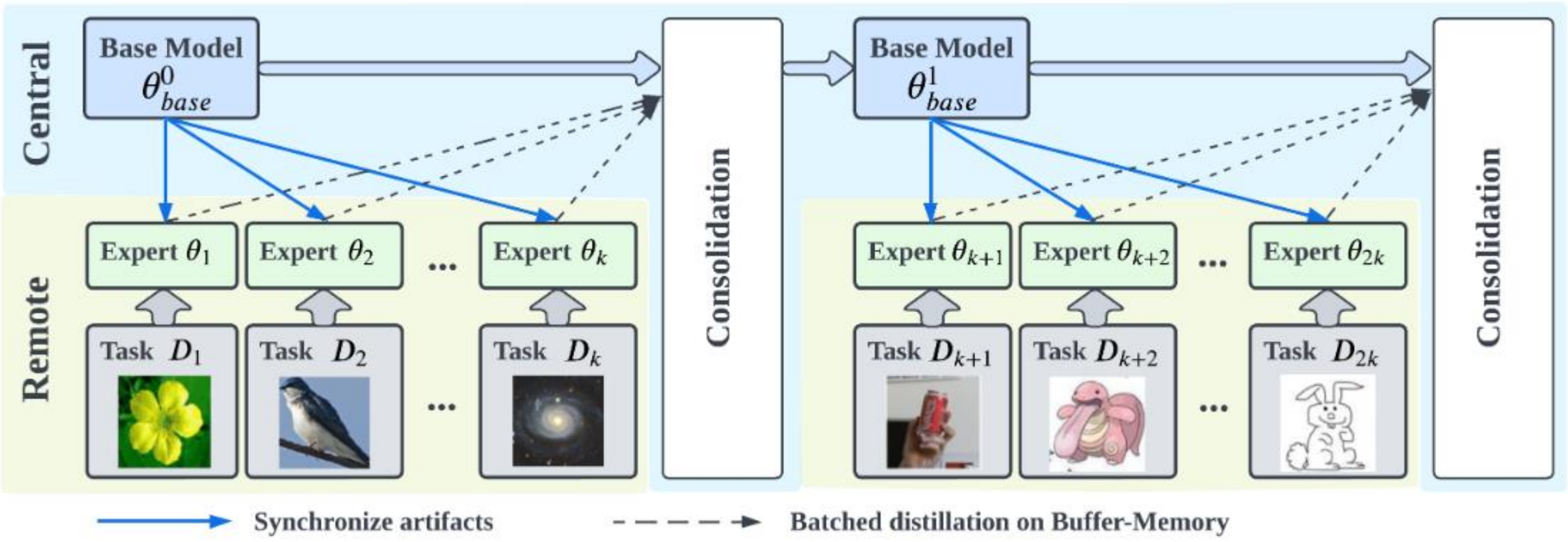
A. Carta et al. "Projected Latent Distillation for Data-Agnostic Consolidation in Distributed Continual Learning." arXiv preprint, 2023

Batch Model Consolidation

BMC:
- Regularized adaptation with distillation on the latent activations (teacher: base model)
- Replay data for batch consolidation

Sparse consolidation allows asynchronous learning in independent agents with light synchronization





I. Fostiropoulos et al. “Batch Model Consolidation: A Multi-Task Model Consolidation Framework.” CVPR ‘23

Conclusion

Main Message

The goal of Continual Learning is to understand how to design machine learning models that learn over time

- on a constrained budget (memory/compute/real-time requirements)
- with non-stationary data

- The goal is much wider than «class-incremental learning» or «finetuning a pretrained model»
- We need to push towards more realistic settings
  - Toy data is fine for research, toy settings not so much
  - CL metrics can be misleading and very easy to abuse
  - Good solutions already exist!

Open Challenges

- **CL Benchmarks**:
  - Real drifts and prequential evaluation
  - Exploitation of temporal coherence
  - Real-time training with infinite memory
- **CL Robustness to**
  - stream parameters
  - (continual) hyperparameter selection
  - stability gap
- **Beyond Single Agents**
  - continual pretraining
  - ex-model / distributed continual learning

CL and Reproducibility - Avalanche

- PyTorch library for continual learning https://avalanche.continualai.org/
  - A community effort with >30 CL methods, >60 contributors
  - Easy to use and extend
- **Reproducible baselines**: https://github.com/ContinualAI/continual-learning-baselines
- **CIR**: https://github.com/HamedHemati/CIR
- **OCL survey**: https://github.com/albinsou/ocl_survey

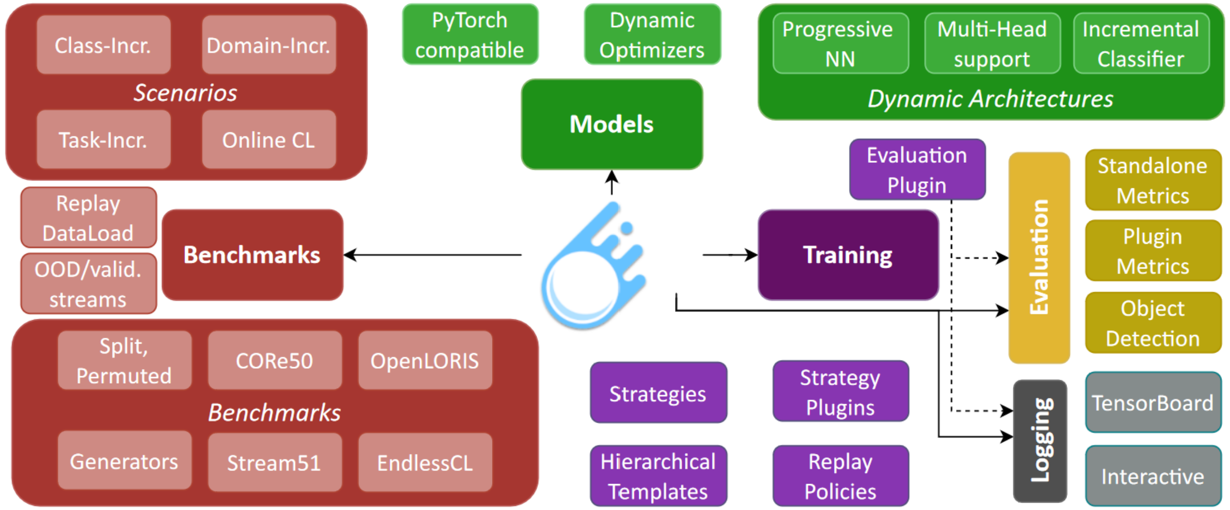


Figure 1: Avalanche main functionalities and modules.

Avalanche

powered by

ContinualAI

A. Carta et al. "Avalanche: A PyTorch Library for Deep Continual Learning." 2023