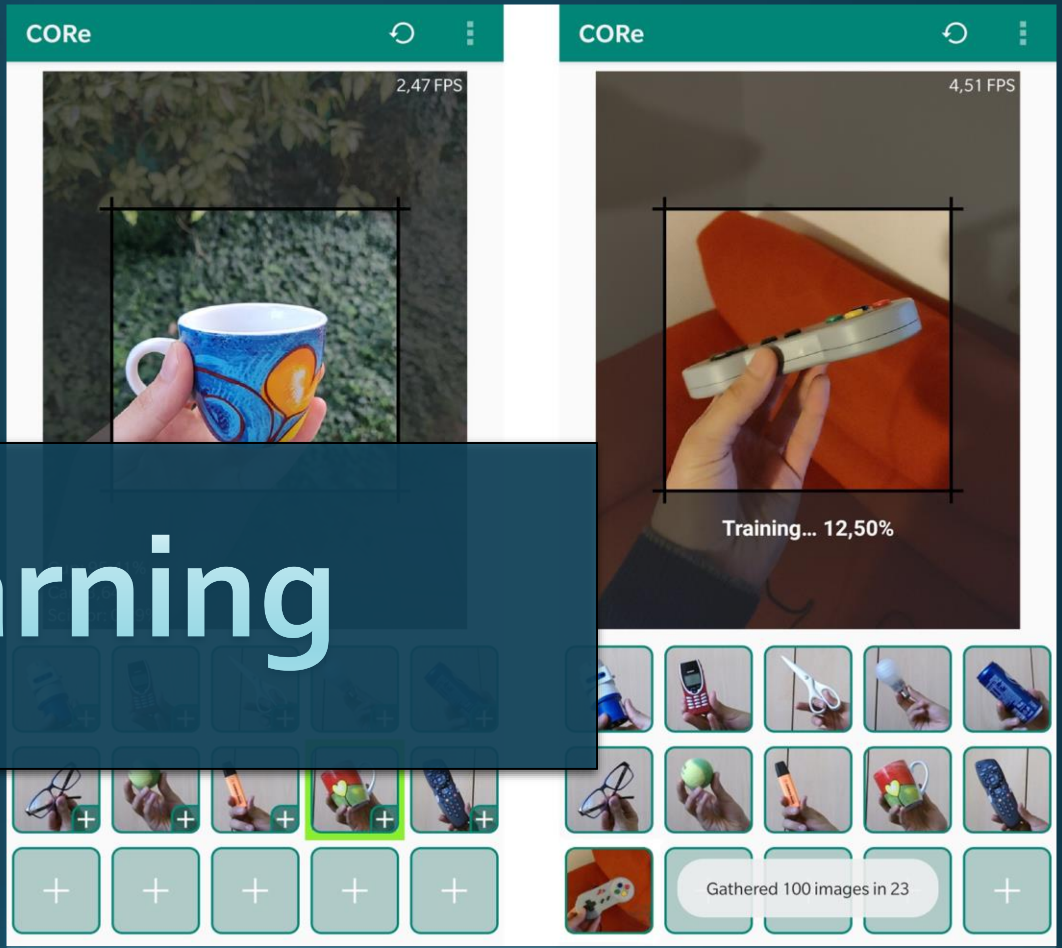


Continual Learning

Davide Maltoni
University of Bologna
davide.maltoni@unibo.it

Machine Intelligence @ BioLab

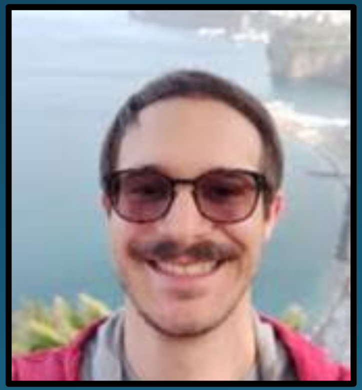
Computer Science Department - University of Bologna

Cesena - Italy

People involved in this research

Davide Maltoni

Vincenzo Lomonaco



Lorenzo Pellegrini

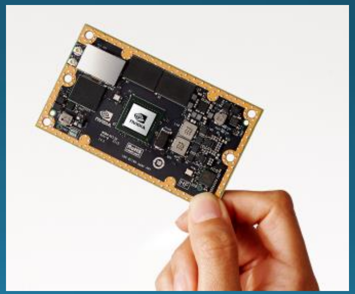
Gabriele Graffieti

Guido Borghi

Training vs Inference

- Today most of the ML models are batch trained on **huge** and **fixed** datasets.

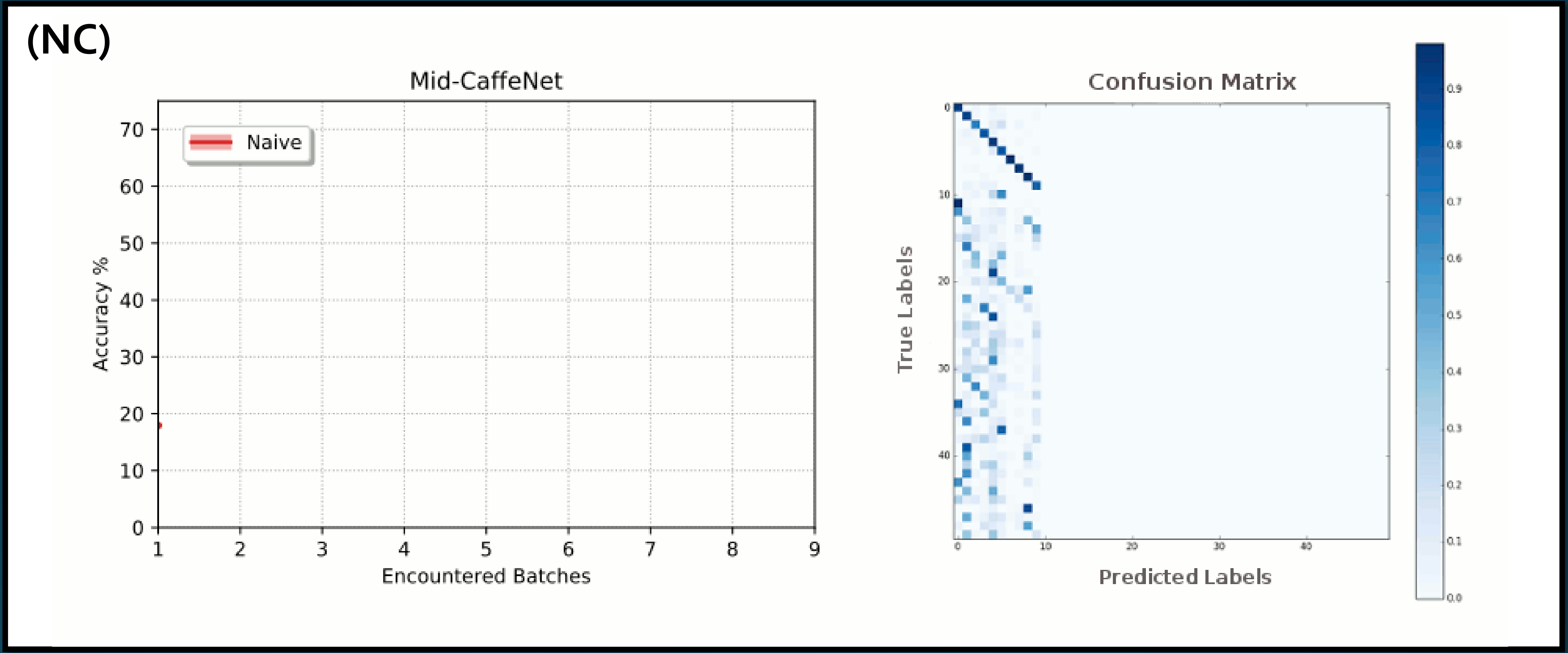
- Strong separation between
  - **Training** (server side with GPUs)
  - **Inference** (also on light architectures)



Why Continual Learning ?

- Many real-world scenarios (e.g. object recognition in robotics) are **incremental** and training data available as subsequent batches (or from streaming sources).
- Storing the whole past data and **retrain the system from scratch** is often **unfeasible**.
- Imperative to **build on top** of previously learned knowledge.
- Toward **natural learning**: lifelong, mostly unsupervised, multimodal
- Problem: **Catastrophic forgetting** (McCloskey & Cohen 1989)

Catastrophic Forgetting



Strategies to mitigate forgetting

- **Architectural:** specific architectures, layers, activation functions, and/or weight-freezing strategies.
- **Regularization:** loss function is extended with terms promoting selective consolidation of the weights which are important to retain past memories.
- **Rehearsal:** past information is periodically replayed to the model, to strengthen connections for memories it has already learned.

Recently some interesting approaches have proposed (e.g. EwC, LwF), but **tested on simple benchmarks**.

D. Maltoni, V. Lomonaco. ***Continuous learning in single-incremental-task scenarios***, Neural Networks, 2019.

Our Aims

Online Continual Learning

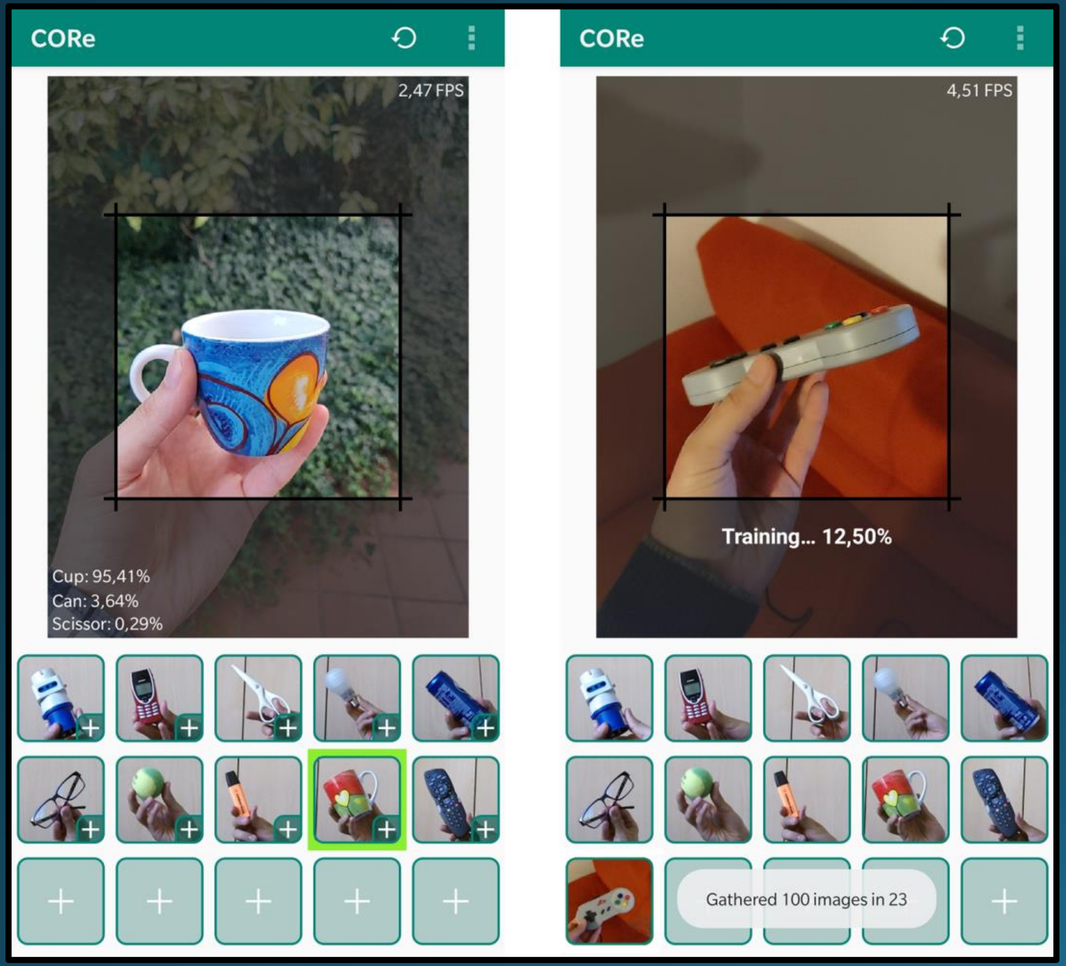
- Limited computation and storage, real-time updates, small non i.i.d. batches
- Training (once deployed) at the edge without network connection
- Privacy friendly (no server-side processing, no storage of row data)

Focus on real computer vision applications

- Robotics*
- Smart cameras (e.g. surveillance)
- Vision apps on mobile devices

*T. Lesort, V. Lomonaco, et al. ***Continual Learning for Robotics***, Information Fusion, 2020.

CORe Android App



- **Demonstrates CL at the edge.**
- **Android smartphone with no hardware acceleration.**
- Near **real-time training of a MobileNet** (less than 1 sec. to update the model after each 20 sec. video).

L. Pellegrini, G. Graffieti, V. Lomonaco and D. Maltoni. ***Latent Replay for Real-Time Continual Learning***, IROS 2020.

This Video: https://www.youtube.com/watch?v=Bs3tSjwbHa4
Download the App: https://github.com/lrzpellegrini/CL-CORe-App

A new Dataset and Benchmark

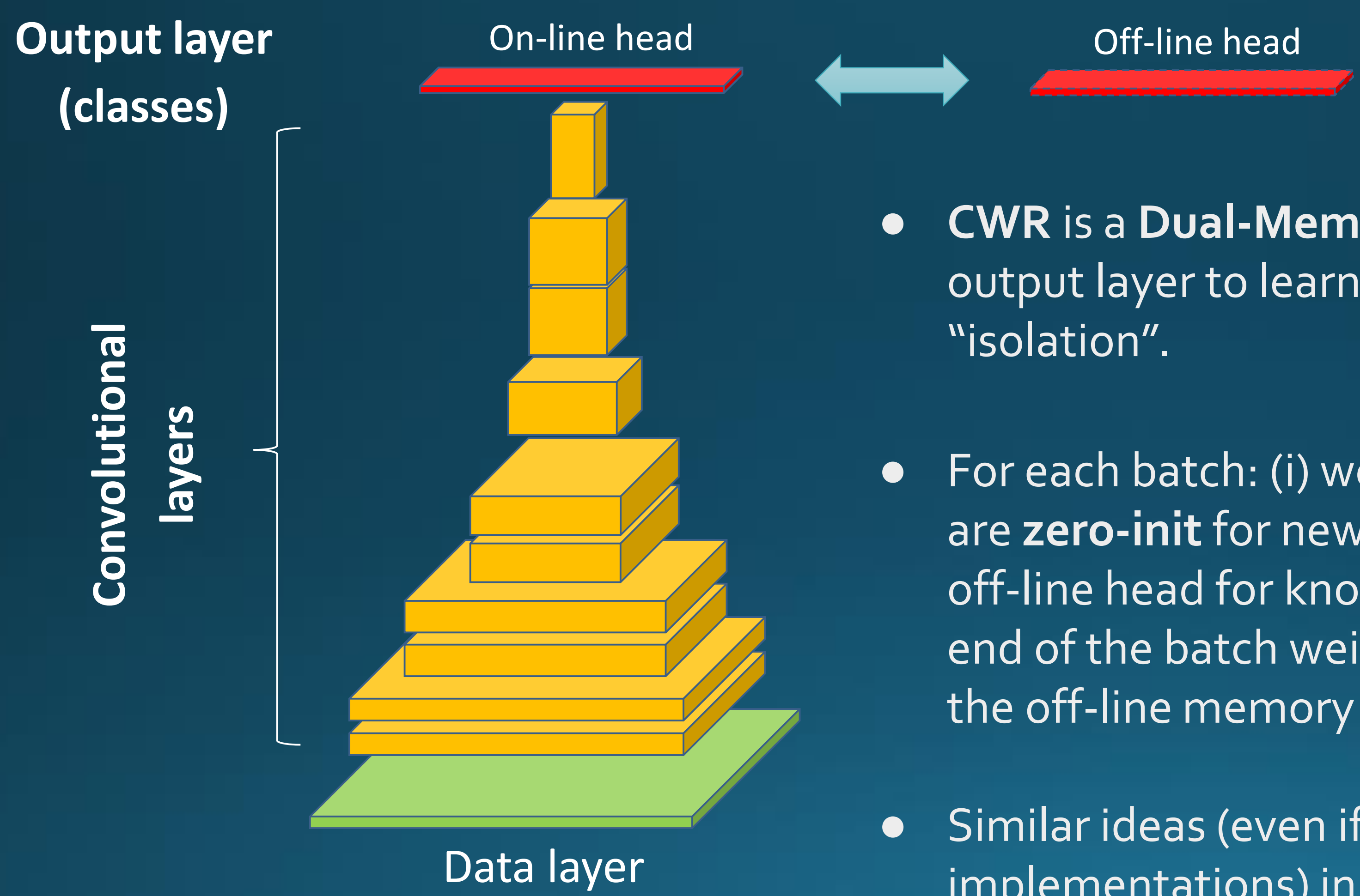
Categories

Objects

# Images	164,866
Format	RGB-D
Image size	350x350 128x128
# Categories	10
# Obj. x Cat.	5
# Sessions	11
# img. x Sess.	~300
# Outdoor Sess.	3
Acquisition Sett.	Hand held

Vincenzo Lomonaco and Davide Maltoni. CORe50: a new Dataset and Benchmark for Continuous Object Recognition, 1st Conference on Robot Learning, Mountain View (CA), 2017.

AR-1: Architectural component (CWR)

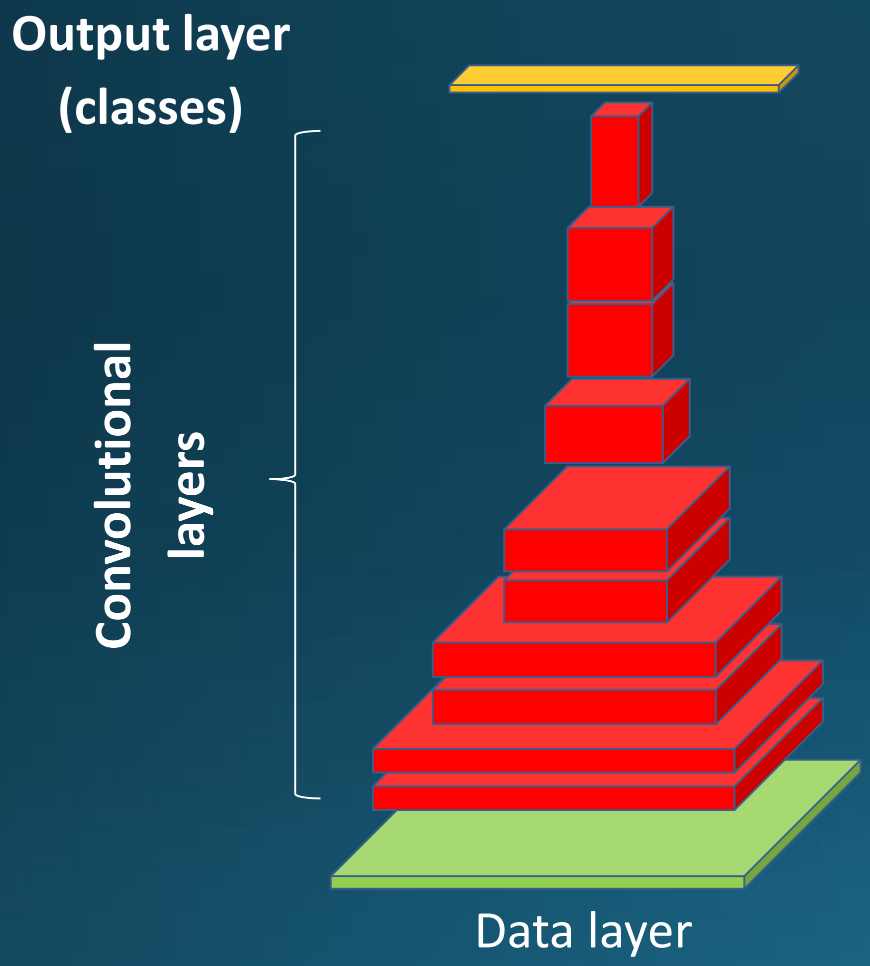


- **CWR** is a **Dual-Memory** system for the output layer to learn the current batch in "isolation".

- For each batch: (i) weights in the online head are **zero-init** for new classes and **reload** from off-line head for known classes; (ii) at the end of the batch weights are consolidated in the off-line memory by weighted averaging.

- Similar ideas (even if more complex implementations) in recent **Rebalance** and **ScalL** approaches.

AR-1: Regularization (SI)



- To control forgetting in the lower layers we use **Synaptic Intelligence** (EWC like approach) to limit update of *important* weights.
- **SI** exploits information made available by SGD and does not require further gradient propagations to compute the weight importance.
- We proposed a variant of the EWC weight update that does not require to store the old weights and reduce the risk of divergence.

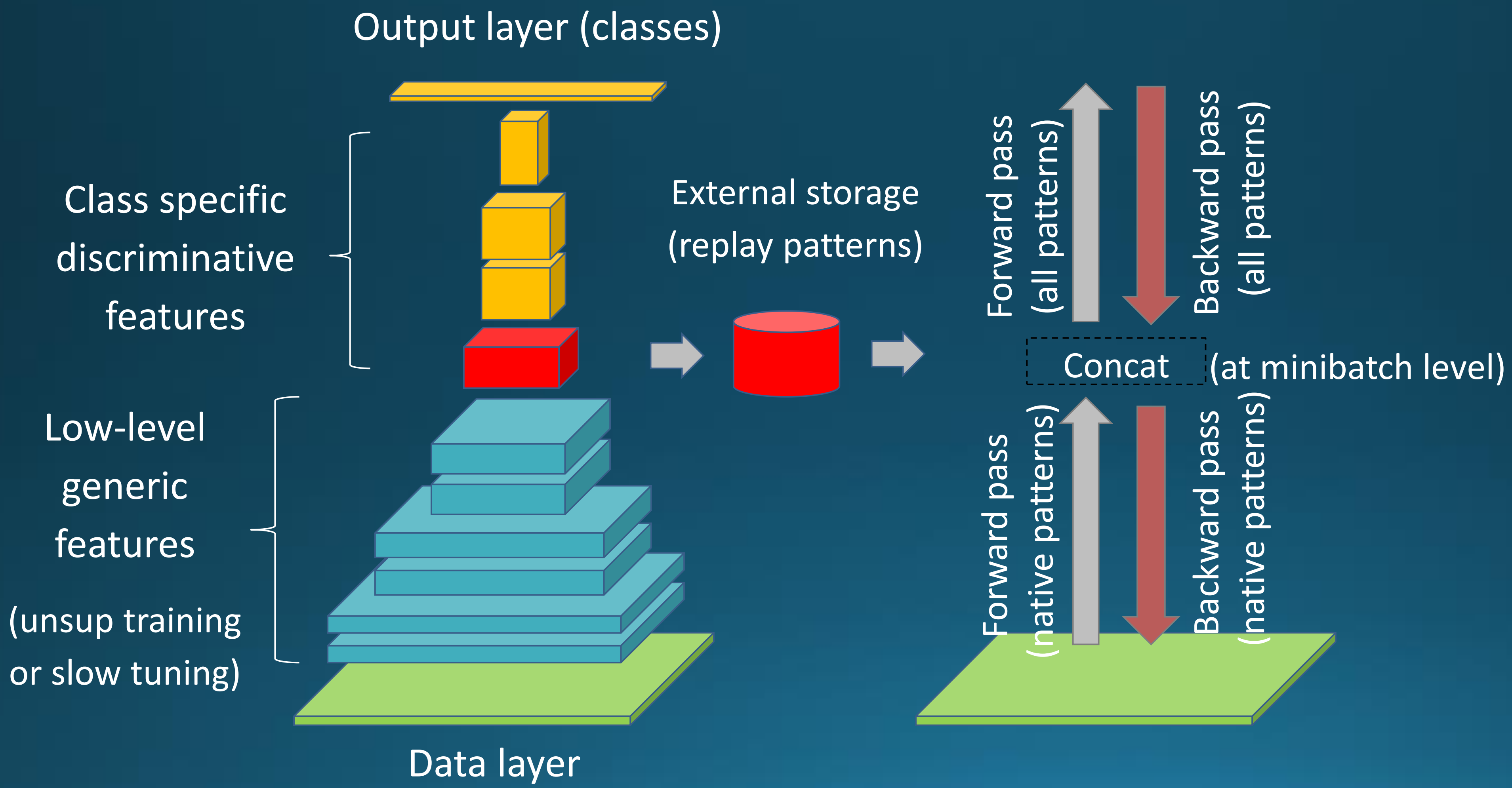
Practical issues with rehearsal

1. **Rehearsal (or replay) requires to store some representatives of old batches**
   - ICARL is one of the best-known techniques.
2. **Requires extra storage**
   - For example for ImageNet, if we store 20 patterns per class, the total storage is about 3.8 GB.
3. **…and extra forward/backward steps!**
   - When mixing new and old patterns more iterations for epoch.

Idea: storing activations at some intermediate level and not raw images.

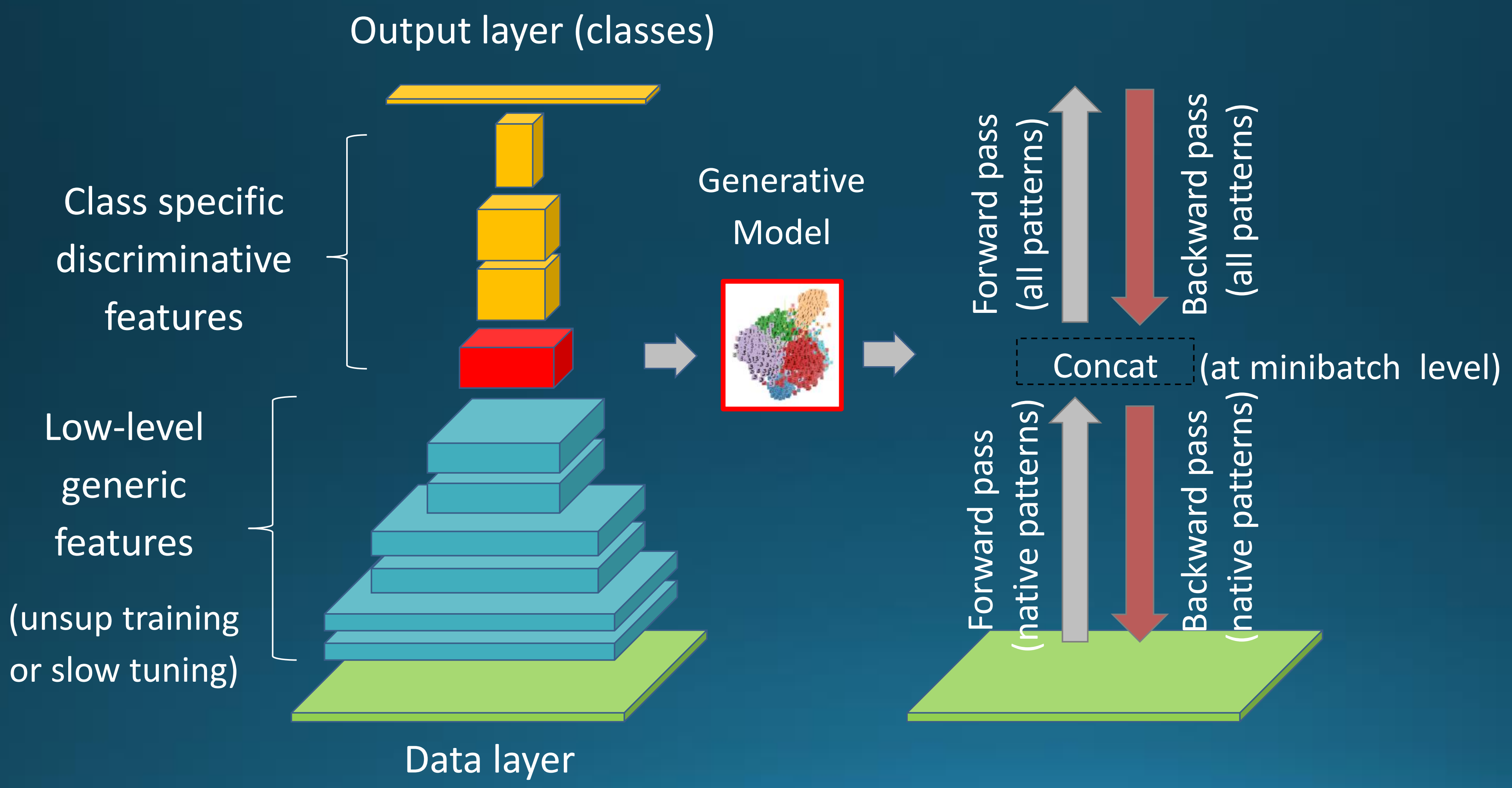
AR-1 with Latent Replay

this is the solution used in the CORe demo App



Work in progress

AR-1 with Latent Generative Replay



Future work

Move toward unsupervised training

- Self-training by exploiting temporal coherence*.
- Openset classification (automatic discovery of new classes).
- Sparse human supervision (active learning).

Applications

- On device adaptation (e.g., Recycling code application)
- Loop Closure in Robotic SLAM.

*D. Maltoni and V. Lomonaco. ***Semi-supervised Tuning from Temporal Coherence***, ICPR 2016.