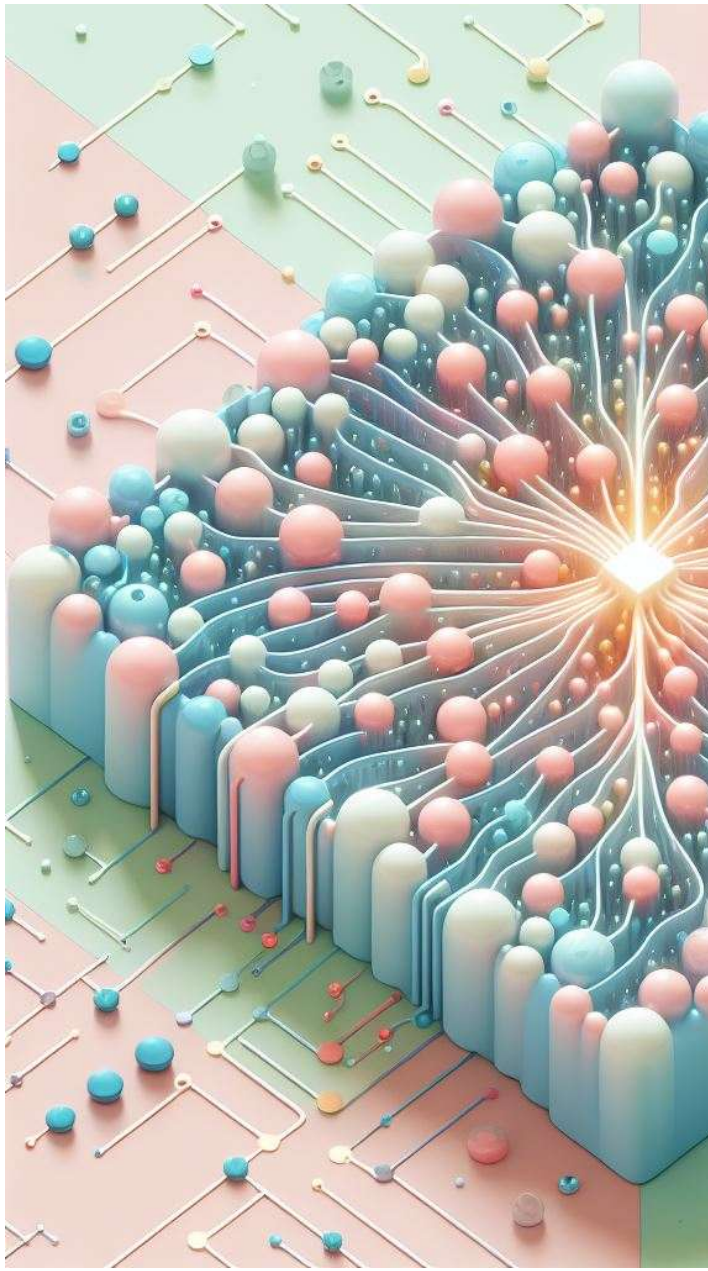
# On Class-Incremental learning for Fully Binary Neural Network

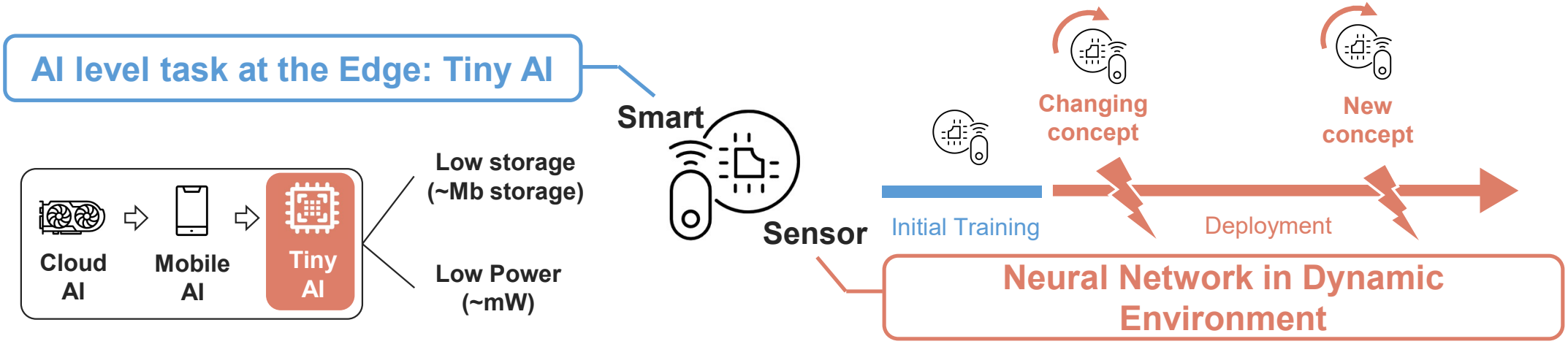
Yanis BASSO-BERT, William GUICQUERO, Anca MOLNOS, Romain LEMAIRE, Antoine DUPRET

*This work is part of the IPCEI Microelectronics and Connectivity and was supported by the French Public Authorities within the frame of France 2030.*



# 1. Why BNN and Incremental Learning ?

# Context & Motivation




**Bring AI level task close to sensor can lead to breakthrough functionality :**

- Send only relevant information
- Adapt the sensor functioning point
- Wake up signals

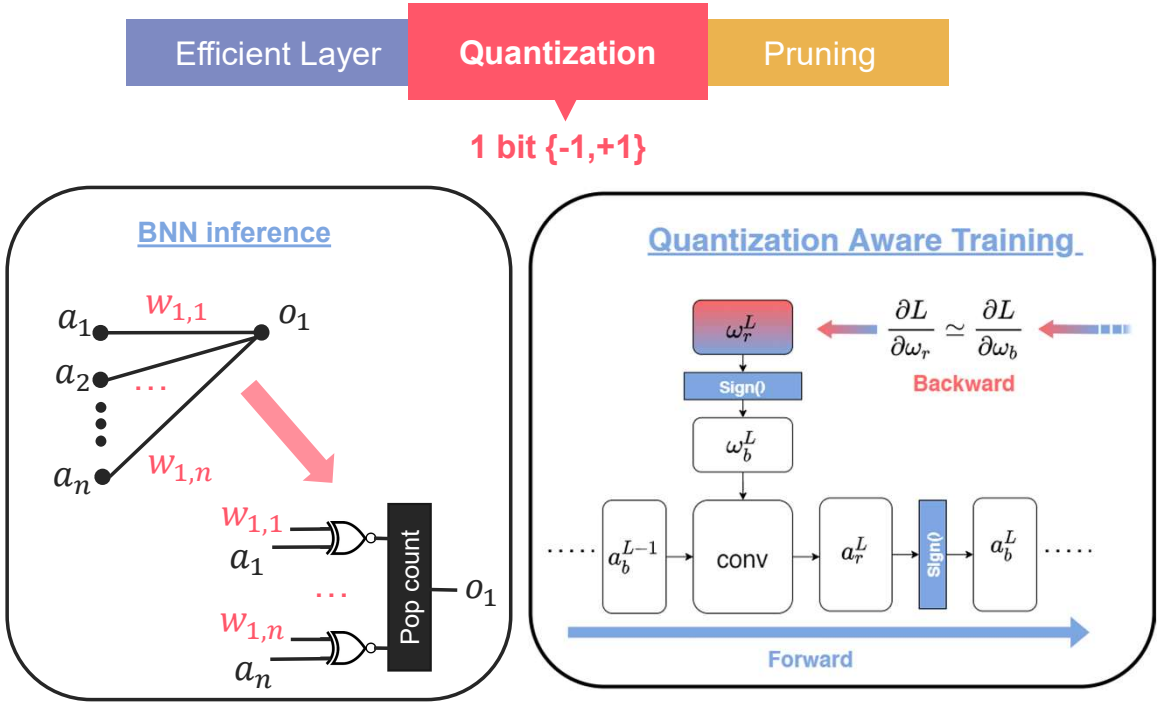
**However in real-life scenario the deployment environment is permanently changing**

- Need to detect changes and to **adapt** while **accumulate** knowledge
- Need an agile way to retrain the network

**How to learn incrementally on resource-constrained hardware platform ?**

# BNN – The solution for Embedded AI on emerging technology…

**Compact Neural-Network co-design** [1]




**Versatile Network thanks to simple arithmetic**

BNN use **logic-gate** type of arithmetic which allows to use a wider variety of technology and design approach for **low-power hardware design**

→ Stay true if there is no Mixed Precision operation
**Fully Binary Neural Network**

**However they show several drawbacks**

- drop in performance
- hard to converge
- lack of expressivity

→ Training once is complex, so several times…

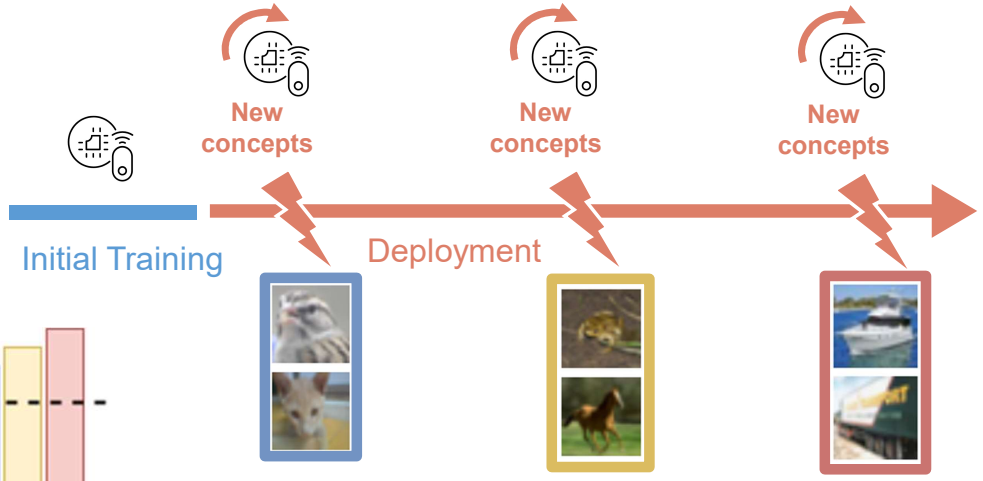
**… but not easily compatible with IL**

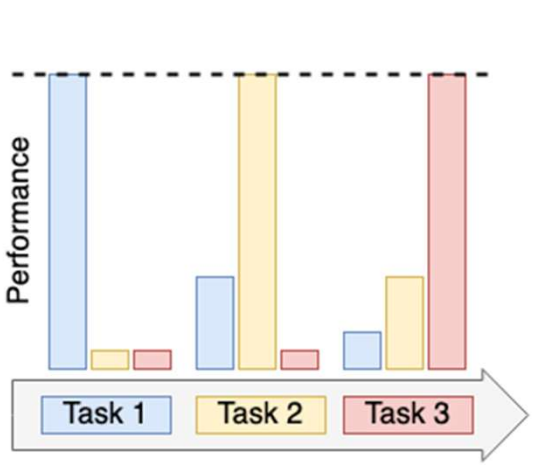
# Class-Incremental Learning is challenging on its own

**( = New concepts)**

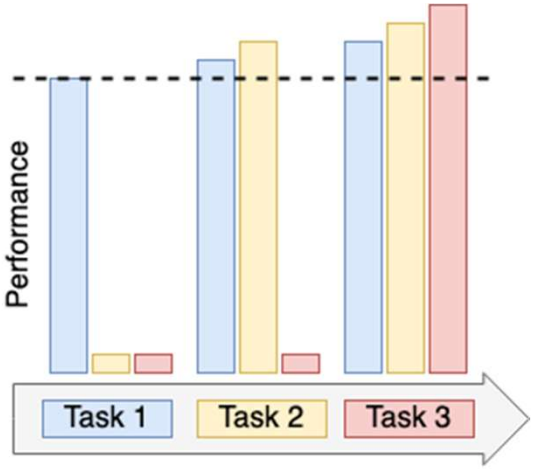
## Plasticity-Stability dilemma [2]

- Neural network tend to forget…
- … and to much regularization and it cannot learn




**Vanilla Case**

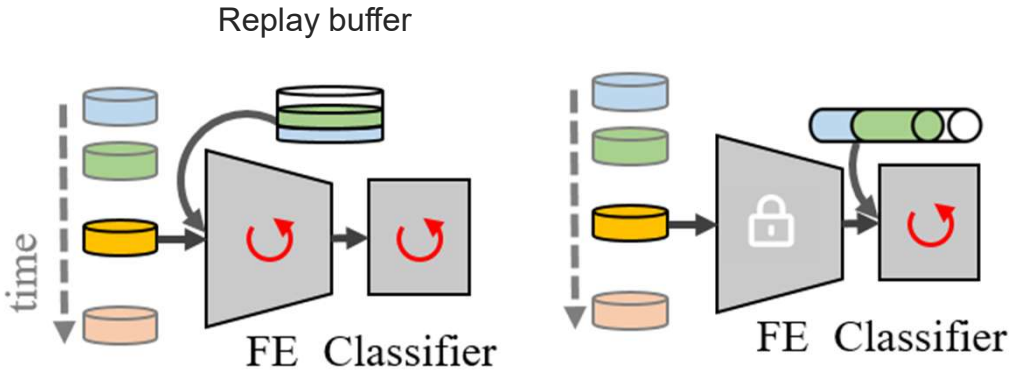



**Ideal Case**

# How to overcome forgetting ?

**Strategy develop in the ML literature** [3]

- Architecture Growing
- Weights regularization
- Activity regularization
- Training regularization
  - **Replay method**




**Best performance in the literature are obtained with replay method but at the cost of a large memory footprint**[3]

**Solution: Buffer Size Reduction**

**Feature Extractor act as dimensionality reduction + Latent Replay** [4][5]

# Open questions

1. How to design a Compact Fully BNN for CIL?
2. Is it possible to retrain it with a (latent) replay buffer?
3. Impact of the Replay Buffer size on successive performance?
4. BNN compare to an equivalent Full Precision Network?

# Contributions

- **Learning system design**
  - A 4Mb BNN compute graph and its training protocol for CIL
  - CIL evaluation of Latent Replay and Native Replay on CIFAR100

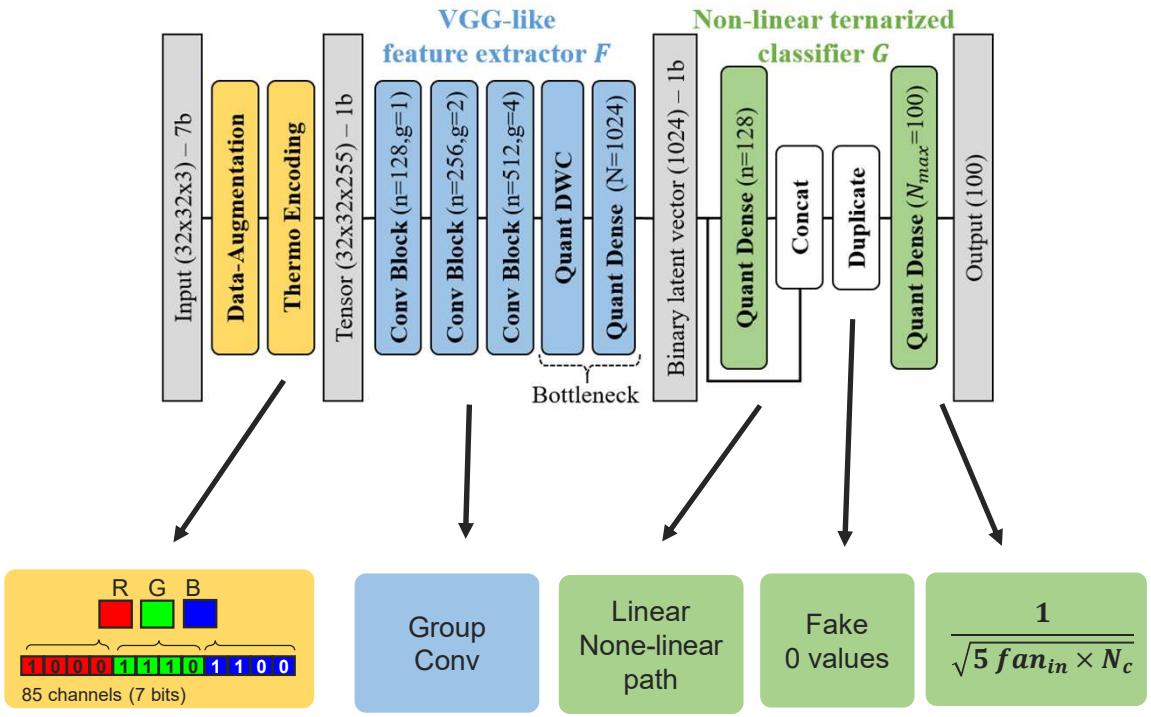
- **Results**
  - Effect of Replay Memory Buffer footprint
  - Comparison with Full Precision network on an equal footprint (x32 less weights) in different scenarios

# 2. Learning System Design

1 – Fully Binary Neural Network
2 – Replay Strategies and their compromises

# Network Design and Training Protocol for binary only arithmetic

## 4Mb Fully-binary NN architecture



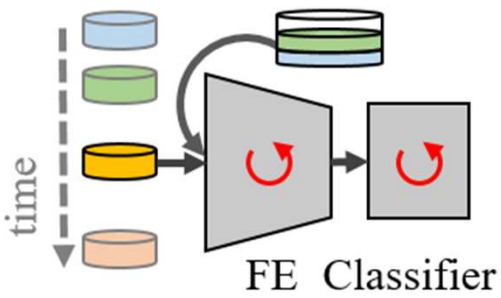

## Adaptive BN-less training protocole

- Adpative lr scheduler for autonomous and terminal convergence
- Remove Batch Normalisation Phase in training
- Replace by scaling factor that ensure unit activation variance

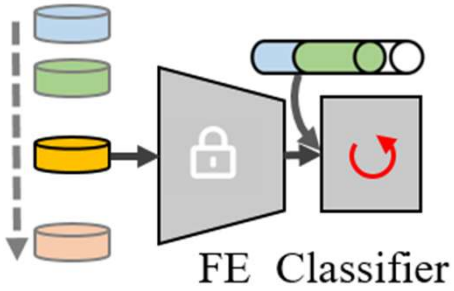
$$K = \frac{1}{\sqrt{fan_{in}}}$$

| | FP I/O layers | Parameters | CIFAR100 |
|---|---|---|---|
| **BNNwoBN** [6] | ✓ | 28M | 55,0% |
| **Ours** | ✗ | 4M | 53,3% |

# Latent Replay compared to Native Replay




Native Replay (NR)




Latent Replay (LR)

- **More diversity**
  - **Sample storage per Mb:** LR allowed to store x24 samples per Mb
- **Less expressiveness**
  - **Information contained in a sample:** Image are more informative than compressed latent representation
  - **Data augmentation:** There is no canonical Data augmentation in latent space
  - **Dependance on pretraining:** Feature extract is forced to be fixed in latent replay

# 3. CIL Results

# Artificial Benchmark scenario

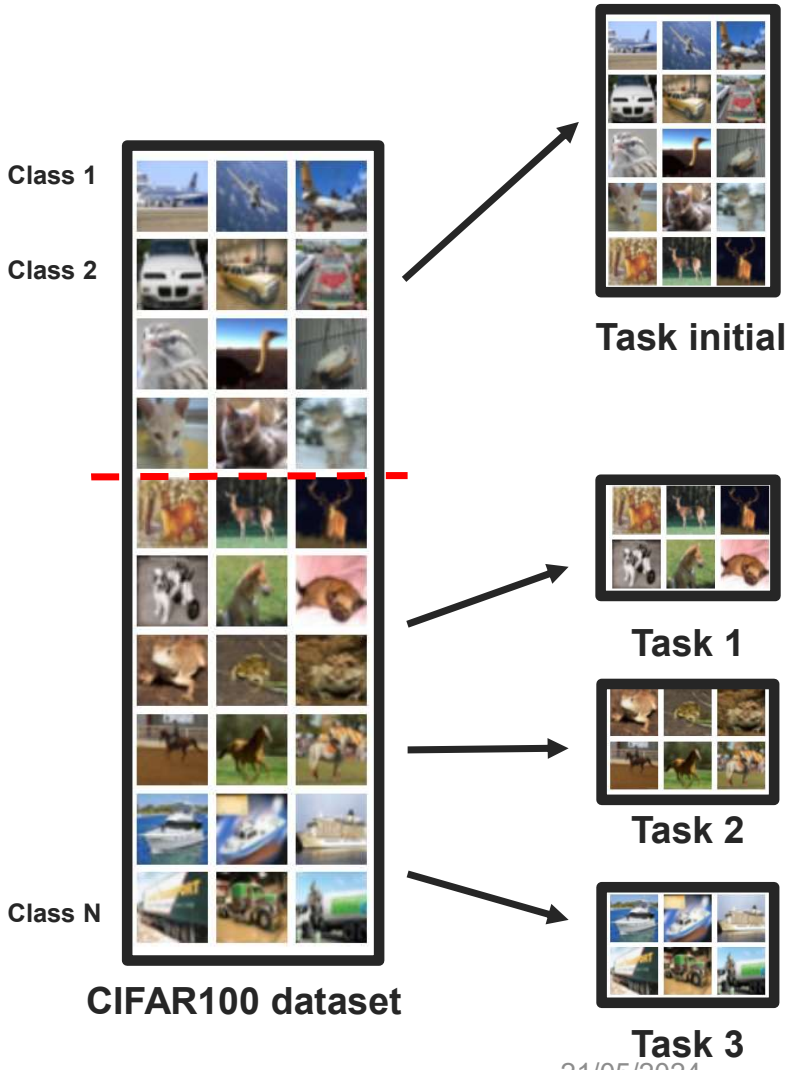
## Evaluation benchmark : Split-CIFAR100

- **Pre-training:** first 50 classes
- **Deployment:** tasks with last 50 classes

- Classifier is reset after pre-training

- **Native replay**
  - Feature extractor is set trainable
  - Examplars are stored from the input space

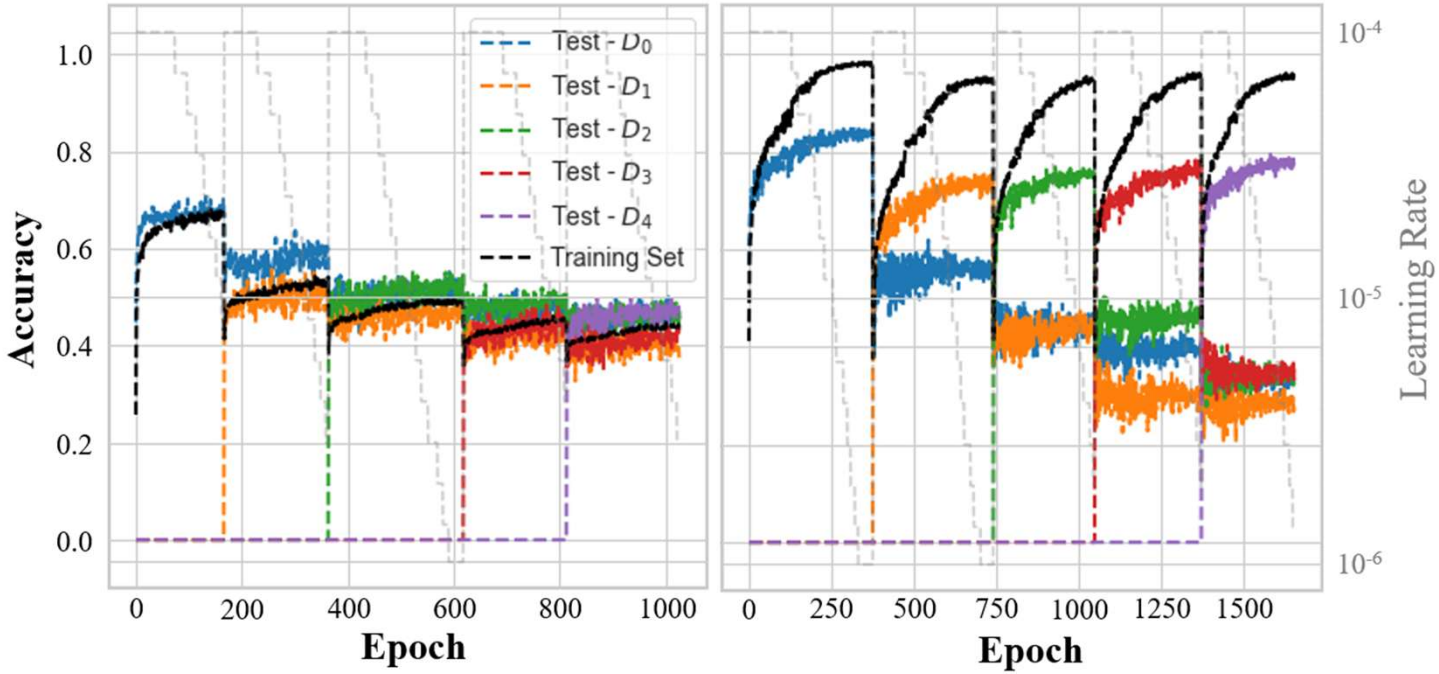
- **Latent replay**
  - Feature extractor is frozen after pre-training
  - Examplars are stored from latent space





This work is part of the IPCEI Microelectronics and Connectivity and was supported by the French Public Authorities within the frame of France 2030.

# Successfull retraining with a 25Mb Memory Buffer

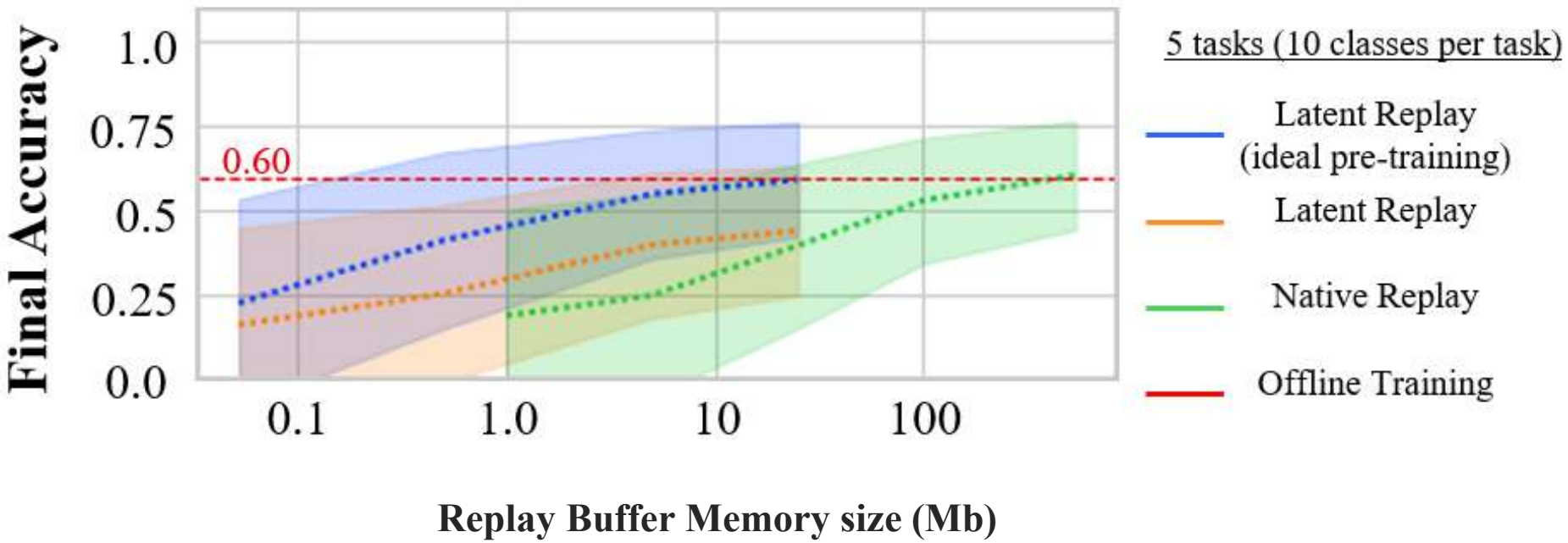



## Key take-aways

- Adaptative training protocole allows robust convergence across task
- Overfitting is controled in both cases


This work is part of the IPCEI Microelectronics and Connectivity and was supported by the French Public Authorities within the frame of France 2030.

# Comparison: Latent Replay vs Native Replay

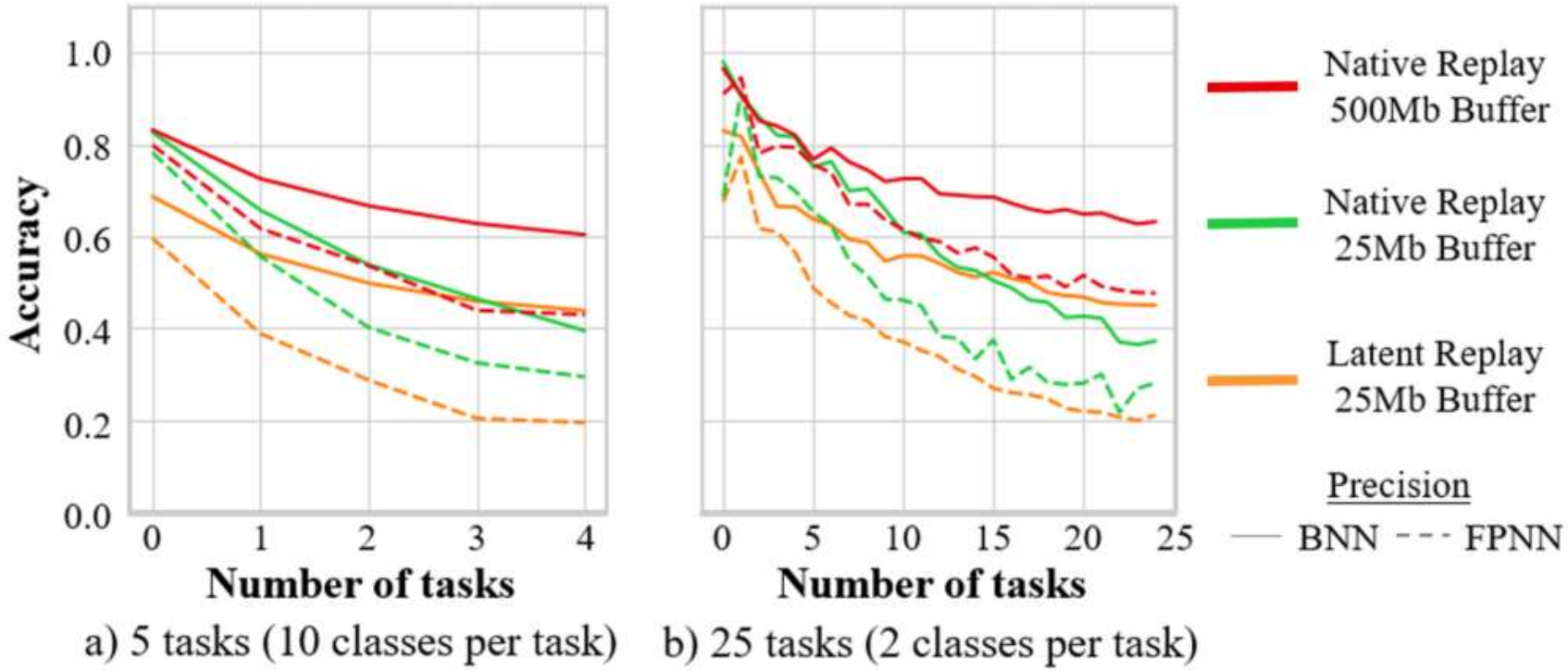



## Key take-aways

- Latent Replay can outperform Native Replay for a limited replay buffer memory size
- The main bottleneck of Latent Replay performance remains the Feature extractor pre-training


This work is part of the IPCEI Microelectronics and Connectivity and was supported by the French Public Authorities within the frame of France 2030.

# Robustness to retraining




a) 5 tasks (10 classes per task) b) 25 tasks (2 classes per task)

**Key take-aways**

- Increasing the number of re-training steps do not impact the final performance.
- BNN is 10% higher to FPNN of *same memory footprint* .


This work is part of the IPCEI Microelectronics and Connectivity and was supported by the French Public Authorities within the frame of France 2030.

# 4. Conclusion

# Conclusion and take-aways

- Presented good practices for Fully BNN and training protocol designing in CIL setting
- Show that Buffer Memory Footprint has an impact on the Strategy Choice
- Show that Pre-training quality is the main limitation for latent replay
- Show that on an equal footprint BNN can outperform with FPNN

# Thank you for your attention

## On Class-Incremental learning for Fully Binary Neural Network

Yanis BASSO-BERT, William GUICQUERO, Anca MOLNOS, Romain LEMAIRE, Antoine DUPRET

*This work is part of the IPCEI Microelectronics and Connectivity and was supported by the French Public Authorities within the frame of France 2030.*

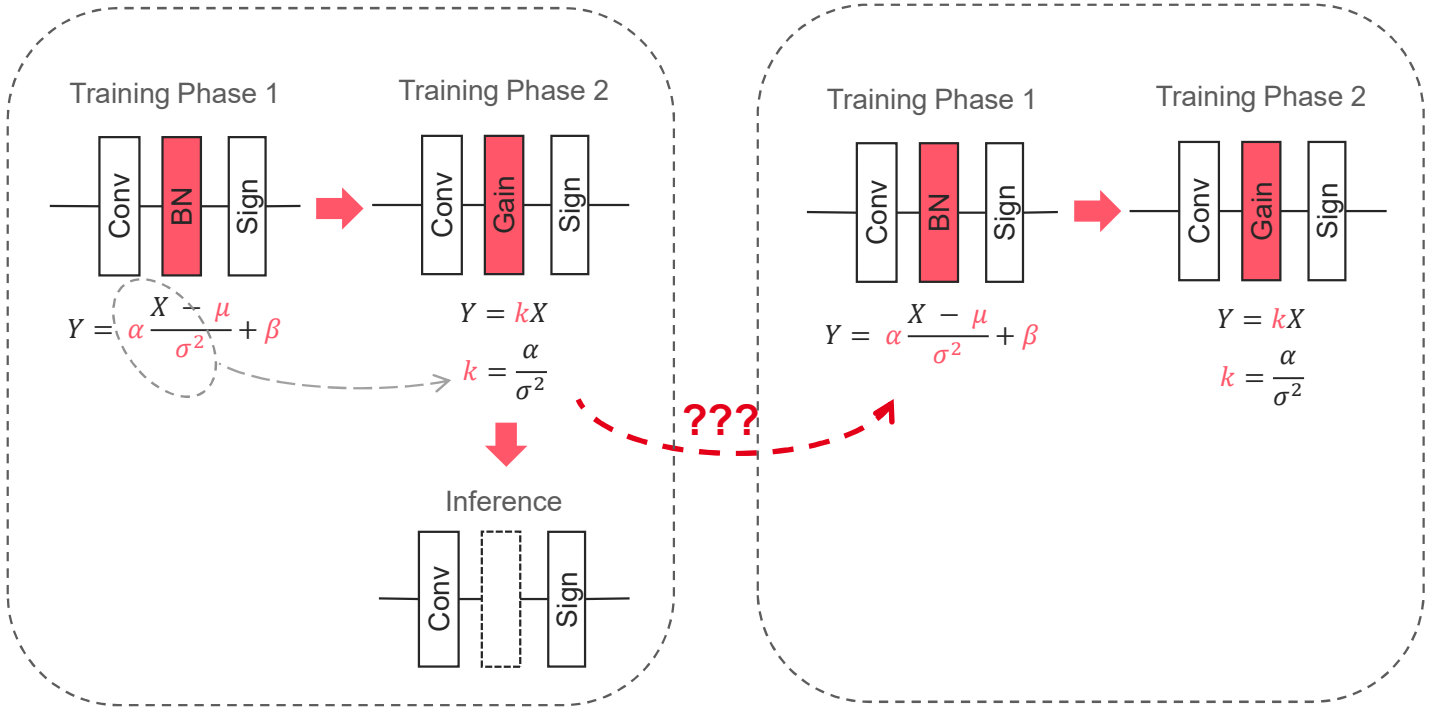
# 5. Annexes

# Future directions

- Pre-training protocole for transferable representation
- From Replay to pseudo-replay
  - Are generative model a better option ?
- On-Device training
  - Quantized gradients
  - Alternative to back-propagation
- Considering more realistic Scenario
  - No task boundary : anomaly detection
  - No labels : OpenSet Recognition

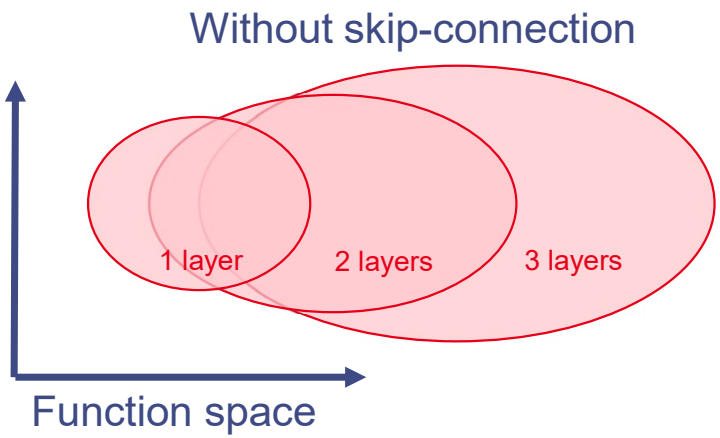
# From “single-phase” to “two-phase” learning

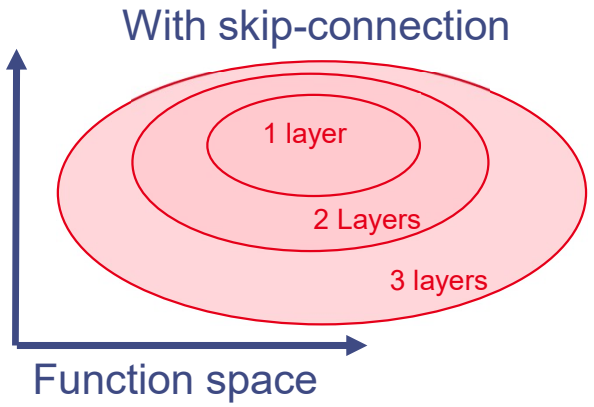
Batch Normalisation (BN) warm-up is a common methodology to help convergence in offline training. Its hardly applicable in an incremental setting




**Solution :** Get rid of the warm-up phase and find the right initialization for the Gain factors

# Non linear ternarized classifier




$$A_{out} = W_2 \times sign(W_1 \times I + I) = W_2 \times sign(W_1 \times I) + W_2 \times I$$