Embedded Continual Learning for HEP

Marco Barbone, Christopher Brown, Georgi Gaydadjiev, Alexander Howard, Wayne Luk,
Tom Maguire, Mikael Mieskolainen, Benjamin Radburn-Smith, Andy Rose, Alexander Tapper

m.barbone19@imperial.ac.uk

Imperial College London

Continual Learning (CL)

Concept: Train a model with a continuous stream of data

Learns from a sequence of partial experiences rather than all the data at once

Advantages:

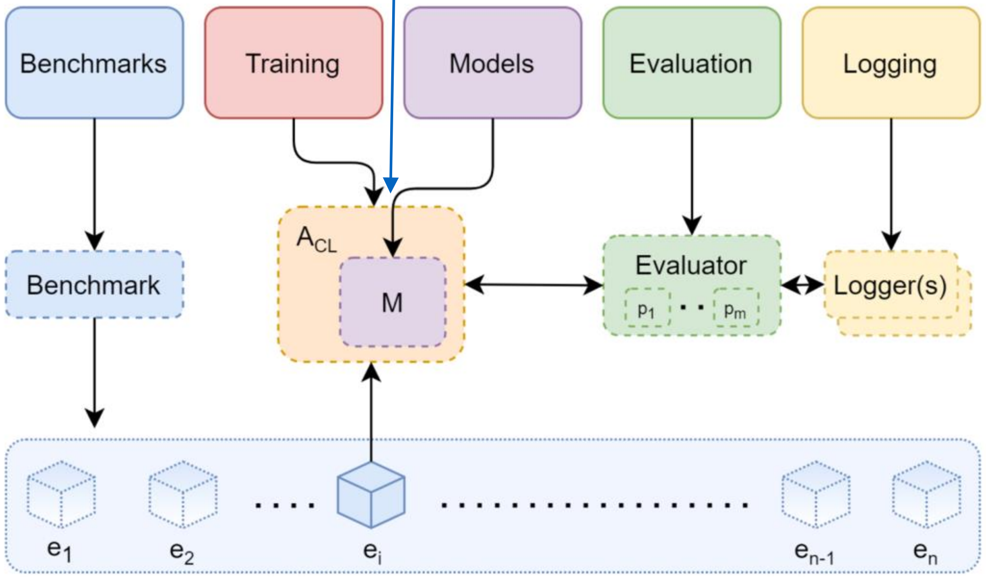
- Avoids **catastrophic forgetting** → initial training is not disregarded
- **Adapts** to a **changing data stream** → don't need to quantify how the environment changes
- Don't need to store previous training data or retrain model

Disadvantages:

- For supervised learning need a ~continual stream of labelled data which might not be accessible

CL Algorithm

[ref]



Non-stationary stream of experiences

Imperial College London

Motivation

- Changes to the input distribution of a model can lead to a model being invalid or sub optimal
- Having to retrain a model offline means there is a loss of accuracy until the newly retrained model is online
- For the duration of the retraining of the model you are working with a model that you know is suboptimal
- This is can be relevant in the context of CERN trigger system and other constrained environments

Imperial College London

Timescales for CMS

Seconds	Days	Months
Beam fluctuations	Beam conditions or small degradation	Significant detector changes
No time to create training data or retrain a model L1 can monitor ML robustness Online CL could be explored	Time to collect data directly from detector e.g. scouting or full reconstruction Between fill calibration [1][2]	Time to accurately emulate detector performance in large MC campaigns

This talk

Chris's talk
This afternoon 2:30 PM

Imperial College London

Training

- It is computationally expensive
- Usually done on GPUs
- Embedded systems, constrained environments do not have accelerators

Imperial College London

Goal

Find an alternative to **stochastic gradient descend** (SGD) to allow CL on embedded system

Imperial College London

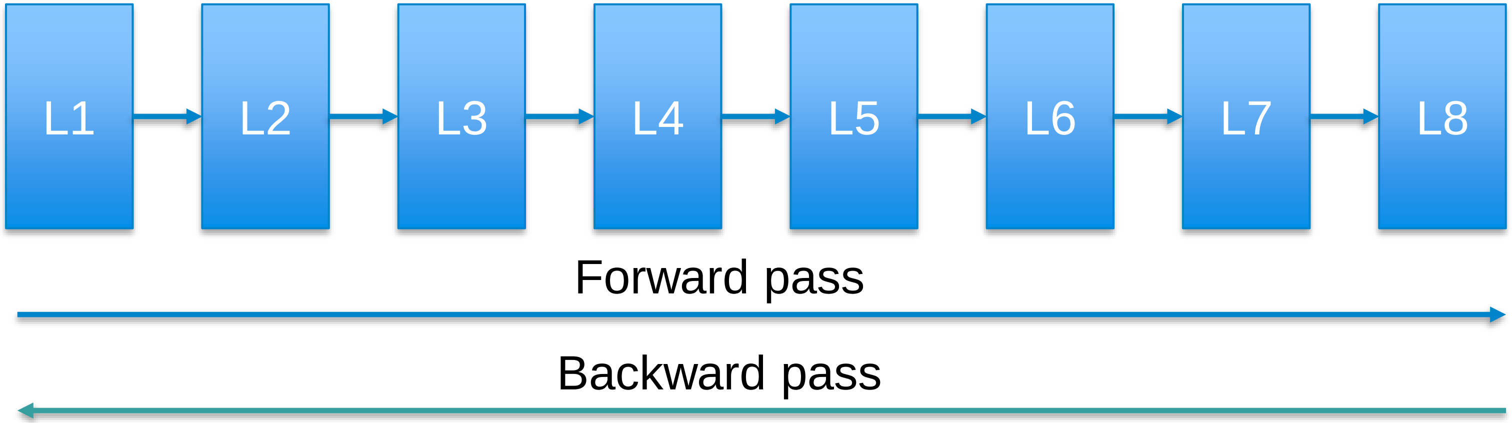
Alternating Minimisation (AltMin)

- An alternative to SGD
- Open source
- Proof of concept
- Experiments are reproducible

Imperial College London

AltMin vs SGD

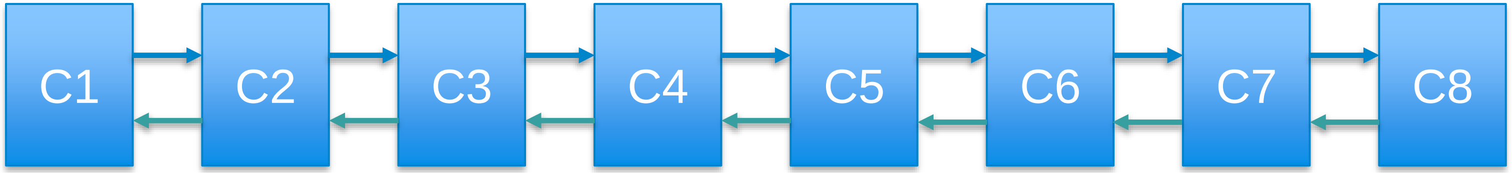
AltMin drops one order of complexity



L1 cost function depends on all the previous layers

SGD

AltMin



- Forward pass populates “codes”
- Cost function depends only on the previous layer **(No chain rule)**

AltMin vs SGD

	Backpropagation	Alternating Minimisation
Vanishing and exploading gradients	Yes	No
Biologically implausible	Yes	Yes (but closer to plausible)
Allows for parallel weight updates	No	Yes
Gives good accuracy on benchmark datasets	Yes	Yes
Speed of initial convergence	Slightly slower	Slightly quicker
Smoothness of convergence	High	Medium

Imperial College London

Dataset

Modeled a "typical" High Luminosity LHC event that emulates the CMS phase-2 upgraded detector

Dataset created from a top quark pair production sample generated in Pythia with an additional 200 soft proton-proton interactions overlayed on top

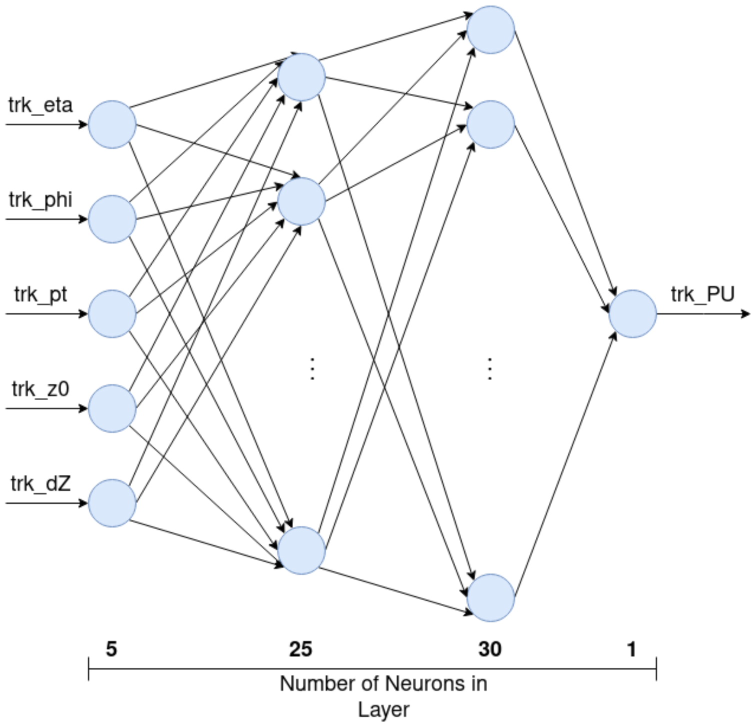
Tracks were generated using Delphes running a simulation of the high lumi CMS detector, tracks kinematically constrained to pT > 2 GeV, |eta| < 2.4 and |z0| < 15 cm to emulate a CMS level-1 tracking set of tracks

Tracks were then reprocessed in python to generate two datasets, an unsmeared dataset of 10,000 events, taken directly from the Delphes output and another 10,000 events that were smeared using a gaussian smear on each track parameter. This smearing was gradually increased across a set of 10 separate smearings to give a set of 10 individual experiences for the CL algo. Smearings emulate a worsening of detector resolution over time

Imperial College London

Network

- The model takes 5 features as input
- There are two hidden layers with 25 and 30 neurons respectively
- Track eta, phi, pt and z0 are track helix parameters taken from the delphes simulation and smeared using a guassian smear for the smeared datasets
- Track dZ is the distance between the track z0 and a primary vertex found in the event using a simple histogram based vertex algorithm
- The target is whether the track originated from the underlying top quark pair production or from the additional soft pileup

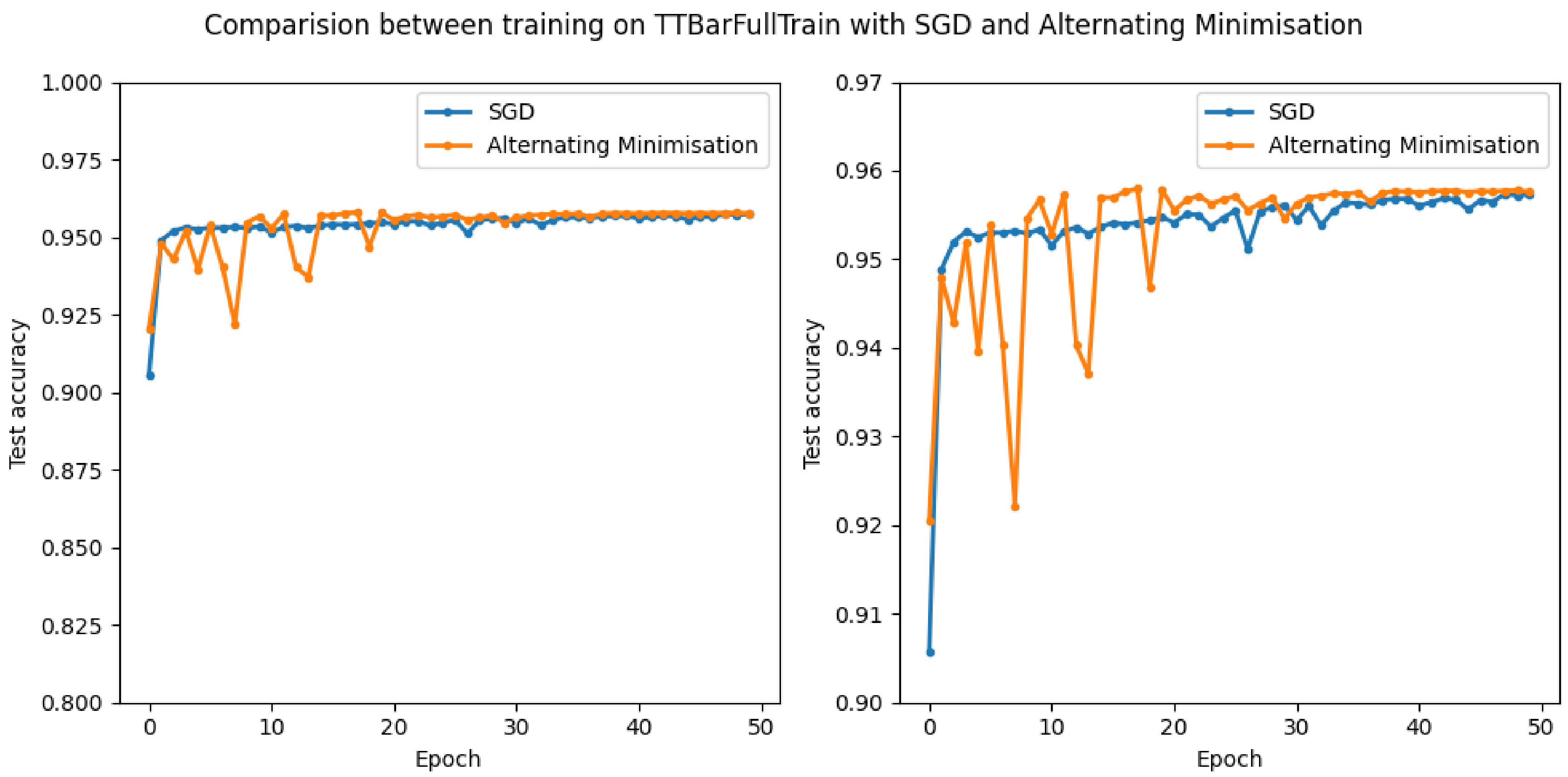


Experimental setup

- Trained on non smeared data
- Executed the CL model on smeared data
- Compared against SGD

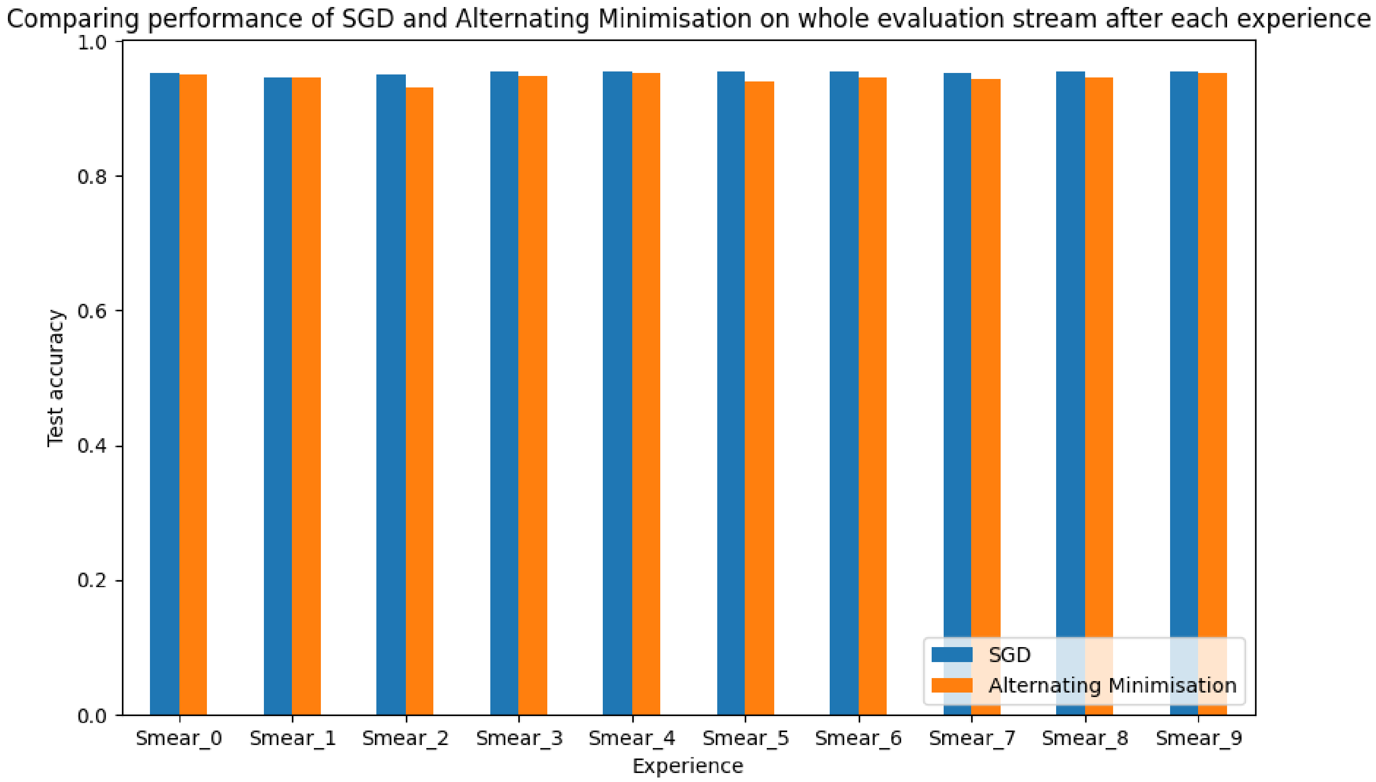
Imperial College London

Training convergence

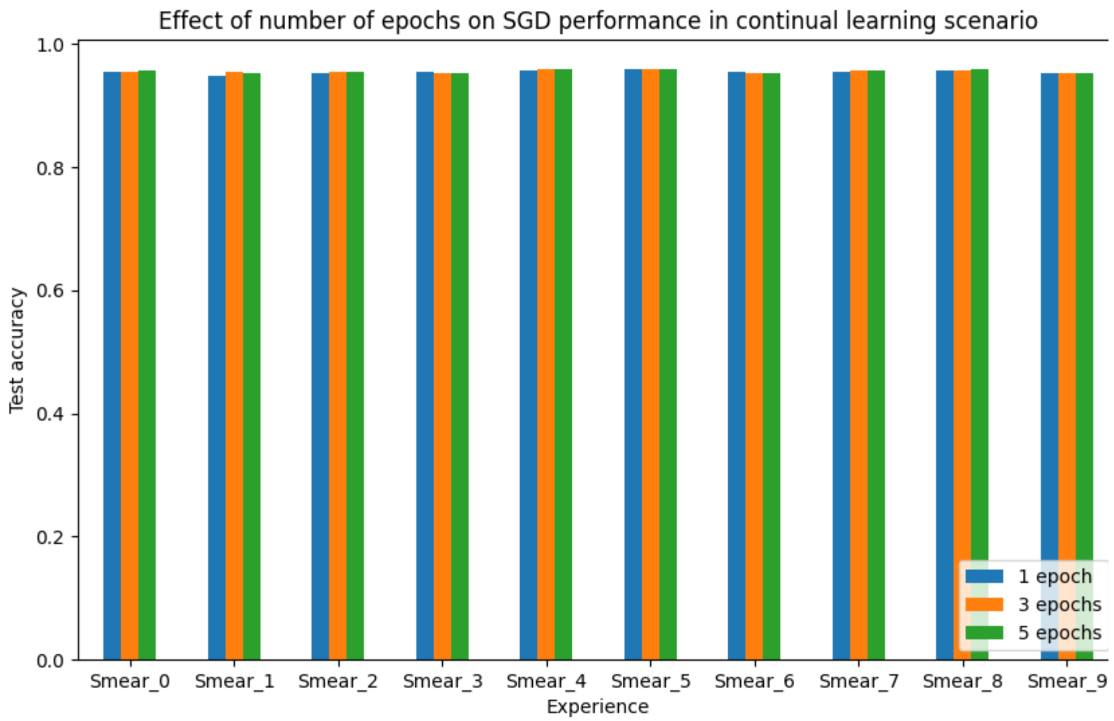


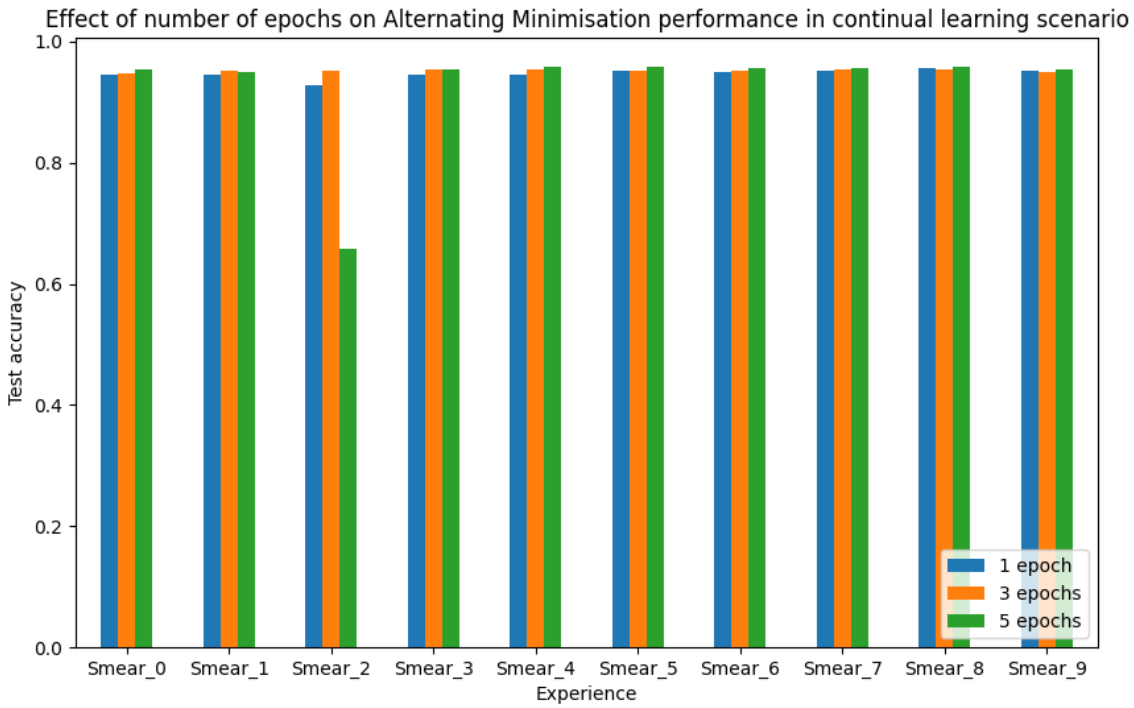
Imperial College London

Network accuracy (CL)



Number of epocs





Imperial College London

Conclusions

- AltMin is a valid alternative to SGD
- Experiments show that both techniques have comparable performance
- Future work: aim to implement AltMin in C++/CUDA

Backup

Imperial College London

Alternating Minimisation (continued)

- The algorithm stores code variables in the forward pass that are equal to a linear transformation of the previous-layer activations
- These code variables are used to propagate the error backwards
- These code variables are then used to update the network's weights after each iteration