SMARTHEP

REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

ESR12: Accelerated Anomaly Detection

Pratik Jawahar

Supervisors:
Caterina Doglioni, Jiri Masik, Alex Oh, Maurizio Pierini

Overview:

- Heterogenous Tracking
- AD for DQM
- Knowledge is Overrated - Fast Inference
- Other Activities

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824 The University of Manchester

SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

Qualification Task: Heterogenous Track Reconstruction

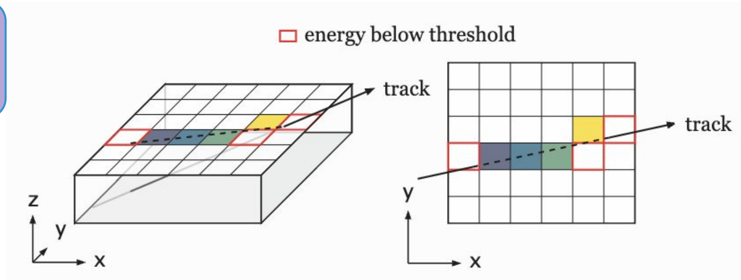
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824
The University of Manchester

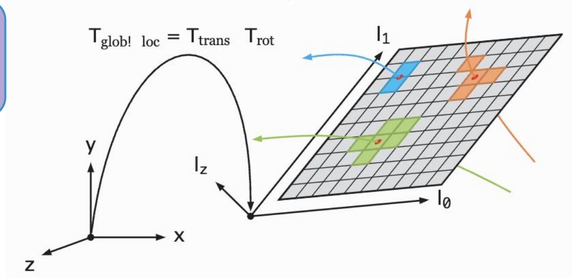
SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

Track Reconstruction

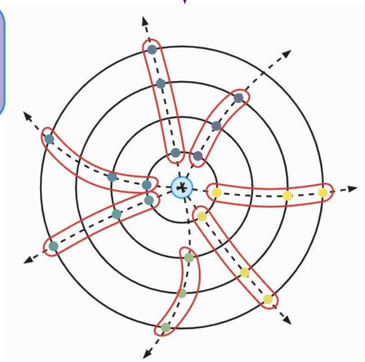
Clusterization



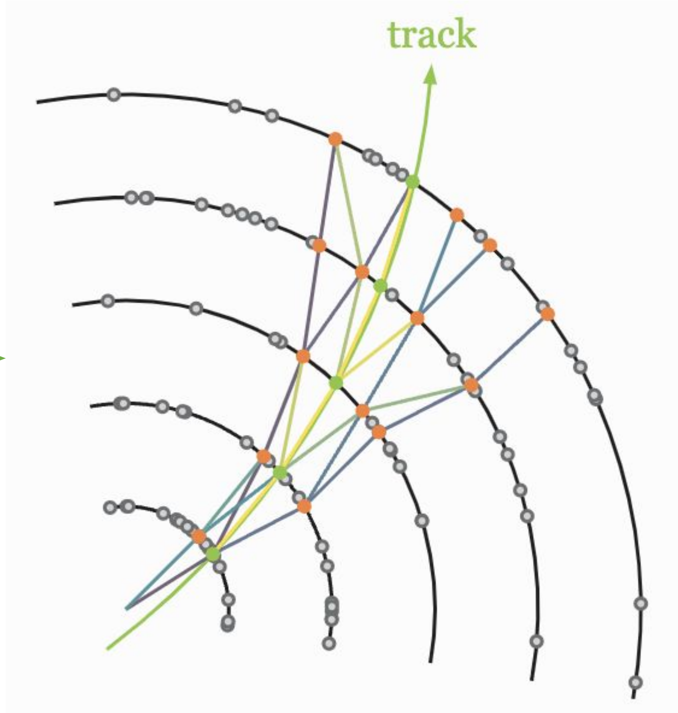
Spacepoints



Seeding



Track Finding



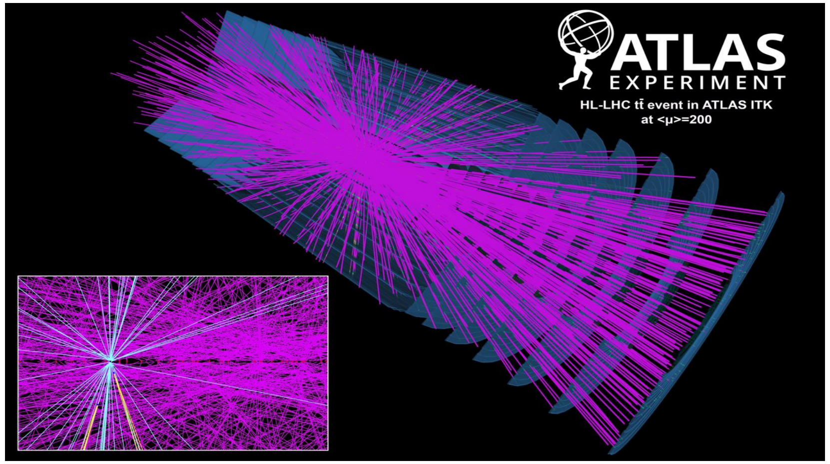
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824
The University of Manchester

SMARTHEP Problem(s)!

REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

- No. of tracks per event for HL-LHC, expected to increase 2.5x
- Current R&D: Use GPUs for speedup via parallel computation
- However, sequential algorithms like CKF do much better on CPUs than GPUs!
- Heterogenous track reco? (CPU-GPU)



acts

traccc

Demonstrator tracking chain for accelerators.

Computing term for specific purpose architectures (eg. GPU, TPU, IPU etc.)

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

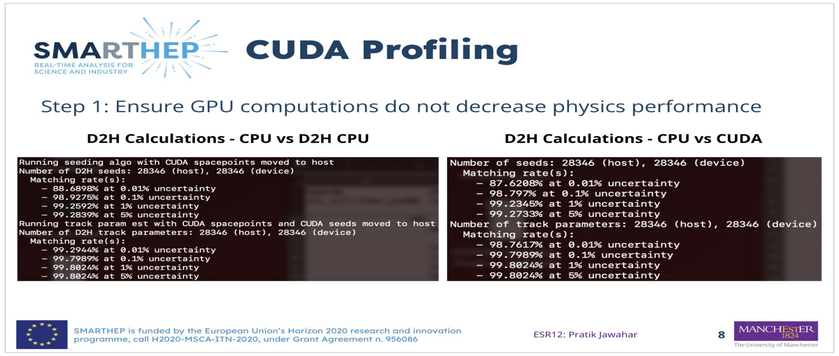
Heterogenous Track Reconstruction:

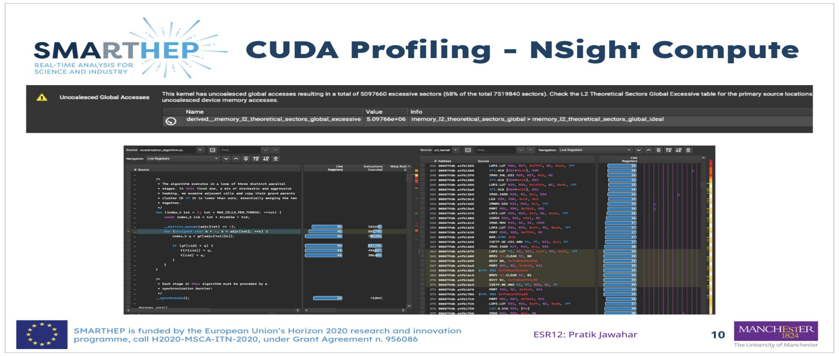
- Step 1: Profiling CPU and GPU code to identify run-time speed-up
    - Ideally without drop in tracking efficiency
- Step 2: Identify bottlenecks
    - Points where one architecture outperforms the other
- Step 3: Calculate data-transfer latencies at bottlenecks
    - Data transfer latencies between host (CPU) and device (GPU) eat up speed-up

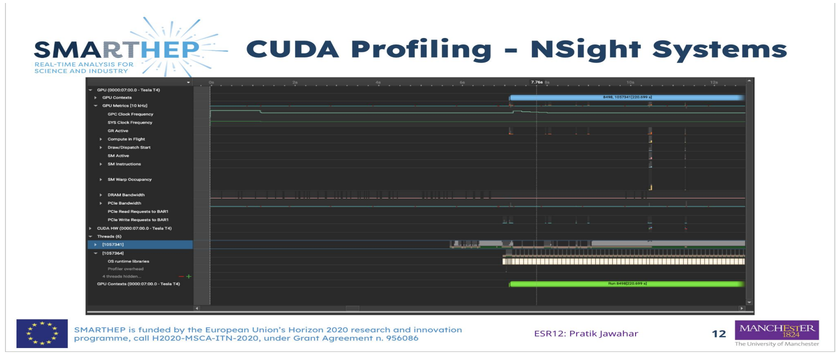
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824
The University of Manchester

CUDA Profiling (MetaInfo ♞)







CUDA Profiling - TRACCC (mu200)

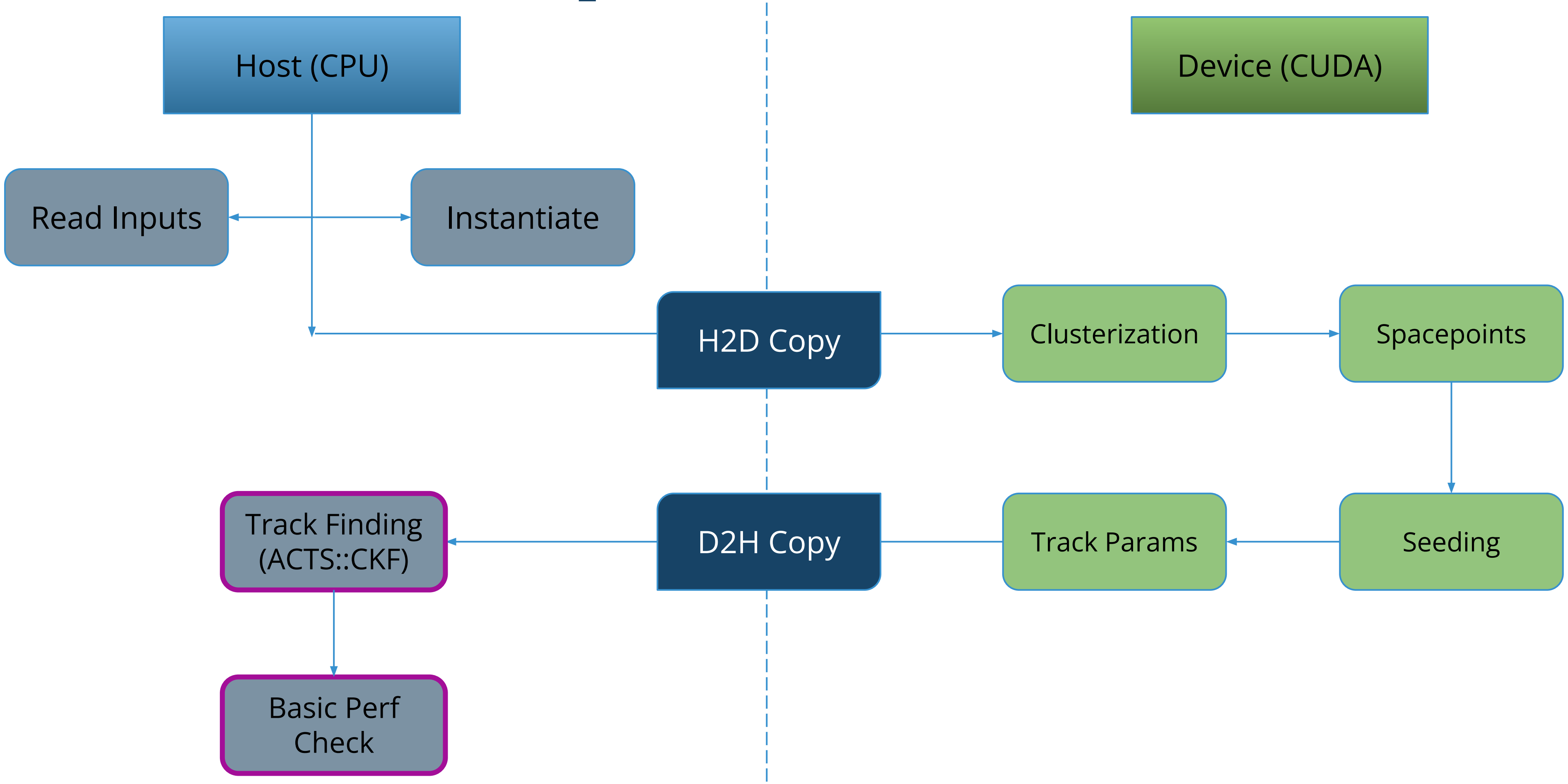
- POC feasibility example:
  - Clusterization, Spacepoint formation, Seeding are significantly faster on Device
  - Considering Host-Device and Device-Host wall-time overheads,
    - there is still a speedup of ~5800 msec until the seeding step of the chain

Data File	Detector Geometry	No. of Events
tml_full/ttbar_mu200	tml_detector/trackml-detector	10

CPU				GPU		
	Parent process	Duration/ Event [mu-sec]			Parent process	Duration/ Event [mu-sec]
	Container Instantiation	13			Container Instantiation	4
	File reading	3,825,015			File reading	NA
	Clusterization	118,703			Clusterization	NA
	Spacepoint Formation	22,413			Spacepoint Formation	NA
	Clusterization + Spacepoints	141,116			Clusterization + Spacepoints	3,832
	Seeding	5,715,996			Seeding	16,365
	Track param est	32,824			Track param est	365

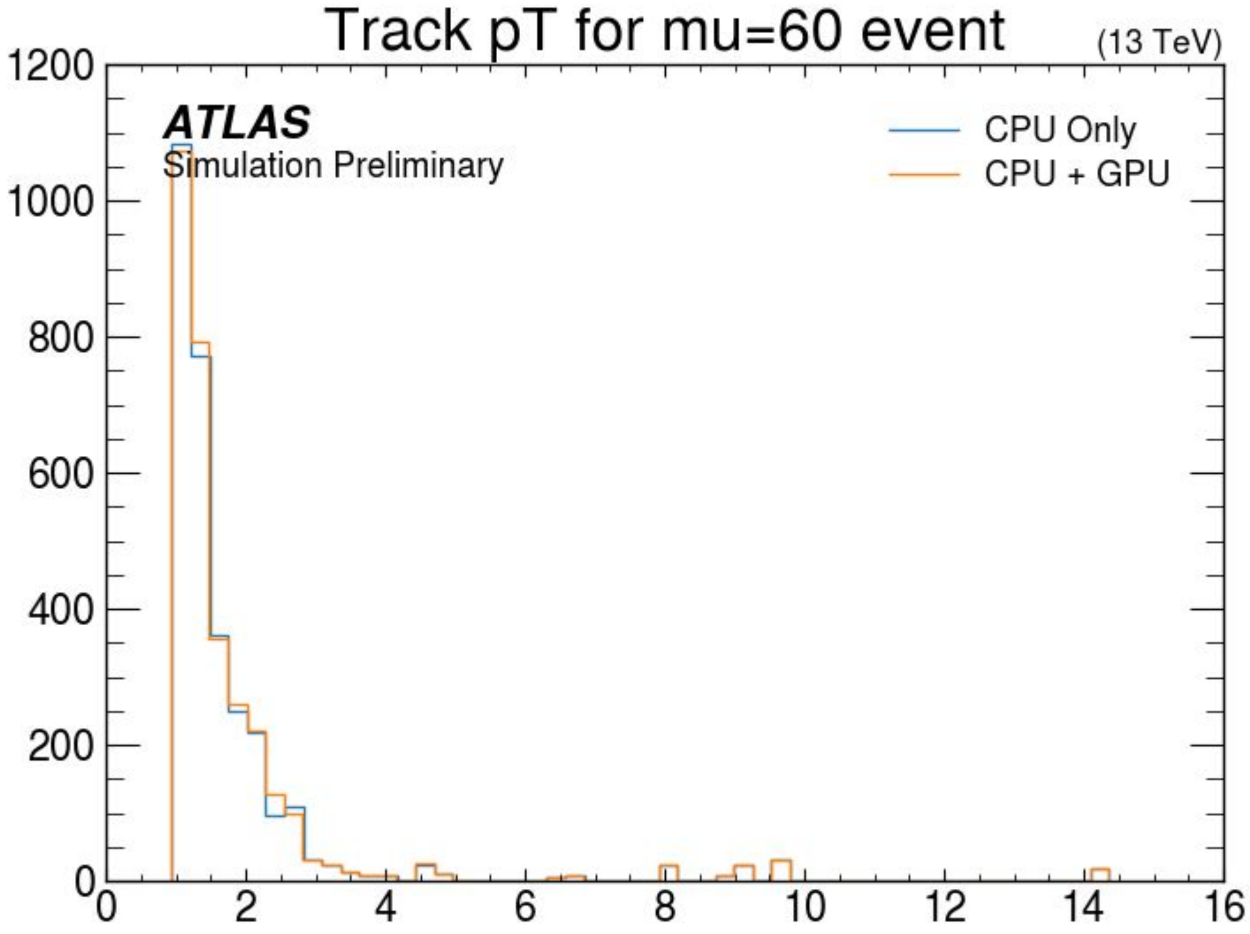
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

TRACCC + ACTS POC Example



POC Track Quality

- Compare POC tracks with tracks produced by ACTS for
  - same detector geometry
  - similar config options
- Plot shows track pT distributions after the track finding step before resolving ambiguities for 1 event
- Distributions checked for 10 example events
  - Distributions roughly correspond
    - POC example produces comparable tracks before ambiguity resolution
- Possible reasons for differences:
  - ACTS methods and TRACCC Device methods do not have 1-1 correspondence
    - Measurements, Seeds, Params are slightly different between the two
  - Minor differences in Config option setups b/w ACTS and TRACCC



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Wall Time Values for POC (mu200)

- POC example runs via a single executable
  - Easier to profile
- ACTS::TrackFinding is the most compute intensive step as expected
- TRACCC::TrackFinding on Device is faster BUT this result is not for the same event or detector geometry
  - TRACCC::TrackFinding measurement comes from a toy example
  - Only meant as a ball-park (placeholder) comparison until TRACCC has a full chain implementation

Data File	Detector Geometry	No. of Events
tml_full/ttbar_mu200	tml_detector/trackml-detector	10

CPU			GPU	
	Parent process	Duration/Event [mu-sec]		Duration/Event [mu-sec]
	Container Instantiation	13		4
	File reading	3,825,015		NA
	Clusterization + Spacepoints	141,116		3,832
	Seeding	5,715,996		16,365
	Track param est	32,824		365
	ACTS::TrackFinding (CKF)	13,501,182		
	TRACCC::TrackFinding (CKF)			NA

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Conclusions

- Profiling tools can provide a lot of useful metrics - Tools are consistently improving
- Apples-Apples comparison of performance of this work requires a corresponding TRACCC::TrackFinding implementation within the full chain example
  - Caveat: ACTS and TRACCC are not 1-1 replicas for Host nor Device implementations
- Case:
  - TrackFinding on Device is slower: A heterogeneous solution could be considered
    - With performance opitmizations
    - Better build integration between TRACCC-ACTS as opposed to brute-forcing the build
  - TrackFinding on Device is faster: The heterogeneous solution could still provide flexibility downstream in terms of net throughput optimization
- Overall, track reconstruction on Device is promising. Heterogeneous operation is a potential solution to consider for further development

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

SMARTHEP

REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

Anomaly Detection for Data Quality Monitoring

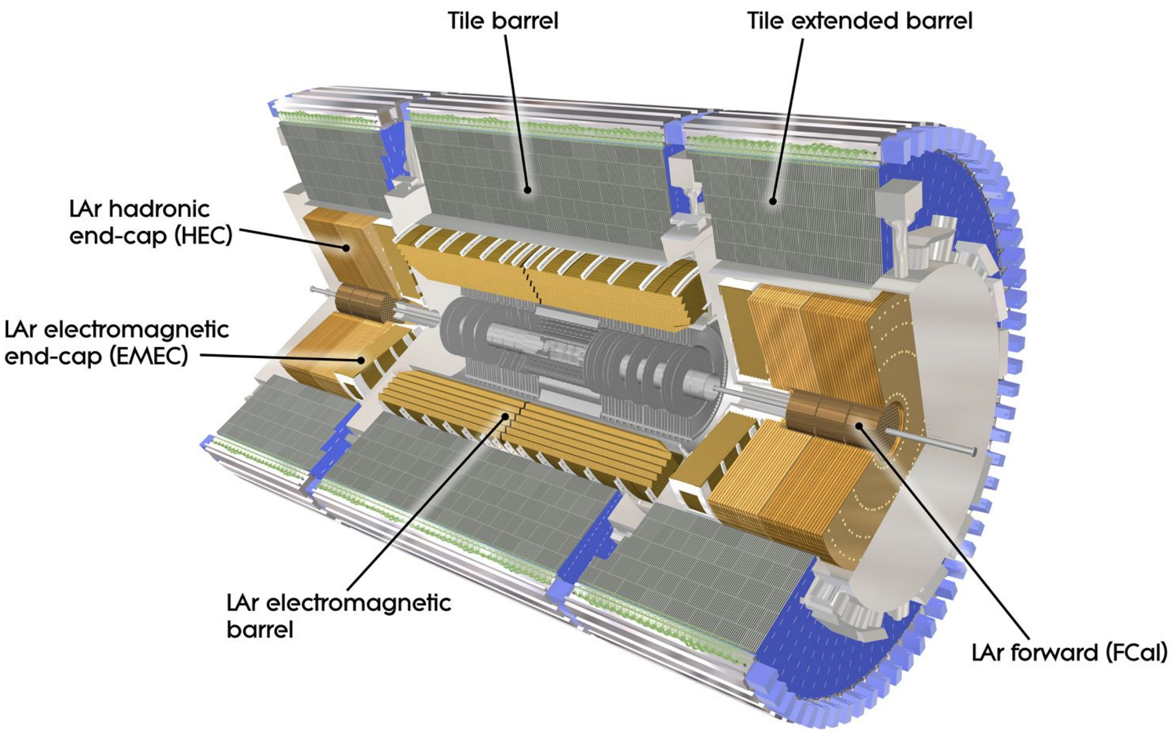
SMARTHEP is funded by the European Union’s Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824

The University of Manchester

Liquid Argon (LAr) Calorimeter

- Particles passing through the LAr create electromagnetic showers, inducing ionization in the liquid argon, which is collected by electrodes under high voltage
- Structure:
  - Divided into four main sections:
    - Electromagnetic Barrel **(EMB)**: Covers $|\eta| < 1.5$.
    - Electromagnetic Endcap Calorimeters **(EMEC)**: Covers $1.4 < |\eta| < 3.2$.
    - Hadronic Endcap Calorimeter **(HEC)**: Covers $1.5 < |\eta| < 3.2$ and uses copper as passive material.
    - Forward Calorimeter **(FCal)**: Covers the high pseudorapidity region ($3.1 < |\eta| < 4.9$), using copper and tungsten.



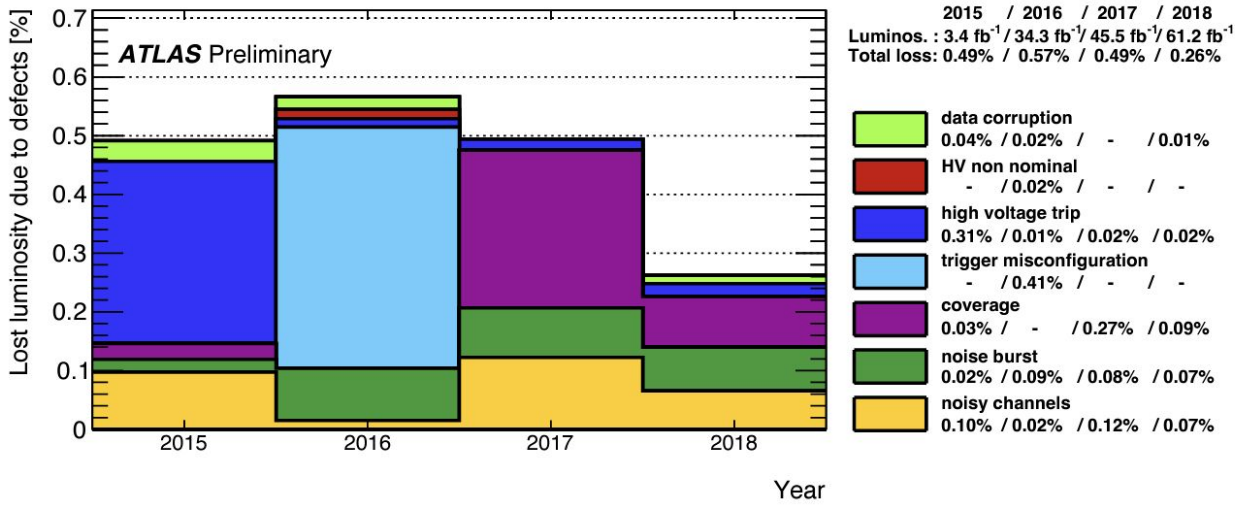
Ref

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

LAr Data Quality Issues

- High Voltage (HV) Trips:
  - Sudden voltage drops, affecting signal collection.
- Data Corruption:
  - Desynchronization errors between FEB and clocks.
- Noisy Channels:
  - Identified during calibration, corrected using neighboring cells.
- Noise Bursts:
  - Correlated with luminosity, detected using LArNoisyRO algorithm.
- Trigger and Coverage Misconfigurations:
  - Misconfigurations leading to reduced data quality.

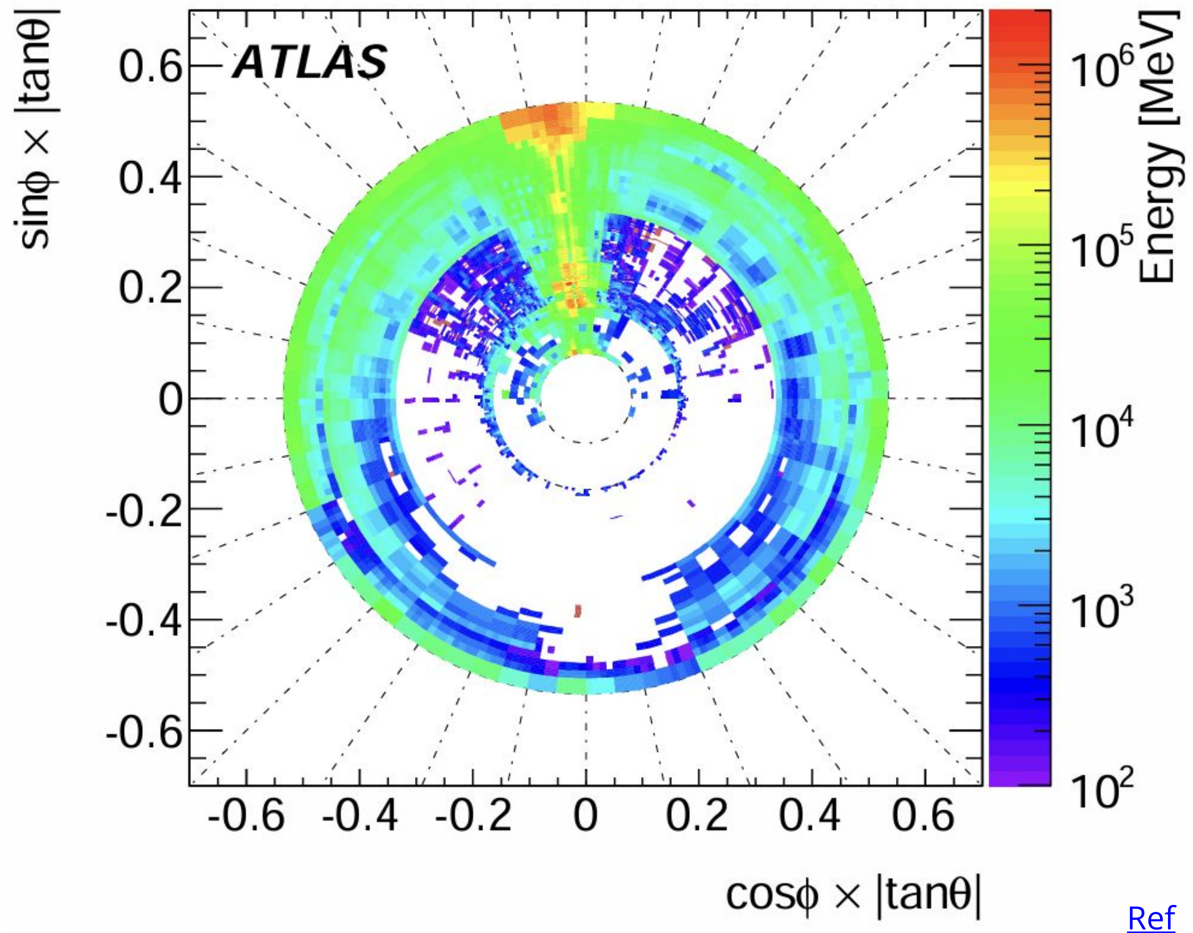


[Ref]

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

LAr Data Quality Issues

- High Voltage (HV) Trips:
  - Sudden voltage drops, affecting signal collection.
- Data Corruption:
  - Desynchronization errors between FEB and clocks.
- Noisy Channels:
  - Identified during calibration, corrected using neighboring cells.
- **Noise Bursts:**
  - Correlated with luminosity, detected using LArNoisyRO algorithm.
- Trigger and Coverage Misconfigurations:
  - Misconfigurations leading to reduced data quality.
- However, there could be other unlabelled detector effects that affect the LArs

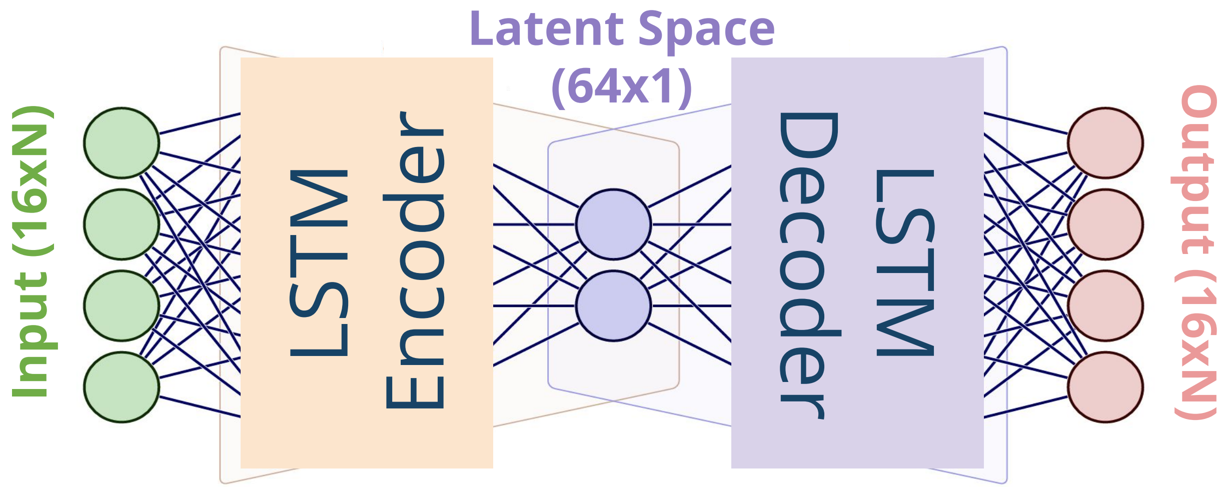


Ref

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Unsupervised AD

- A single algorithm sensitive to all known and unknown LAr issues
- Events do not need to be tagged in most cases since they are usually discarded if any DQM check is not met
- Autoencoder approach:
  - Train on "good" events
    - LumiBlocks with no known flagged issues
  - During inference, detector issues result in high reconstruction loss
    - MSE between AE input and output
- Current setup uses LSTM networks in the encoder and decoder
  - Enables time-series feature extraction



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

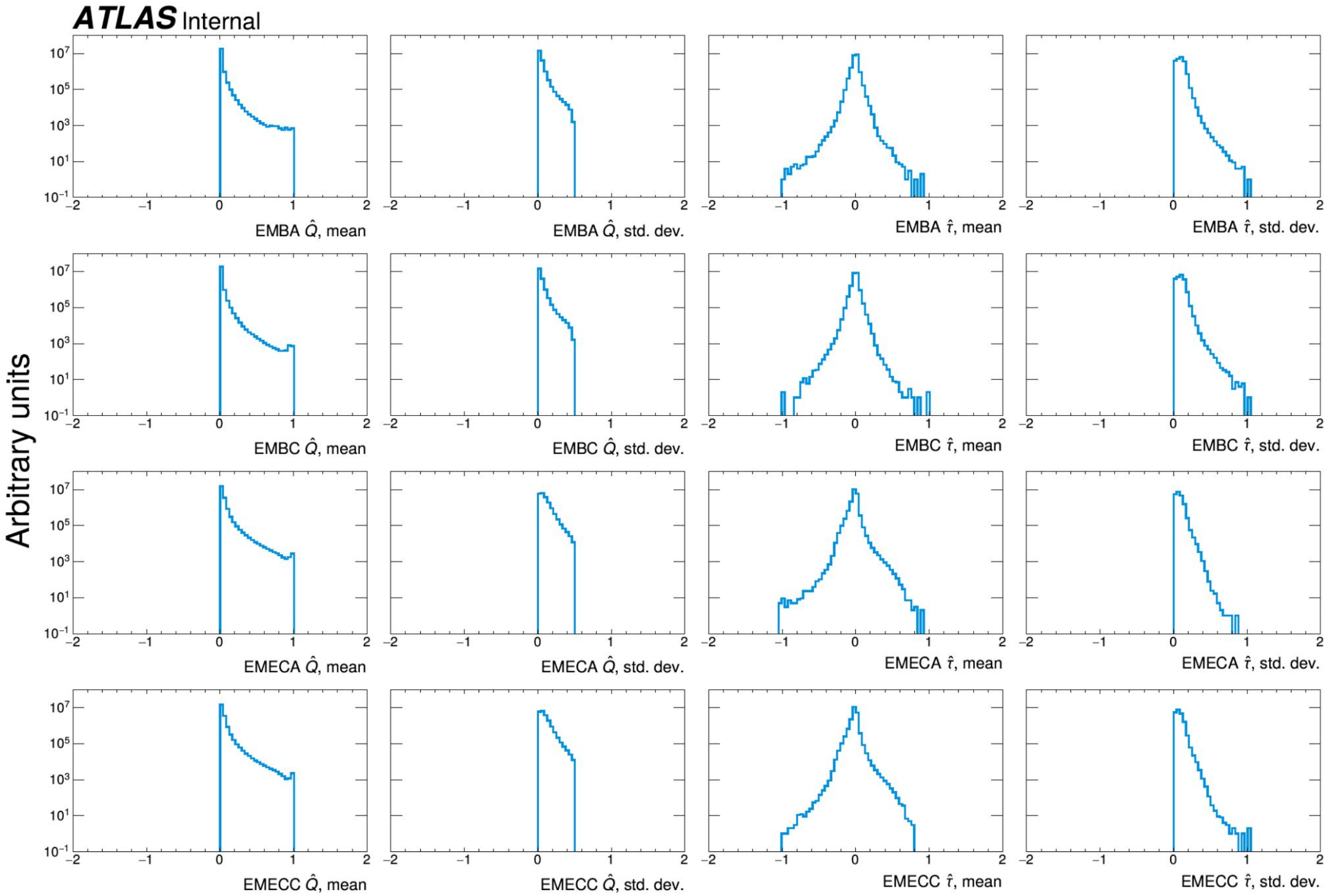
Input Data

- Source of Input Data:
  - The data comes from topocluster moments, which are aggregated features of clusters of calorimeter cells.
  - The two primary topocluster properties used are:
    - Q-factor: Indicates how well the signal pulse shape matches the expected ideal shape.
    - Timing (τ): Refers to the timing of the signal relative to the event, helping detect out-of-time signals or anomalies.
  - For each of these properties, we consider the mean and std. dev as the AE inputs
- Two regions considered for both Barrel and End Cap resp.:
  - Barrel C: $-1.5 \leq \eta \leq 0$
  - Barrel A: $0 < \eta \leq 1.5$
  - Endcap C: $-3.2 \leq \eta < -1.5$
  - Endcap A: $1.5 < \eta \leq 3.2$
- As a result each input point to the AE is 16 dimensional considering p-p collisions

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Input Data

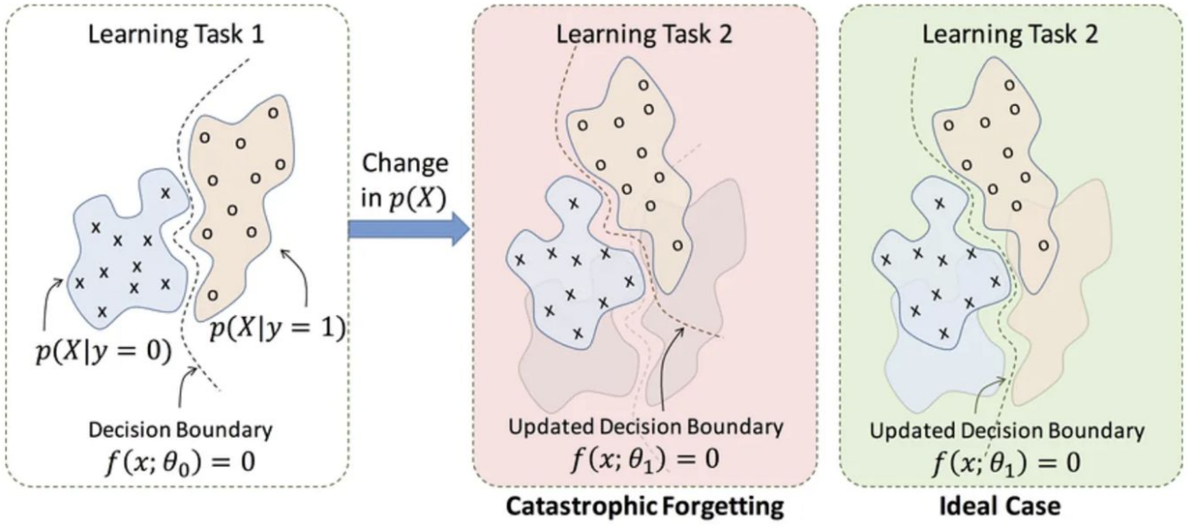
- Source of Input Data:
  - The two primary topocluster properties used are:
    - Q-factor: Indicates how well the signal pulse shape matches the expected ideal shape.
    - Timing (τ): Refers to the timing of the signal relative to the event, helping detect out-of-time signals or anomalies.
  - For each of these properties, we consider the mean and std. dev as the AE inputs
- Two regions considered for both Barrel and End Cap resp.:
- Each input point to the AE is 16 dimensional considering p-p collisions



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Problems with LSTMs

- LSTMs are known to suffer from the catastrophic forgetting phenomenon
  - Evident when network trained on p-p collisions is then trained on Heavy Ion data
    - Fixed by small tweak in the code disconnecting mem gates for both tasks
- Sequence length suitable for LSTM-AE is considerably small
- LSTMs are also very memory heavy
  - Intermediate contexts need to be stored for backprop.
- LSTMs are hard to parallelize

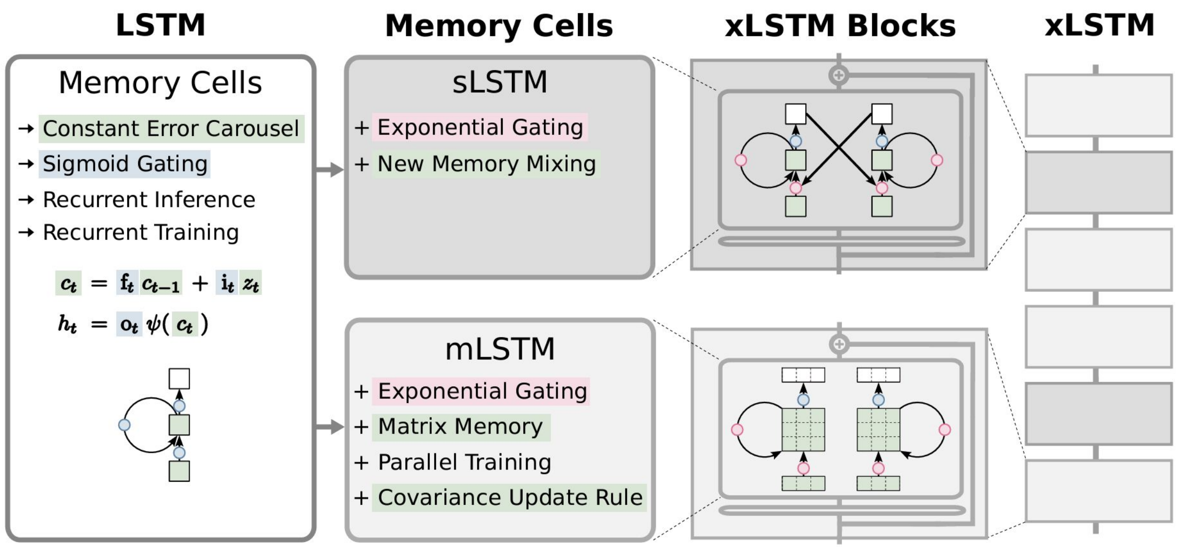


Ref

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

xLSTM

- Motivation: solves catastrophic forgetting
  - Specialized memory handling
- Produces richer hidden representations of much longer sequences
  - More sensitive to harder to find detector issues
- Still memory intensive and non-parallelizable
  - Train a student network to predict xLSTM loss
  - Accuracy gained over the baseline LSTM is traded off in the student network for speed
- Current status: xLSTM implementation and code testing done
  - Repeat tests using same dataset
  - Merge events from dataset to form larger input sequences and compare LSTM-xLSTM

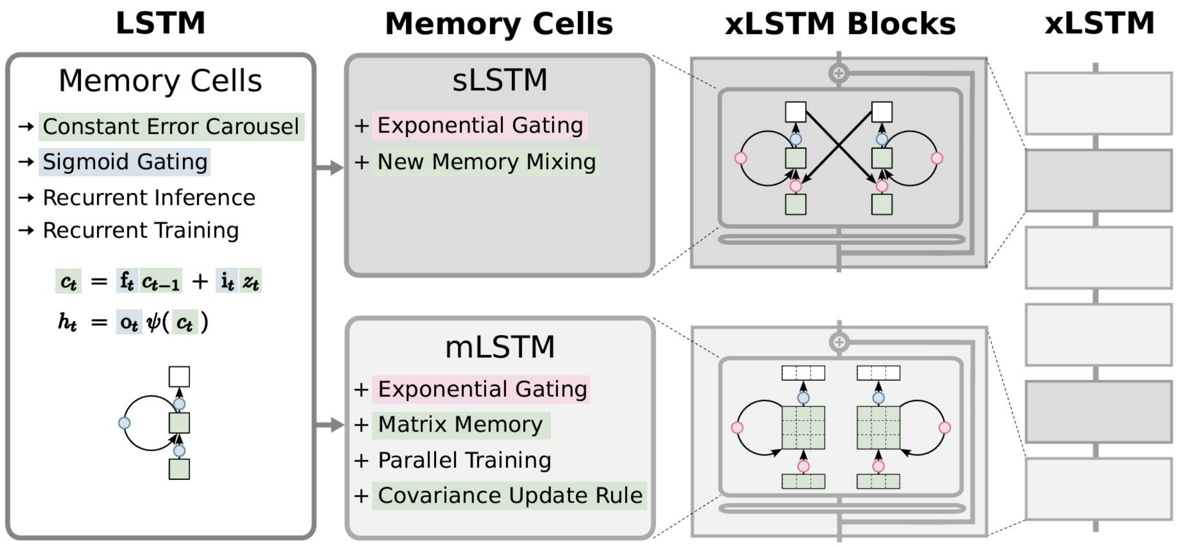


Ref

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

xLSTM

- xLSTM is a better 1-1 comparison to attention based models such as transformers
  - (potentially what Laura might look at with the same dataset)
- Better memory management
  - The architecture of XLSTM allows it to allocate memory more effectively, improving performance on tasks that require long-term sequence retention.
  - It can selectively forget less useful information while preserving key details for future use.
- However xLSTM inference is much more compute intensive than LSTM
  - KD is essential!



Ref

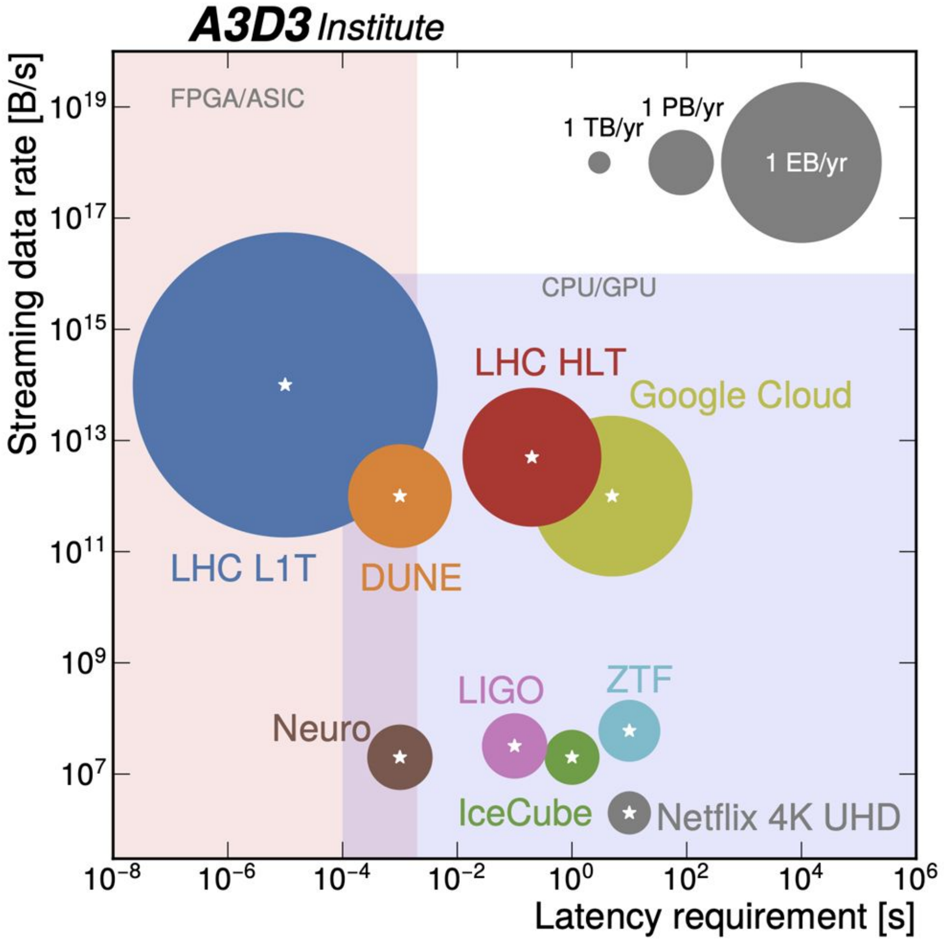
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Secondment: Knowledge is Overrated

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Fast Inference (EdgeML '24)

- High demand in HEP and many other fields for:
  - Fast execution of algorithms
    - Low latency
    - Low compute
    - Low power
    - Low memory
- Ideally without losing performance on the task
- Something that wasn't spoken about in much detail:
  - Fast yet **SUSTAINABLE!**
    - Low power IS fast (FastML '23)



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Fast ML Inference (EdgeML '24)

SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

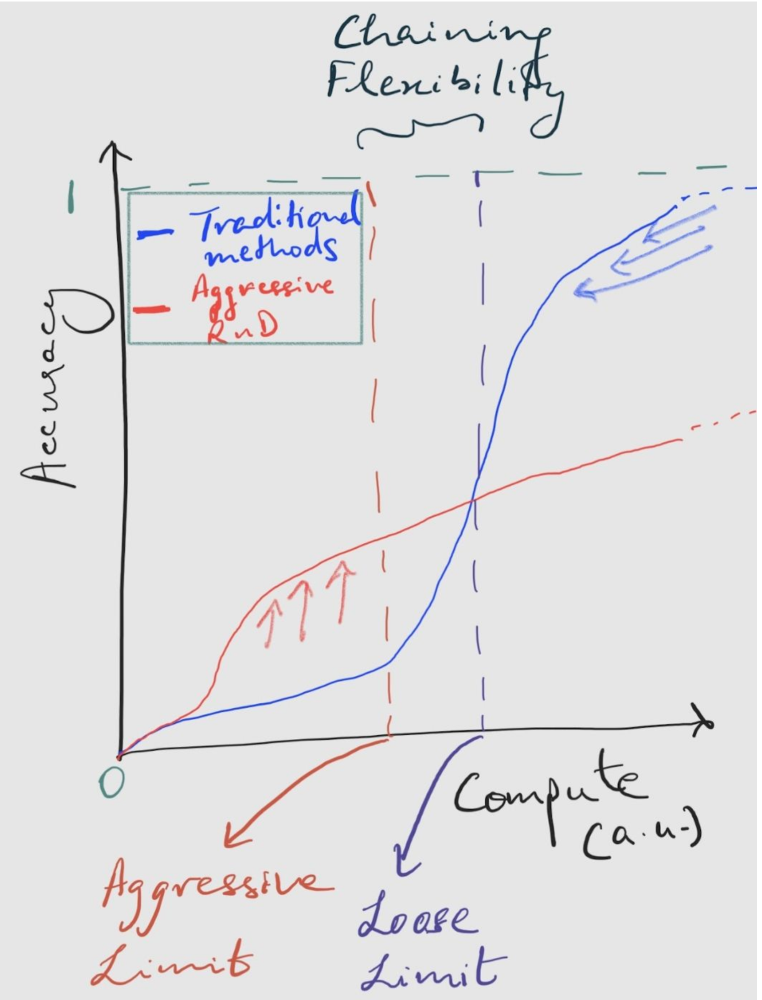
- Use specialized hardware
  - GPUs
  - FPGAs
    - Requires special model management driven by hardware specs
      - Pruning
      - Quantization
  - ASICs
  - NPUs
- Knowledge Distillation
  - A large network is trained on the required task
  - A much smaller network is trained to predict the loss of the larger one
  - Only the smaller network is deployed on the Edge device

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824
The University of Manchester

What if…

- We could distill a larger network like in KD but regress to:
  - An arbitrary metric as opposed to loss of a complicated network (loss landscapes can be extremely complex high-dimensional manifolds themselves)
    - Would enable significantly larger inference-size reduction
  - While being:
    - Verifiable
    - Scalable
- Trade off some more performance for… speeeeeed
  - Quantize not just weights but also inputs
- The pareto line still lies at the {accuracy lost - speedup line}
  - But now instead of
    - pushing the line down with resistance from accuracy loss
  - We are:
    - pushing the line up with resistance from loss in speedup



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Project Title Proposals

- Project Status:
  - Built codebase with toy MNIST examples
  - Identified ways of calculating net amount of computations required for inference
    - Needs improving
  - Need to design tests using HEP data and tasks
- “Knowledge is overrated: Fast ML Inference”
  - Knowledge: richness of the learned prior
    - i.e. how descriptive the algo is
      - NNs are designed to give as rich prior approximations as possible
  - Over {Rated} : rate of compute
    - We don't want a high compute rate
  - So we are trading away knowledge for lower compute rates (high speed)
- “DUMBHEP ”
  - Trading away knowledge makes the algorithm inherently “dumb”

SMARTHEP is funded by the European Union’s Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

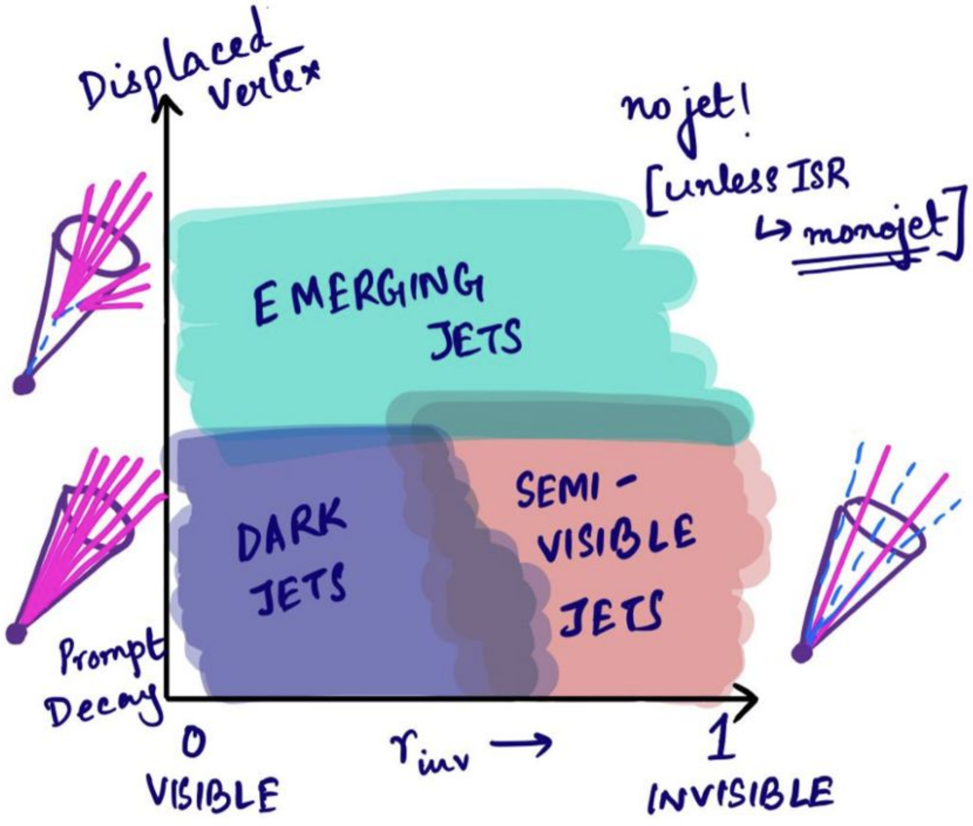
Other Activities

SMARTHEP is funded by the European Union’s Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824
The University of Manchester

Quests

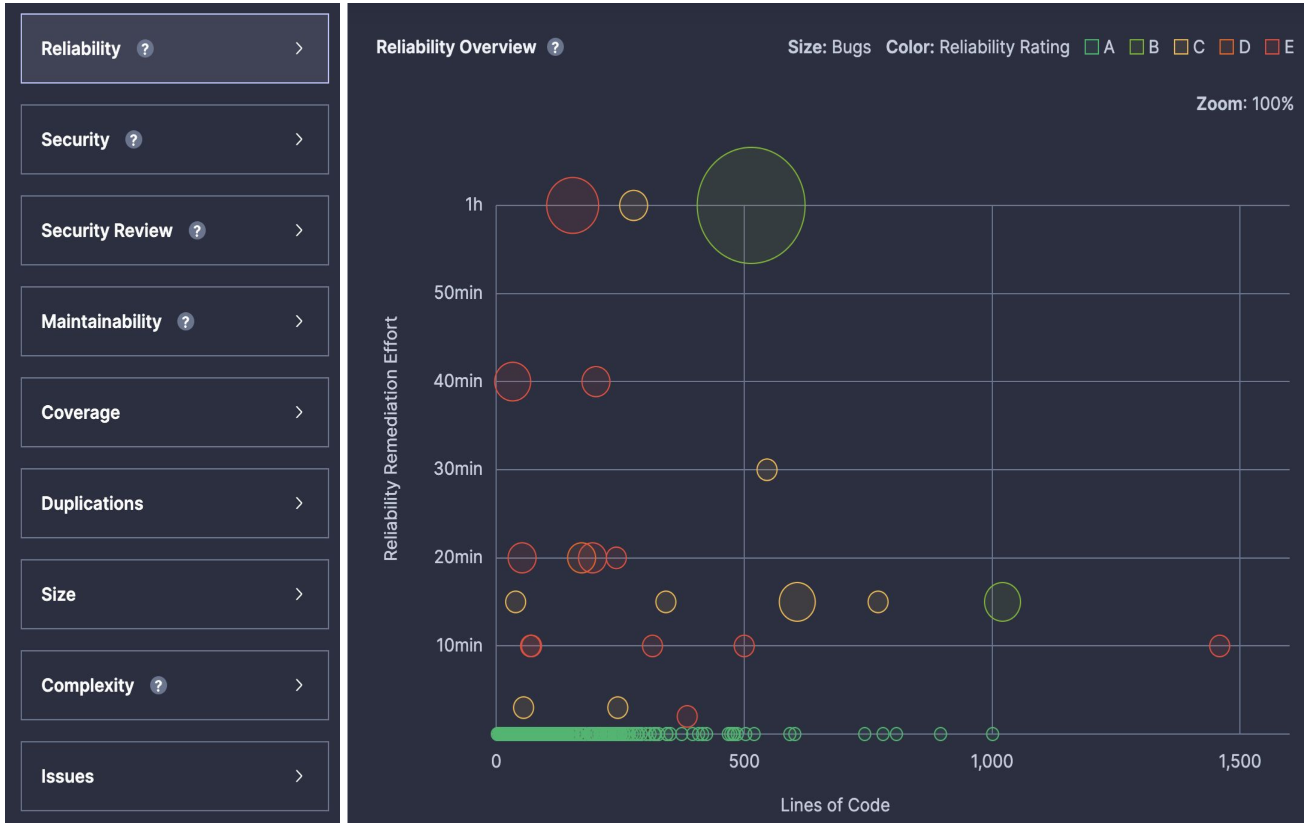
- Analysis: SVJ+leptons (potential DM signature)
  - Jets with MET aligned along jet axis
- Consulting for AD at the Trigger for new physics
  - VAE based approach inspired by AXOL1TL (CMS)
- Pheno project: AD using richer bkg representations by combining generator tunes



SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

Side Quests

- EVERSE Project: Software sustainability
  - WP4 pilot (ACTS)
    - Used static analysers (SonarCloud) to extract code quality metrics
- Used static analysis to help identify code inefficiencies and reduce cyclomatic complexity in:
  - GAPS: GPU-Amplified Parton Showers
  - Project in UniMan theory Dept.



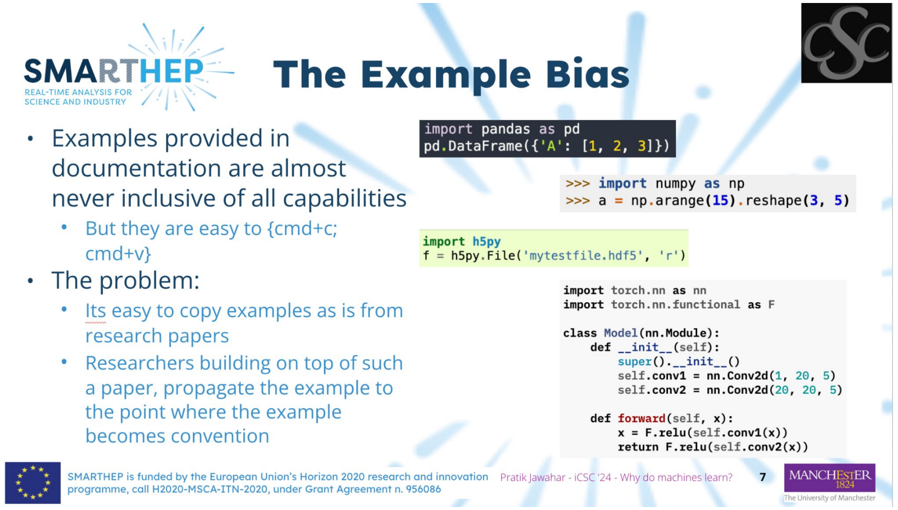
SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

The University of Manchester

Side Quests

- Taught at the iCSC
  - A fundamentals of ML lecture titled “Why do Machines Learn”
    - Dealt with typical misconceptions at every step of a traditional ML pipeline
    - Introduced a partially new idea called example bias
      - Documentation biases people in the way they perceive code
    - Introduced fundamental theoretical ML research via
      - Geometric DL
      - Categorical DL
      - Search for a “Theory of Everything ML”
- Anthology (10/17) + EP (3/7)

58% 42%


SMARTHEP
REAL-TIME ANALYSIS FOR SCIENCE AND INDUSTRY

The Example Bias

- Examples provided in documentation are almost never inclusive of all capabilities
  - But they are easy to {cmd+c; cmd+v}
- The problem:
  - Its easy to copy examples as is from research papers
  - Researchers building on top of such a paper, propagate the example to the point where the example becomes convention

```
import pandas as pd
pd.DataFrame({'A': [1, 2, 3]})
```

```
>>> import numpy as np
>>> a = np.arange(15).reshape(3, 5)
```

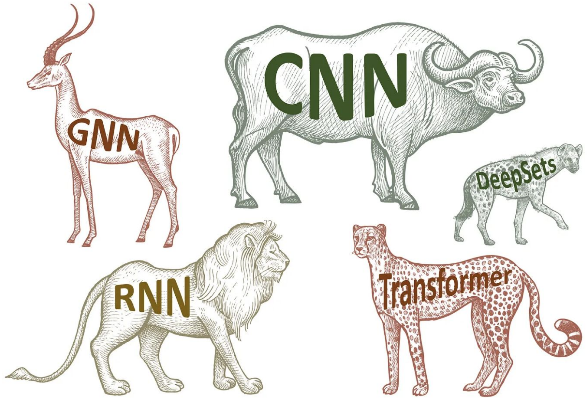
```
import h5py
f = h5py.File('mytestfile.hdf5', 'r')
```

```
import torch.nn as nn
import torch.nn.functional as F

class Model(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(1, 20, 5)
        self.conv2 = nn.Conv2d(20, 20, 5)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        return F.relu(self.conv2(x))
```

SMARTHEP is funded by the European Union's Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086 Pratik Jawahar - iCSC '24 - Why do machines learn? 7 MANCHESTER 1824 The University of Manchester



Deep learning today: a zoo of architectures, few unifying principles. Animal images: ShutterStock.

SMARTHEP is funded by the European Union’s Horizon 2020 research and innovation programme, call H2020-MSCA-ITN-2020, under Grant Agreement n. 956086

MANCHESTER 1824 The University of Manchester

Thank you!