



NextGen
Next Generation Triggers

MLOps Pipeline for Continuous Deployment of Machine Learning Algorithms in the CMS Level-1 Trigger

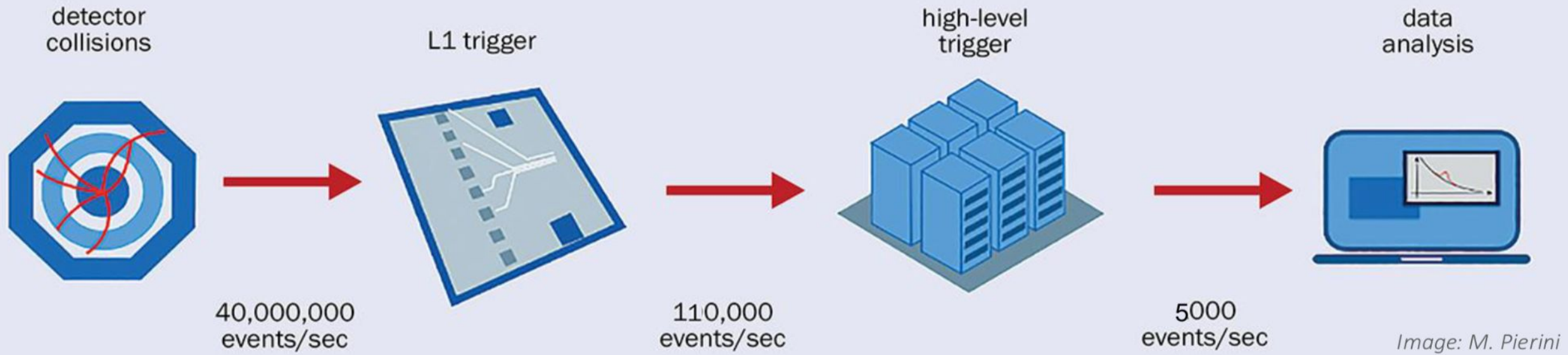
Maciej Głowacki on behalf of the CMS collaboration
ACAT 2025, Hamburg



NextGen
Next Generation Triggers



Introduction to CMS Trigger



The CMS experiment at the LHC deploys a two-step trigger system to filter a 40 MHz proton-proton collision rates down to 100 KHz for offline analysis.

The Level 1 Trigger (L1T) ^[1] filters **99.75%** of collision events
If we don't identify interesting events in trigger, we lose them forever!

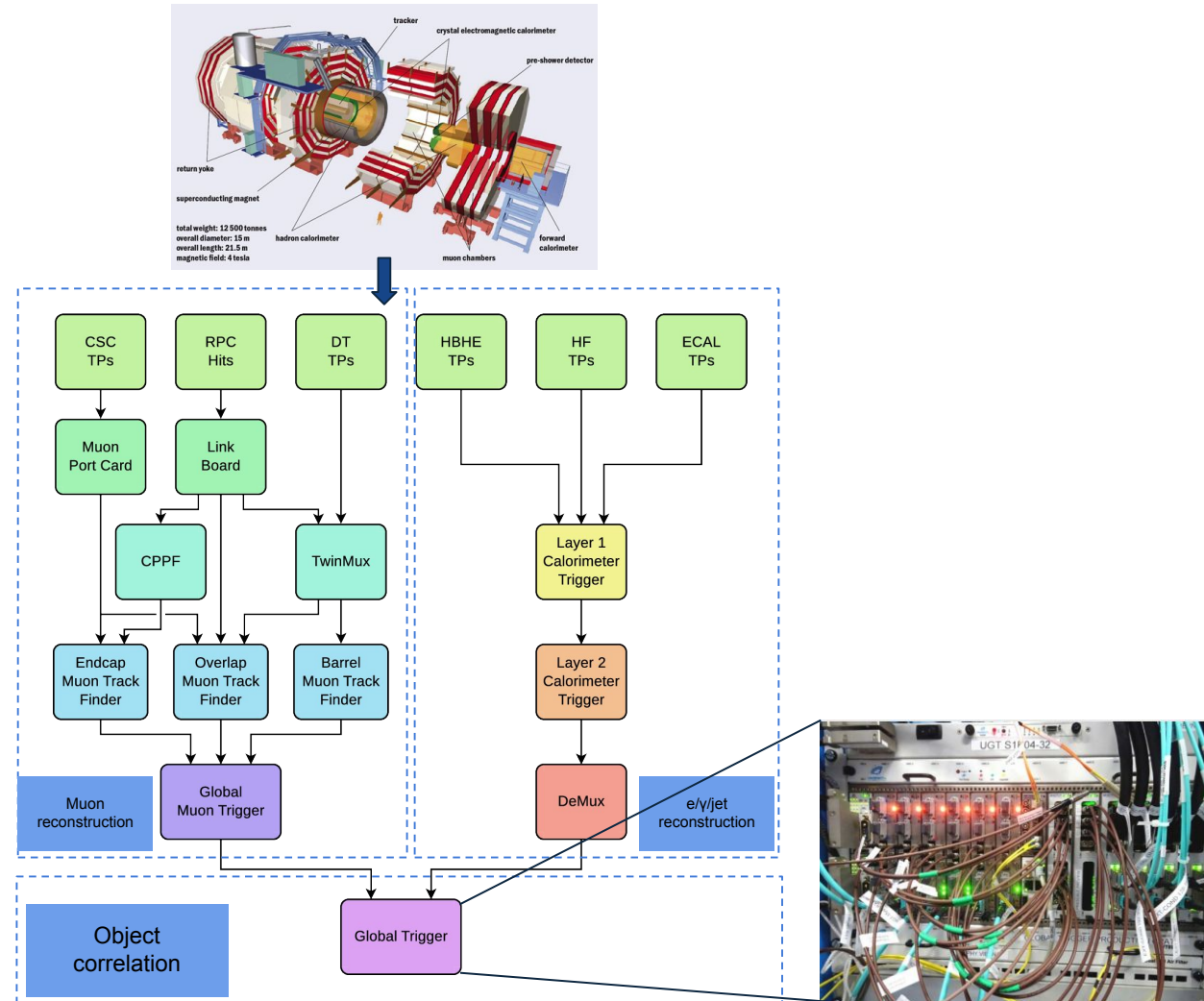
Introduction to Level 1 Trigger

The L1T receives condensed event information in the form of trigger primitives.

Dedicated subsystem modules reconstruct physics objects from varying detectors and/or regions.

L1T is an all FPGA design hosted on custom boards interconnected via GB/s optical links.

The final L1T decision (L1T accept/reject) is propagated to the Data Acquisition System (DAQ).



6 MP7^[2] cards hosting
Xilinx Virtex 7 FPGA

Introduction to Level 1 Trigger

How is the L1T decision generated?

- The Level 1 Accept/Reject is an “OR” operation between all seeds in the L1T menu.
- L1T menu: set of handcrafted conditions to select events which align with our physics priorities (as well as balancing efficiency to selecting signal versus the total data rate).

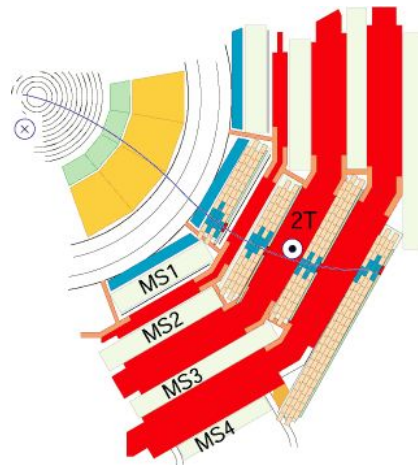
Seed	Condition	Rate (kHz)	signal efficiency for process X (%)
Double Jet	$P_T(1), P_T(20) > 110, 90 \text{ GeV}$	20	50%
Single Electron	$P_T > 10 \text{ GeV}$	15	45%
ML algorithm	Cut on output distribution	10	40%

Machine Learning at L1T

ML at the L1T since ~2016.

- Machine Learning is currently deployed to expand L1T acceptance while maintaining a manageable total data rate.
- P_T measurement from muon tracks and anomaly detection at varying levels of trigger reconstruction.

Overlap Muon Track Finder
Muon Energy (P_T) regression [3]

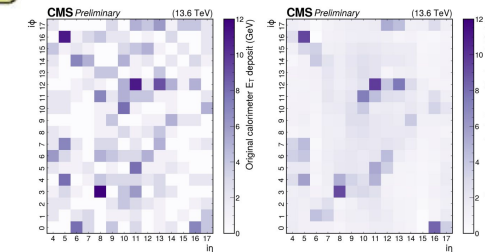


Naive Bayes principle to extract energy measurement from tracks in Muon stations.

$$P(\Delta\phi_l \mid p_T, \text{refLayer}, \text{layer})$$

Layer 1 Calorimeter Trigger

CICADA Anomaly detection [4]



Unsupervised algorithm trained to detect anomalous energy deposits in the calorimeter.

Global Trigger

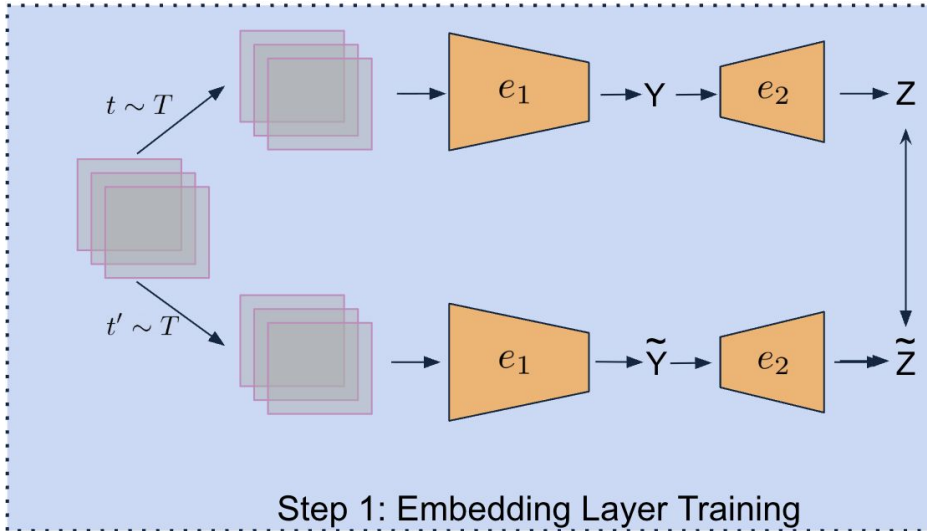
AXOL1TL Anomaly detection [5]

Unsupervised algorithm trained on all accessible reconstructed objects in the Global Trigger to detect anomalous cross-object correlations

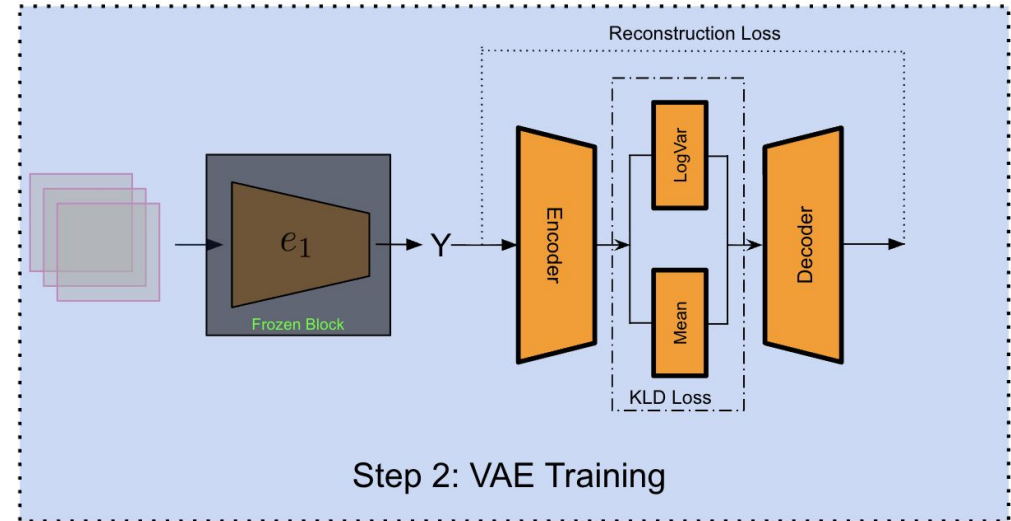


AXOL1TL

- AXOL1TL is deployed in the Global Trigger as Variational Autoencoder (VAE), and takes as input all available reconstructed objects: (P_T , η , ϕ) of 4 e/γ , 4 μ , 10 jets, and $P_T(\text{miss})$.
- AXOL1TL is a two-step algorithm which firstly embeds incoming events into a **representation space** on which anomaly detection is performed.



Step 1: Pre-training of encoder block using augmented views of the same event and the VICREG ^[6] loss.



Step 2: Train VAE on encoder output:

$$S_{\text{Anomaly}} = \text{MSE}(X, \tilde{X}) = \frac{1}{n} \sum_{i=1}^n (x_i - \tilde{x}_i)^2$$

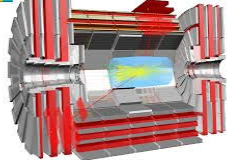
Deployment of



AXOLITL

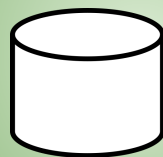


CMS Experiment at the LHC, CERN
Data recorded: 2015-04-16 10:22:56.0000 GMT
Run: 271523.000 - 271523.000

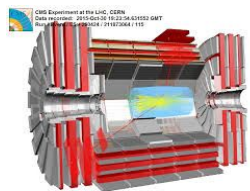


Entry Point

Storage



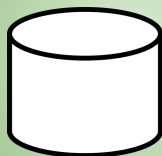
Deployment of AXOLITL



Entry Point



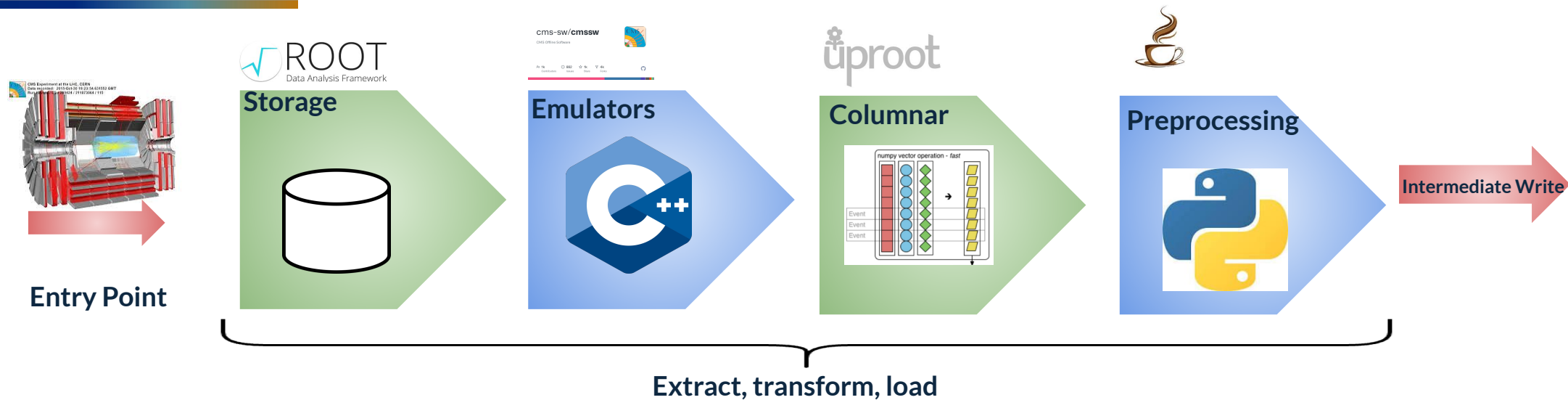
Storage



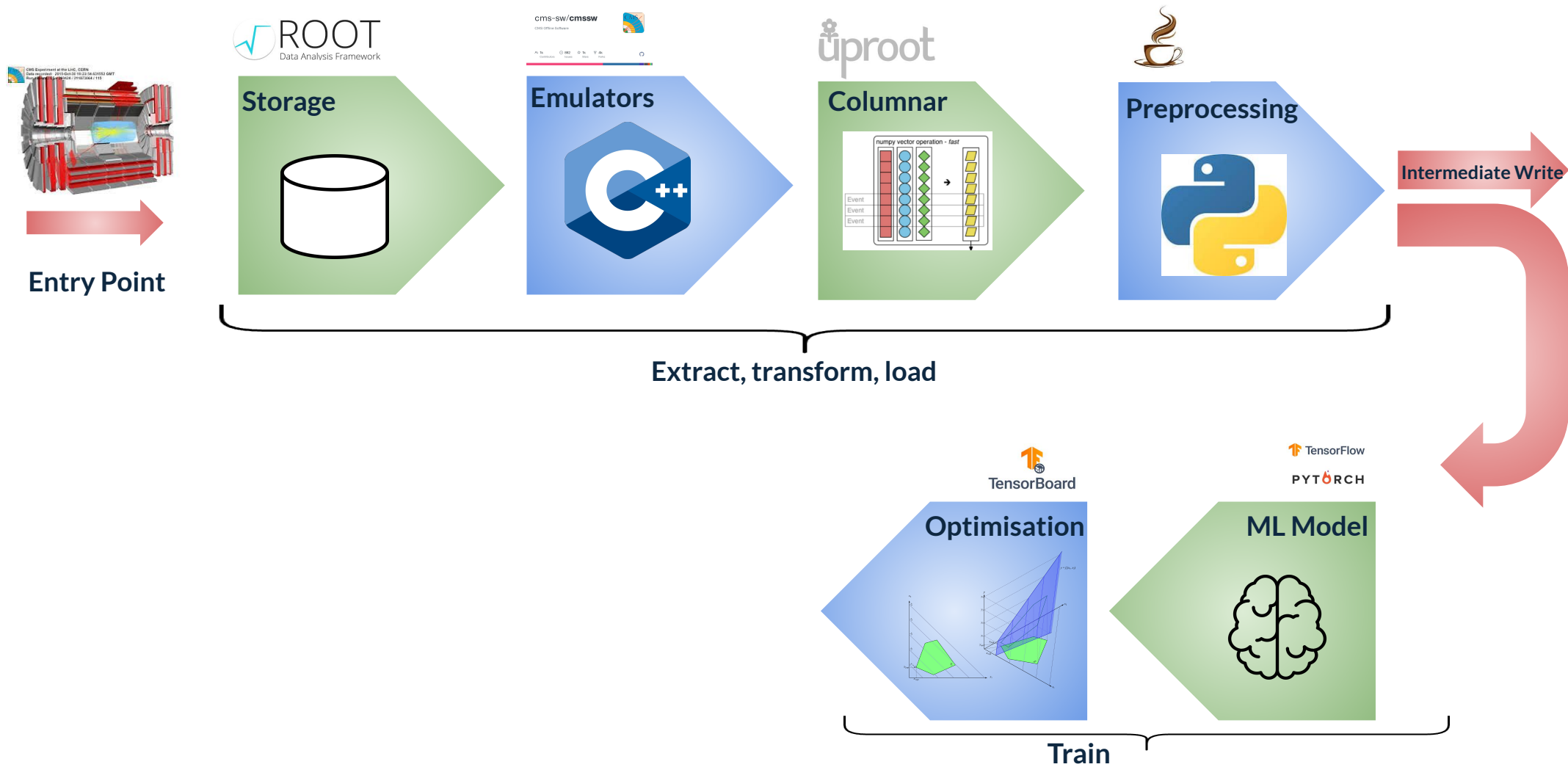
Emulators



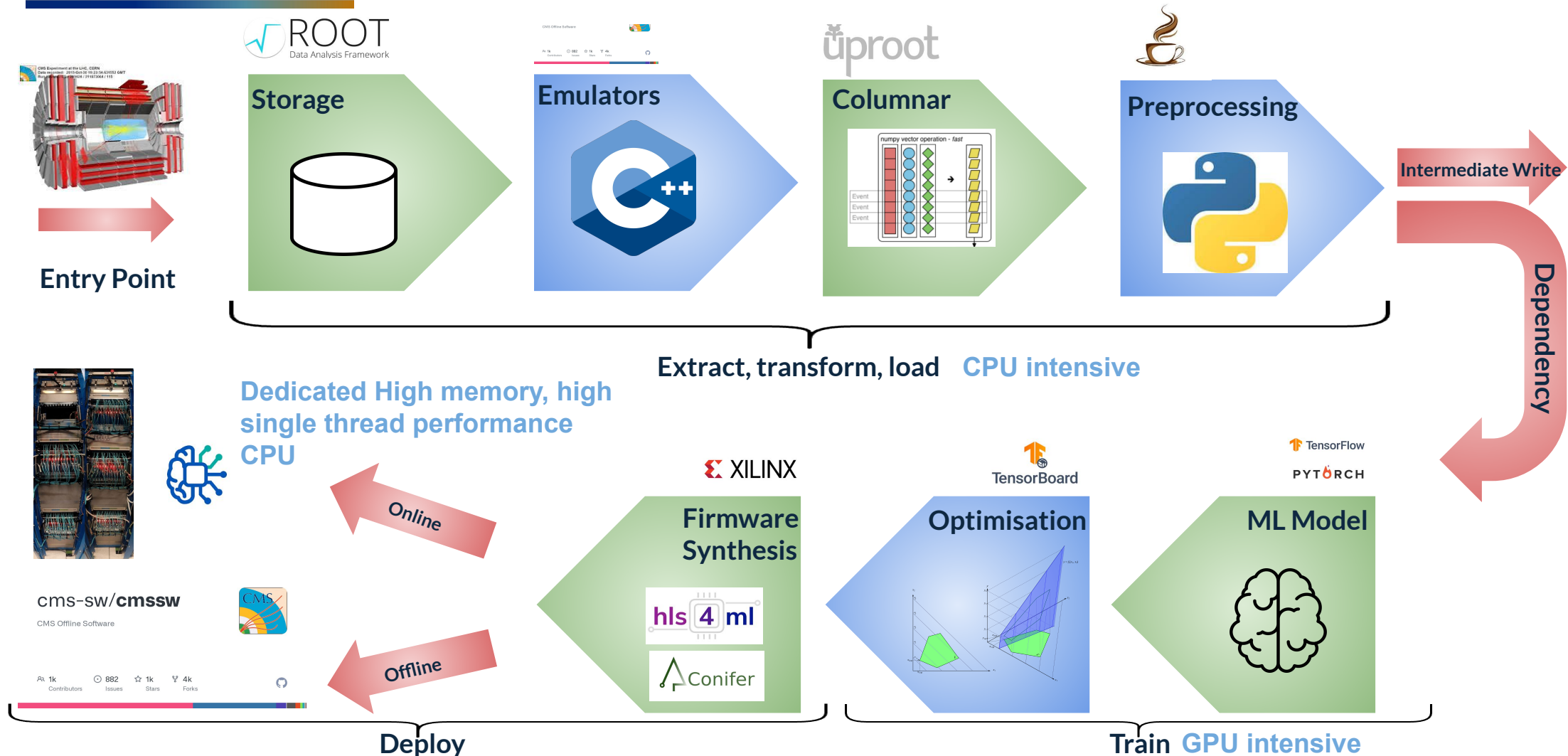
Deployment of AXOLITL



Deployment of AXOLITL



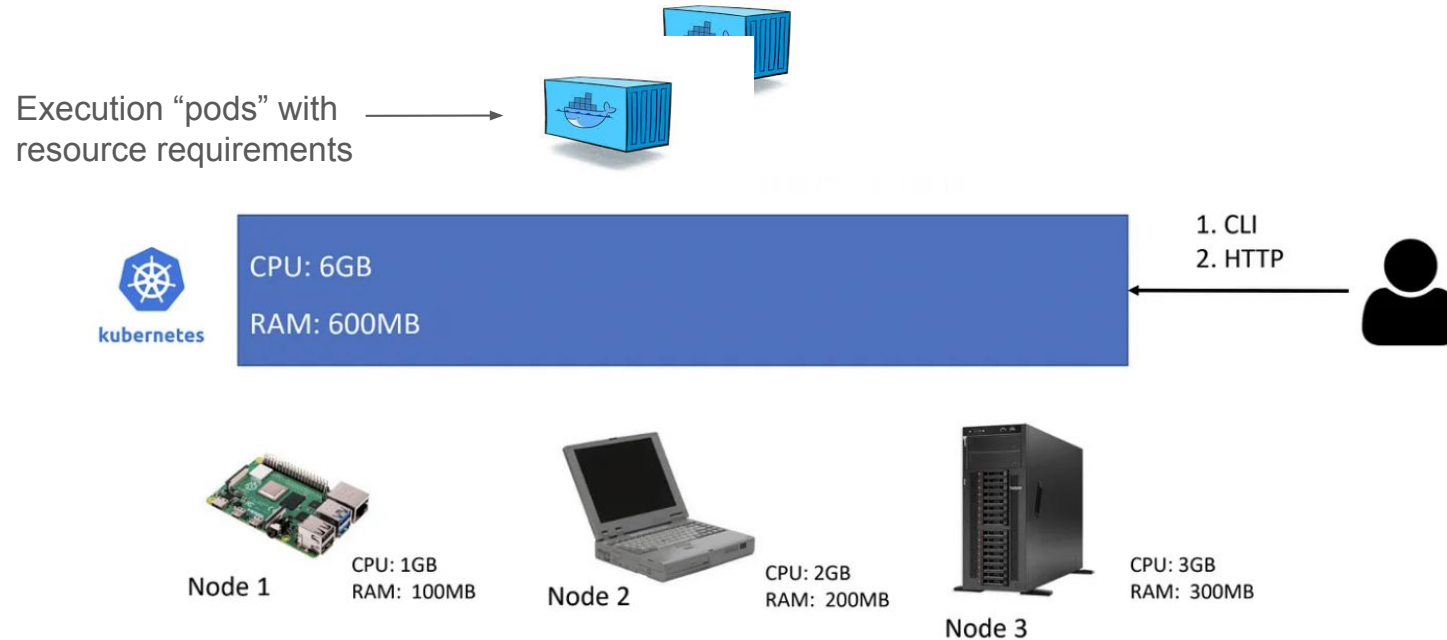
Deployment of AXOLITL



Kubernetes


Big data processing, Model training, Firmware synthesis.

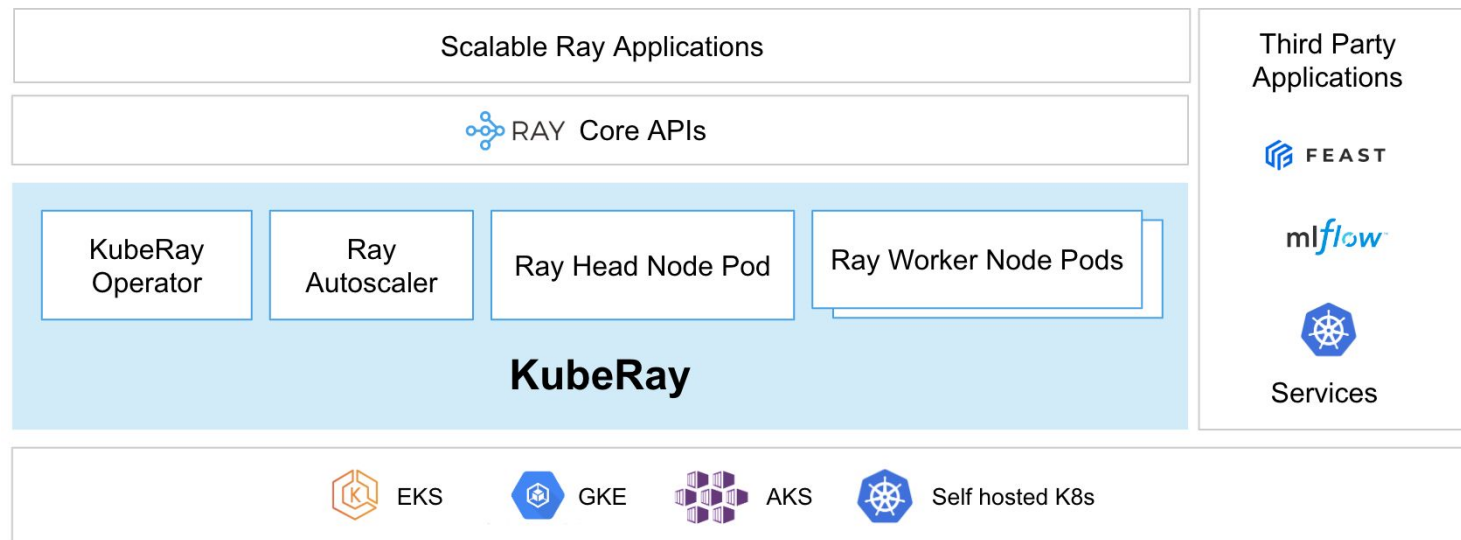
- Complete deployment pipelines consist of data processing, model training and firmware synthesis demands **heterogeneous** compute.
- Kubernetes is used to orchestrate the underlying hardware, provisioning resources dependent on pipeline component.



Kubernetes Operators

Please, no more YAML!

- A Kubernetes Operator is a custom controller that manages applications. Once deployed (e.g., with Helm), users interact with applications from their code.
-  **AXOLITL** uses the Ray Operator to scale out HPO to multiple GPUs.



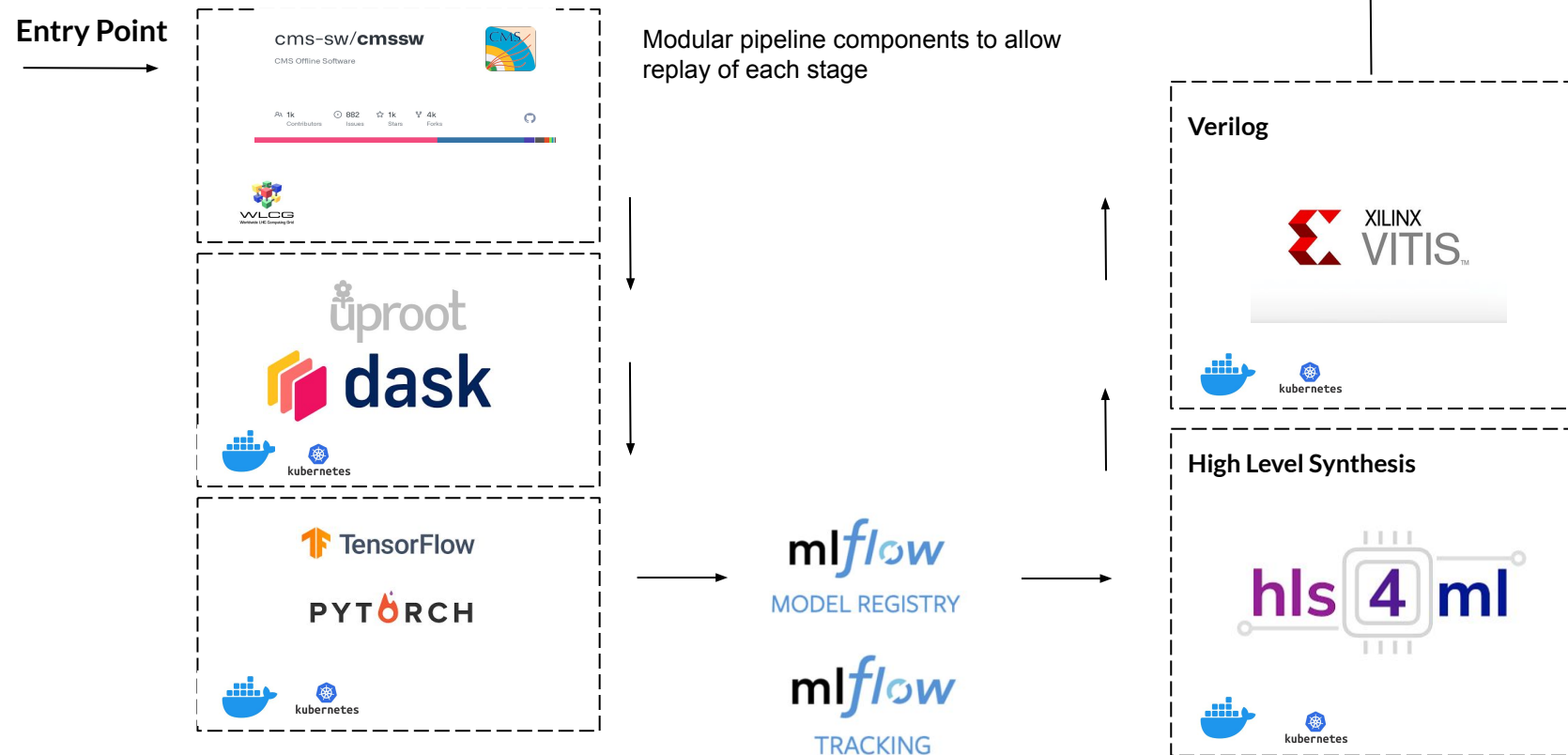
```
search_alg = OptunaSearch(  
    space=search_space,  
    metric='raw-pure/haa-4b-ma15-POWHEG',  
    mode='max'  
)  
  
scheduler = ASHAScheduler(  
    metric='raw-pure/haa-4b-ma15-POWHEG',  
    mode='max',  
    max_t=480,  
    grace_period=32,  
    reduction_factor=2  
)
```

```
analysis = tune.run(  
    train_wrapper,  
    config={},  
    search_alg=search_alg,  
    scheduler=scheduler,  
    storage_path='/axovol/raytune',  
    num_samples=1000,  
    resources_per_trial={'cpu': 5, 'gpu': 0.5},  
    keep_checkpoints_num=1,  
    checkpoint_score_attr='raw-pure/haa-4b-ma15-POWHEG',  
    reuse_actors=False  
)
```

Service deployed and managed at CERN by NGT WP1.1

Monitoring & Versioning

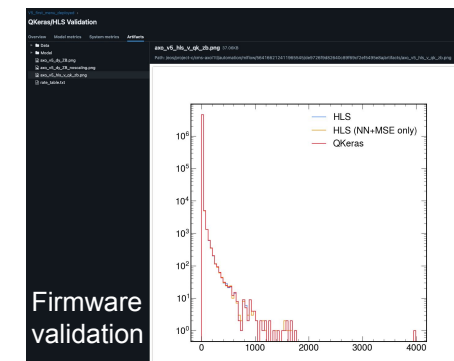
- MLFlow used as the centralised monitoring and versioning system.
- Ensures data, model, firmware provenance throughout.



Parameters (3)

Parameter	Value
completion_status	Success
monitoring_url	https://monit-grafana.cern.ch/d/cmsTMDetail/cms-task-m-task=260422_074002%3Aamlglowac_crab_haa-4b-ma1
workflow_directory	crab_projects_mc_run3/crab_haa-4b-ma15-POWHEGL

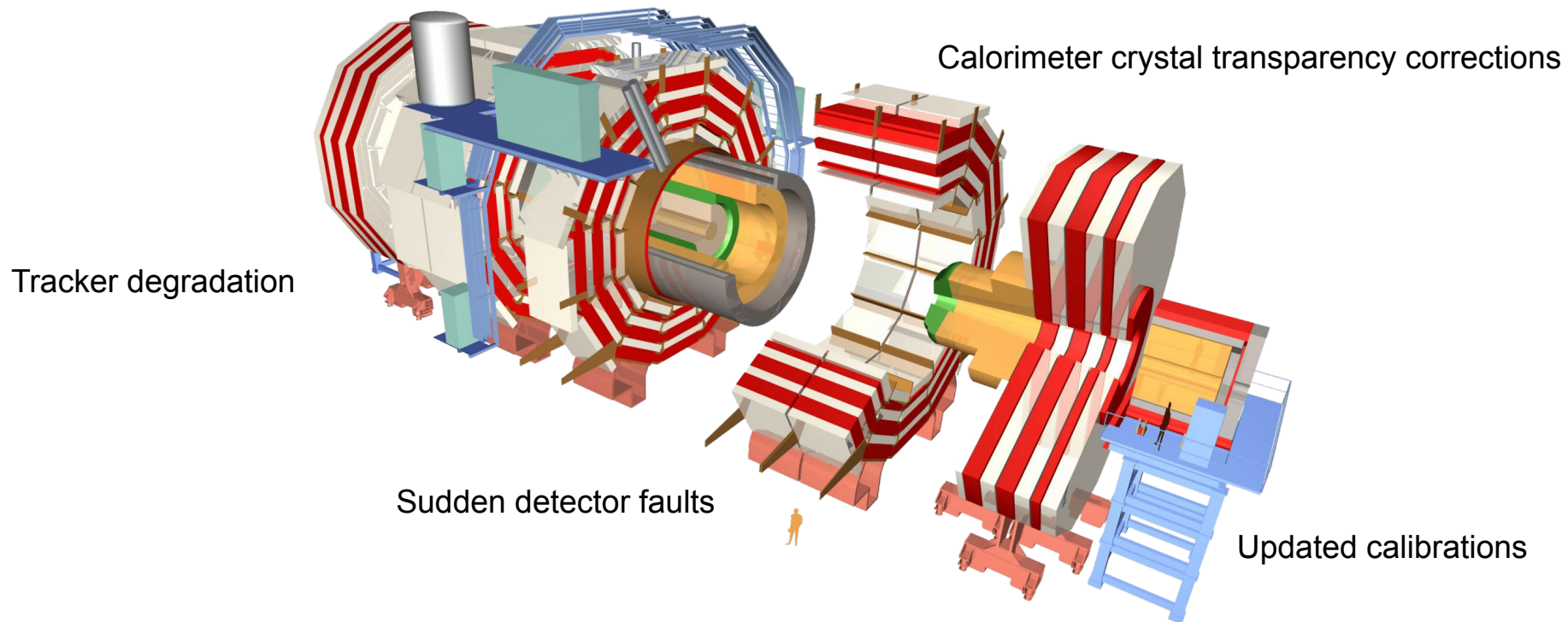
GRID job submission



ML in Evolving Environments

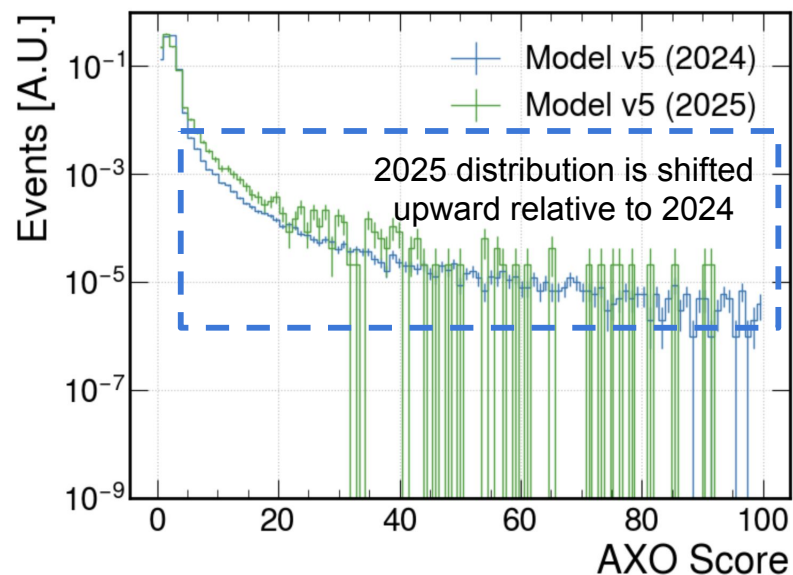
The challenges of data drift.

- The CMS detector is an example of a changing environment;
 - changes can be sudden and unexpected or evolve as a function of time.
 - Models will encounter changing conditions w.r.t. training time.

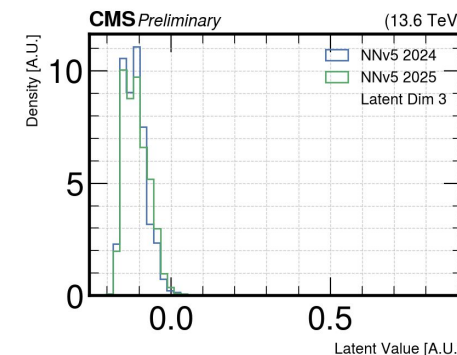
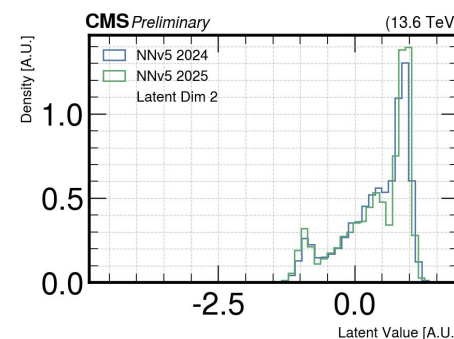
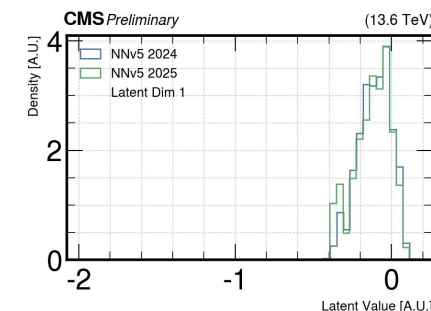
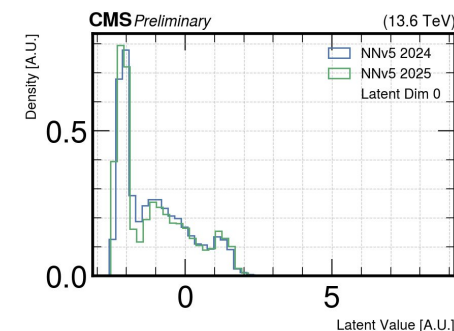


A case study of changing environments with AD trigger.

- Two data taking period studied: 2024 (used for training) and 2025
 - Between these periods, several electromagnetic calorimeter settings and object calibrations were updated.



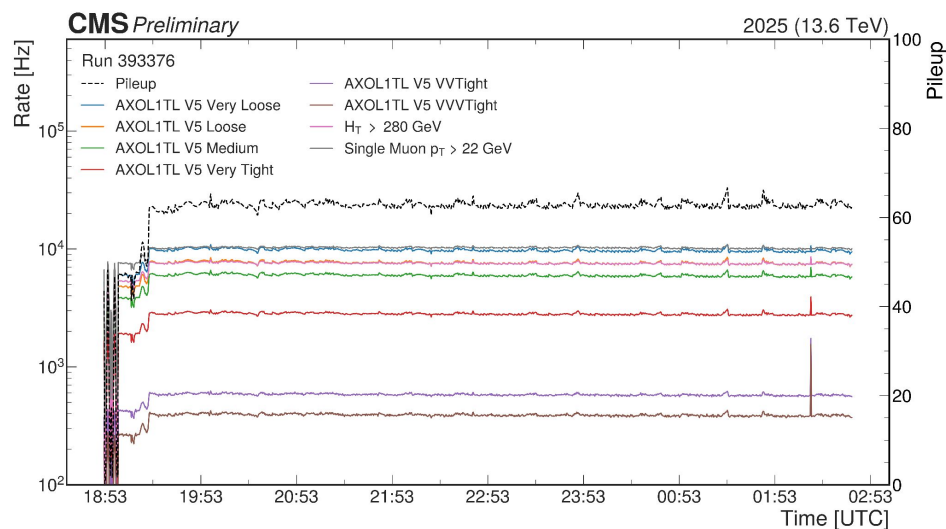
Output distribution of AXOL1TL NNv5 for 2024 data and 2025 data^[5].



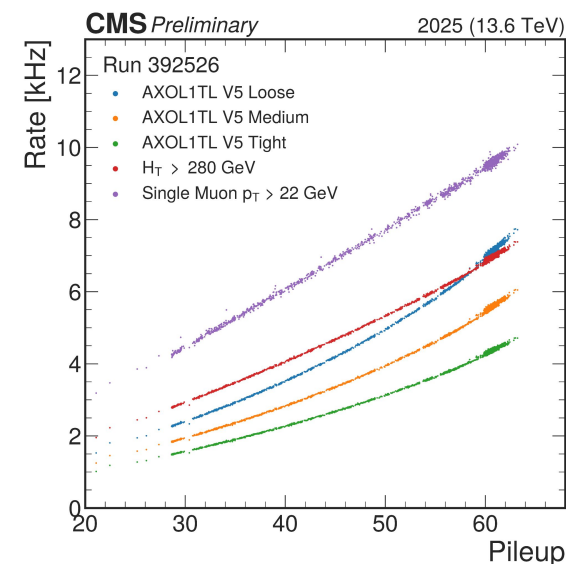
Latent distribution of the VAE^[5]. Since the data semantics are unchanged, the probability density is expected to remain constant.

AXOL1TL Operation

- AXOL1TL exhibits a stable total data rate during data taking run 393376 at CMS;
 - small fluctuations are also recorded by non-ML trigger paths and are likely attributed to transient noise in the detector.
- Equally the AXOL1T rate shows a predictable dependence on the number of pile up (PU) collisions.



Total output rate of AXOL1TL V5 trigger paths^[5] show stability over data taking run during luminosity levelling. Small fluctuations are also observed in traditional trigger paths and attributed to detector noise.



Total output rate of AXOL1TL trigger paths as a function of PU interactions^[5]. Non-linear dependence is observed and also displayed by traditional triggers such as $H_T > 280$ GeV.

Re-training (re-calibrating?)

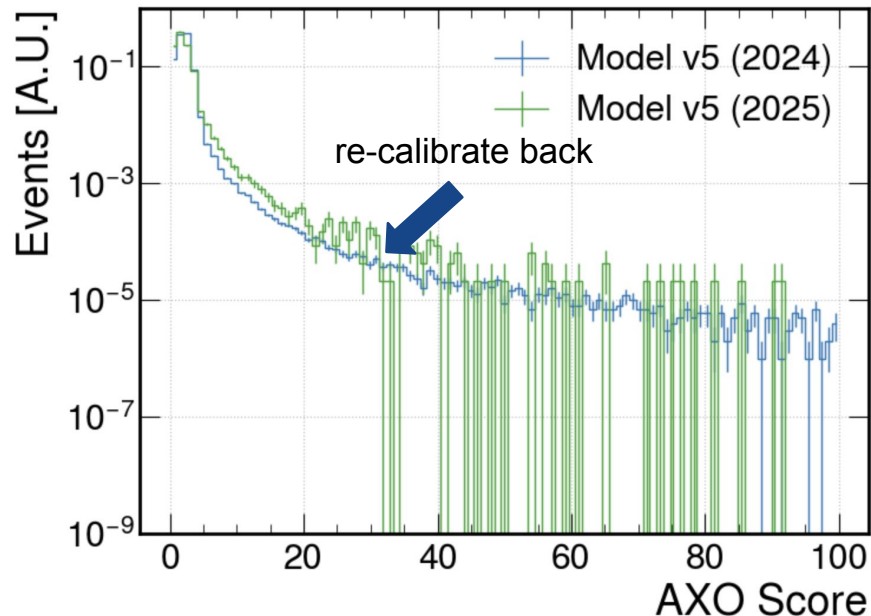
Data will drift over time (and/or unexpectedly). What are the options?

1. **Collect a new dataset and retrain the model (our initial approach with AXOL1TL).**
 - Each retraining changes the algorithm's characteristics, effectively selecting a new phase space.
 - The new behaviour must then be tested, validated, presented: very lengthy process.
2. **Collect new dataset and re-calibrate the model to a reference distribution.**
 - The reference is already well understood in terms of trigger characteristics.
 - This can be done quickly to respond to sudden drifts in the data.

Re-training (re-calibrating?)

Going back.

- A strategy for aligning ML models exposed to evolving input data;
 - an updated loss function regularised by divergence from reference distribution (obtained at training time) and clipping mechanism to ensure stable gradients.
 - Largely inspired by reinforcement learning strategies (GPRO).



output (new model/old model) on new data

Kullback–Leibler divergence (KLD) between new model on new data and reference distribution (from original training)

$$L(\theta) = \sum_{i=1}^N \rho_{\delta}(s_{\theta}(x_i) - s_{\text{ref}}(x_i)) + \beta D_{\text{KL}}(p_{\theta}(s) \parallel p_{\text{ref}}(s)) \quad [7]$$

where $\rho_{\delta}(x) = \text{clip}(x, -\delta, \delta)$

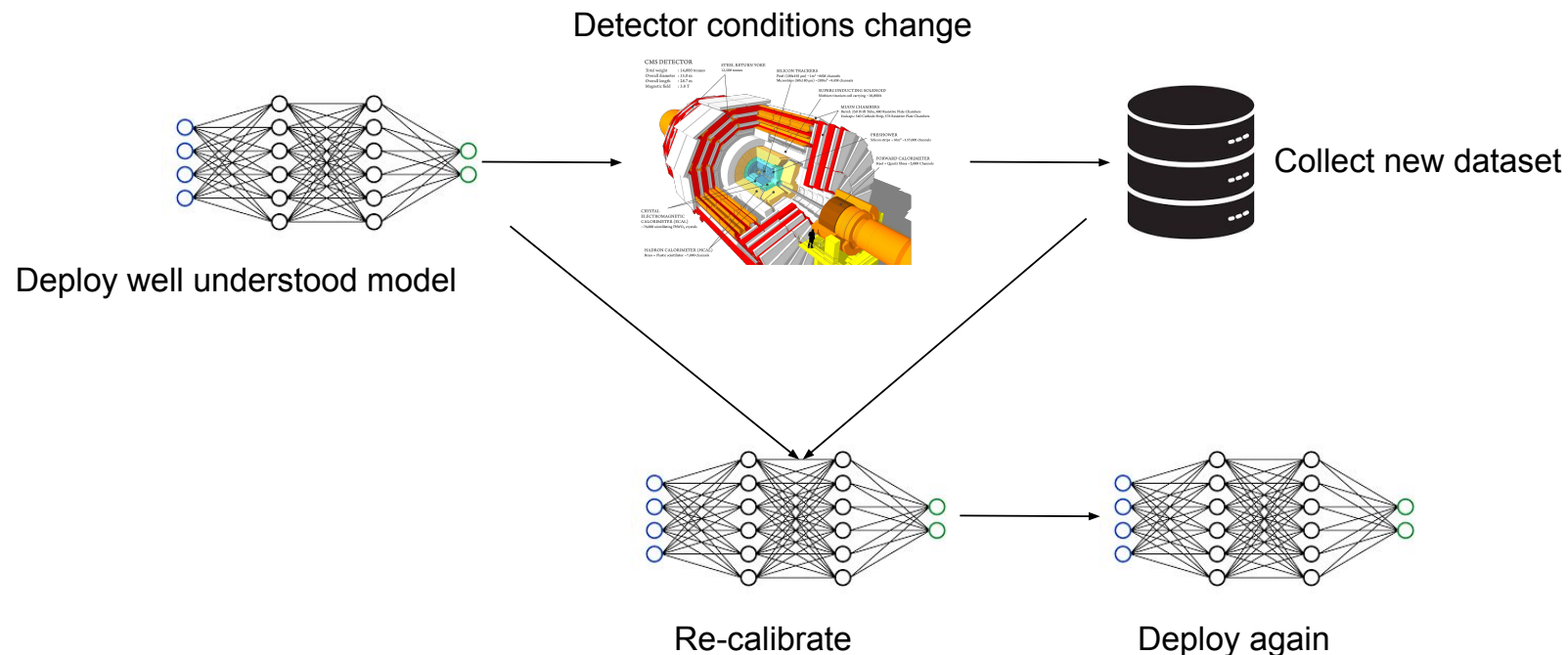
Clipped to a “trust region” where model can explore

└ early stopping when KLD is minimal

Re-training (re-calibrating?)

Continual Learning in 2026?

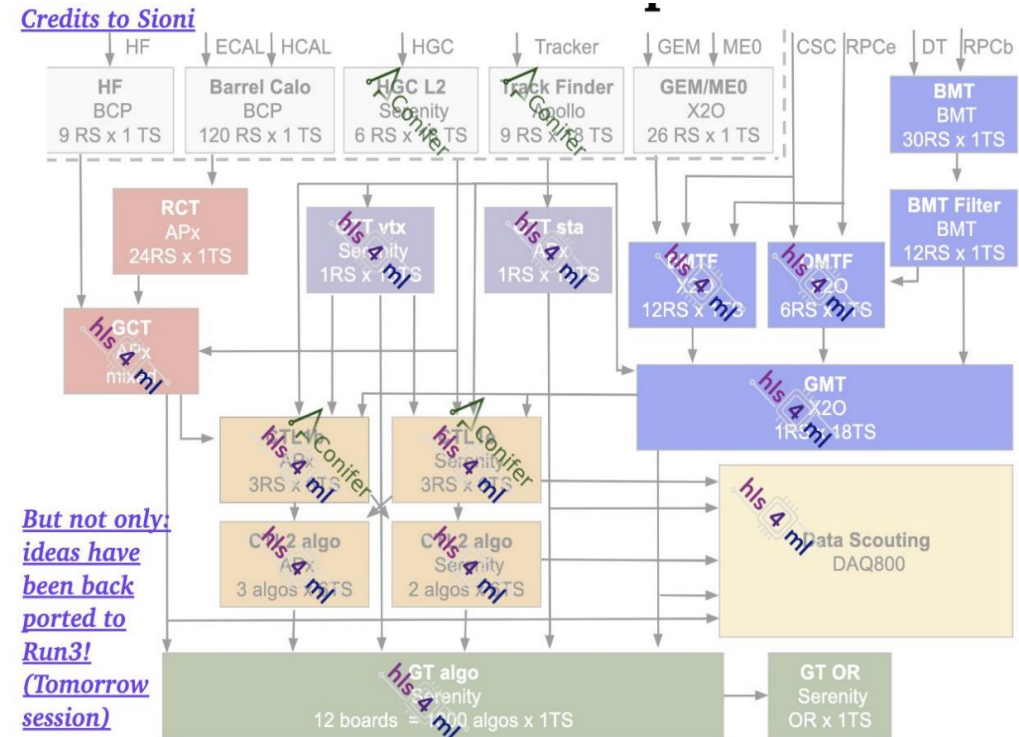
- Considered strategy for AXOL1TL during 2026 data taking;
 - deploy understood model under known detector conditions,
 - as conditions change, use new data to re-calibrate model to known distribution,
 - continue taking data without change in rates or efficiency losses.



High-Luminosity LHC

Trained, not programmed.

- During High-Luminosity LHC, the L1T system^[8] will see a large shift from heuristic algorithms to Machine Learning.
 - *Cascades* of ML models and dependencies.
- Operating such a trigger will be challenging; we will need to leverage all lessons learned from current ML triggers to maintain stable data taking.
 - Centralised training & deployment platform built on Kubernetes.
 - Lockstep Software & Firmware update schedules.
 - Leveraging continual learning strategies.



CMS Level 1 Trigger for
High-Luminosity LHC
**Order of ~billion
inferences / second**

High-Luminosity LHC MLOps

From R&D to production.

- Joint CMS & ATLAS workshop on MLOps for Hardware triggers:
<https://indico.cern.ch/event/1543741/>
- The following is a summary of requirements for both experiments

Data: Algorithms to have on-demand access to training data

- Standardised ML data formats across the project to reduce I/O bottlenecks
- GPU utilisation across the whole chain (NVIDIA Rapids for data pre-processing)
- Central production of “ML-friendly” datasets within experimental software
- Enable data tracking, traceability & audits of training data

Emulation: ML Models in the online-to-offline framework

- Define a model “object” which travels through all software and firmware transitions for clear model lineage
- Add ML models to centralised offline database with “Interval Of Validity” for bookkeeping
- Allow for just-in-time compilation / runtime interpreters when evaluating models converted into High Level Synthesis code

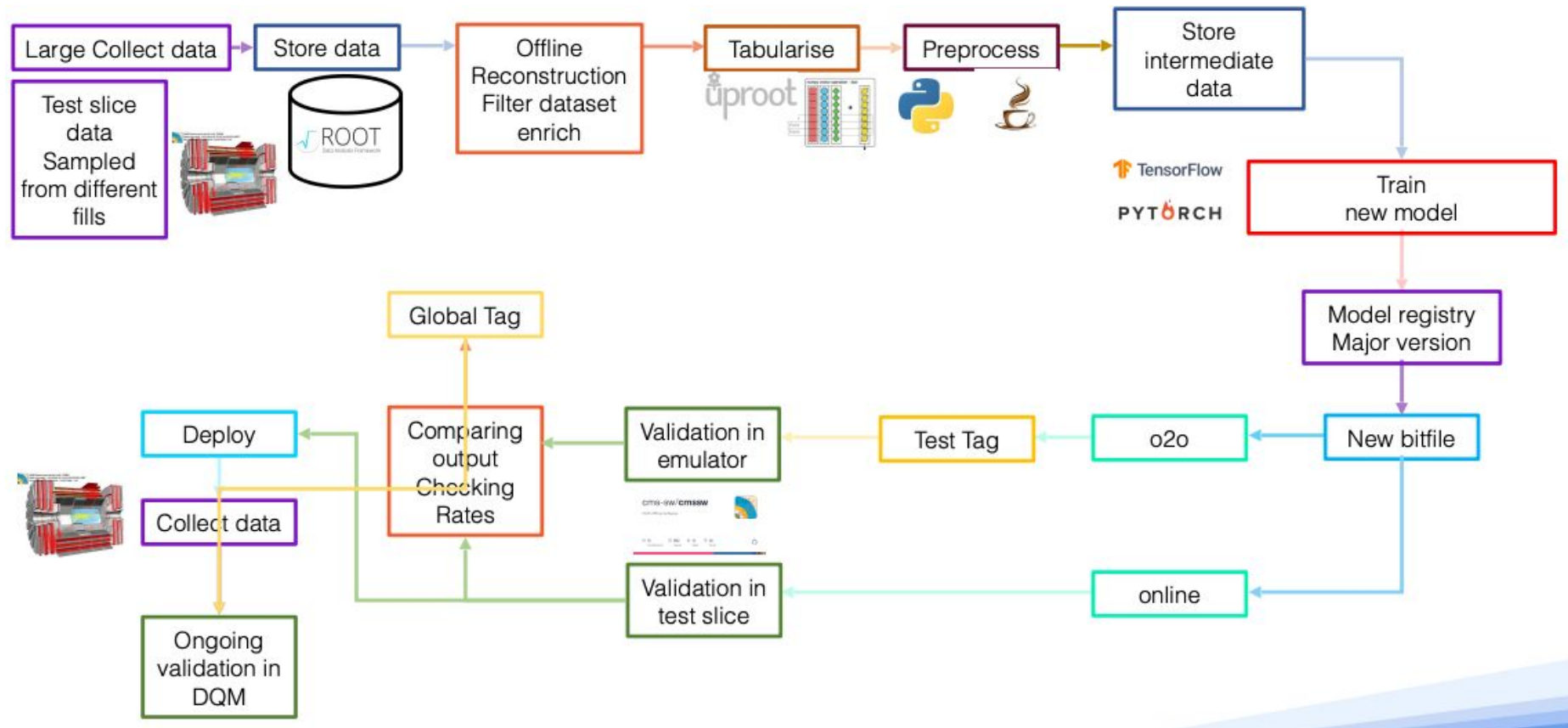
Training: Re-training & Calibrating models

- GPU capacity
- Robust training schemes (Lipschitz networks)
- Monitoring data drifts
- Strategies for continual learning



High-Luminosity LHC MLOps

Projected training & deployment flow for Machine Learning triggers during CMS High-Luminosity LHC data taking.



Summary

- High Energy Physics Experiments are transitioning towards trigger systems which are trained rather than explicitly programmed.
 - Machine Learning is already deployed at the CMS Level 1 Trigger.
 - Use of Machine Learning will only increase during the High-Luminosity Phase: $\mathcal{O}(\text{billion})$ inferences /second.
 - This brings about a new set of operational challenges for the L1 system.
- How do we orchestrate the deployment of Machine Learning at scale?
 - Flexible Kubernetes platform for heterogeneous pipelines.
 - Maintain a common model object through all software to firmware transitions.
 - Interface with centralised databases to aid bookkeeping.
 - Centralised control over algorithms.
- Changing environments
 - Detector evolutions mean data drift can occur.
 - Model output will also drift over-time and/or suddenly.
 - On a short time scale, re-calibration to a reference distribution is possible.
 - Train robustly from the outset.

CMS is planning for these operational demands now.

References

- [1] CMS Collaboration, "CMS Technical Design Report for the Level-1 Trigger Upgrade," CERN-LHCC-2013-011, CMS-TDR-12, 2013
- [2] K. Compton et al., "The MP7 and CTP-6: multi-hundred Gbps processing boards for calorimeter trigger upgrades at CMS," JINST, vol. 7, no. 12, 2012
- [3] CMS Collaboration, "Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV," *Journal of Instrumentation*, vol. 15, no. 10, P10017, 2020, doi:10.1088/1748-0221/15/10/P10017.
- [4] CMS Collaboration, "Level-1 Trigger Calorimeter Image Convolutional Anomaly Detection Algorithm," CMS-DP-2023-086, 2023, <https://cds.cern.ch/record/2879816>
- [5] CMS Collaboration, Public Results, <https://twiki.cern.ch/twiki/bin/view/CMSPublic/L1TriggerDPGResults>
- [6] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning" 2022. Available: <https://arxiv.org/abs/2105.04906>
- [7] DeepSeek, "DeepSeek-V3 Technical Report," arXiv:2412.19437 [cs.CL], 2025.