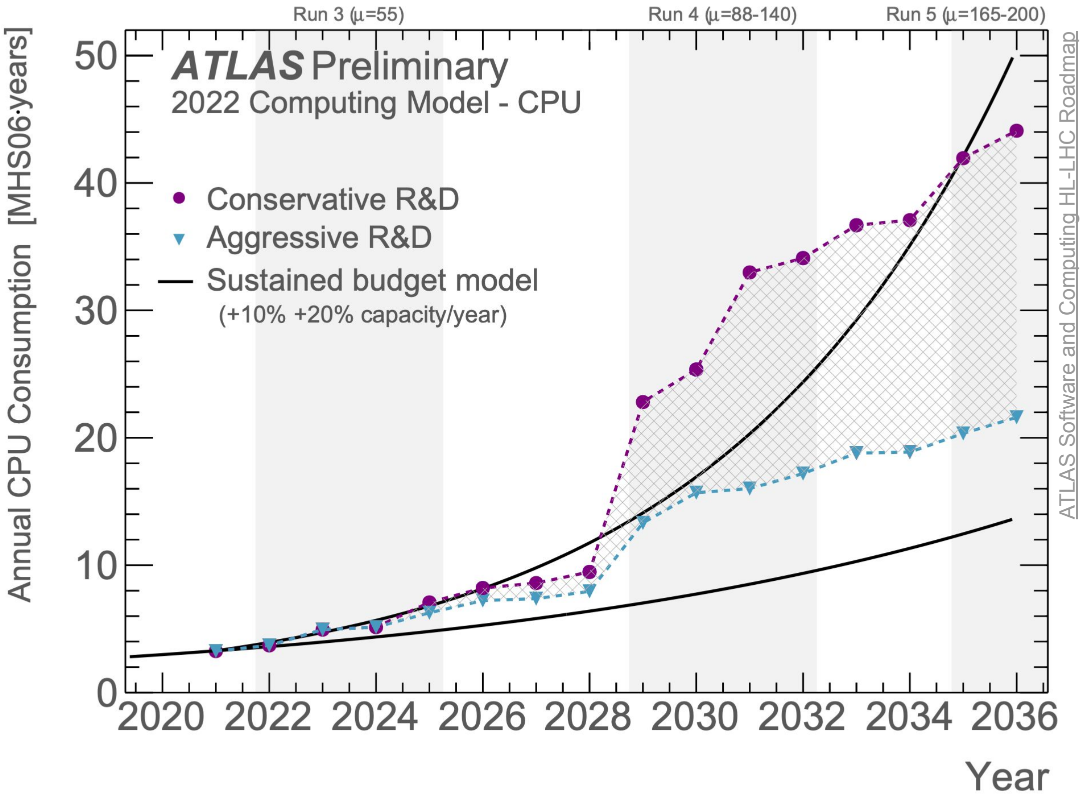
interTwin is funded by the European Union
Grant Agreement Number 101058386

Calorimeter shower simulation with machine learning

Corentin Allaire, Vera Maiboroda, David Rousseau, Minh Tuan Pham

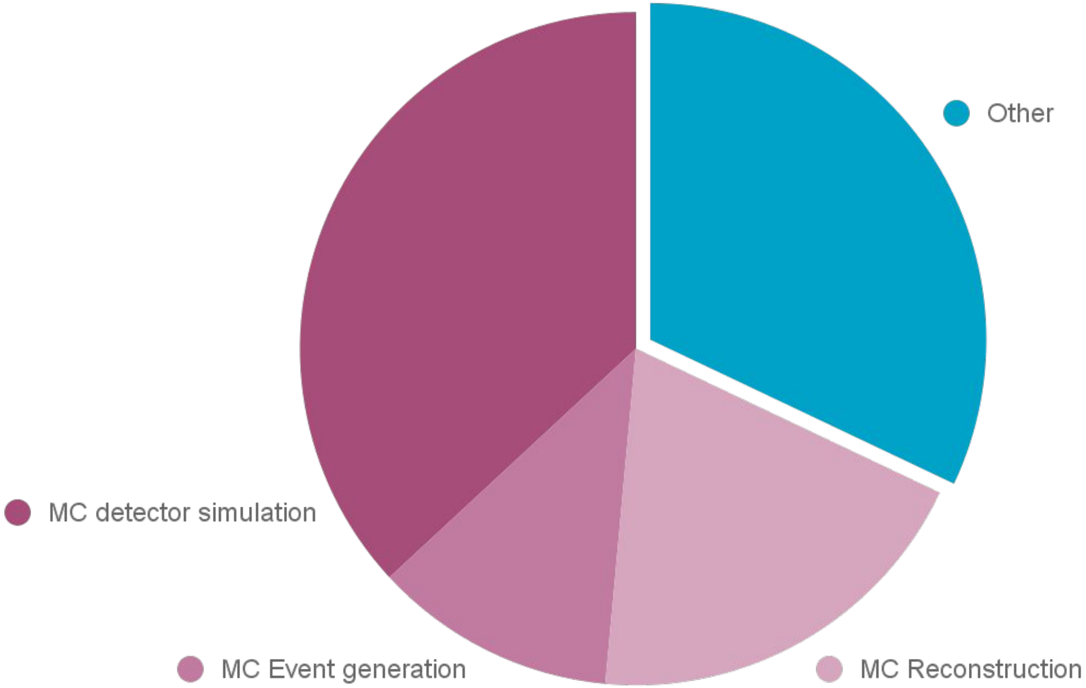
Computational resources for simulations

- Computing budgets are limited — not scaling with Run 4 requirements
  ⟶ need optimisations, fast simulation, use of HPCs and GPUs



Computational resources for simulations

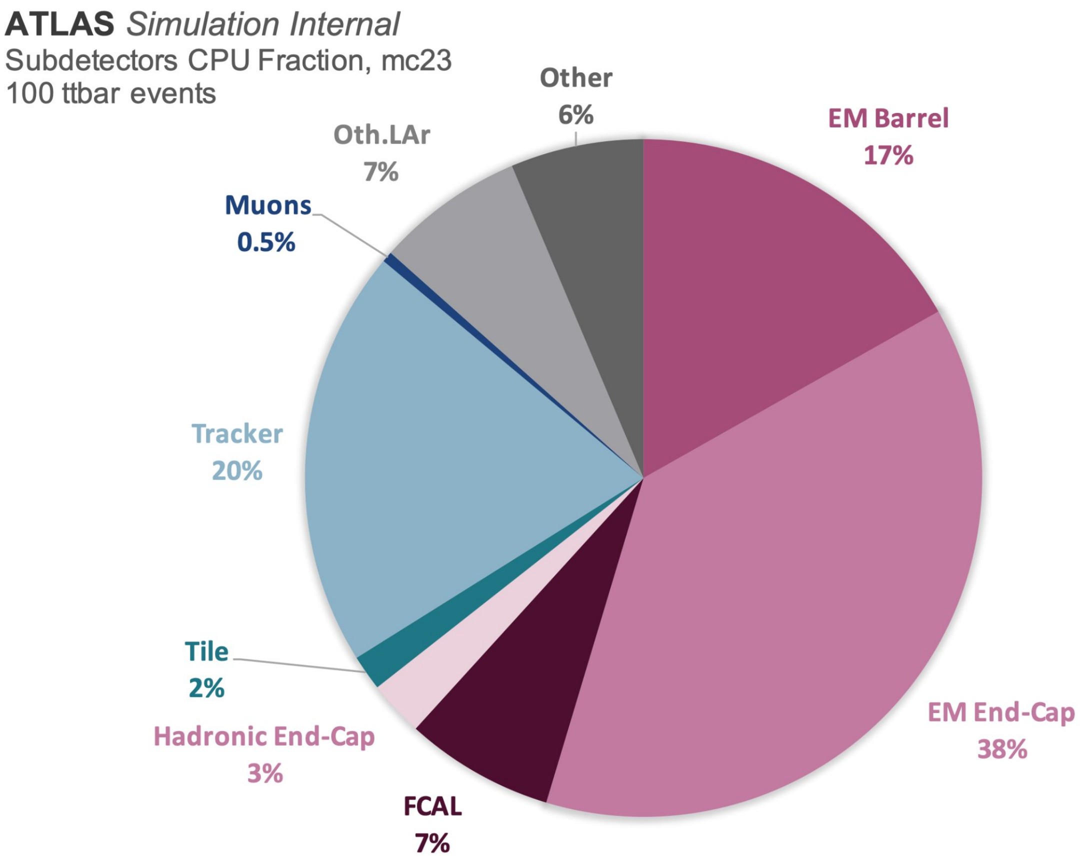
- Computing budgets are limited — not scaling with Run 4 requirements
  ⟶ need optimisations, fast simulation, use of HPCs and GPUs
- MC production takes ~70% of the GRID CPU time in ATLAS
- Dominated by MC detector simulation



ICHEP 2024

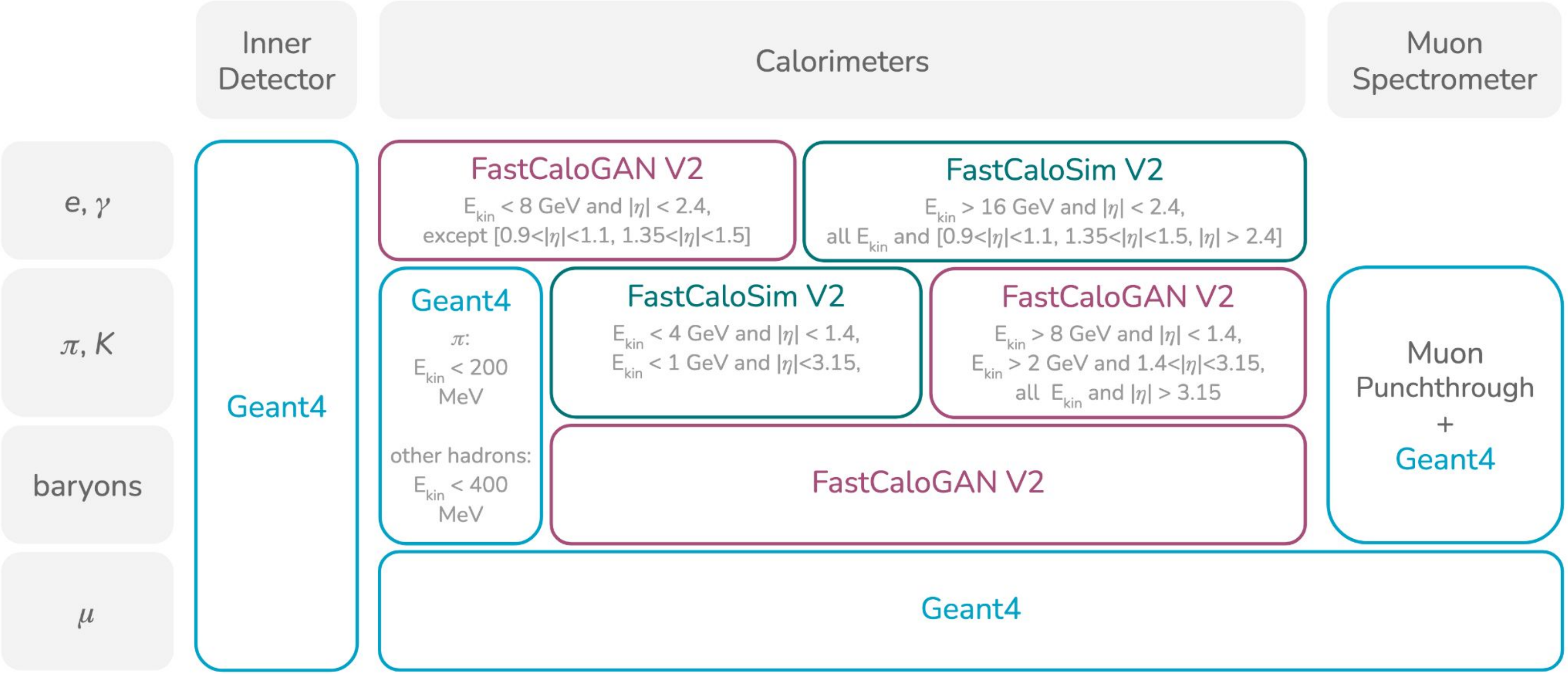
Computational resources for simulations

- Computing budgets are limited — not scaling with Run 4 requirements
  ⟶ need optimisations, fast simulation, use of HPCs and GPUs
- MC production takes ~70% of the GRID CPU time in ATLAS
- Dominated by MC detector simulation
- Dominated by calorimeter simulation



SIM-2024-003

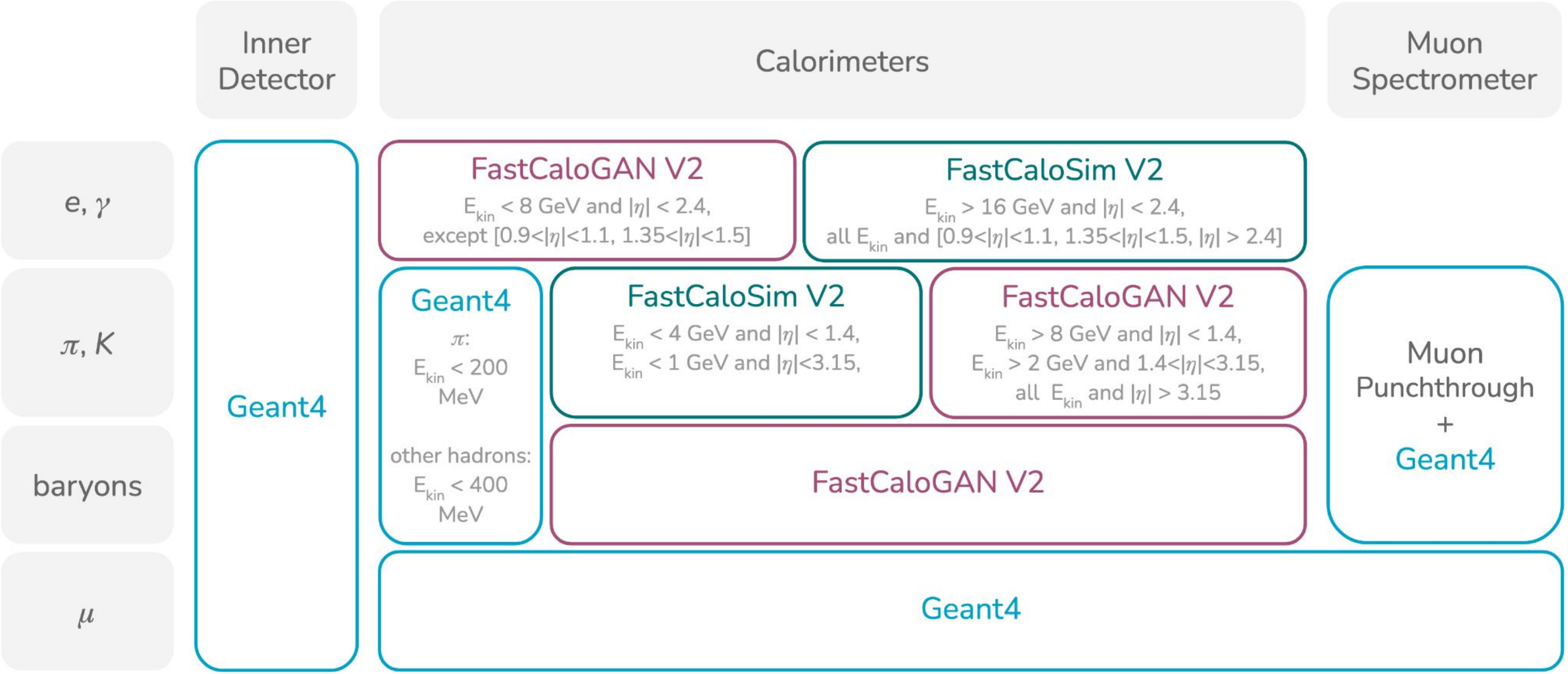
Fast calorimeter simulation for Run-3 within AtlFast3



arxiv.org/abs/2404.06335

- Geant4: all particles in Inner Detector, low energy hadrons in calorimeters, and muons

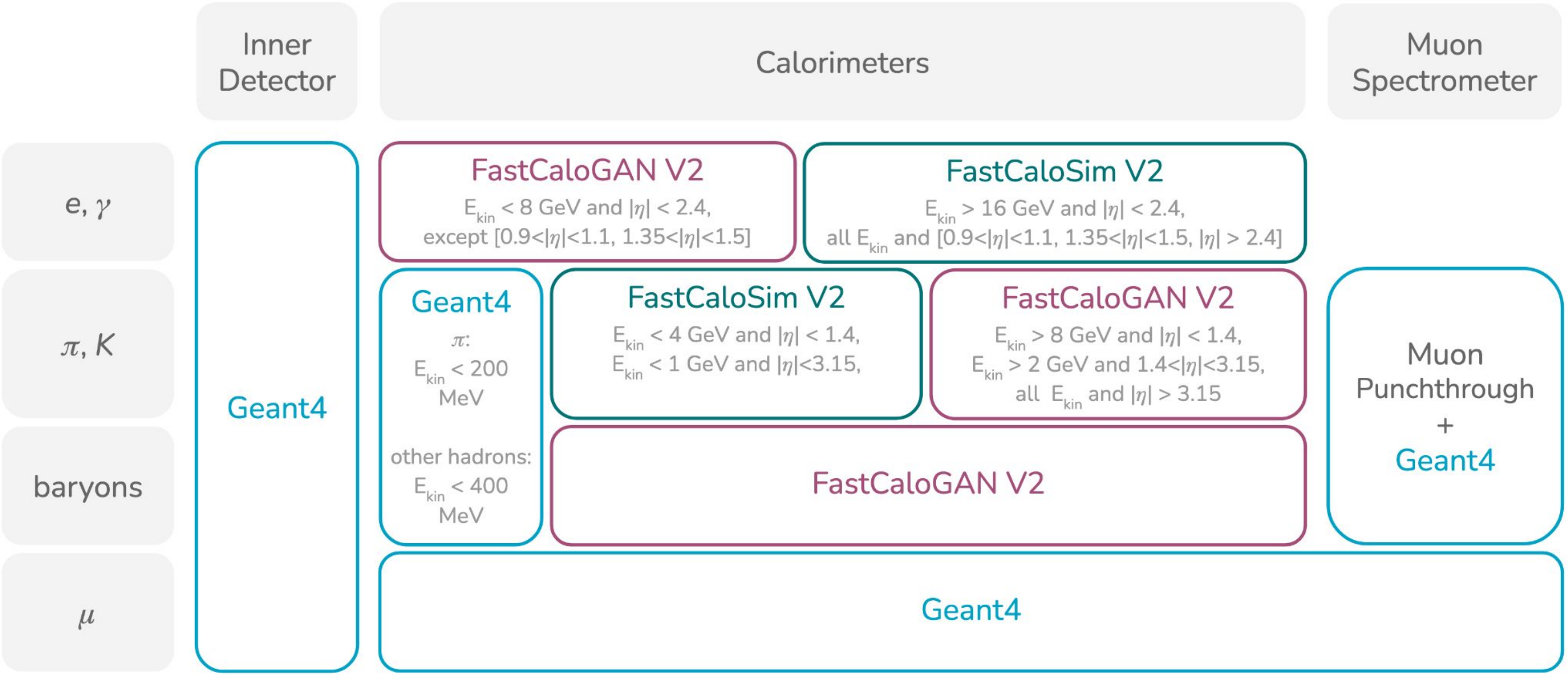
Fast calorimeter simulation for Run-3 within AtlFast3



arxiv.org/abs/2404.06335

- Geant4: all particles in Inner Detector, low energy hadrons in calorimeters, and muons
- FastCaloSim V2: parametrization of shower development

Fast calorimeter simulation for Run-3 within AtlFast3

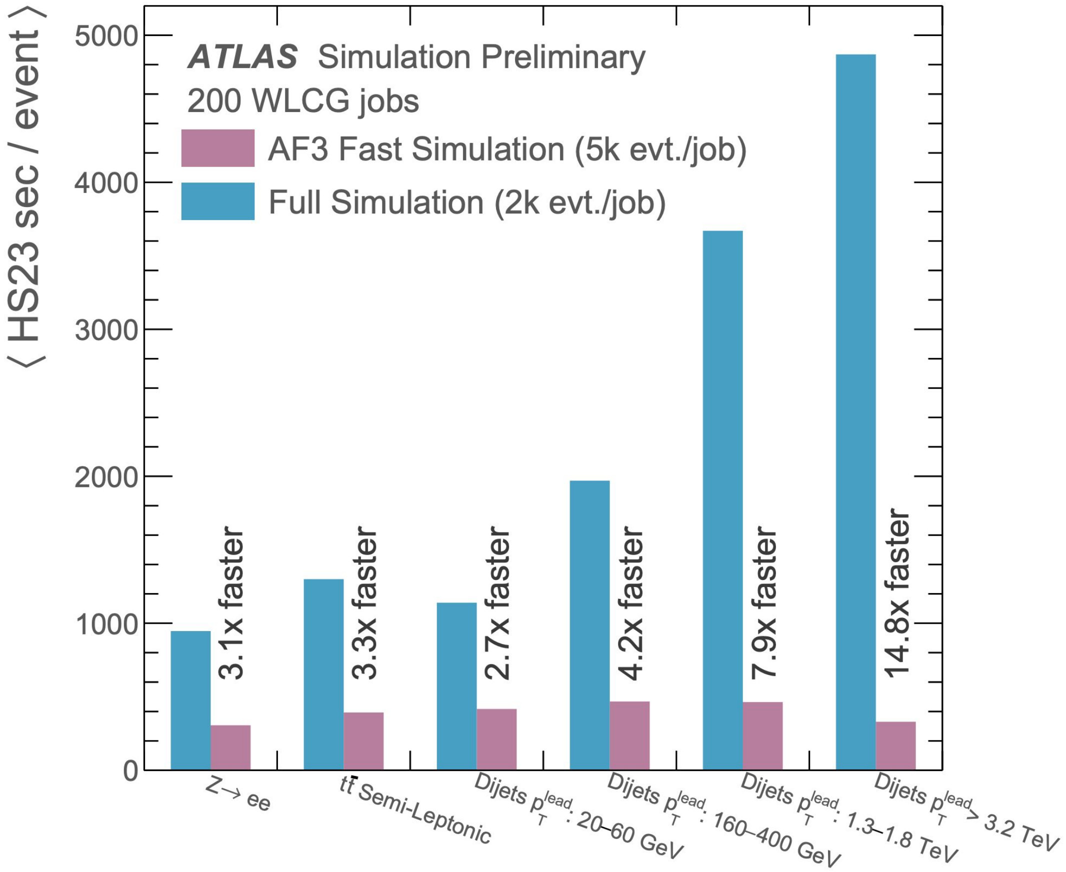


arxiv.org/abs/2404.06335

- Geant4: all particles in Inner Detector, low energy hadrons in calorimeters, and muons
- FastCaloSim V2: parametrization of shower development
- FastCaloGAN V2: generative adversarial networks

AtlFast3 quality: speed

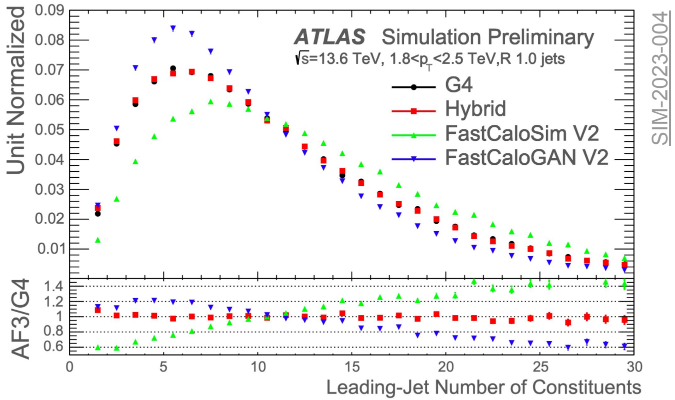
- AtlFast3 is 3 (for $Z \rightarrow ee$ events) to 15 (for high-pT dijet events) times faster than the Geant4 simulation of the ATLAS Run 3 detector mc23c

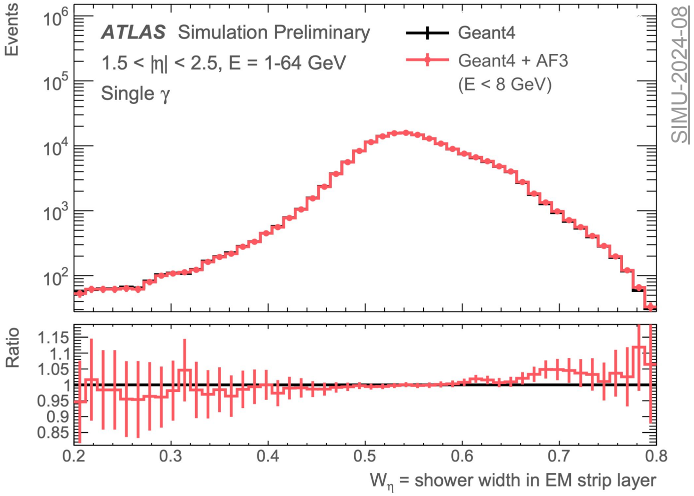


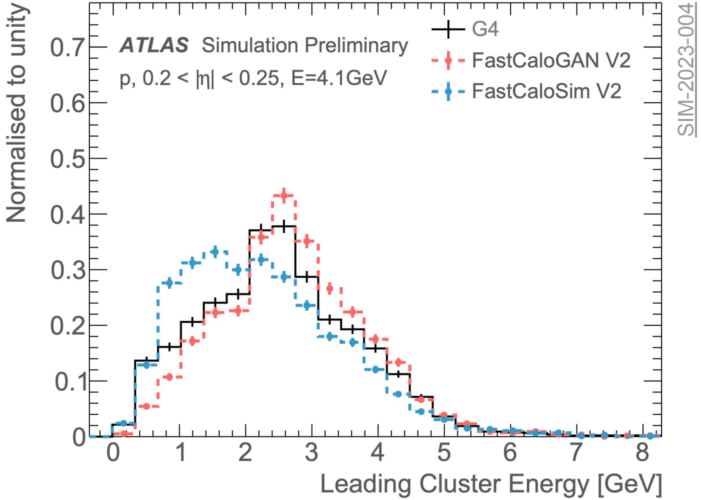
SIM-2023-005

AtlFast3 quality: accuracy

- AtlFast3 is 3 (for Z → ee events) to 15 (for high-pT dijet events) times faster than the Geant4 simulation of the ATLAS Run 3 detector mc23c
- For most observables used in physics analyses, AtlFast3 and Geant4 agree within a few %







CaloChallenge

CaloChallenge:

- Setting for the calorimeter simulation ML models comparison
- Develop a model that samples from $p(\text{shower} \mid E_{\text{incident}})$

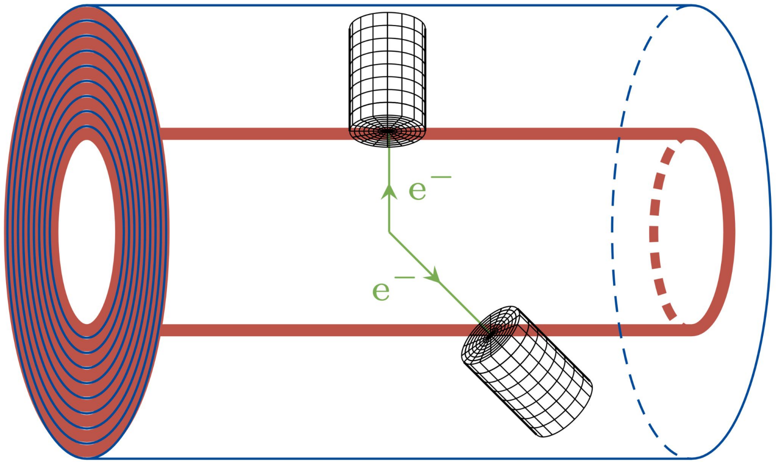
CaloChallenge

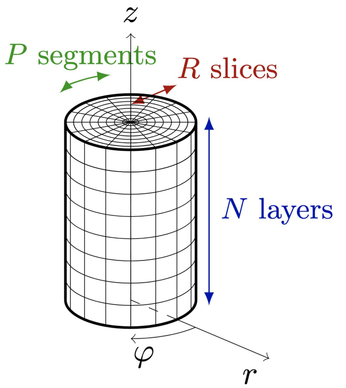
CaloChallenge:

- Setting for the calorimeter simulation ML models comparison
- Develop a model that samples from $p(\text{shower} \mid E_{\text{incident}})$

Datasets:

Origin	Particle type : # of voxels	E_{incident}
AtlFast3 training data	γ: 368 π: 533	[256 MeV, 4.2 TeV]
Par04 from Geant4	e^-: 6480	[1 GeV, 1 TeV]
Par04 from Geant4	e^-: 40500	[1 GeV, 1 TeV]





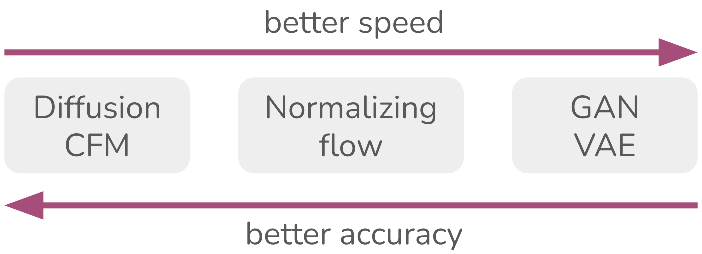
arxiv.org/abs/2410.21611

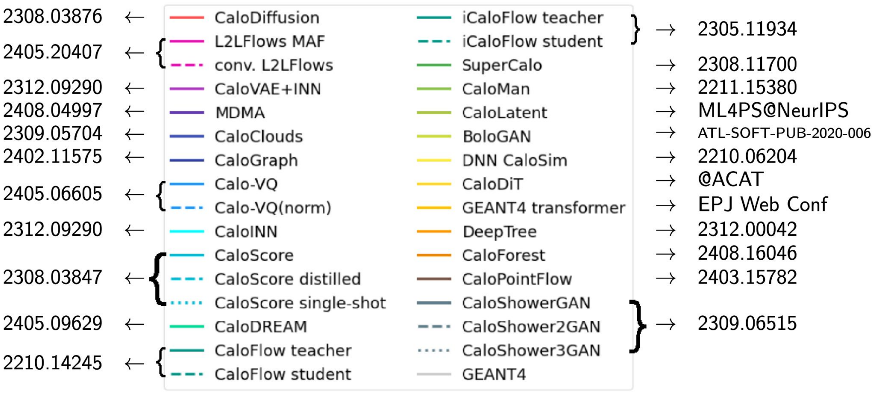
CaloChallenge:

- Setting for the calorimeter simulation ML models comparison
- Develop a model that samples from $p(\text{shower} \mid E_{\text{incident}})$

Results:

- 30+ submissions
- Observed speed/accuracy tradeoff:

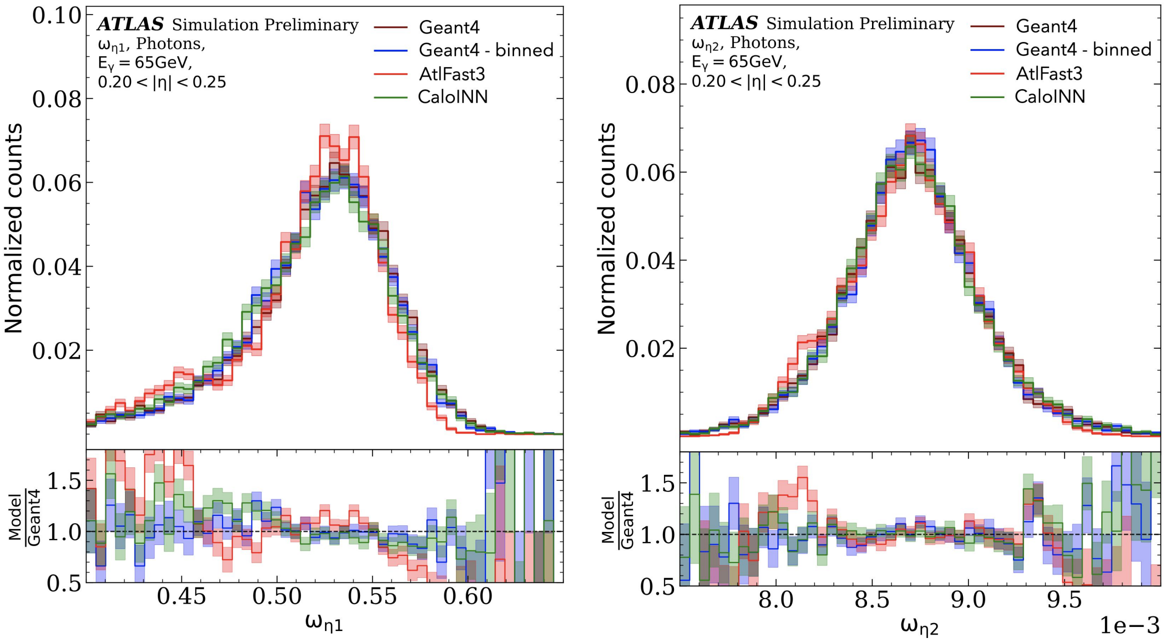




arxiv.org/abs/2410.21611

Towards Run 4

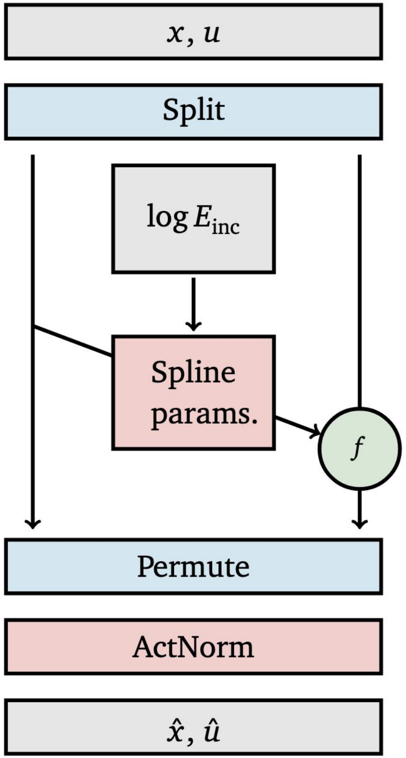
- Successful insertion of ML-based calorimeter simulation into Run 3 production
- Ongoing development of AtlFast4:
  - Improved voxelisation
    to reduce bias due to calorimeter geometry
  - Research diffusion models and Invertible Neural Networks (INNs)
    add to / replace AtlFast3 GANs



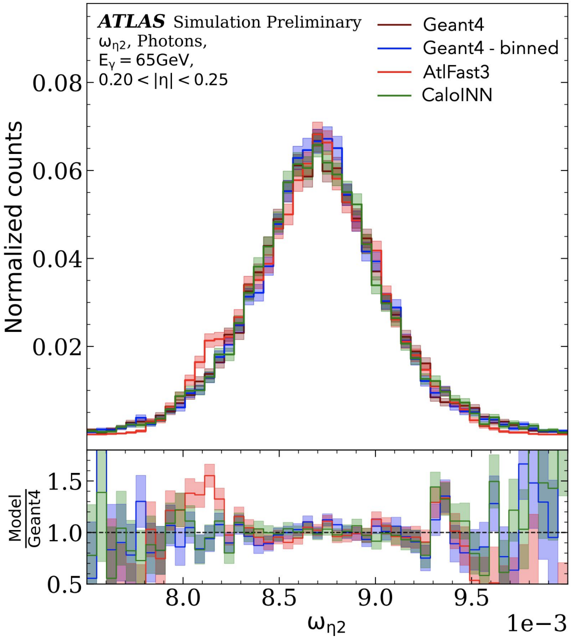
SIMU-2024-10

Towards Run 4

- Successful insertion of ML-based calorimeter simulation into Run 3 production
- Ongoing development of AtlFast4:
  - Improved voxelisation
    to reduce bias due to calorimeter geometry
  - Research diffusion models and Invertible Neural Networks (INNs)
    add to / replace AtlFast3 GANs



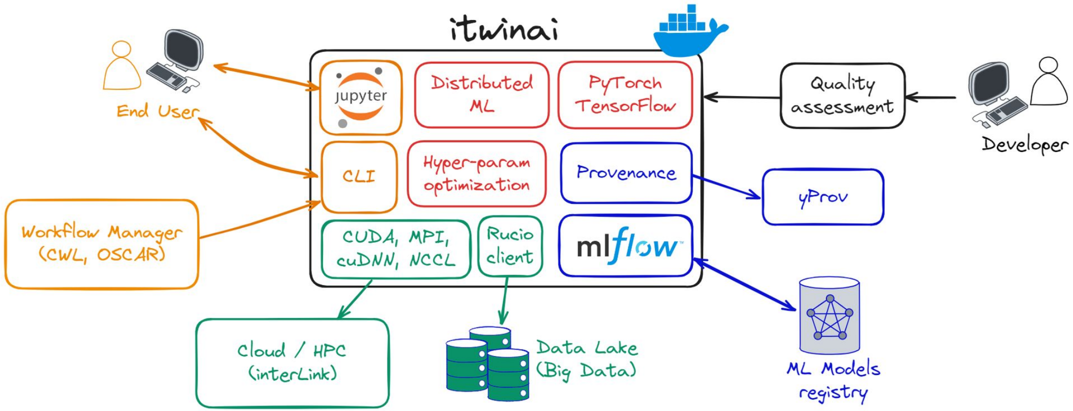
CaloINN diagram



SIMU-2024-10

IJCLab activities

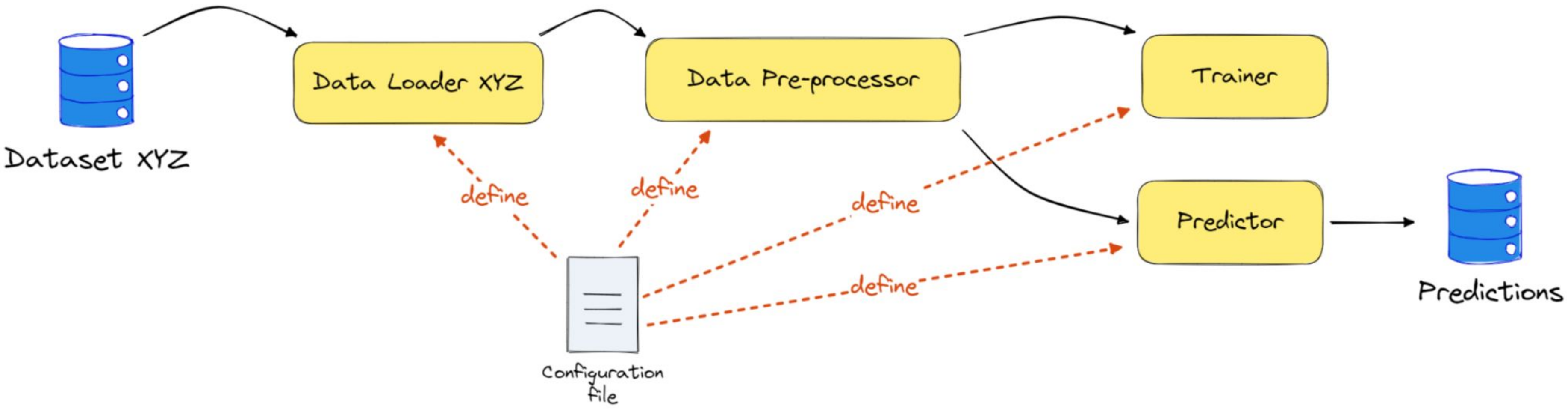
- Research INN chosen from CaloChallenge:
    - Full integration with interTwin project



interTwin project

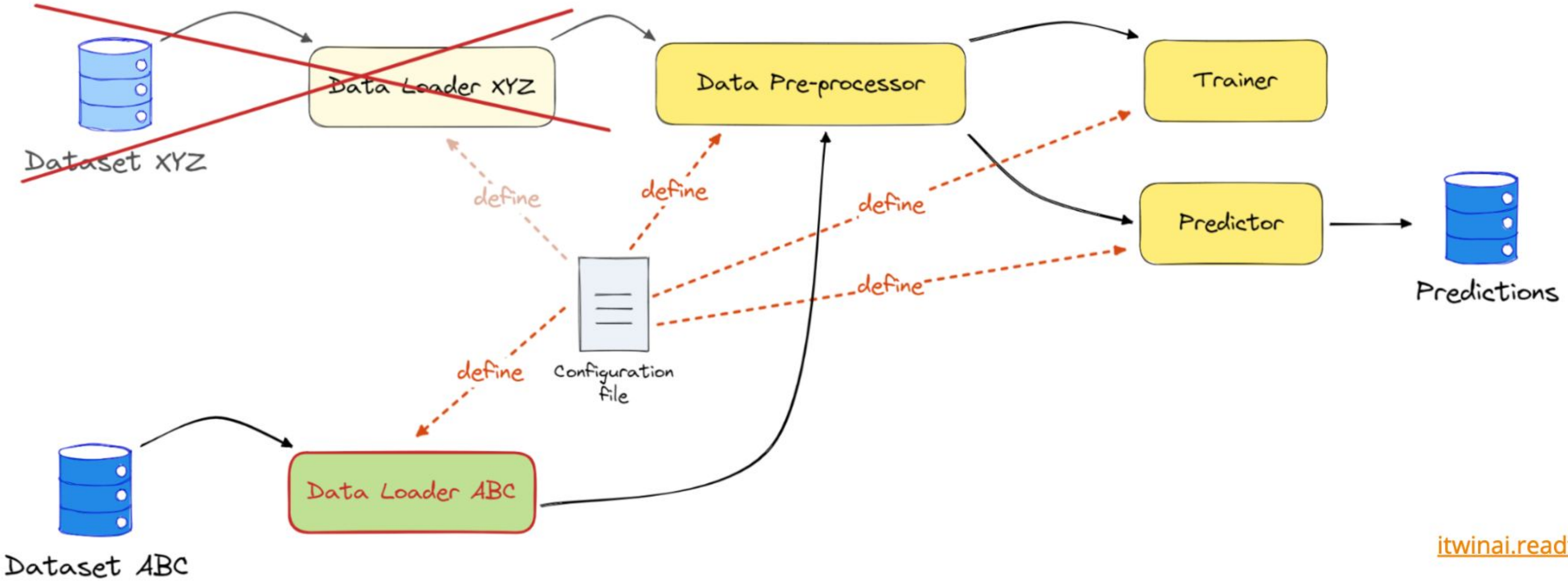
Sometimes ML workflows can be **monolithic** scripts, **hard to maintain**.

itwinai helps to break the code in **modular** and **reusable** components.



Sometimes ML workflows can be **monolithic** scripts, **hard to maintain**.

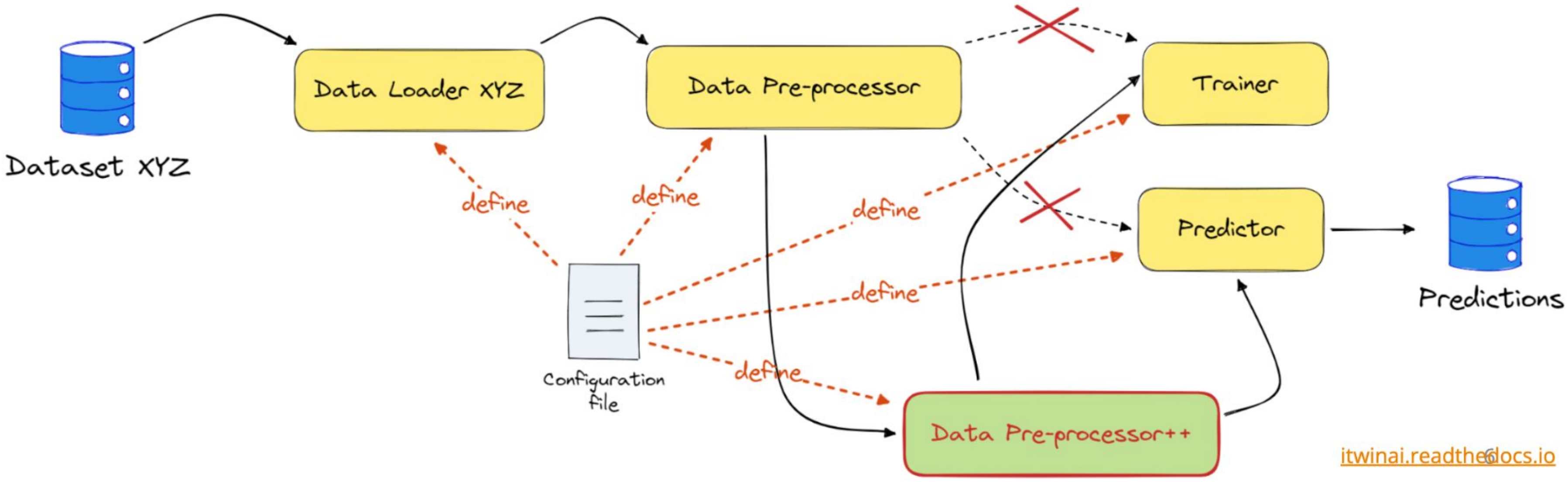
itwinai helps to break the code in **modular** and **reusable** components.
Example: **replace** data connector.



itwinai.readthedocs.io

Sometimes ML workflows can be **monolithic** scripts, **hard to maintain**.

itwinai helps to break the code in **modular** and **reusable** components.
Example: **add** steps.

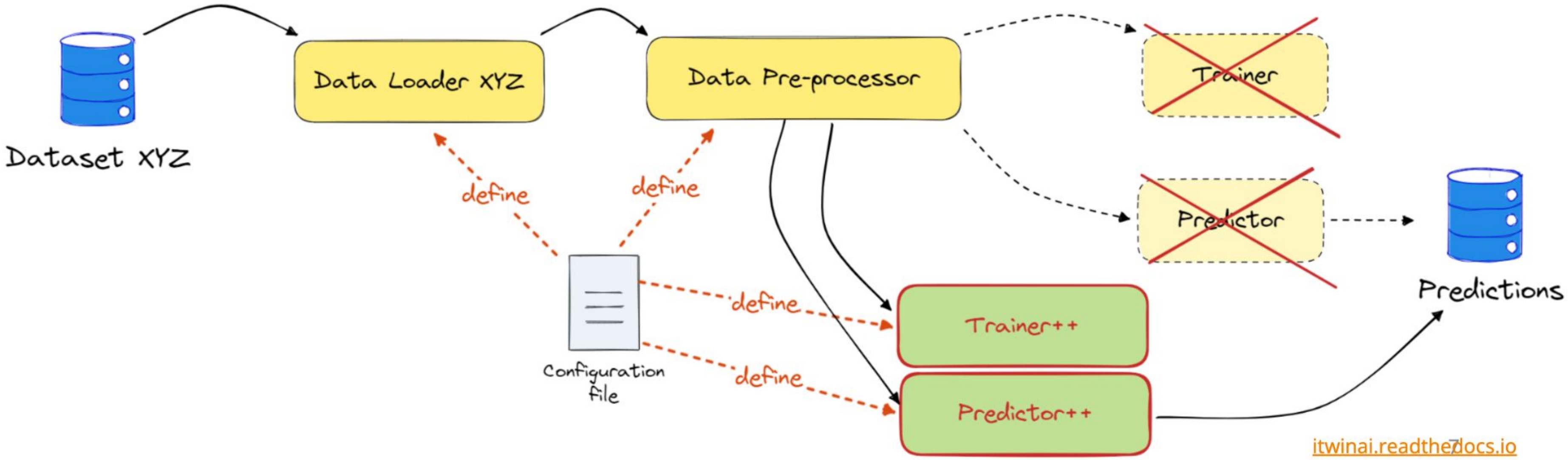


itwinai.readthedocs.io

InterTwin: itwinai

Sometimes ML workflows can be **monolithic** scripts, **hard to maintain**.

itwinai helps to break the code in **modular** and **reusable** components.
Example: **replace** the ML model (training and/or inference).

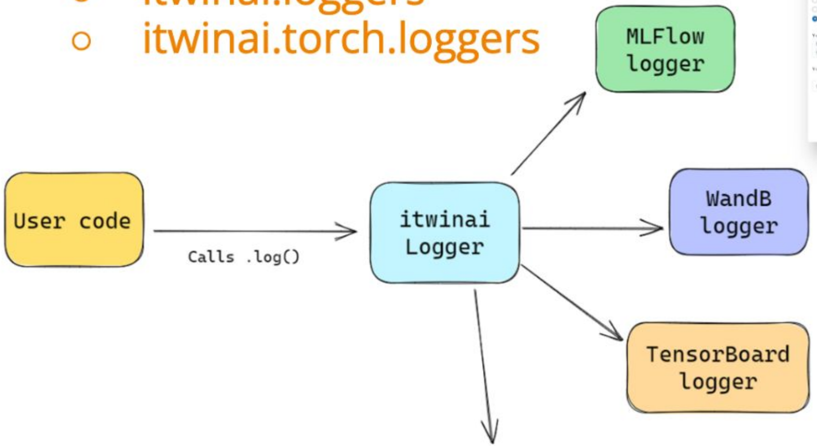


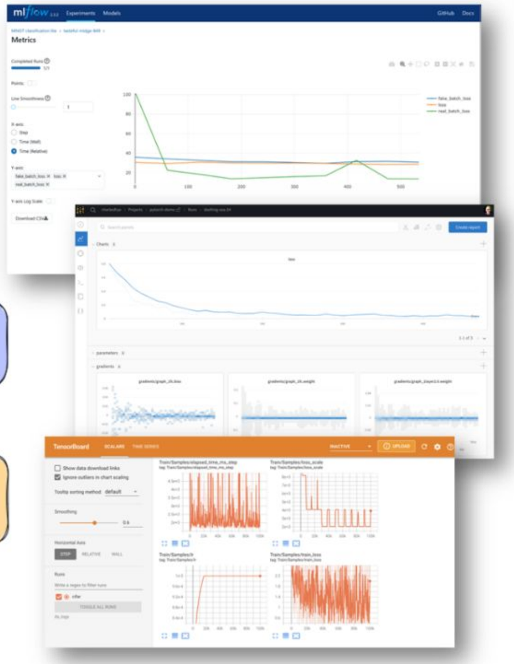
itwinai.readthedocs.io

InterTwin: itwinai

itwinai machine learning toolkit

- Unified logger interface (MLFlow, TensorBoard, WandB)
    - itwinai.loggers
    - itwinai.torch.loggers



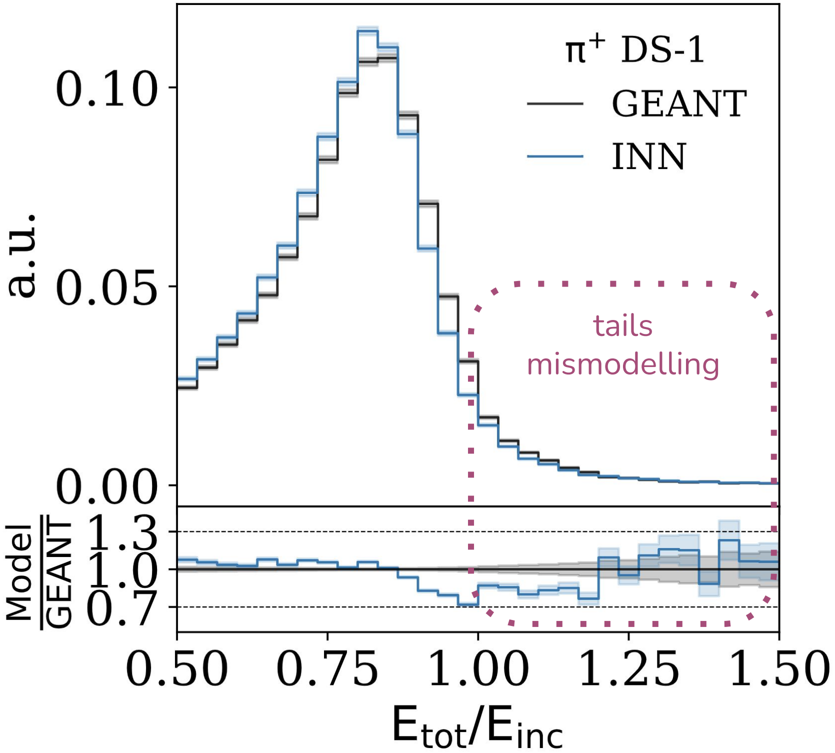


yProvML
https://github.com/HPCI-Lab/yProvML
(UNITN)

itwinai.readthedocs.io

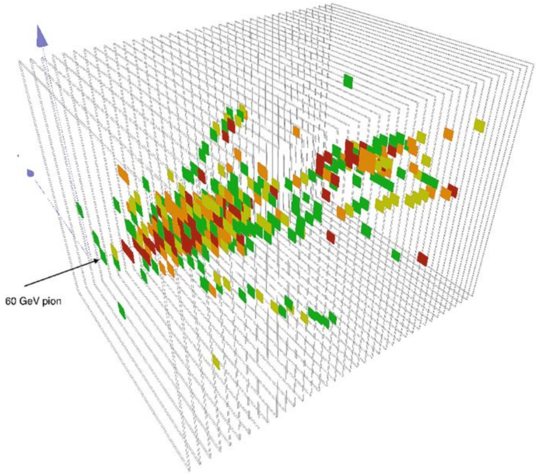
IJCLab activities

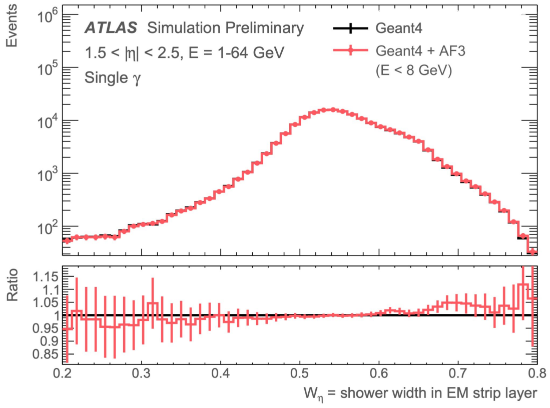
- Research INN chosen from CaloChallenge:
  - Full integration with interTwin project
  - Validation with Open Data Detector
    more realistic detector geometry, open-source
  - Refine distribution tails
    introduce postfix to the most problematic areas
    and/or improve model training strategy
    → better quality keeping speed

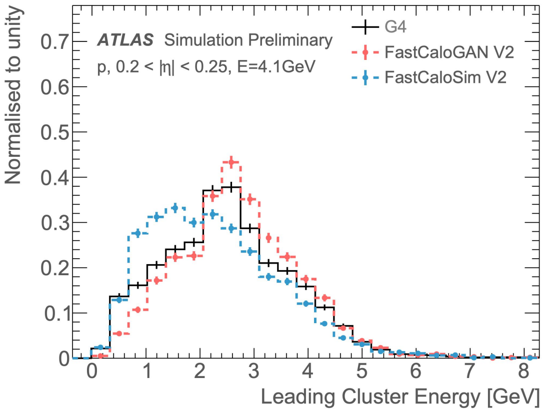


Refine distribution tails

- Research INN chosen from CaloChallenge:
  - Full integration with interTwin project
  - Validation with Open Data Detector
    more realistic detector geometry, open-source
  - Refine distribution tails
    introduce postfix to the most problematic areas
    and/or improve model training strategy
    → better quality keeping speed

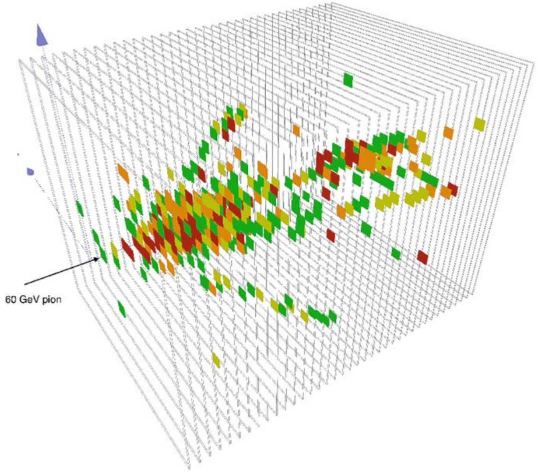






Refine distribution tails

- Research INN chosen from CaloChallenge:
  - Full integration with interTwin project
  - Validation with Open Data Detector
    more realistic detector geometry, open-source
  - Refine distribution tails
    introduce postfix to the most problematic areas and/or improve model training strategy
    → better quality keeping speed



Already existing solution - FastPerfekt: correction of the "high level" features

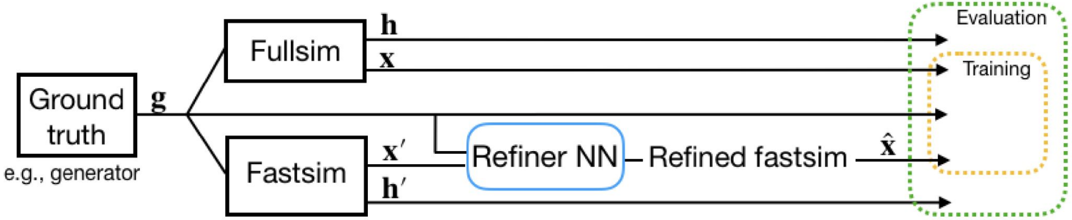
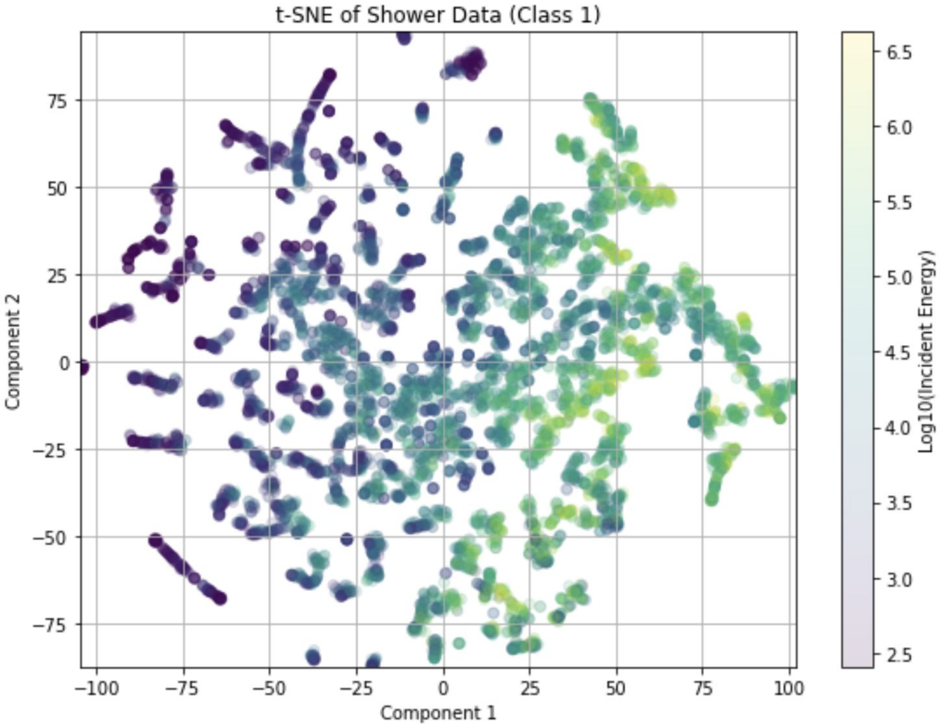


Figure 1: A schema of the Fast Perfekt training and evaluation. The fullsim features $\mathbf{x}$ and $\mathbf{h}$ and fastsim features $\mathbf{x}'$ and $\mathbf{h}'$ share the same ground truth $\mathbf{g}$. The refiner network is trained to apply a residual correction to obtain the refined fastsim data set $\hat{\mathbf{x}}$. The hidden features are used for an evaluation meta-study (green box) but are not incorporated into the training procedure (yellow box).

Fast Perfekt

Refine distribution tails: CaloChallenge dataset with pions

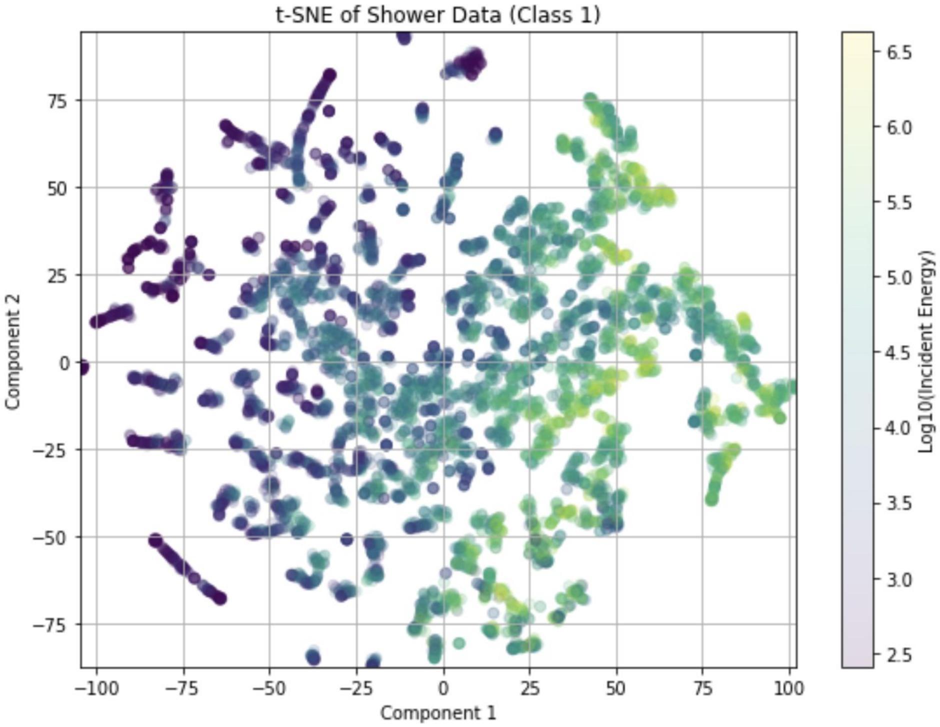
- Select not well simulated showers
  - Dimensionality reduction for visualisation
    500 voxels (normalized by total shower energy) ->
    50 features after U-map ->
    2 features after TSNE



Geant4 showers

Refine distribution tails: CaloChallenge dataset with pions

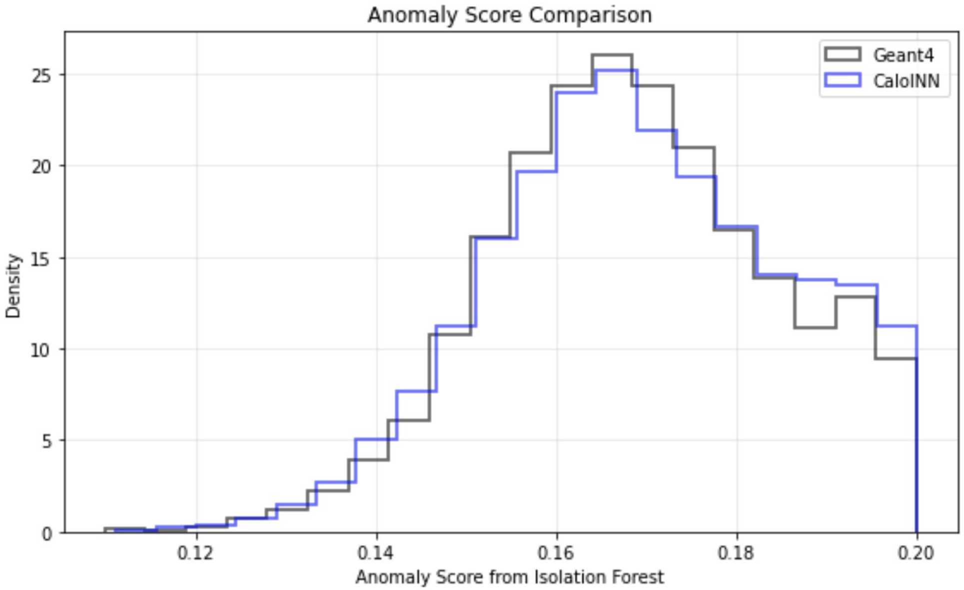
- Select not well simulated showers
    - Dimensionality reduction for visualisation
      500 voxels (normalized by total shower energy) ->
      50 features after U-map ->
      2 features after TSNE



CaloINN showers

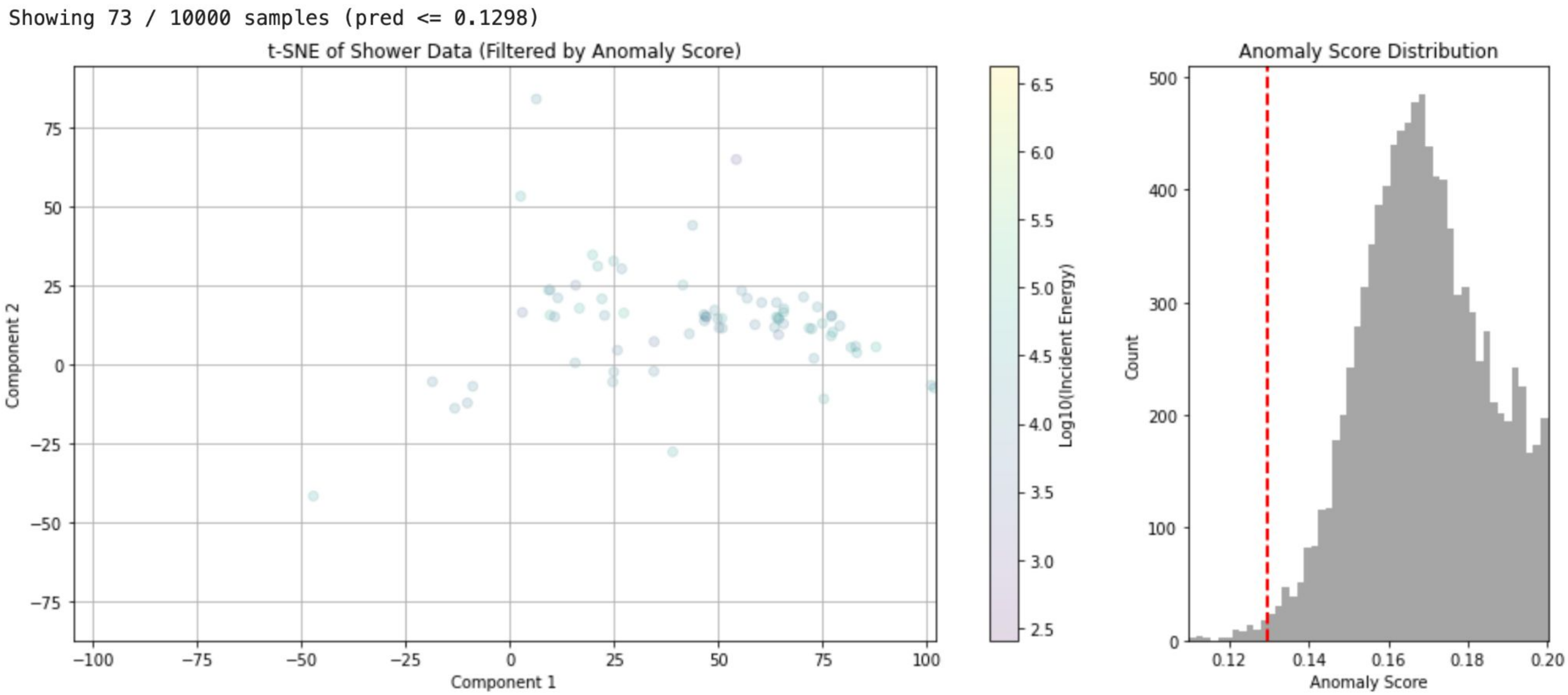
Refine distribution tails: CaloChallenge dataset with pions

- Select not well simulated showers
  - Dimensionality reduction for visualisation
    500 voxels (normalized by total shower energy) ->
    50 features after U-map ->
    2 features after TSNE
  - Use anomalous score from Isolation Forest as a metric of “normality” of a shower
    Trained on the Geant4 data, applied to both Gean4 and CaloINN generated showers



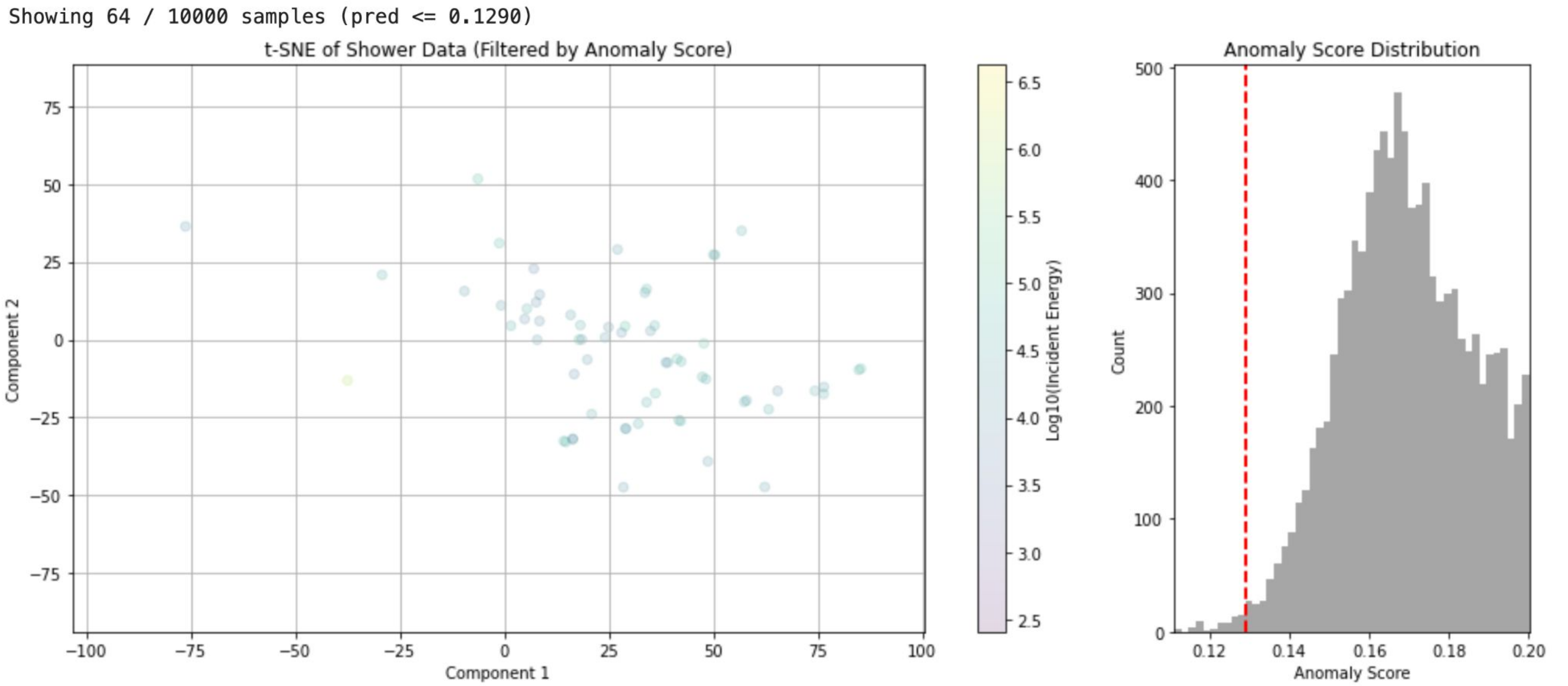
Smaller the score, more anomalous are the showers

Refine distribution tails: CaloChallenge dataset with pions



Most anomalous showers in Geant4

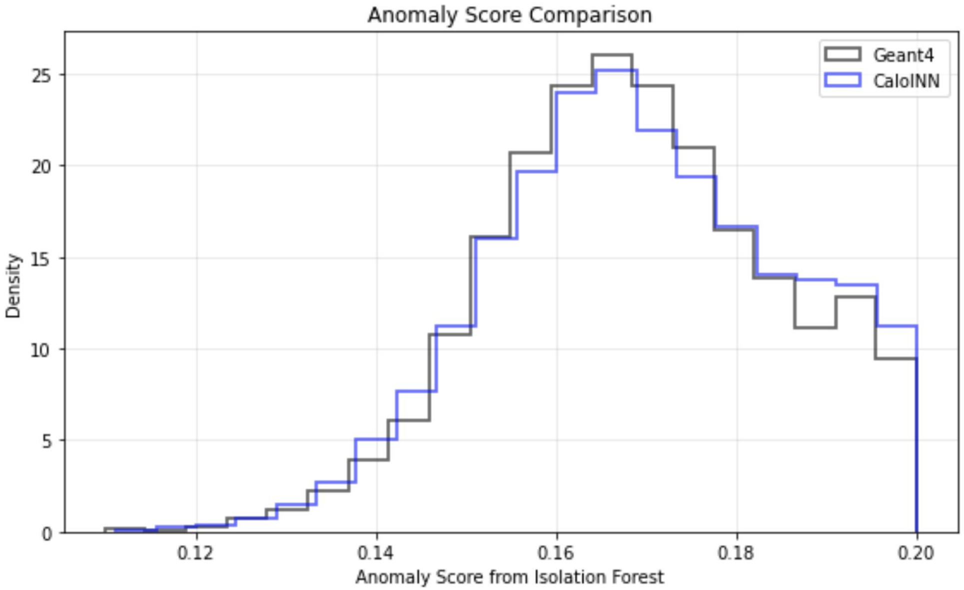
Refine distribution tails: CaloChallenge dataset with pions



Most anomalous showers in CaloINN

Refine distribution tails: CaloChallenge dataset with pions

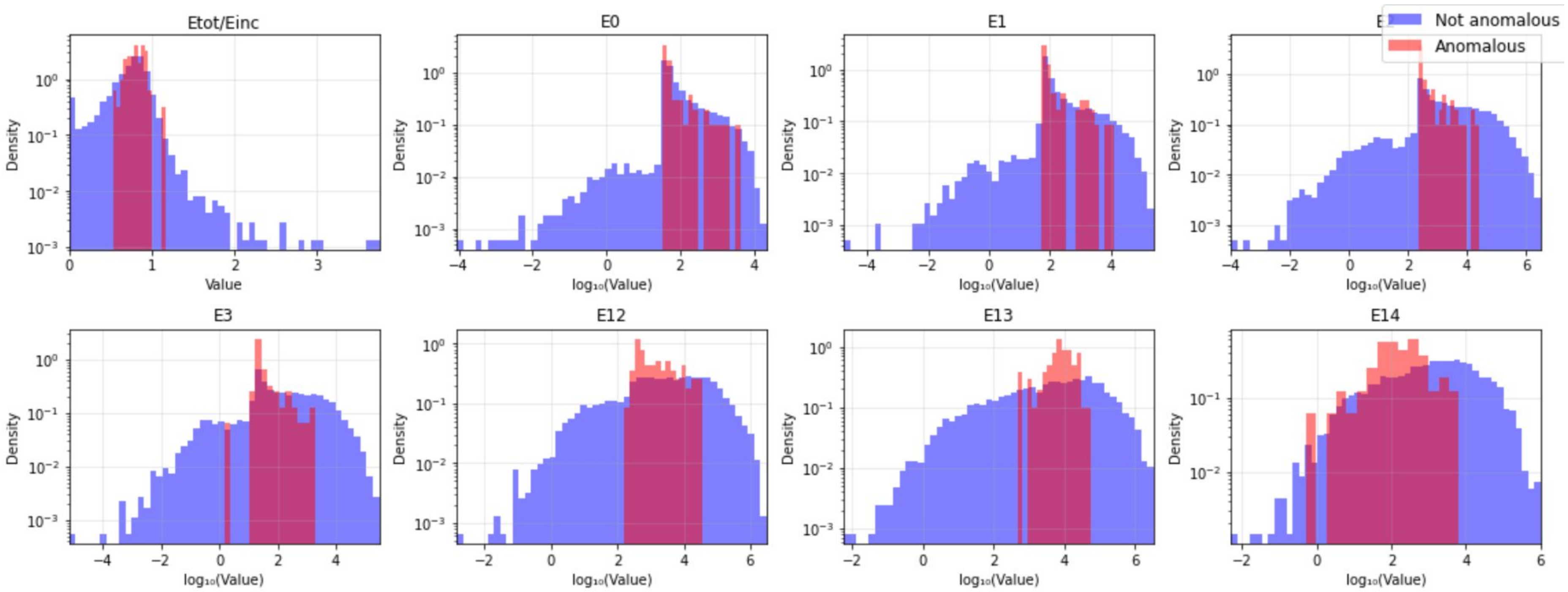
- Select not well simulated showers
  - Dimensionality reduction for visualisation
    500 voxels (normalized by total shower energy) ->
    50 features after U-map ->
    2 features after TSNE
  - Use anomalous score from Isolation Forest as a metric of “normality” of a shower
    Trained on the Geant4 data, applied to both Gean4 and CaloINN generated showers
- Improve the simulaiton



Smaller the score, more anomalous are the showers

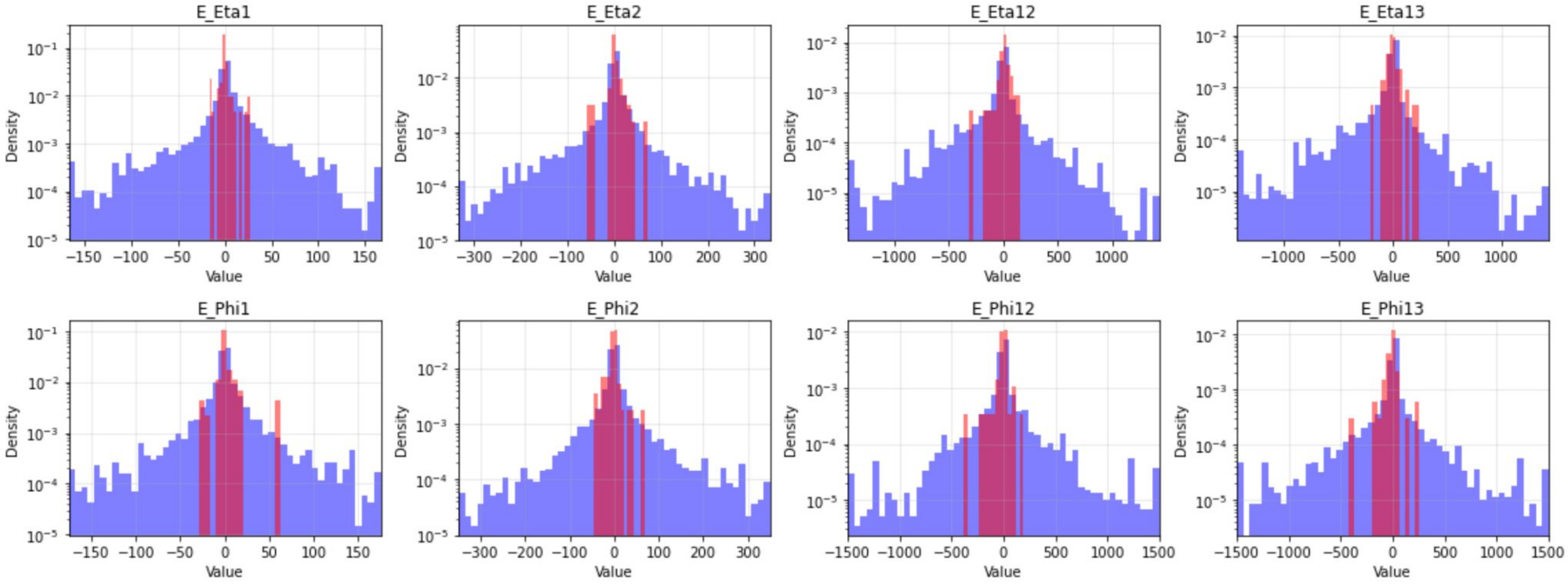
Thank you

Refine distribution tails



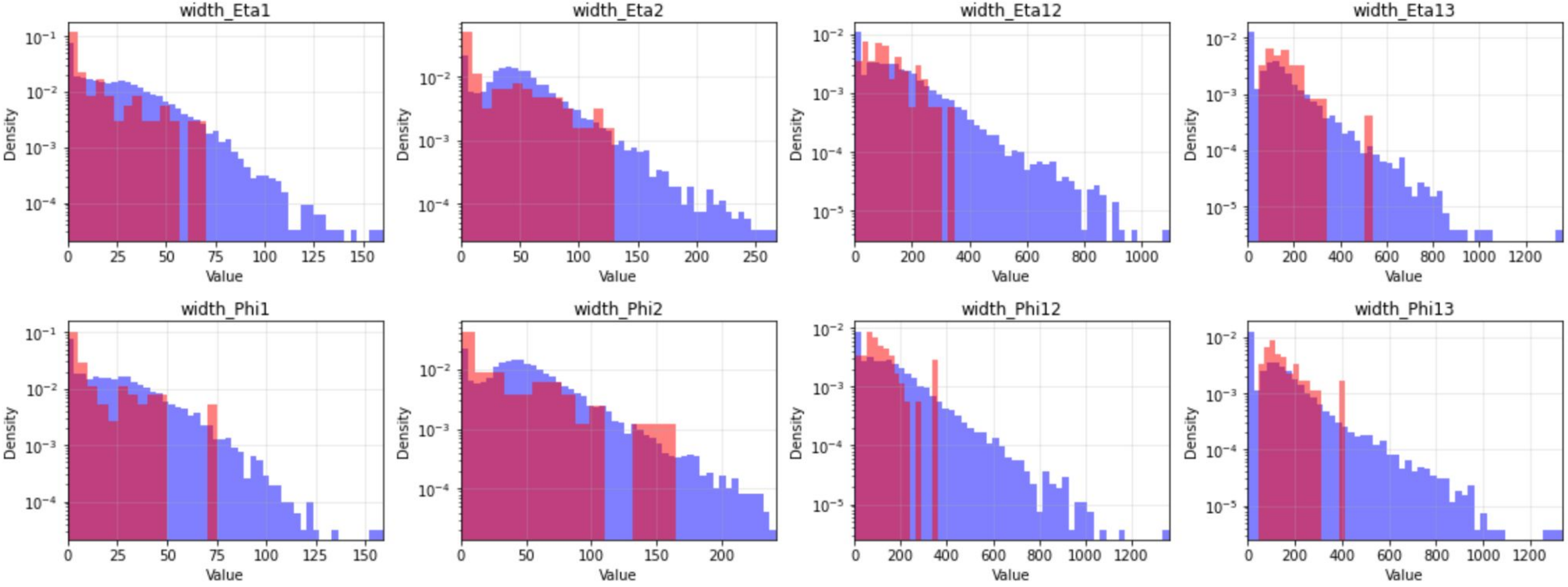
Geant4 shower features for anomalous and non-anomalous showers

Refine distribution tails



Geant4 shower features for anomalous and non-anomalous showers

Refine distribution tails



Geant4 shower features for anomalous and non-anomalous showers