TomOpt:
Differentiable Optimisation of Muon-Tomography Detectors

Giles Strong, on behalf of the TomOpt Authors*

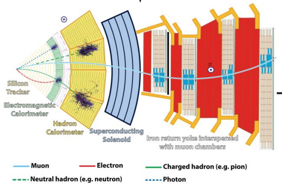
ICHEP 2022, Bologna, Italy - 09/07/22

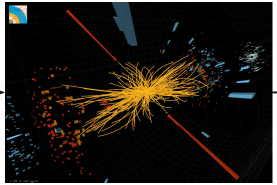
*G Strong, T Dorigo, A Giammanco, P Vischia,
J Kieseler, M Lagrange, F Nardi, H Zaraket,
M Lamparth, F Fanzago, O Savchenko, & A Bordignon.

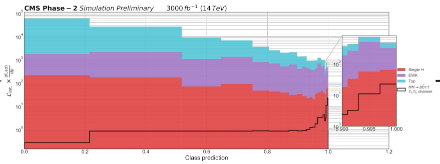
TYPICAL LHC PROCESSING CHAIN

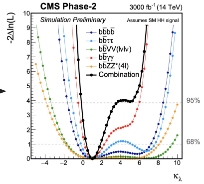
Particles
(real/simulated)

Isolated optimisation → paired / end-to-end optimisation?









Detection
(real/simulated)

- Track first, destroy later
- Kinematic precision
- Dedicated sub-detectors
- Design convenience over analysis convenience

Reconstruction

- Generic optimisation of algorithms
- Fixed working points
- Expert-interpretable data-representations (PID)

Analysis

- Signal/background separation

Measurement

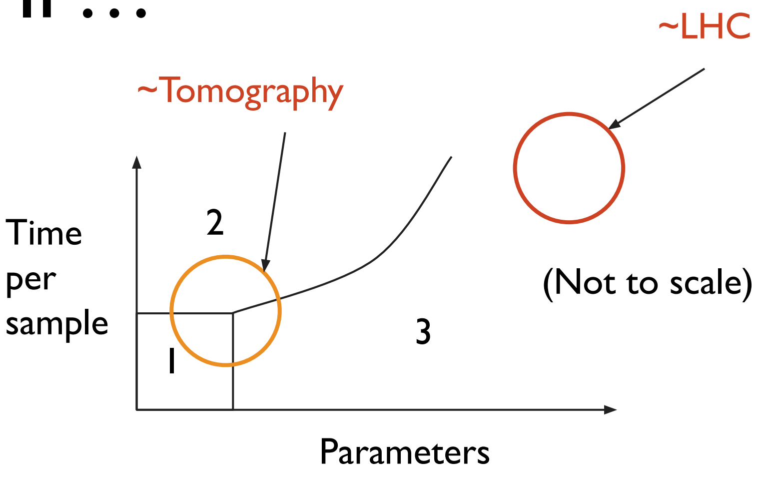
- Domain-driven categorisation
- Separate by decay channel, combine later

Many of these are "necessary evils" for HEP! Time, interpretation, MC corrections, etc.

Sources: detector cds.cern.ch/record/2120661/, reco cds.cern.ch/record/1406073, analysis & measurement CMS-FTR-18-019

WHAT IF…

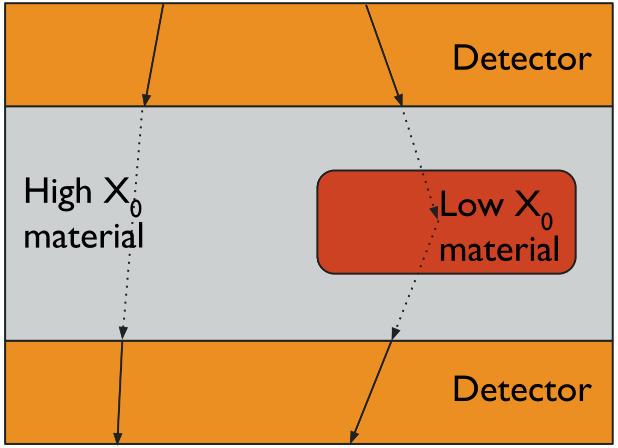
- We can already do measurement-aware analysis optimisation, e.g.:
    - INFERNO (de Castro & Dorigo, 2018)
    - NEOS (Simpson & Heinrich, 2022)
- What about going further?
    - Measurement-aware detector-optimisation
- CERN LHC-style detectors = huge-parameter space + complicated simulation and analysis algorithms
    - Let's start with a simple use-case: muon tomography



1. Grid/random search
2. Bayesian optimisation, Simulated annealing, genetic algorithm, particle swap optimisation, …
3. Gradient-based optimisation: Newtonian, gradient descent, BFGS, …

TOMOGRAPHY VIA MULTIPLE SCATTERING

- Consider a volume with unknown composition
  - E.g. Shipping container, archeological site, nuclear waste, industrial machinery
  - Want to infer properties of the volume:
    - E.g. build a 3D map of elemental composition
- Cosmic muons scattered by volume according to radiation-length (X_0 [m]) of elements in material
  - Measure muons above and below volume
  - Kinematic changes provide info on material composition



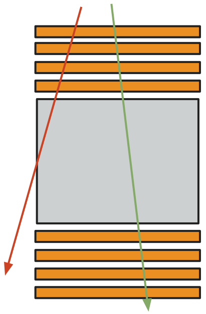
High X_0 = low scattering

Low X_0 = high scattering

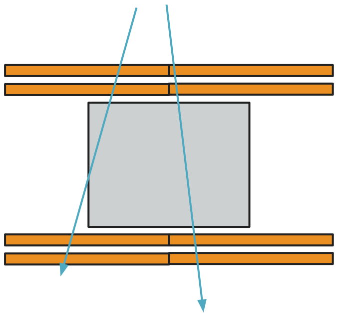
X_0 = average distance between scatterings

PROBLEM

- Each use-case likely to have a budget:
  - E.g. financial, heat, power, spatial, imaging time
- How should detectors be positioned to best function in each use case subject to constraints?
- Domain knowledge, experience, and intuition can help
  - But solutions likely to be based on heuristics and proxy objectives (e.g. lowest uncertainty on muon-path angles)



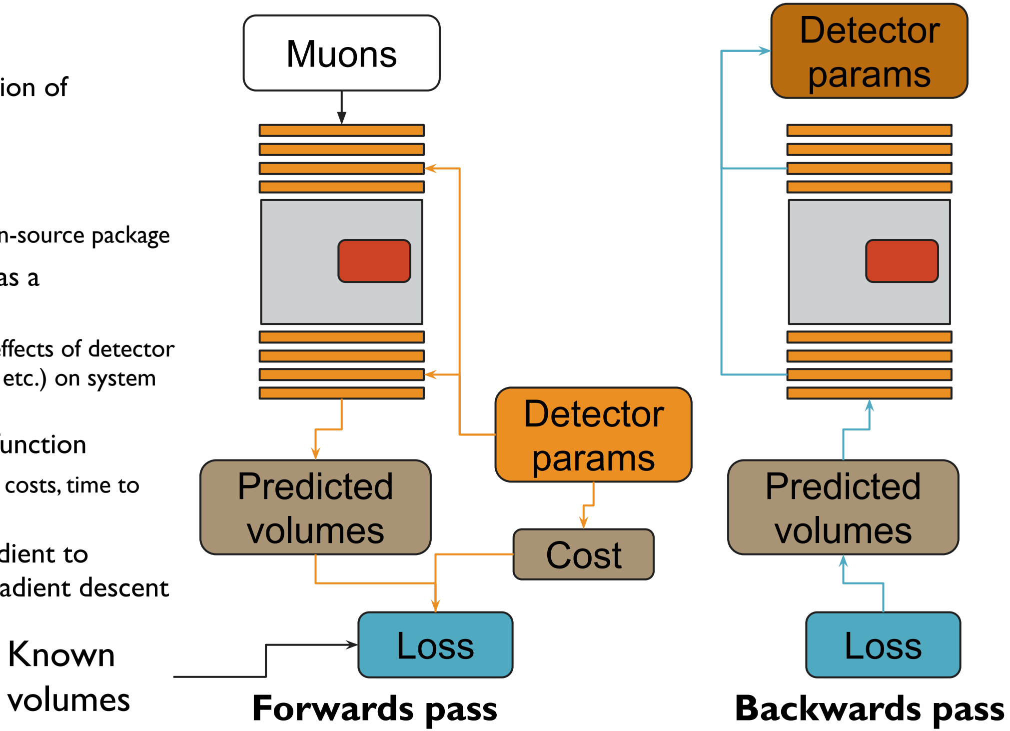
Example 1: Muons measured precisely but less efficiently



Example 2: Muons measured less precisely but more efficiently

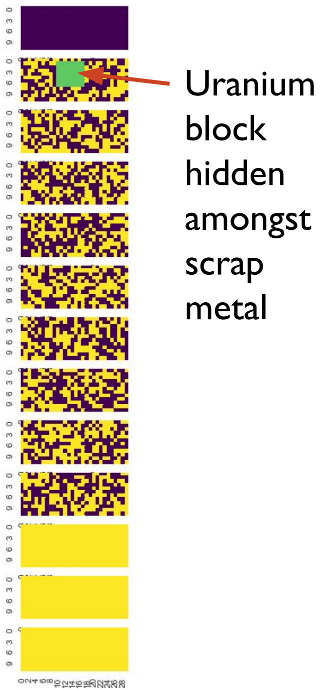
TOMOPT

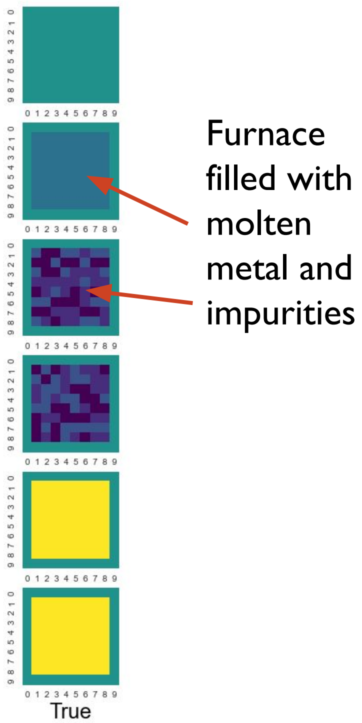
- Python package for differential optimisation of muon-tomography detectors
  - Modular design
  - PyTorch provides autodiff
  - Still underdevelopment; aim is an open-source package
- First, express the entire inference chain as a differentiable system
  - We can now compute the analytical effects of detector parameters (position, size, resolution, etc.) on system outputs
- Now express the desired task as a loss function
  - E.g. error on X_0 predictions, detector costs, time to achieve desired resolution
- We can now backpropagate the loss gradient to detector parameters and optimise via gradient descent
  - Just like a neural network



PASSIVE VOLUME SPECIFICATION

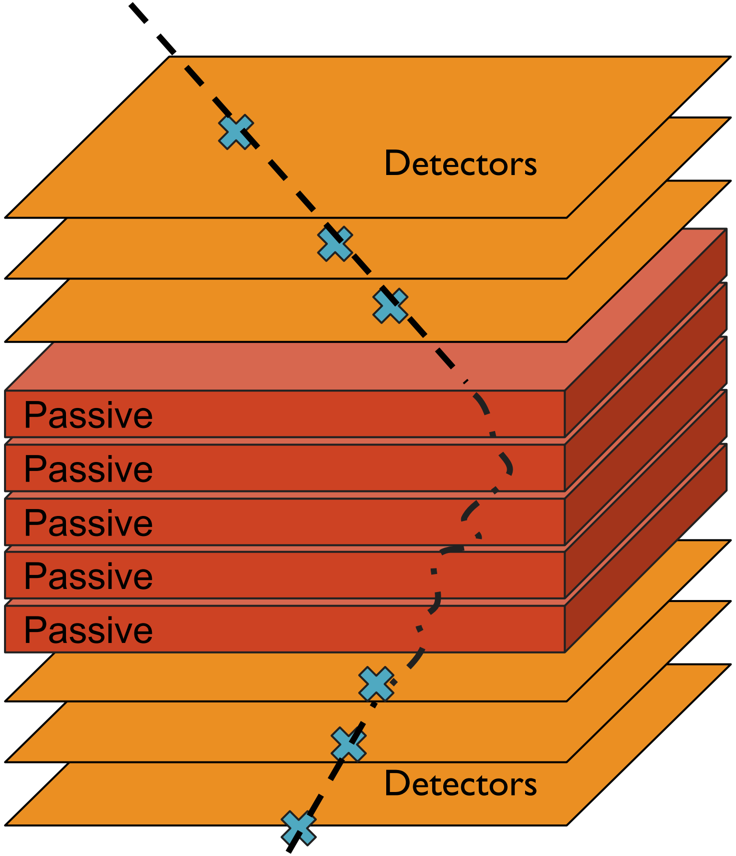
- We can simulate different passive volumes by splitting the space in voxels
  - Each voxel can be a different material
- Muons are scattered according to material density (X_0) of the voxels they pass through
- We can randomly generate typical volumes with pre-specified characteristics
  - These can help simulate different tasks and situations





DETECTOR SPECIFICATION

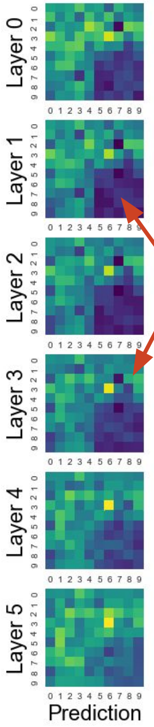
- Detectors panels are placed above and below the volume
- Each panel records a hit when the muons passes through
- We will aim to learn the optimal size and position of each detector panel



WORK IN PROGRESS!

INFERENCE

- First: use the hits to reconstruct the muons entering and exiting the volume
- Then: use the changes in muon trajectory to infer properties of the volume
    - Could simply predict the X0 of each voxel
    - Even better: compute a task-specific summary statistic
    - We can also include deep learning algorithms here

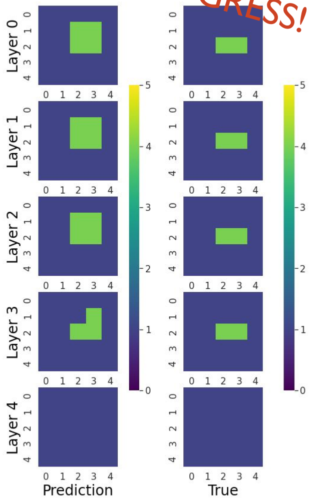


Voxel X0 predictions

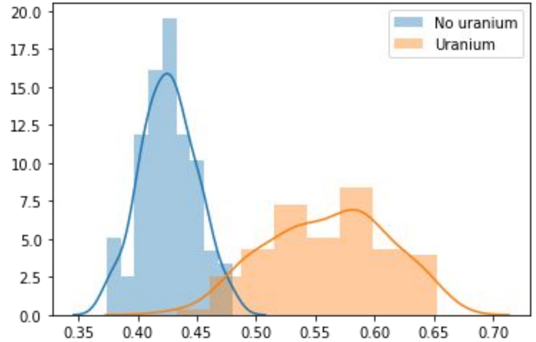
High density block

Low density background

Voxel-wise GNN prediction for material class



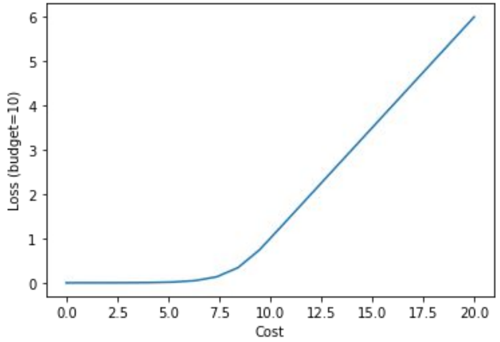
Dedicated summary-statistic for classifying volumes with uranium blocks



LOSSES AND COST

- The loss of the system should contain two components:
  - The error on the predictions
    - E.g. MSE for voxel X_0, or cross-entropy for class predictions
  - The cost of the detectors
    - Cost component smoothly "turns on" near target budget
      - Heavily penalises over-budget detectors
    - Loss scaled according to error loss
- Treat detector just like a neural network:
  - Differentiate the loss w.r.t. the learnable (detector) parameters and update with gradient descent

$$\mathcal{L}_{\text{Error}} = \frac{1}{N_{\text{voxels}}} \sum_{i=1}^{N_{\text{voxels}}} \frac{\left(X_{0,i,\text{True}} - X_{0,i,\text{Pred.}}\right)^2}{w_i}$$

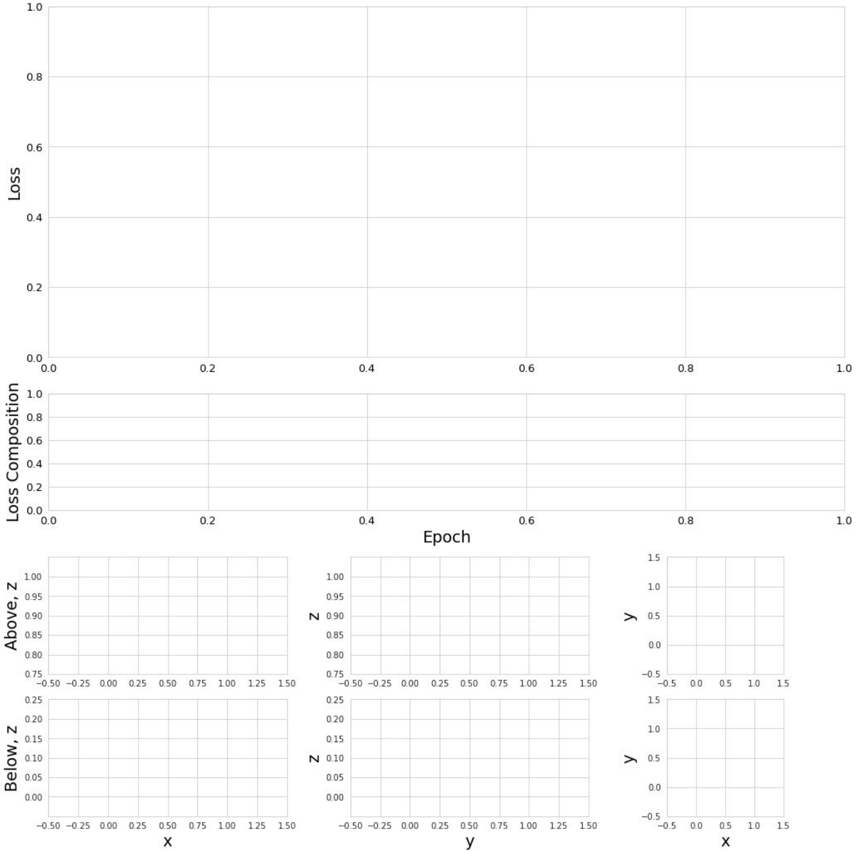


$$\mathcal{L} = \mathcal{L}_{\text{Error}} + \alpha \mathcal{L}_{\text{Cost}}$$

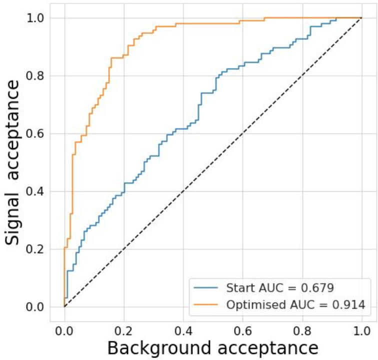
WORK IN PROGRESS!

EXAMPLE

- Task is to infer presence of uranium block in container filled with scrap metal
    - Inference uses a dedicated summary statistic
    - The U block can be anywhere in the volume, so intuitively expect the detectors should be placed centrally in XY over the volume
- Detectors start in corner of volume and optimisation does indeed move them to cover the volume



- Optimised detector provides large improvement to ROC AUC



SUMMARY

- Measurement-aware detector-optimisation = challenging but rewarding task
    - Doesn't aim to replace detector experts; provide tools to make more informed design choices
    - Currently testing on a simplified case: muon tomography
- TomOpt indicates this is possible, and is under rapid development
    - Publications and open-source package this year

GETTING INVOLVED

- MODE Collaboration involved in several other projects:
  - ECal, hybrid HCal, Cherenkov arrays, …
  - Recent whitepaper arXiv:2203.13818
  - Open to new members (contact)
  - TomOpt also welcoming new contributors: giles.strong@outlook.com
- Second MODE workshop on differentiable programming
  - 12-16 September, Crete & online
  - https://indico.cern.ch/event/1145124/

Overview of the sessions:

- Confirmed keynote speakers
  - Adam Paszke (Google Brain): DEX
  - Max Sagebaum (TU Kaiserslautern): High-performance Algorithmic Differentiation
- Lectures and tutorials:
  - Lecture: Differentiable Programming, Gradient Descent in Many Dimensions, and Design Optimization (Pietro Vischia, UCLouvain)
- Special events:
  - Hackathon (Giles Strong, INFN Padova): the challenge will open on 1st August 2022, and submissions will be open until September 5th, 2022. **prizes (see below) will be given to the winners of the challenge!**
  - Poster session: **prizes (see below) will be given to the best posters!**
- Applications in muon tomography
- Progress in Computer Science
- Applications and requirements for particle physics
- Applications and requirements in astro-HEP
- Applications and requirements for neutrino detectors
- Applications and requirements in nuclear physics experiments
- Discussion on the status and needs of the discipline (one parallel session per each of the other sessions)

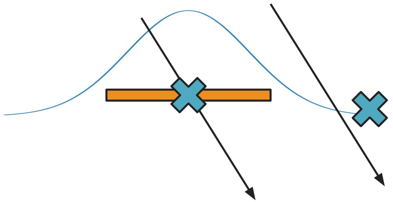
SUMMARY

- Measurement-aware detector-optimisation = challenging but rewarding task
  - Doesn't aim to replace detector experts; provide tools to make more informed design choices
  - Currently testing on a simplified case: muon tomography
- TomOpt indicates this is possible, and is under rapid development
  - Publications and open-source package this year

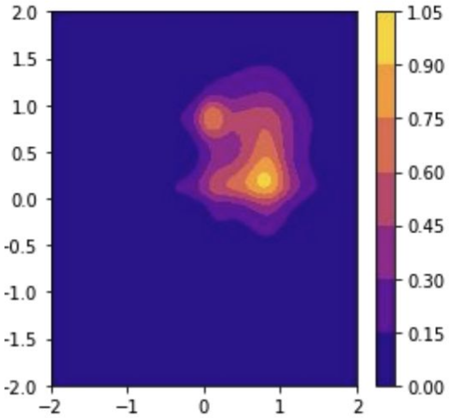
BACKUPS

DETECTOR MODELLING

- Assume commercial detectors ⇒ fixed resolution, fixed efficiency, fixed cost per m^2
- Optimise XYZ position and XY span
- But, muons either hit or miss detectors. How can we make hits be differentiable w.r.t detector parameters?
- Instead, let resolution and efficiency be distributed, e.g. Gaussian centred on panel, with width set by panel span
  - The PDF at the muon position is now diff. w.r.t panel position and span
- Can further generalise by using Gaussian Mixture model



Both muons recorded, but with different resolutions

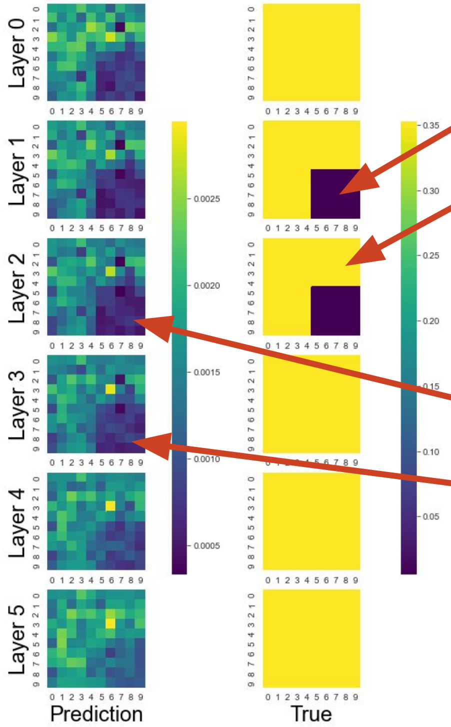


Plot: Max Lamparth

WORK IN PROGRESS!

VOLUME INFERENCE: POCA

- Point of Closest Approach: Assign entirety of muon scattering to single point
  - Invert analytic scattering model to compute X_0
  - Average X_0 predictions in each voxel
- We know, though, that the muon scattering results from multiple interactions throughout the volume
  - Assigning the whole scattering to a single point inherently leads to underestimating the X_0
  - Can slightly improve by weighting muon predictions by their X_0 uncertainty
  - Can also allow muons to predict in multiple voxels according to their PoCA uncertainty



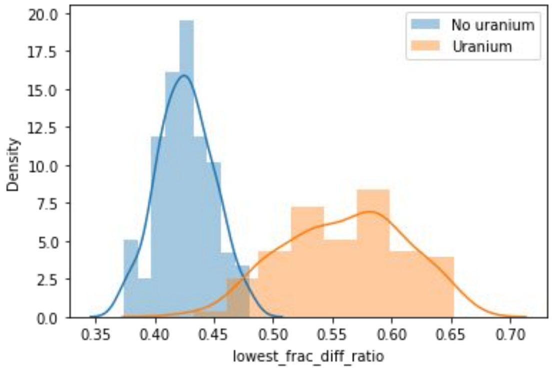
- Block of lead (X_0=0.005612m)
- Surrounded by beryllium (X_0=0.3528m)
- Predictions highly biased to underestimate X_0
- Lead block clearly visible
- but high z uncertainty in scatter location causes 'ghosting' above and below

WORK IN PROGRESS!

VOLUME INFERENCE: SUMMARY STATISTIC

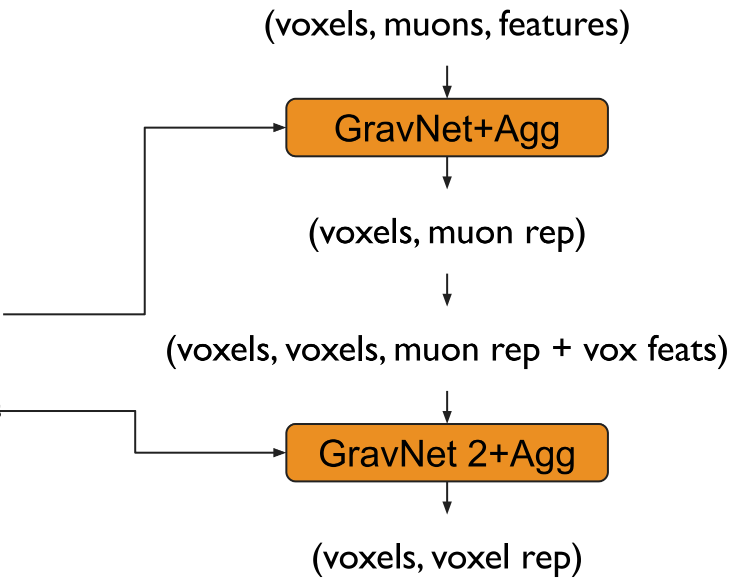
- In some cases, we don't care about predicting voxel X_0 values, but instead determining some higher-level property of the volume
  - E.g. is there uranium located anywhere in the volume?
- For this we can try to construct a summary statistic based on the X_0 predictions
- Statistics must be fully differentiable
  - Ideally, should also be invariant to scale X0 predictions, to mitigate PoCA bias

- E.g. for a uranium-block search, compare the mean of the lowest estimated to X_0 voxels to the mean of the rest
  - No block => small difference
  - Block => bimodal X_0 distribution => large difference



VOLUME INFERENCE: GNN

- Can use a deep learning approach
- Consider two-stage graph:
  - Each voxel has a graph built from muons
    - GNN+aggregation learns a representation of the muons specific to each voxel, by sharing features between muons
  - Each volume has a graph built from voxels
    - Second GNN+aggregation learns a representation of the voxels specific to each voxel, by sharing muon-representations between voxels.



WORK IN PROGRESS!

VOLUME INFERENCE: GNN

- At this point, we have a representation per voxel.
- We can transform these into X_0 predictions (class/value) with a DNN
- We can easily aggregate over the voxels to produce a volume representation.
    - This can then be further transformed into the appropriate prediction shape
- Further details in my [IML talk]

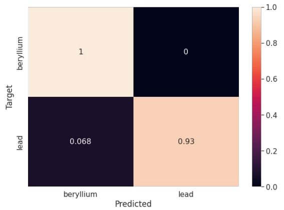
(voxels, voxel representation)

Aggregate over voxels

(volume representation)

DNN

(Volume class)



DNN

(voxels, voxel class)