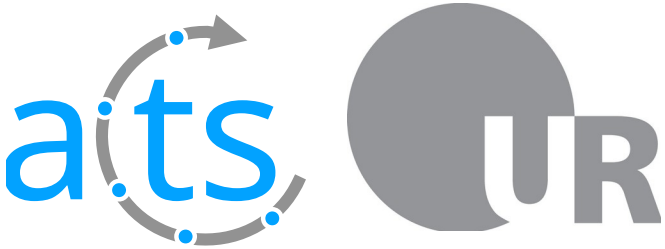
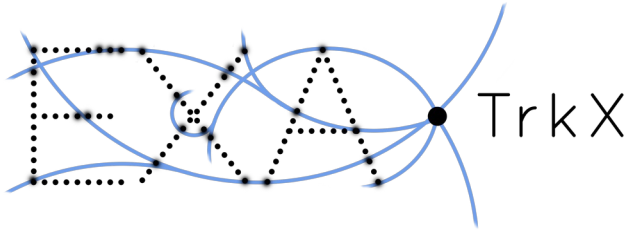
ICHEP 2022

Applying the Exa.TrkX pipeline to the OpenDataDetector with ACTS

08.07.2022





Benjamin Huth
Universität Regensburg

Andreas Salzburger (CERN),
Lukas Heinrich (TU München),
Tilo Wettig (Universität Regensburg),
Exa.TrkX authors: Alina Lazar*, Xiangyang Ju*,
Daniel Murnane*, Paolo Calafiura*
**Lawrence Berkeley National*

Content

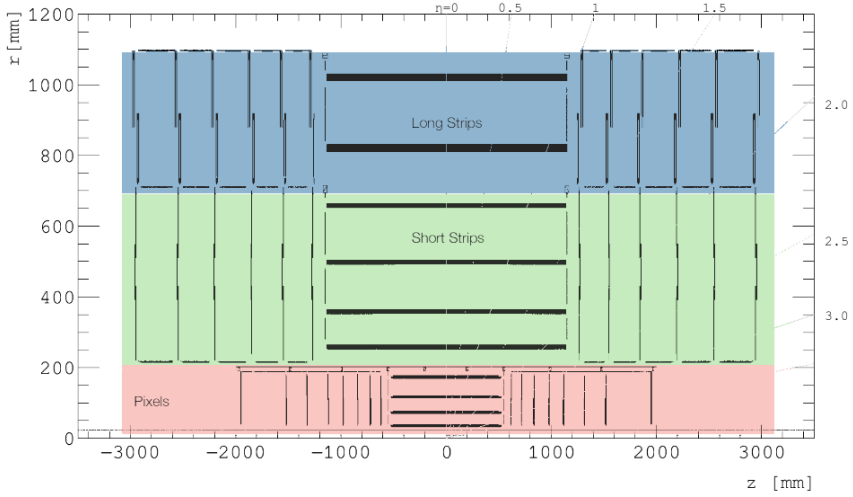
1) ACTS and the OpenDataDetector
2) The Exa.TrkX pipeline
3) Training & Performance
4) Comparison to CKF & Parameter fit
5) Summary & Outlook

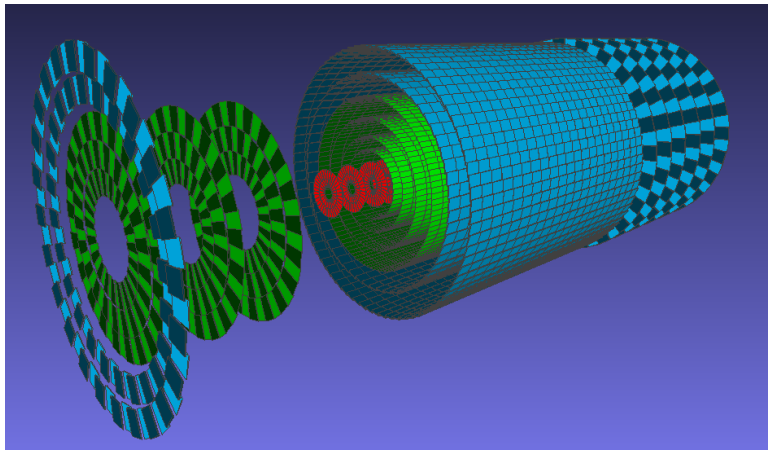
ACTS

- ACTS is an experiment-independent toolkit in modern C++ for charged particle tracking.
  - State-of-the-art implementations of standard algorithms and R&D testbed
  - Used in several experiments: ATLAS, sPHENIX, FASER
  - For more infos see here and on github.com/acts-project/acts

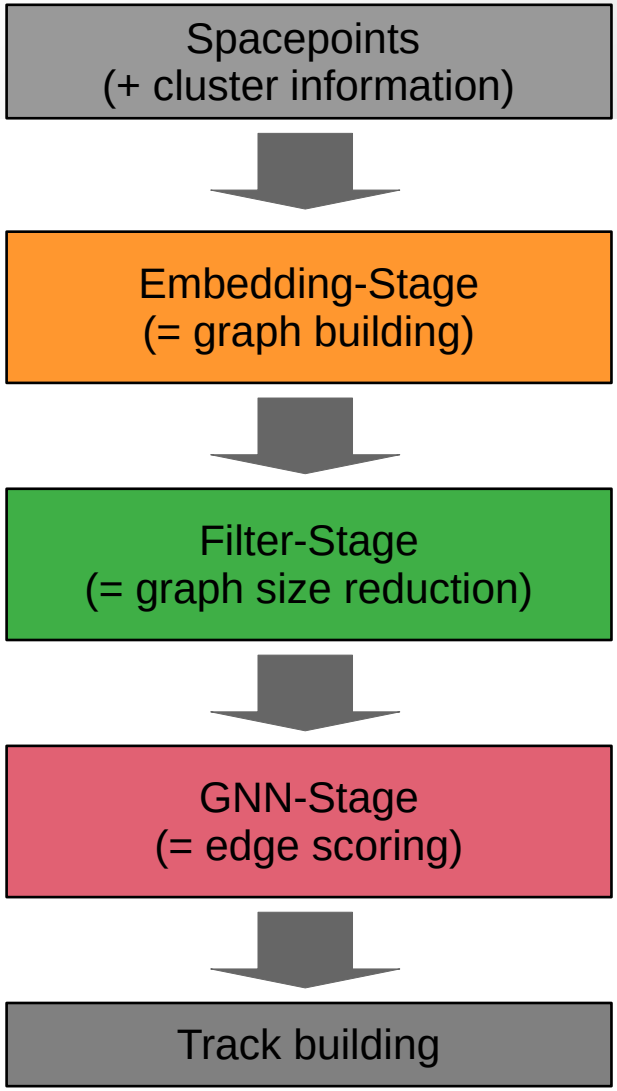
The OpenDataDetector

- The OpenDataDetector is a purely virtual, but realistic detector for testing and R&D purposes
  - Shipped with and interfaces to ACTS
- Changes wrt. the TrackML detector:
  - more precise material description
  - capability of full-simulation
  - based on DD4hep
- For more details see here (Paul Gessinger, ACAT 2021)



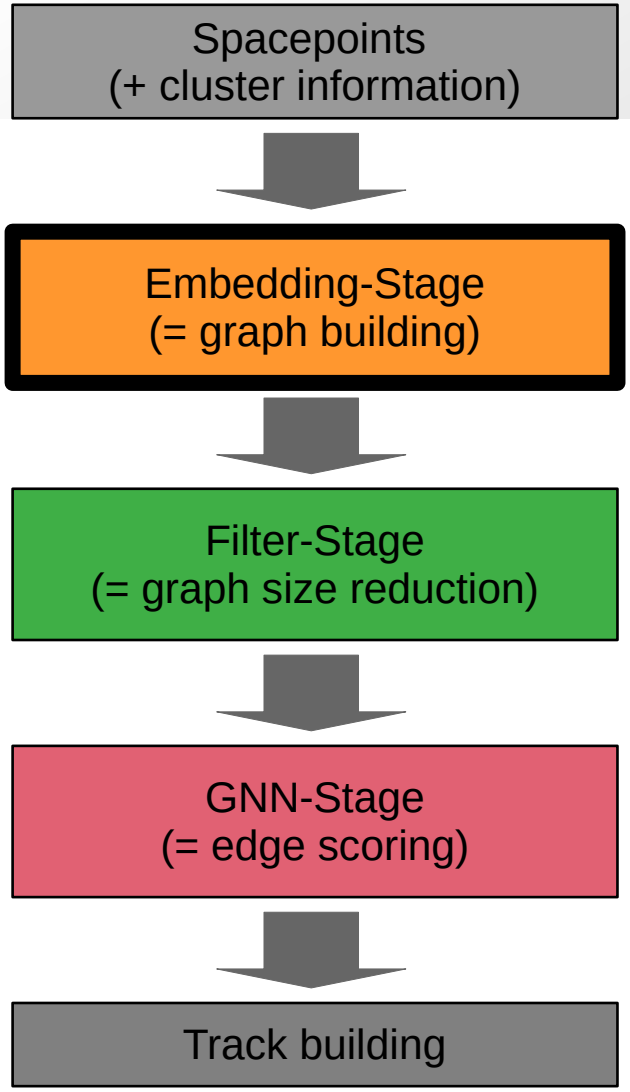


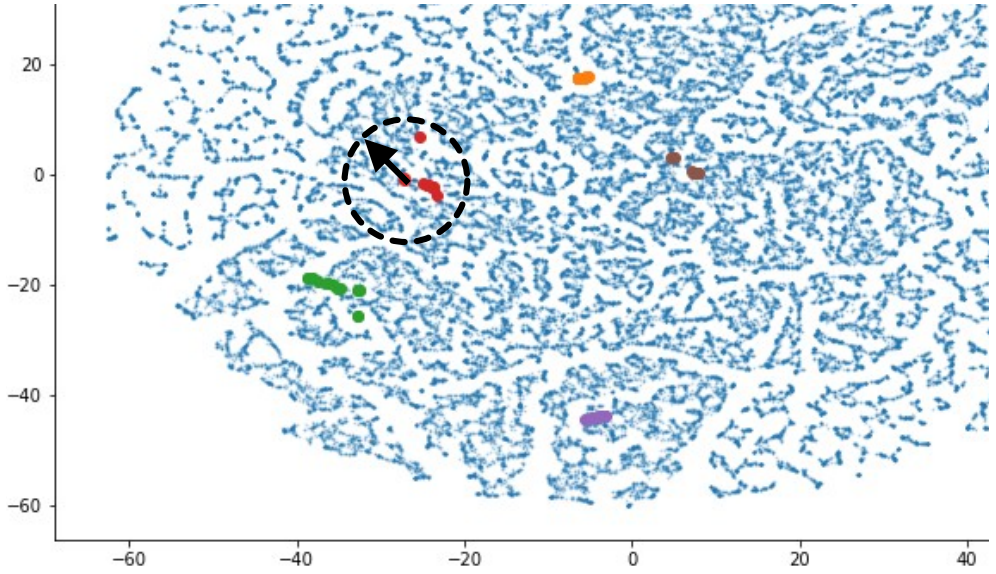
The Exa.TrkX pipeline



- Multi-stage machine-learning pipeline for **track finding**
  - Event as a graph (nodes = hits, edges = potential track segments)
  - Use ML (especially GNNs) to find edges that correspond to track segments
- Track fitting is performed by ACTS
- See e.g. here (Alina Lazar, ACAT 2021)

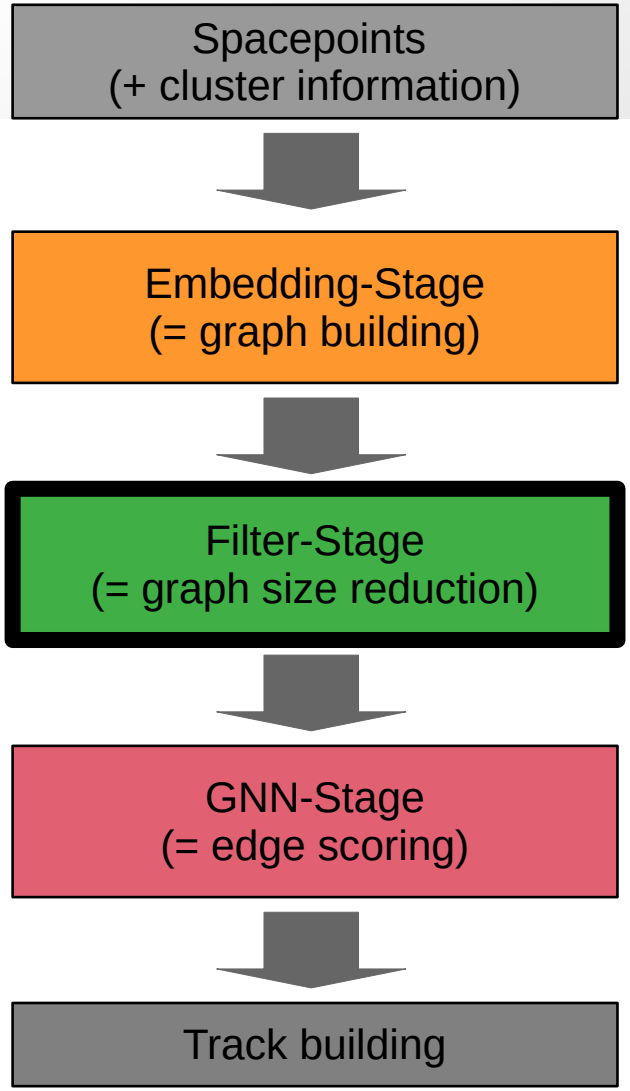
Embedding Stage





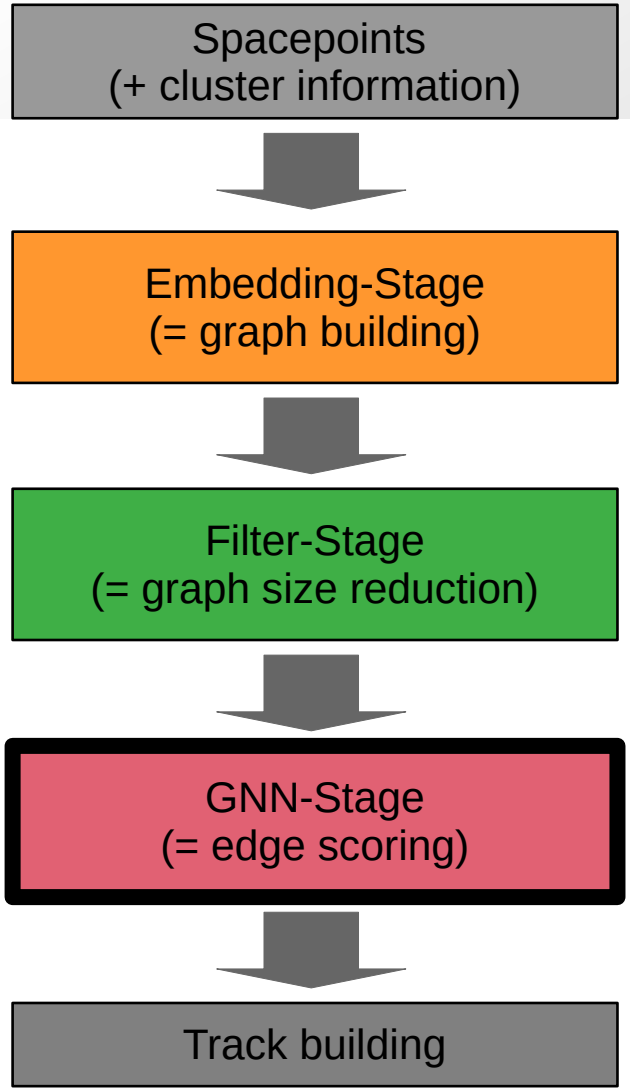
- Learn *d*-dimensional embedding that groups hits of the same track together
- Build graph with fixed-radius k-nearest-neighbor search
- Graph size: O(10M) edges, ~70K nodes

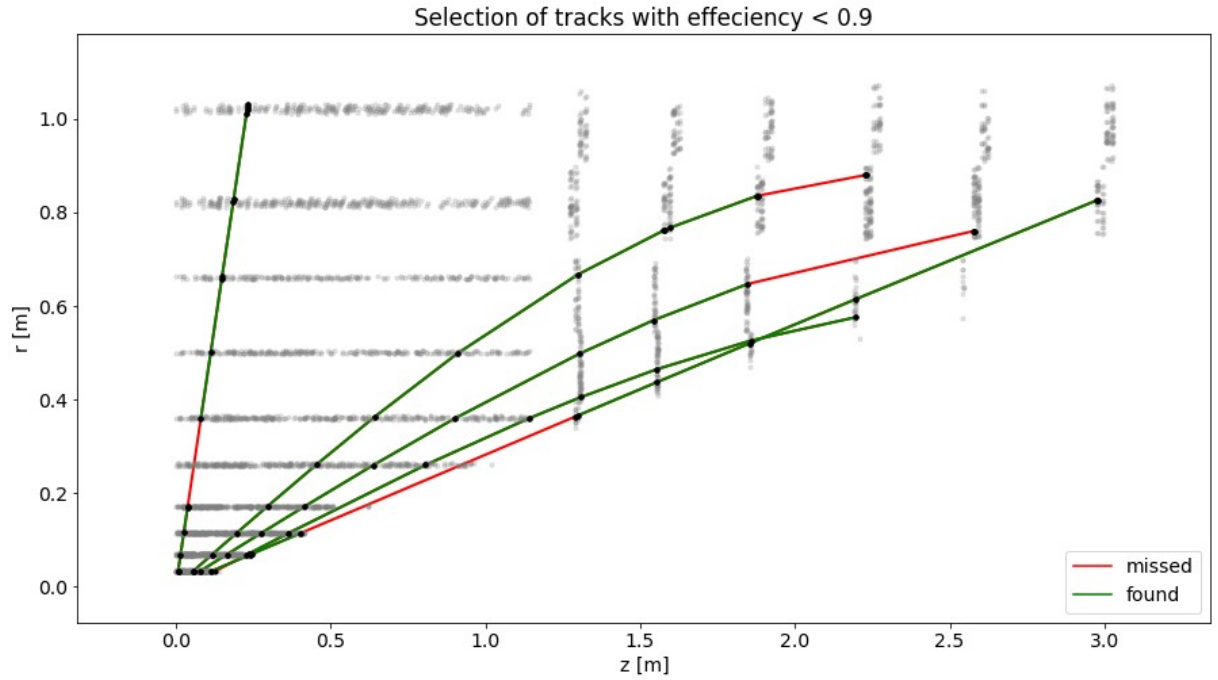
Filter Stage



- Reduce graph size:
  - Neural-Network predicts wether two „nodes“ are likely to be connected by a track
  - Necessary to keep GNN memory consumption managable
- Graph size: O(1M) edges, ~70K nodes

GNN Stage





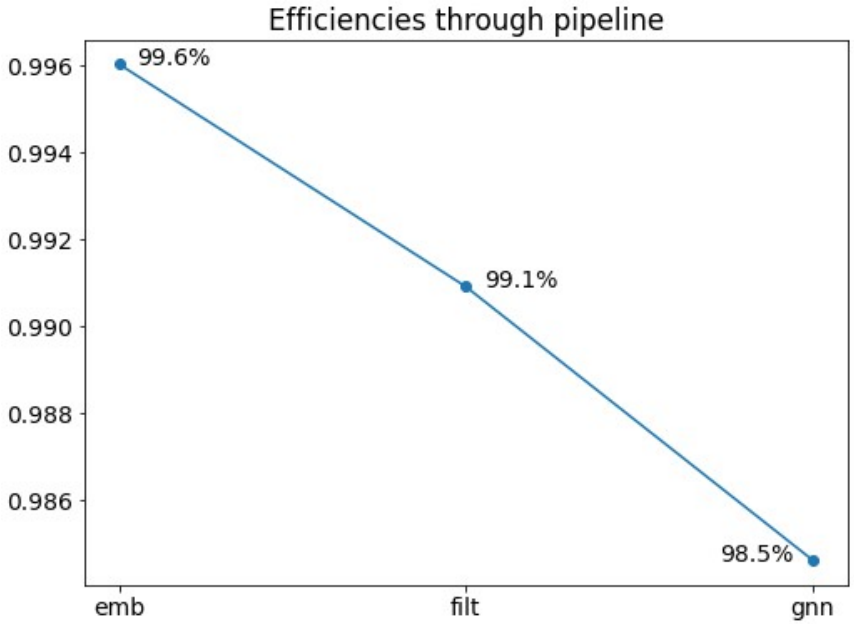
- Message-passing graph neural network + binary edge classification
- Graph size: ~70K edges, ~70K edges

Experiment setup

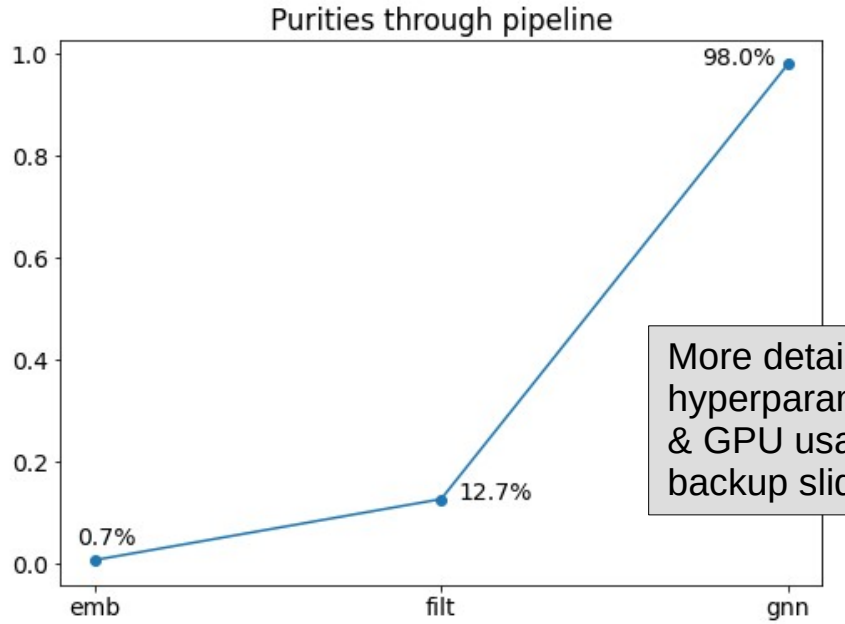
- **Physics process:**
  - Pythia8 generator: HardQCD:all + 200 pileup (HL-LHC conditions)
  - ACTS Fatras Simulation in the OpenDataDetector
  - ~7K particles per event
- **Hardware:** Nvidia A100 with 40 GB memory
- **Training software:** Training pipeline based on Pytorch, Pytorch-Lightning and ‚traintrack‘ (custom pipeline configurator)
  - Easy setup of training via traintrack (resume crashed runs, ...)
  - Adjustments where necessary to support ODD (mainly data import)
- **C++ Inference:** ACTS plugin based on TorchScript

Training

- 1000 events (950/25/25 for training/validation/test)
- Trained with **simulated hit positions** (no readout-uncertainity so far)



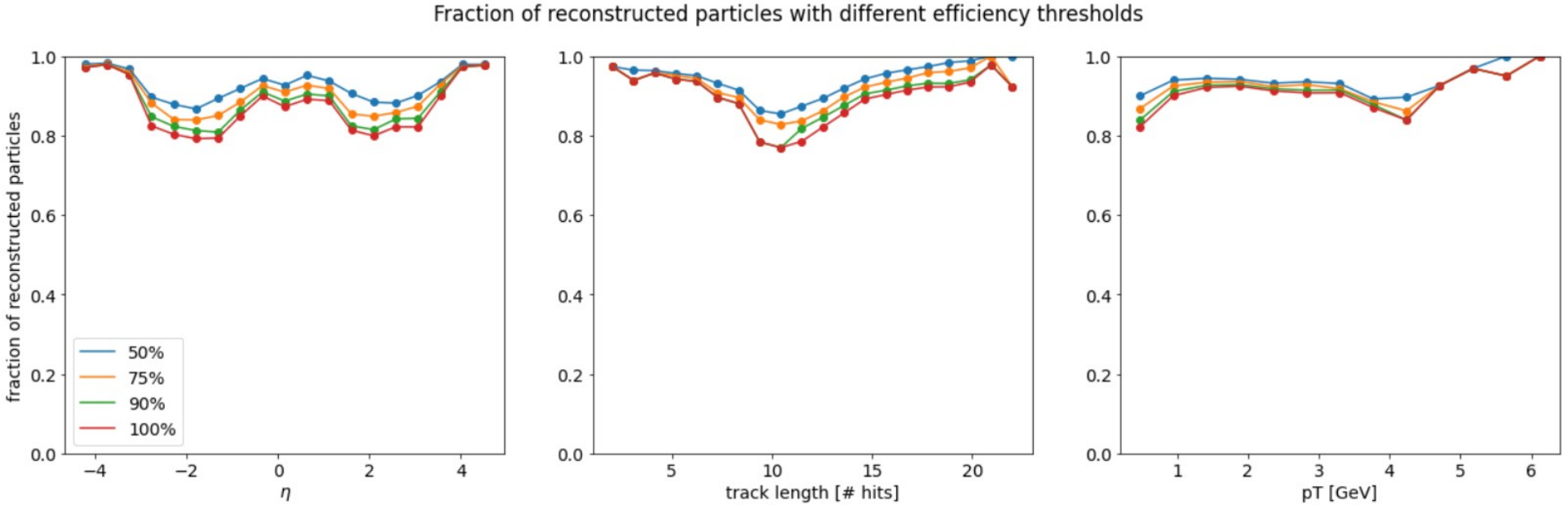
efficiency = #true-edges-in-graph / #true-edges-total



purity = #true-edges-in-graph / #all-edges-in-graph

More details on hyperparameters & GPU usage in backup slides

Inference results in ACTS



- Could be improved by:
  - Adding cluster-information to training data
  - More events / epochs for training
- Results shown for truth hit input

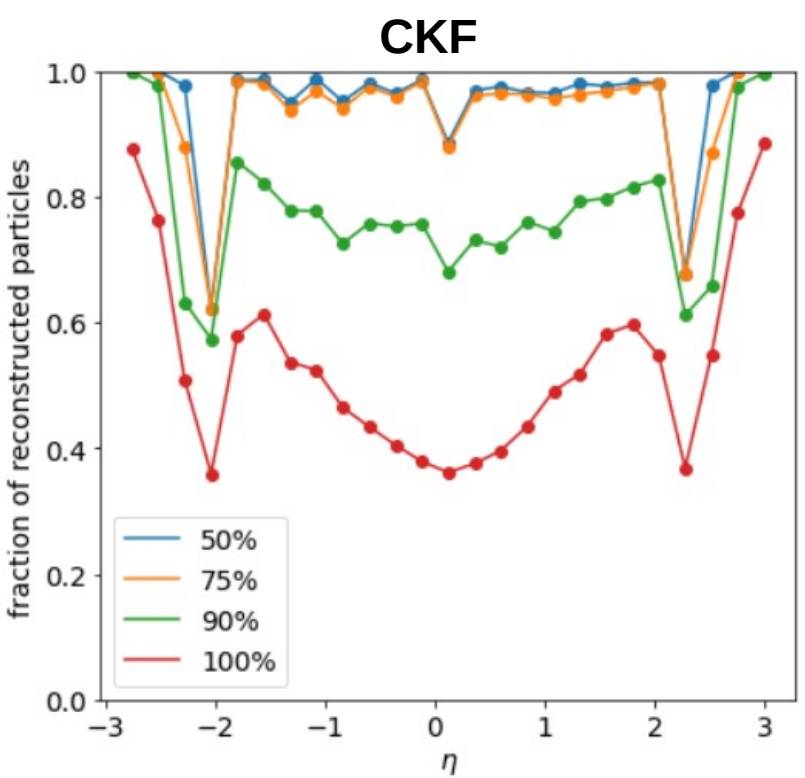
Particle-based **efficiency**:
eff = #hits-in-best-track / #hits-of-particle

Comparison with CKF*

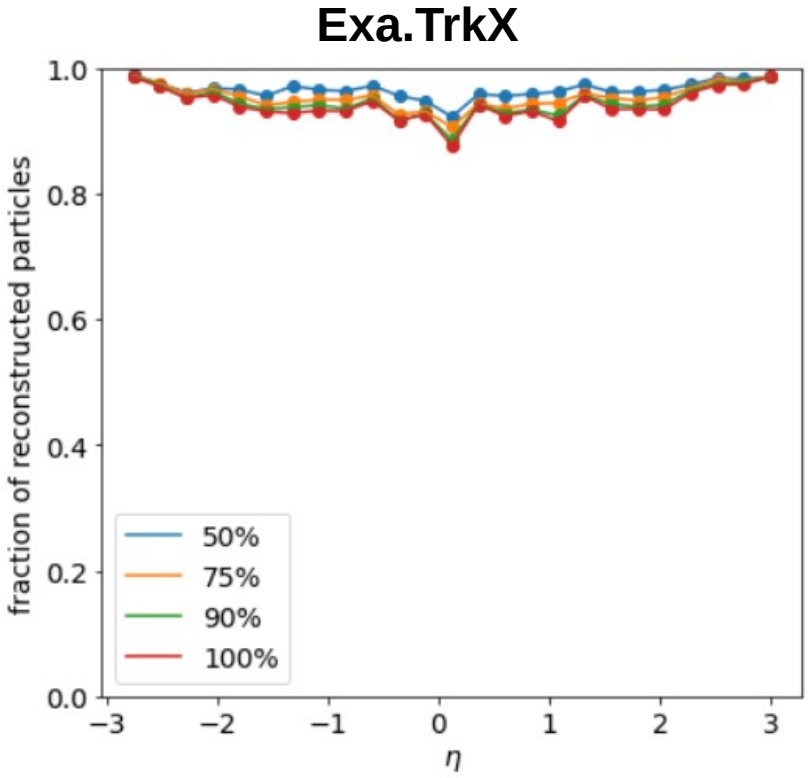
- CKF relies on
  - correct error and material description
  - tuned parameters for initialization & combinatorial search
- For this work: Auto-Tuning of 7 parameter with *Optuna***
- Applied particle pre-selection for both CKF and Exa.TrkX
  - Cut **pT < 500 MeV**
  - Cut **|eta| < 3**
  - Cut **r < 2 mm** (distance from origin in transverse plane)

* Combinatorial Kalman Filter
** See work on automatic optimization for the CKF by Rocky Bala et al here

Reconstruction: true hit positions

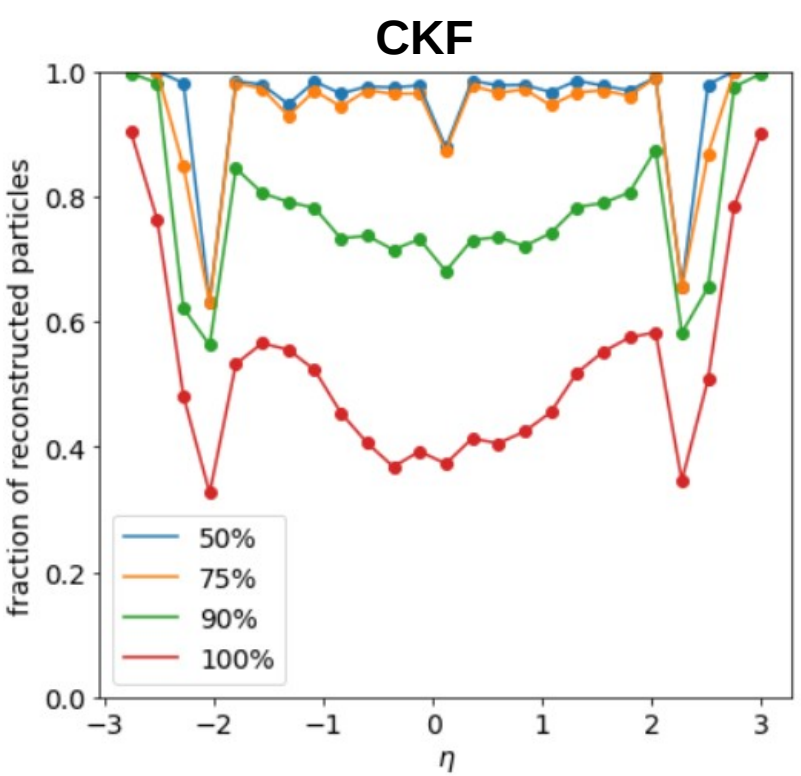


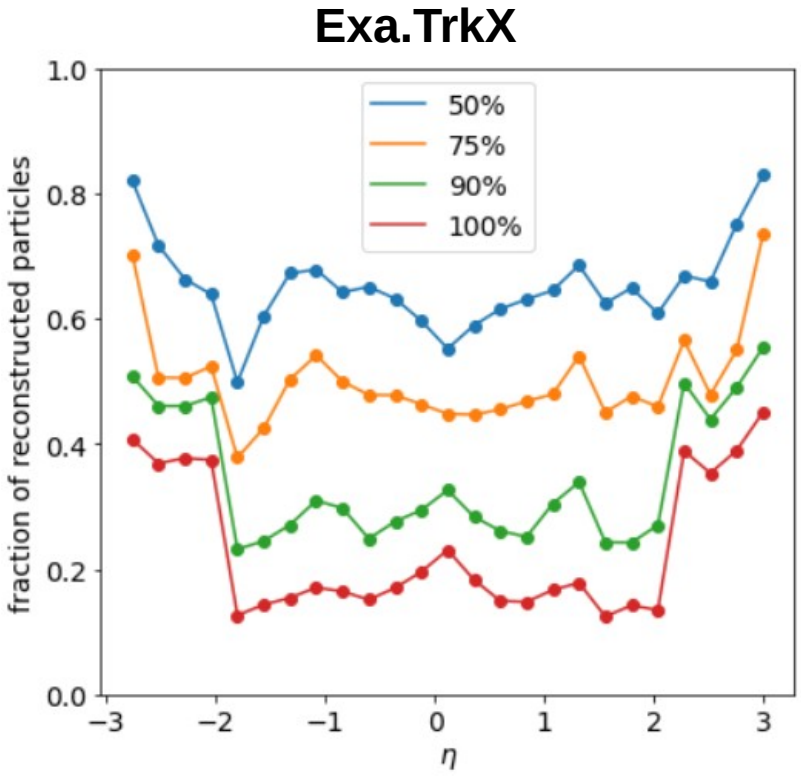
- Error description and parametrization not optimized



- Exa.TrkX profits from applied cuts

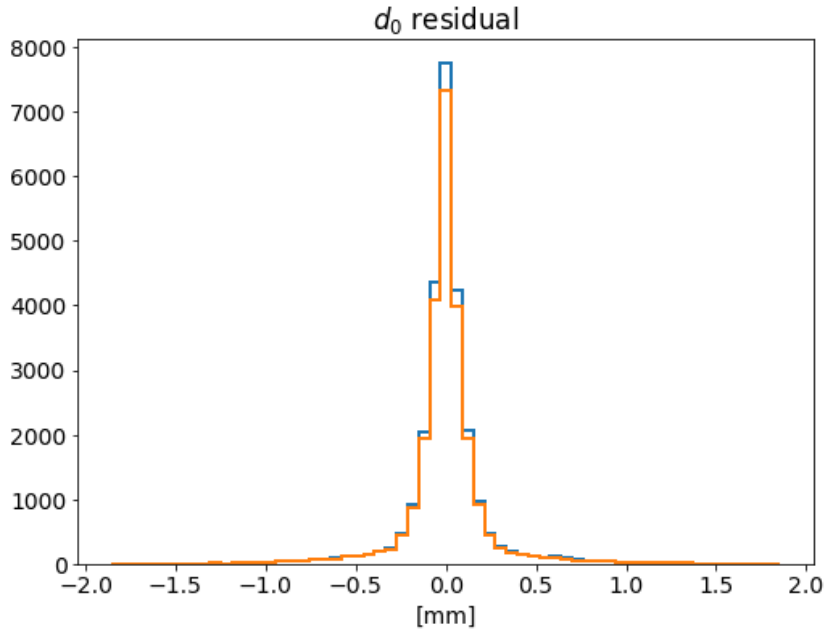
Reconstruction: smeared hits

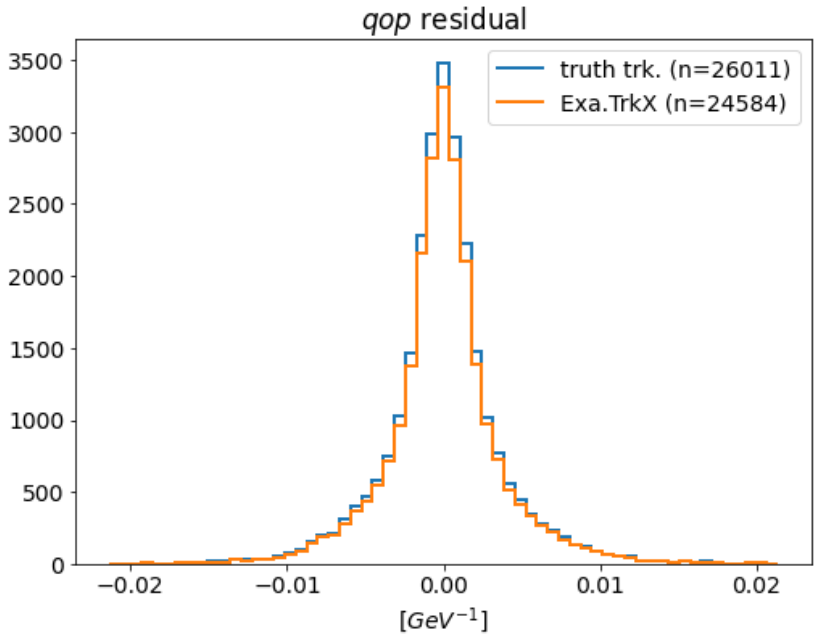




- Almost no sensitivity to hit position changes
- Influence of training with truth positions
  - very sensitive to hit positions

Parameter Fit

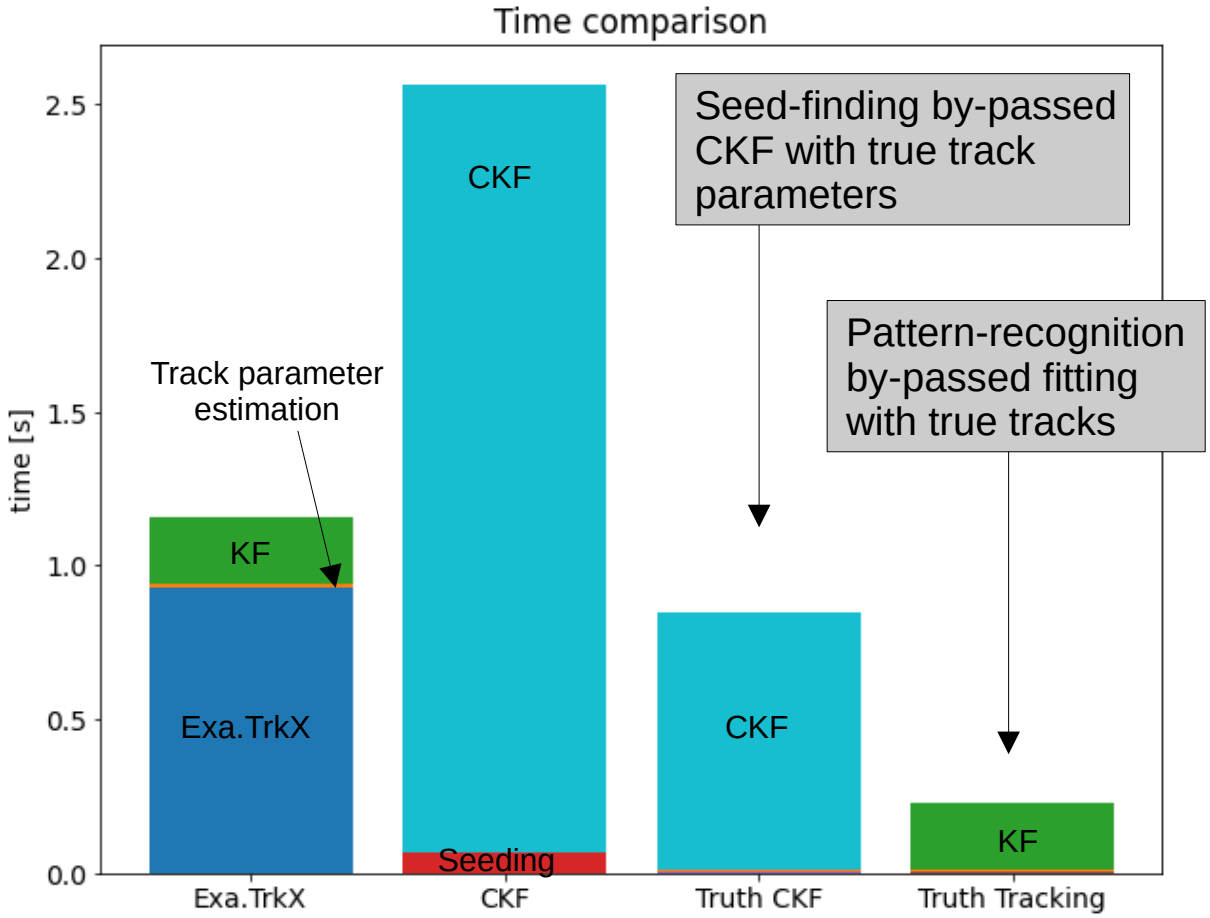




- New in this project: Combination of **Exa.TrkX** and **Kalman-Fitter (KF)**
  - True hit positions as input for Exa.TrkX inference
  - Track parameter resolution of Exa.TrkX pipeline are reaching achievable limit
- CKF not shown here (no ambiguity solution available in ACTS so far)

Timing comparison for full-chain

- GNN-based and combinatorial approach to trackfinding perform similar in **per-event timing**
- CKF & KF run on single CPU core
- Exa.TrkX can exploit available GPU hardware (see backup for details)
- True hit positions as input for CKF and Exa.TrkX inference

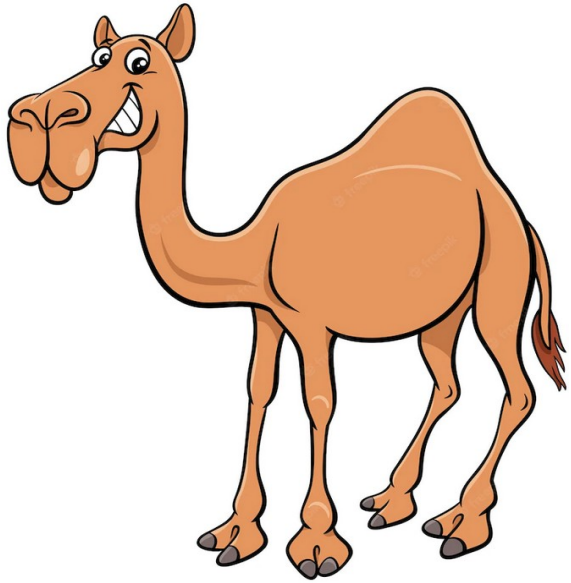


Summary & Outlook

- **Summary:**
  - Applied Exa.TrkX pipeline to the OpenDataDetector
  - Assembeled full-chain example using Exa.TrkX and the ODD
- **Next steps:**
  - Train with simulated measurements instead of true hits (and thus more input uncertainity)
  - Investigate using cluster information from pixel and strip hits
  - Update ACTS Exa.TrkX plugin and add example-code for OpenDataDetector

Thank you for your attention!

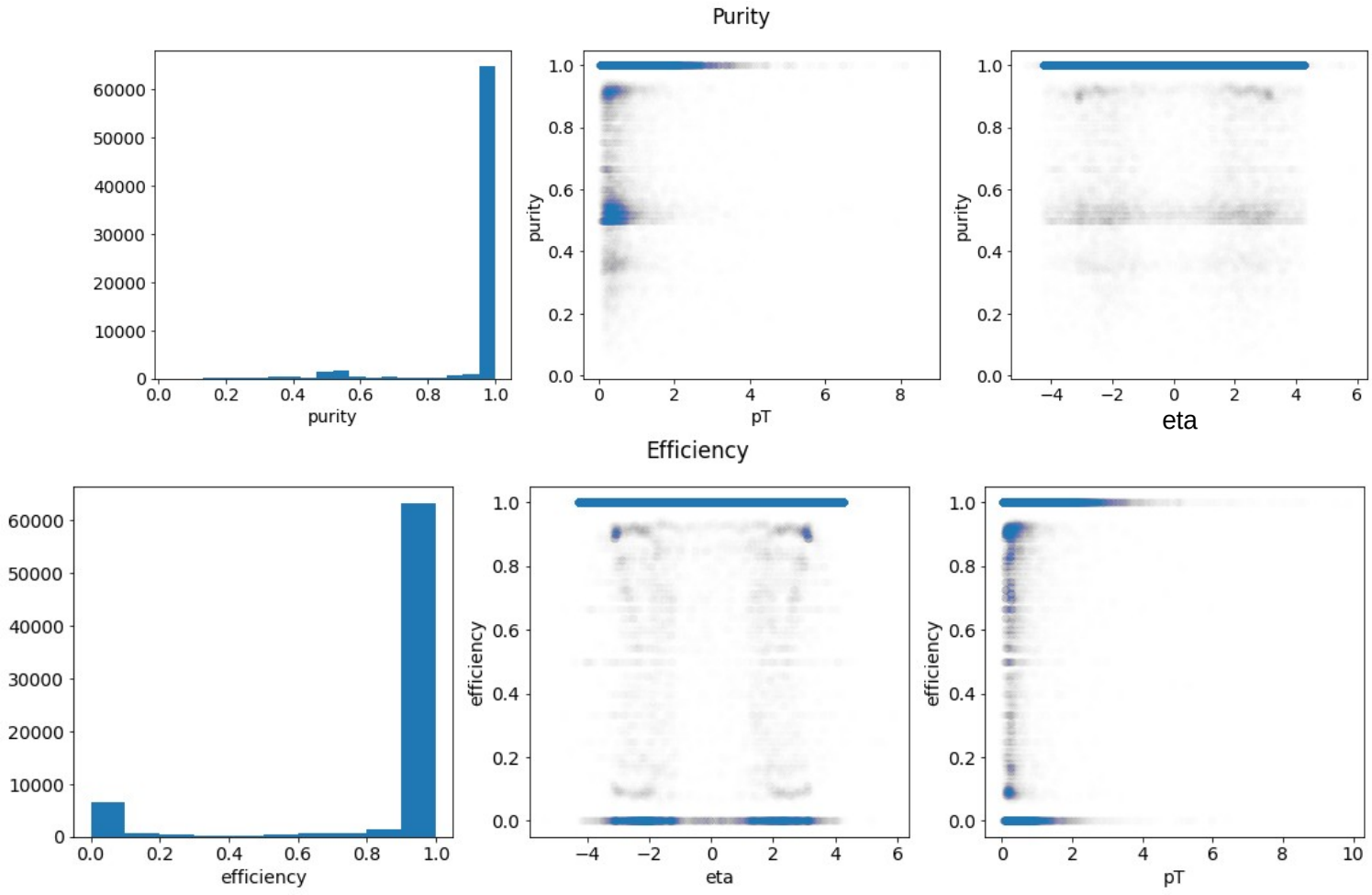
Backup



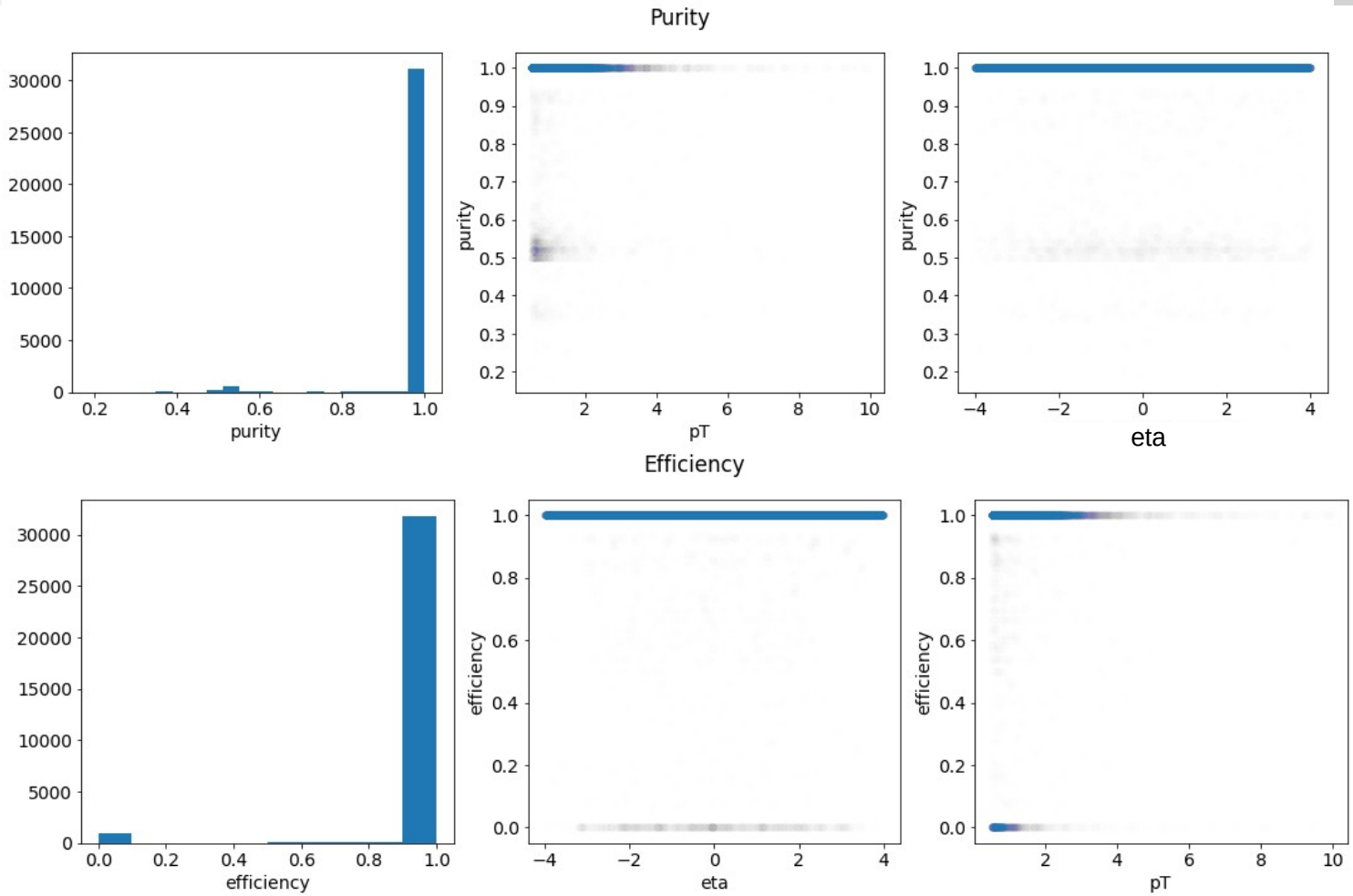
OpenDataDetector hit smearing

- Dependent on detector region:
  - Pixels: stddev_ {x,y} = 10µm
  - Short Strips: stddev_x = 45µm, stddev_y = 1.2mm
  - Long Strips: stddev_x = 60µm, stddev_y = 3.6mm
- Geometric digitization not validated so far

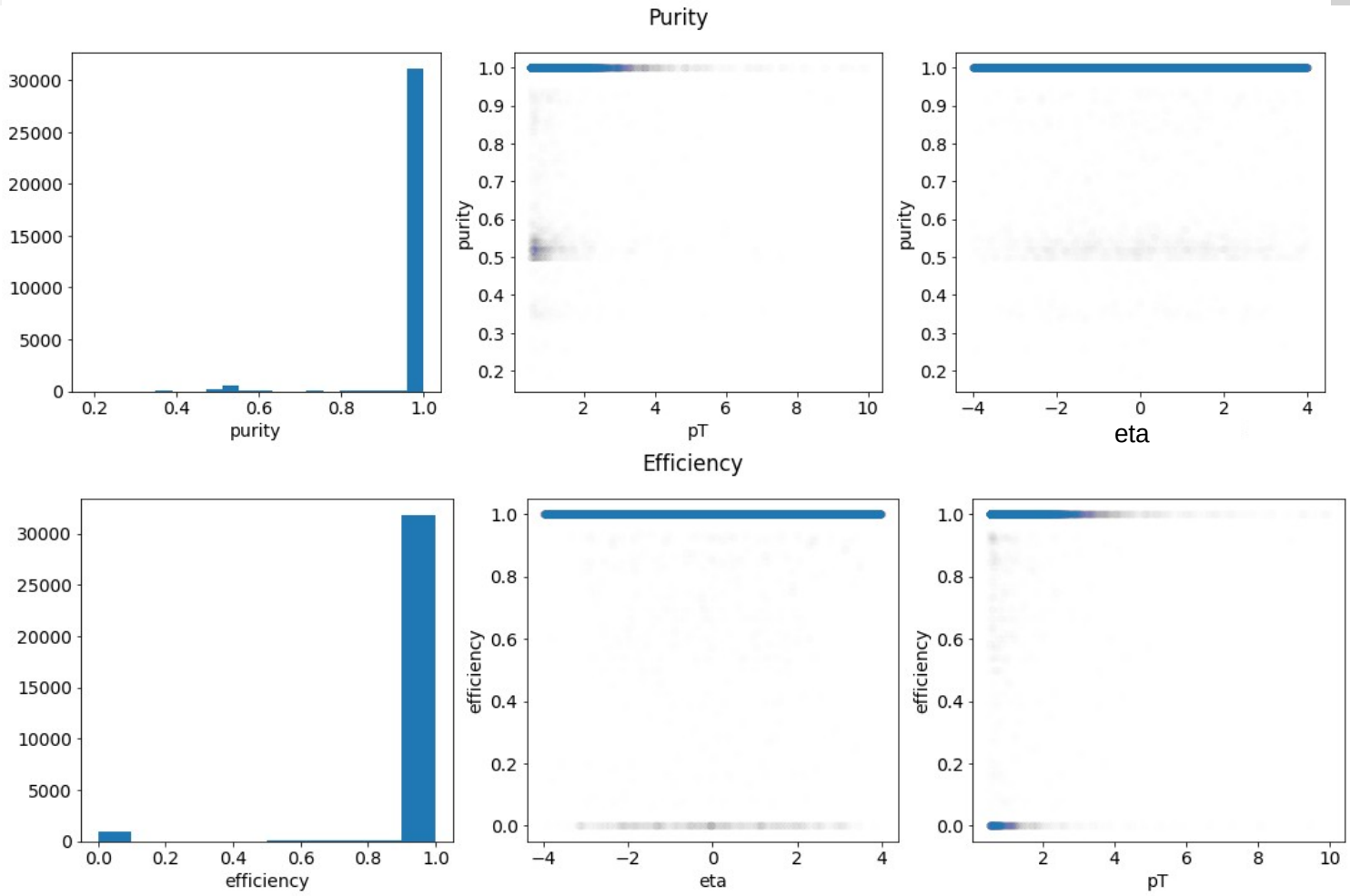
Exa.TrkX Pur & Eff (w/o cuts)



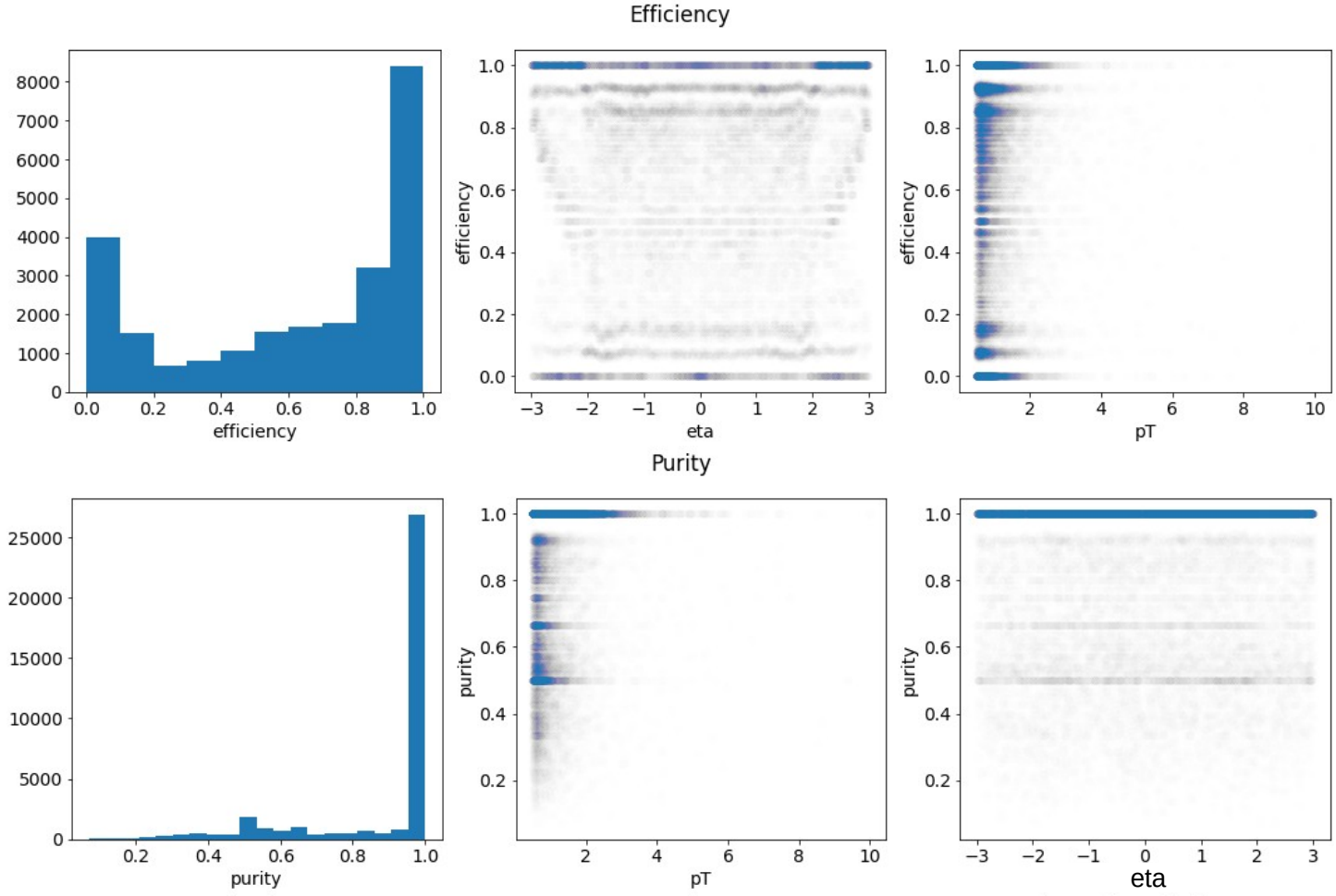
Exa.TrkX Pur & Eff (w/ cuts)



Exa.TrkX Pur & Eff (w/ cuts & true)

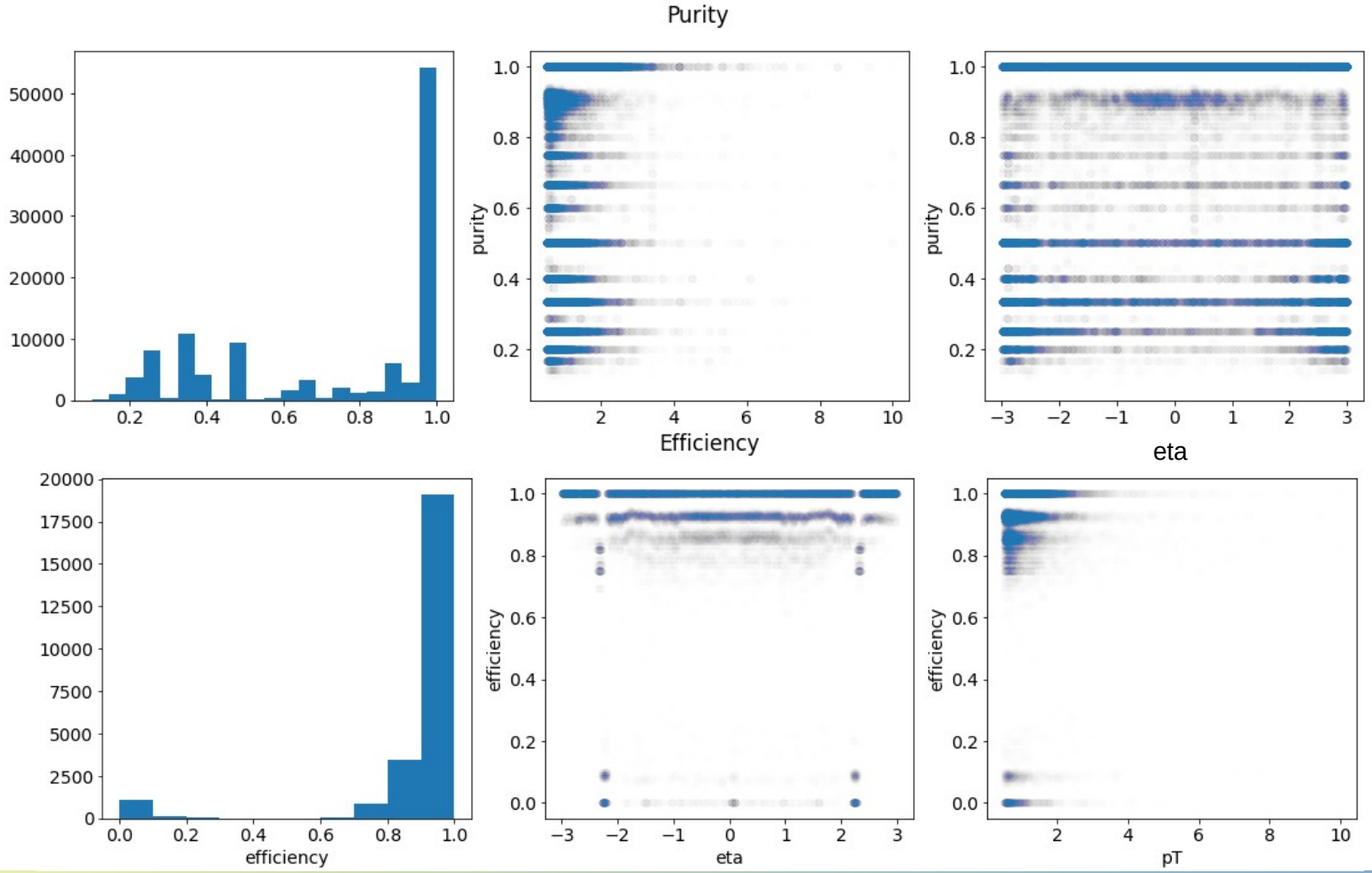


CKF Pur & Eff (w/ cuts & smear)



CKF Pur & Eff

- Smeared hits
- With cut



Hyperparameters

Embedding

emb_dim	8
emb_hidden	1024
knn	50
max_epochs	20
nb_layer	4
points_per_batch	130000
r_{train, test, val}	0.2
true_edges	modulewise
regime	[rp, hnm, norm]
id	`292tlcta`

Filtering

filter_cut	0.01
hidden	512
nb_layer	4
true_edges	modulewise
regime	[ ]
max_epochs	20
id	`267d78hx`

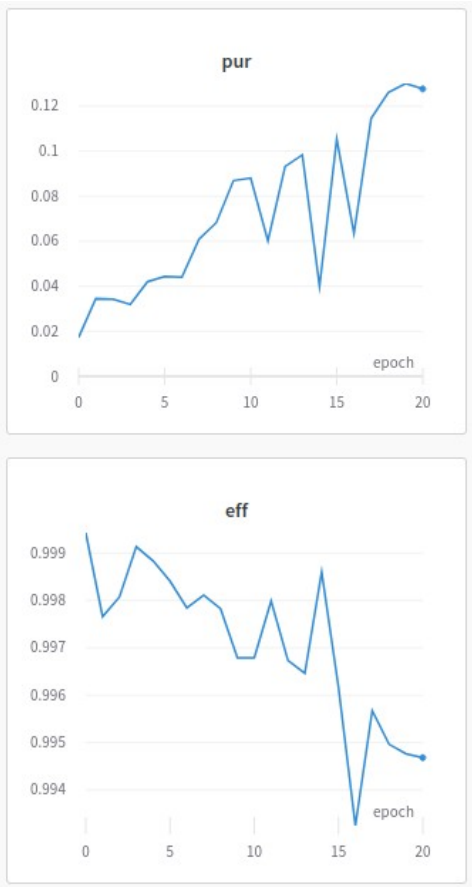
GNN

edge_cut	0.5
hidden	128, ReLu
nb_{edge, node}_layer	5
true_edges	modulewise
n_graph_iters	8
regime	[ ]
max_epochs	25
learning_rate	0.001
id	`3iqwzub9`

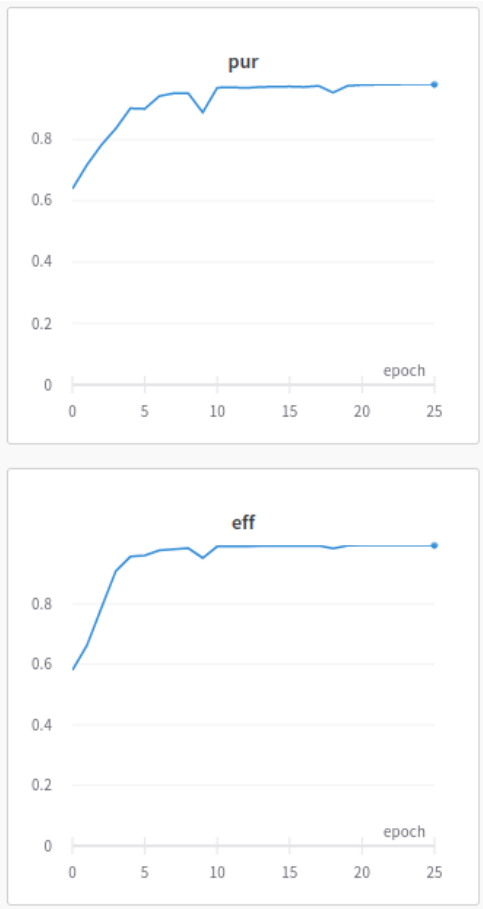
Training metrics

Embedding

Filtering



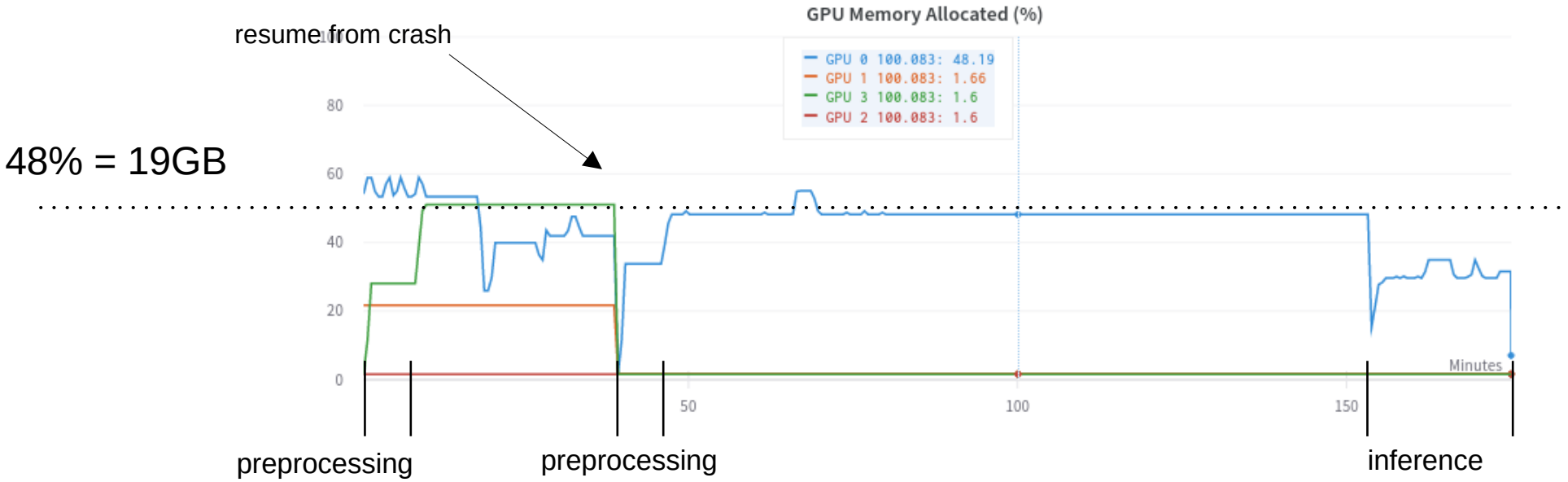
GNN



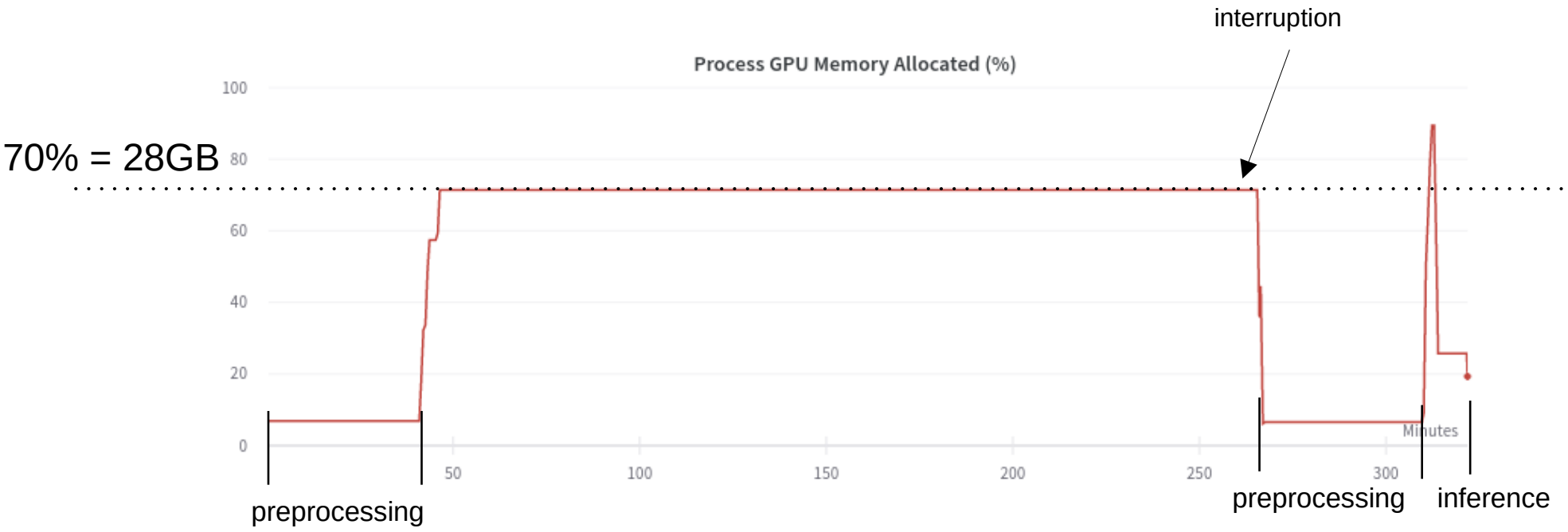
Hardware

- CPU:
  - AMD EPYC 7662 64-Core Processor
  - 1 TB RAM
- GPU:
  - 4x NVIDIA A100-SXM4 with 40GB

Memory profile training embedding



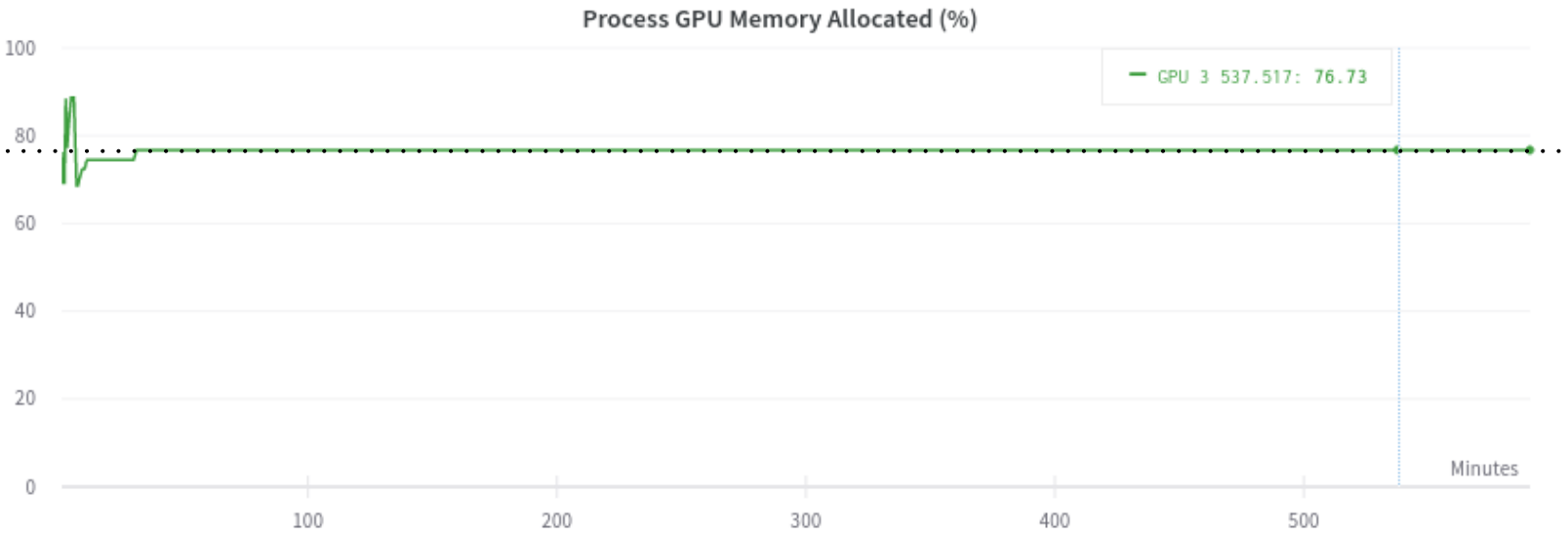
Memory profile training filter



Memory consumption controllable by „chunking“ graph

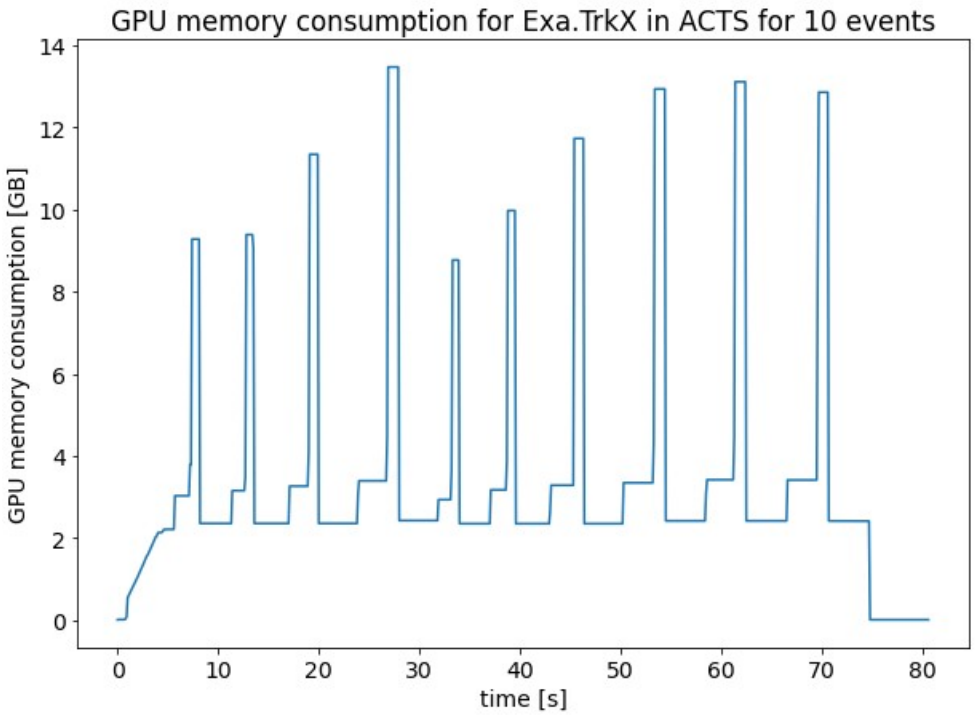
Memory profile training GNN

76% = 30GB



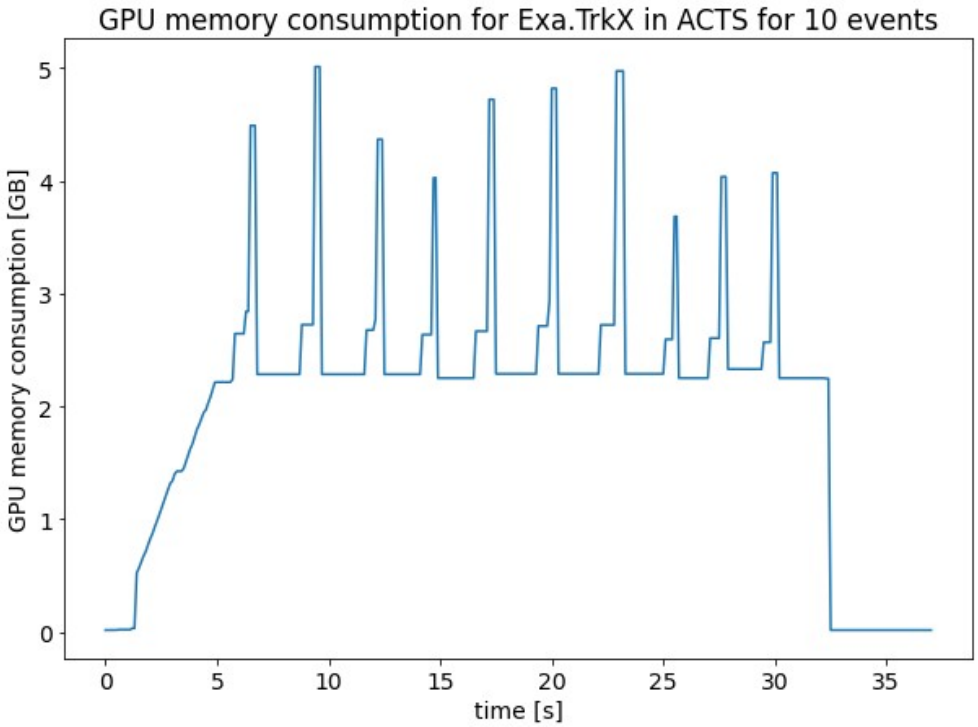
Memory profile inference in ACTS

No cuts on particles



Max = 13.5 GB

With cuts on particles



Max = 5 GB

More residuals