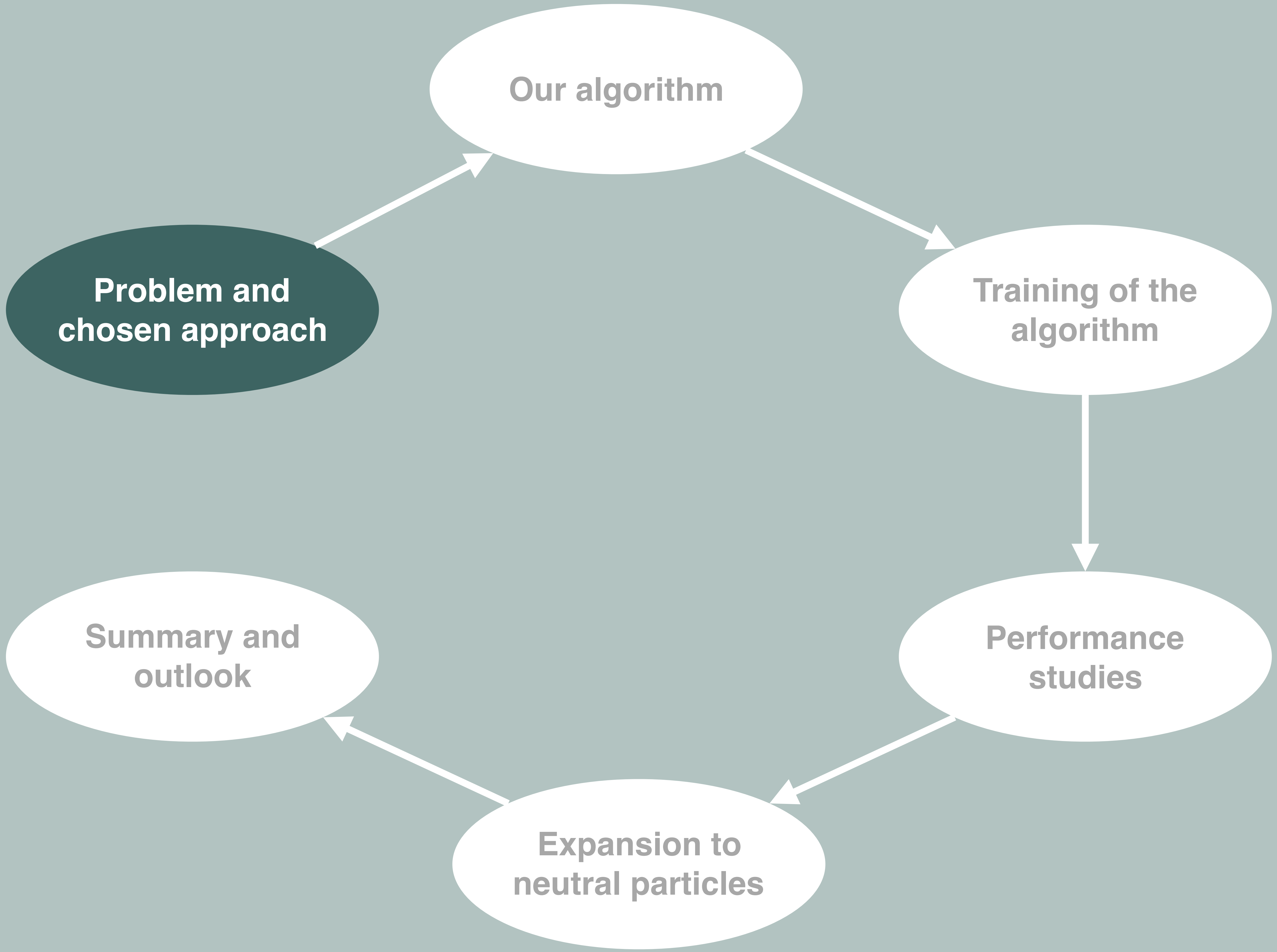
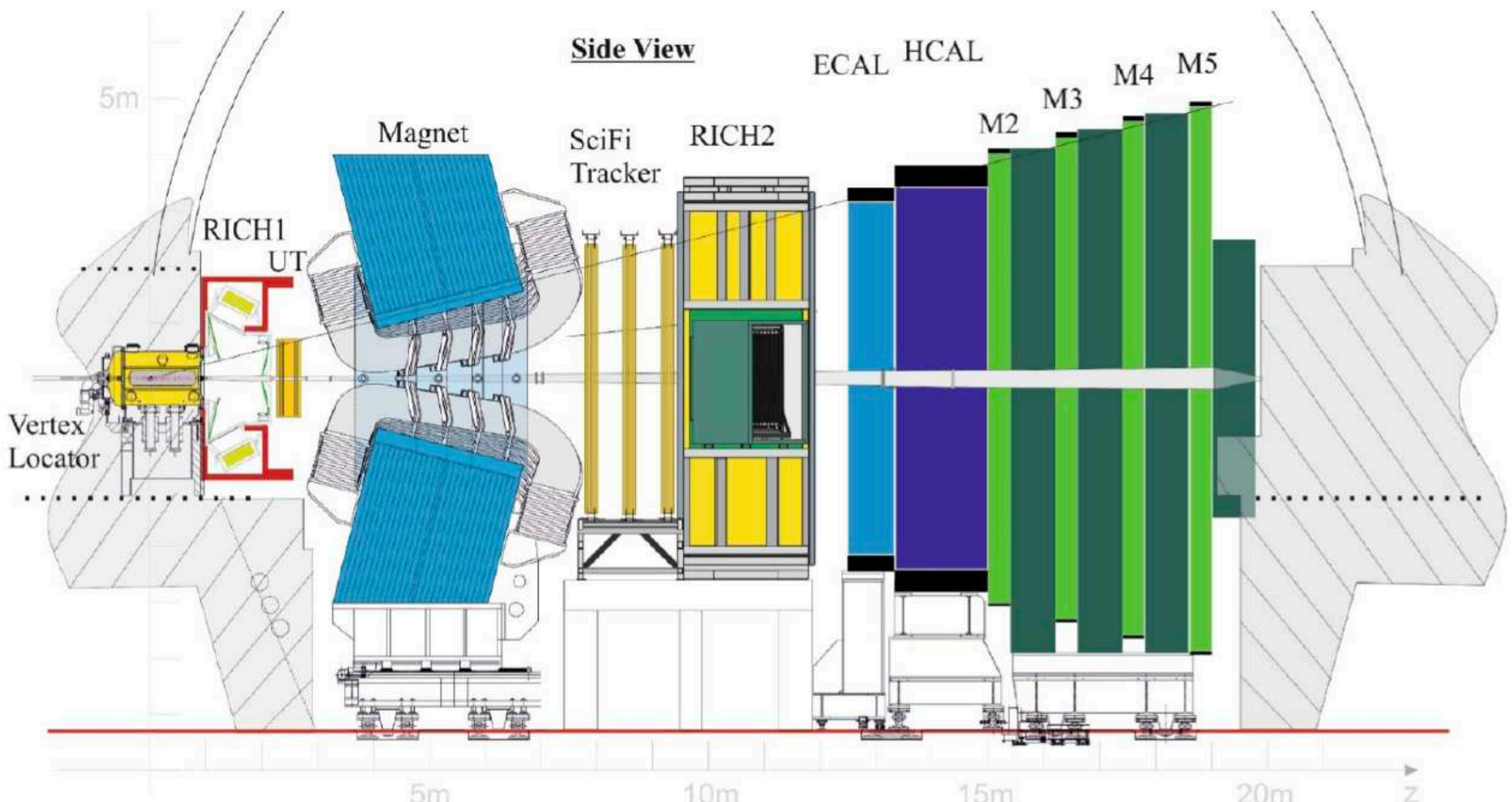
GNN-based algorithm for full-event filtering and interpretation at the LHCb trigger

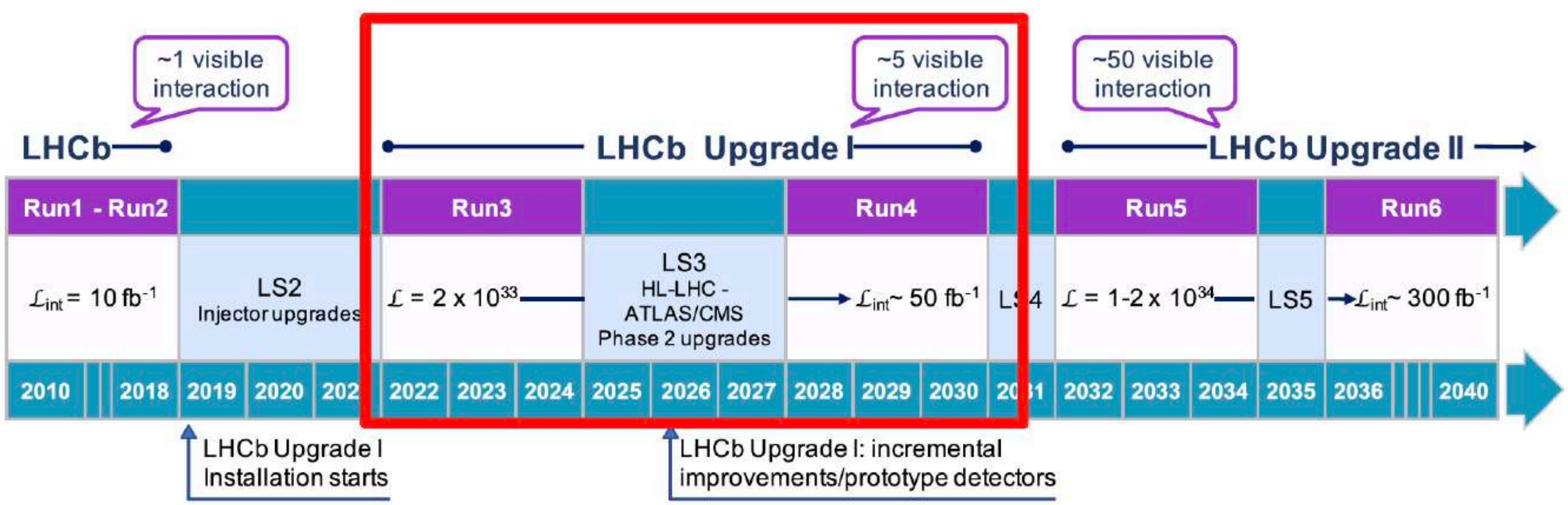
Julián García Pardiñas[1], Andrea Mauri[2], Marta Calvi[1], Jonas Eschle[3], Simone Meloni[1], Nicola Serra[3]

1 University and INFN Milano-Bicocca, Italy
2 NIKHEF, The Netherlands
3 University of Zürich, Switzerland


Our algorithm
Training of the algorithm
Performance studies
Expansion to neutral particles
Summary and outlook
Problem and chosen approach

Introduction





LHCb detector: forward spectrometer studying decays of beauty and charm hadrons.

Upcoming LHC Run3 @LHCb:

- Increase in instantaneous luminosity by a factor 5.
- Hardware part of the trigger removed.
  - ➥ Fully-software trigger.
    - ➥ Higher efficiencies.

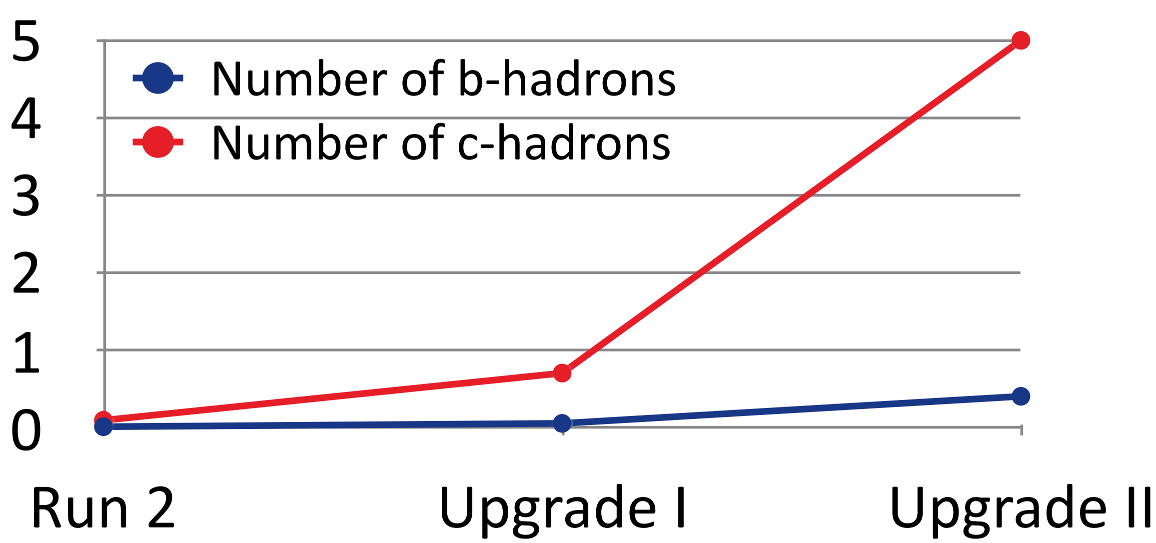
⇓

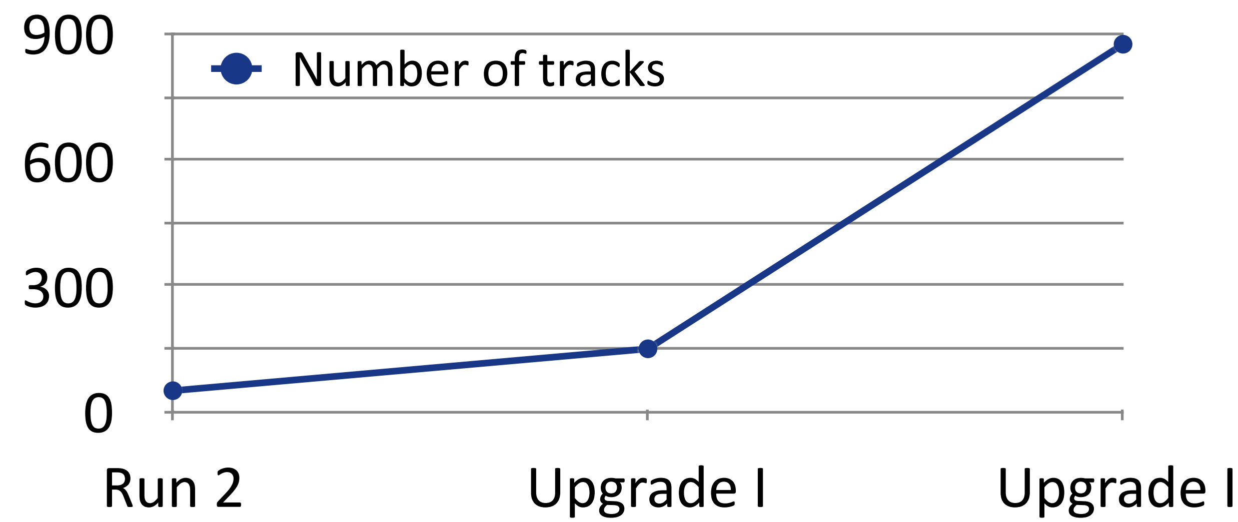
Huge challenge of combinatorics for the trigger.

➥ Much bigger challenge for the Upgrade 2!

Event complexity and evolution of the trigger paradigm

Evolution of average quantities per event(*):





During **Run1 and Run2**, a small fraction of the events had a signal (either b- or c-hadron)

Upgrade 1 (Runs 3-4): about 1 charm hadron per event.

Upgrade 2 (Runs 5-6): about 1 beauty hadron every other event, several charm hadrons per event.

(*) Numbers obtained from a private simulation based on PYTHIA, described later in this talk.

Event complexity and evolution of the trigger paradigm

Evolution of average quantities per event(*):

900

e II

Limited trigger bandwidth [LHCB-TDR-018]:

Bandwidth [MB/s] ∝ Trigger output rate [kHz] × Average event size [kB]

From: **which events are interesting?** We can save a lot of information for many of them!

To: (almost) all of the events are interesting … but they have too much information!
Which parts of each event are interesting?

This problem had already been anticipated in 2014 [LHCb-PUB-2014-027], and LHCb has done several crucial steps to set the basis for an eventual solution.

other event, several charm hadrons per event.

(*) Numbers obtained from a private simulation based on PYTHIA, described later in this talk.

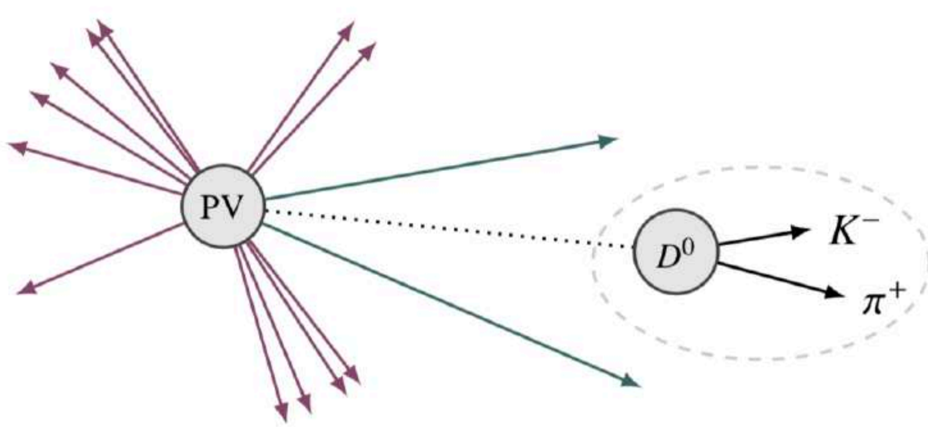
Signal-based trigger vs Full Event Interpretation (FEI)

Signal based

The current LHCb trigger is an **OR between many decay-mode selection lines**.

Since Run2, to reduce the event size, some lines **store only parts of the event which are related** to the specific signal. [JINST 14 (2019) 04, P04006]

E.g.: store the signal + the tracks in the same primary vertex (PV).



FEI

New proposal: try to **reconstruct the b- and c- hadron decay chains in the event**, in a hierarchical-clustering manner (cluster → unstable particle), **and discard the rest**.

<u>Advantages:</u>

- **Exploit extra correlations** between objects in the event.
- **Bandwidth oriented**: focus on storing as much "useful" information as possible.
- Case of several signals per event as an integral part of the approach.
- Establishment of a basis for an expanded functionality of the trigger: inclusive selections, study of anomalous events …

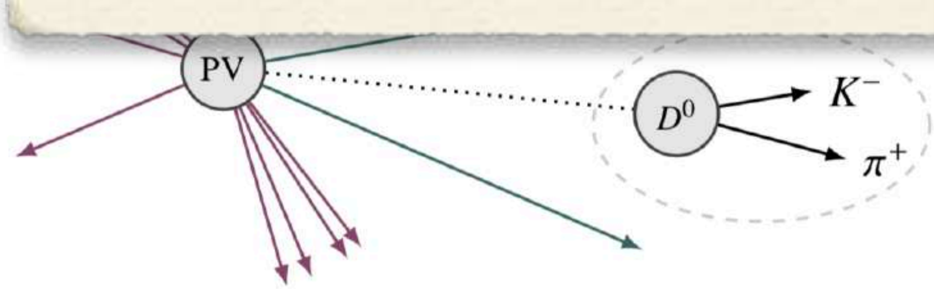
Signal-based trigger vs Full Event Interpretation (FEI)

Signal based

The current LHCb trigger is an **OR between many decay-mode selection lines**.

Sinc
line
are
(201

E.g.:
sam



FEI

New proposal: try to **reconstruct the b- and c- hadron decay chains in the event,** in a hierarchical-clustering manner (cluster
e rest.

n

ing as
le.
s an

- Establishment of a basis for an expanded functionality of the trigger: inclusive selections, study of anomalous events …

Complementary area of interest: offline analysis

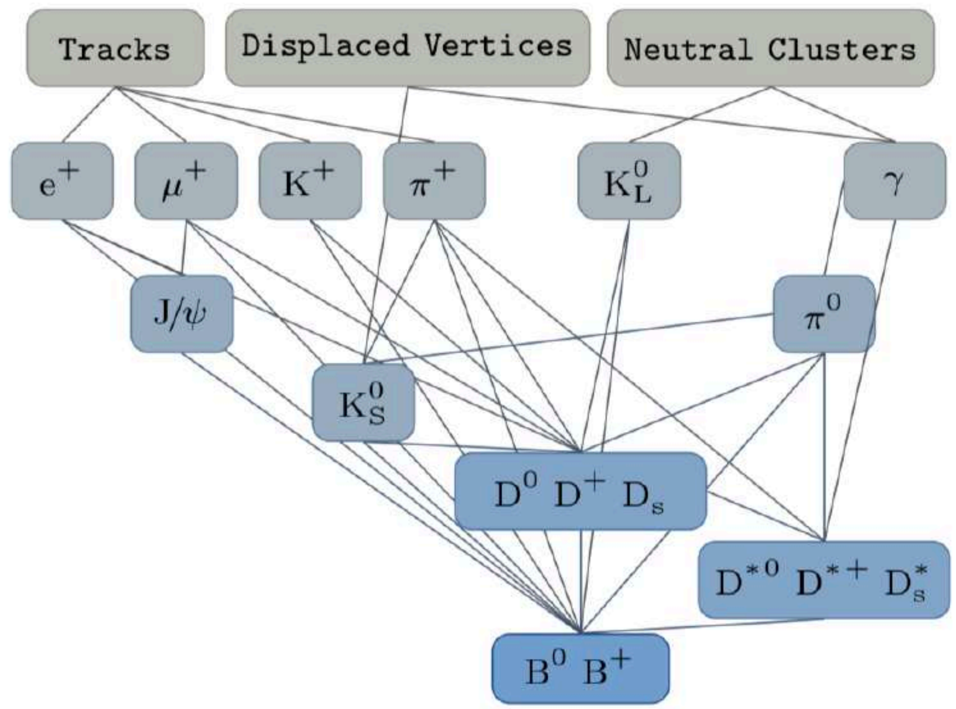
A larger luminosity also implies increased levels of background.

The usage of a FEI approach can help identifying different sources of background, to either suppress them or enhance them in control regions.

Connection with other experiments

From a physics perspective, the closest example from a different experiment is the **FEI algorithm** developed for **Belle II** [Comput.Softw.Big Sci. 3 (2019) 1, 6].

Reconstruction of the tag-side $B^0/B^{\pm}$ decay chain in a hierarchical way, using a **fixed collection of possible decays, with independent classifiers** for each of them.



Recently started exploring an alternative **DL-based inclusive approach with GNNs** [BELLE2-MTHESIS-2020-006].

Our project also has strong technical similarities with many developments done at **CMS**. They have investigated the usage of **GNNs in online computing** for **pileup mitigation** [arXiv: 1810.07988] and to develop a modernized version of the **Particle Flow** algorithm [Eur.Phys.J.C 81 (2021) 5, 381].

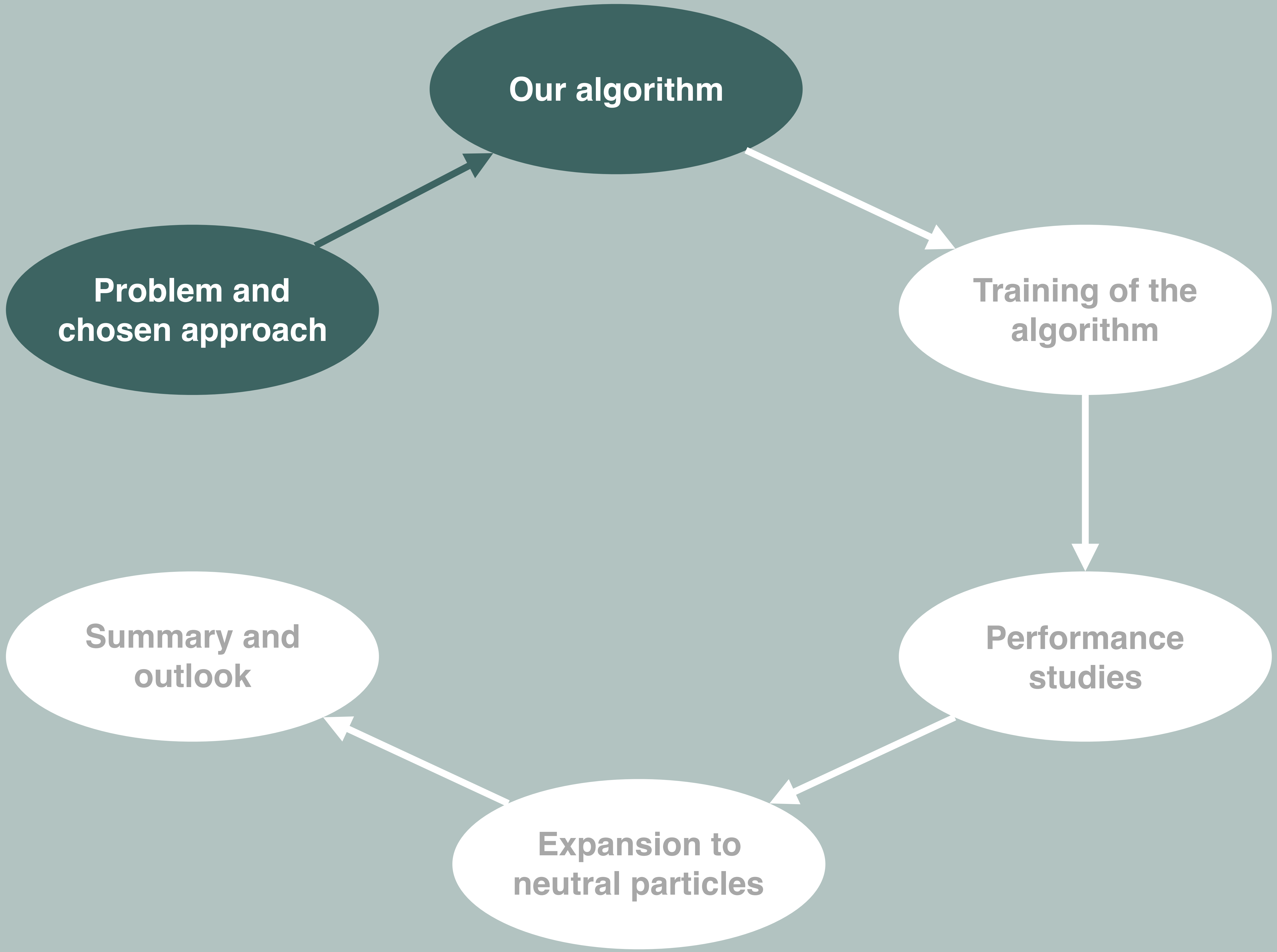
They have also done the **first FPGA-compatible implementation of a GNN** [Frontiers in Big Data 3 (2021) 44], to run the algorithms online fitting their computing requirements.

Differences between Belle II and LHCb

Belle II		**LHCb**
Only $B^0/B^\pm$ hadrons	→	All b-hadron species (+ c-hadrons)
List of possible decays large but limited and well identified.	→	Enormous list of possible decay chains.
e+e- collisions: clean environment.	→	pp collisions: important contamination from the rest of the event.
Hermetic detector: all the information (except for neutrinos) available.	→	Non-hermetic detector: partially-reconstructible decays in many cases.

⇓

Leverage the full power of **deep-learning** and try to reconstruct as much as possible.
➥ **DFEI: Deep-learning based FEI for LHCb.**


Our algorithm
Training of the algorithm
Performance studies
Expansion to neutral particles
Summary and outlook
Problem and chosen approach

General description of our algorithm

Input: **reconstructed stable particles** (charged tracks and neutral clusters).

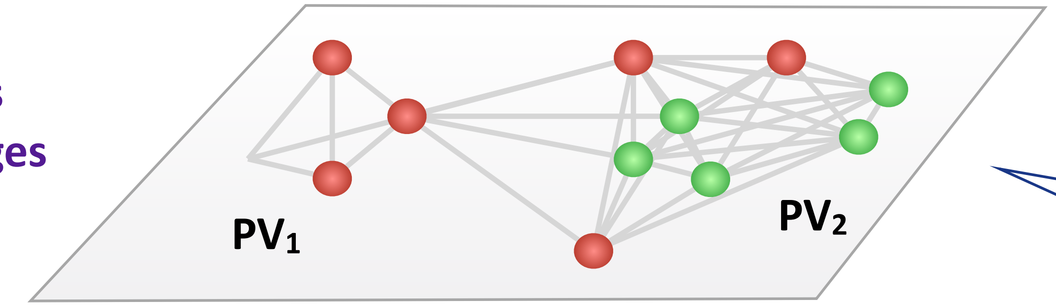
Basic deep-learning units: **Graph Neural Networks (GNN)**. → We are using the *graph_nets* package from DeepMind [arXiv:1806.01261].

Algorithm structure: **chain of modules with increasing "granularity"**, each of them filtering away part of the event.

Current prototype: use only **charged stable particles** and focus on the reconstruction of **beauty-hadron decays**.

DFEI algorithm for b-hadrons, charged particles

~140 nodes
~10000 edges



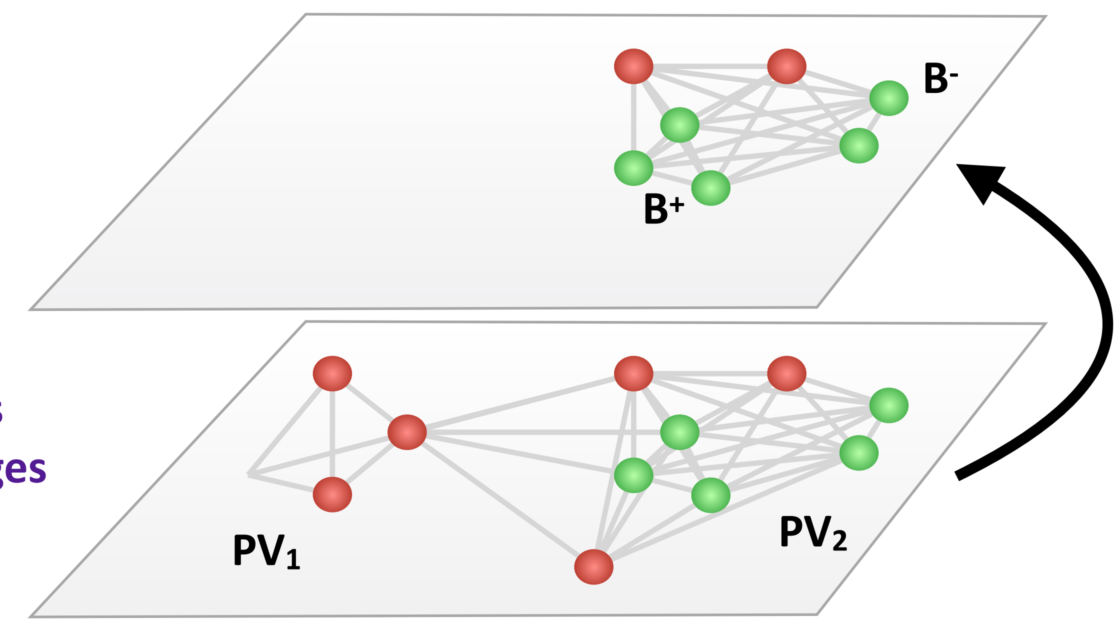
Green: particles from a b-hadron
Red: particles from the rest of the event

Create a graph with the final state particles
Connect particles which are topologically close ("cone"-like requirements, see backup).

DFEI algorithm for b-hadrons, charged particles

~30 nodes
~300 edges

~140 nodes
~10000 edges



1. Node pruning
Kinematics&topology GNN for node classification.

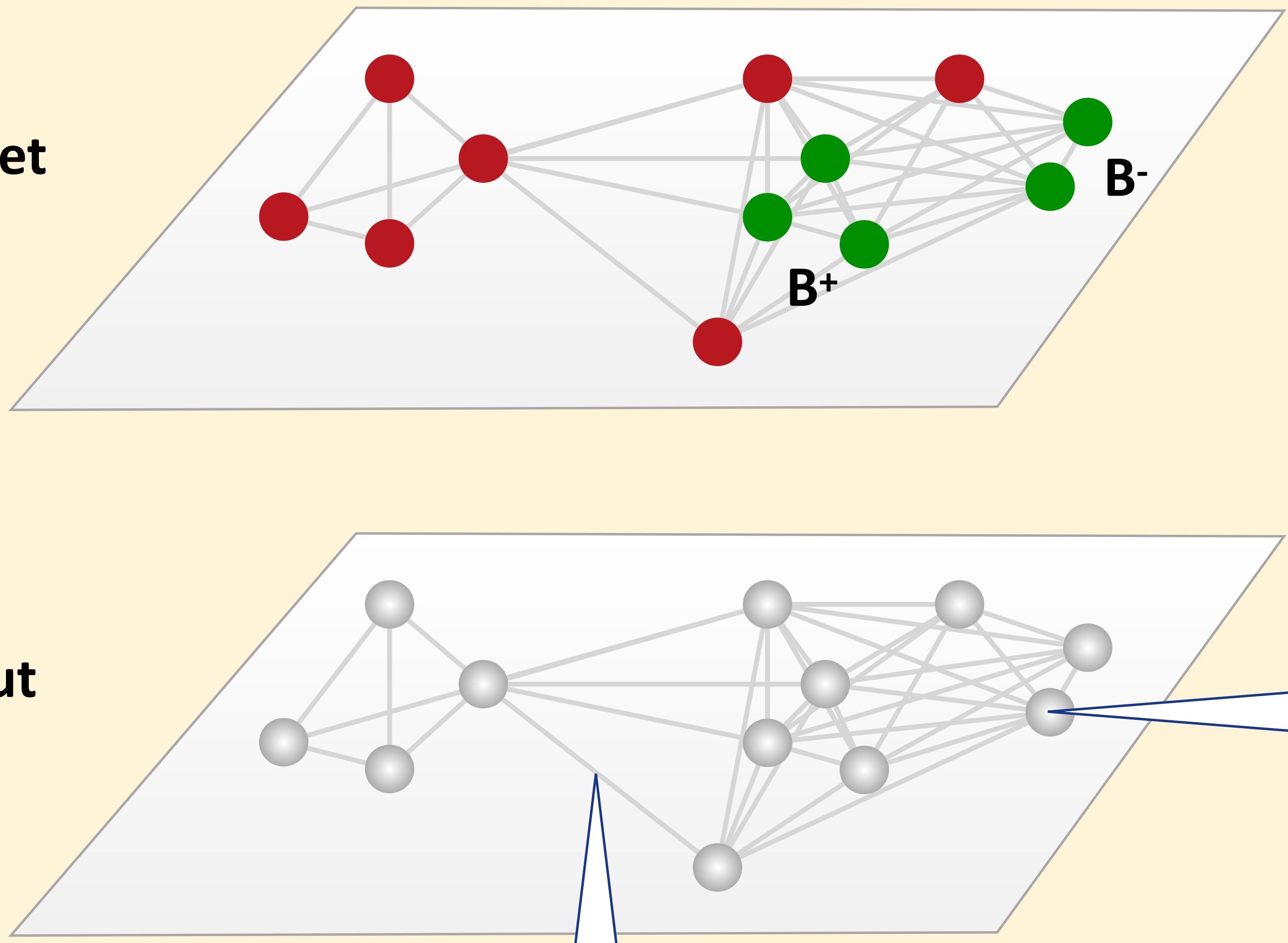
Green: particles from a b-hadron
Red: particles from the rest of the event

1) Node pruning

Signal nodes: particles from a b-hadron (any of them)
Background nodes: particles from the rest of the event

Target

Input

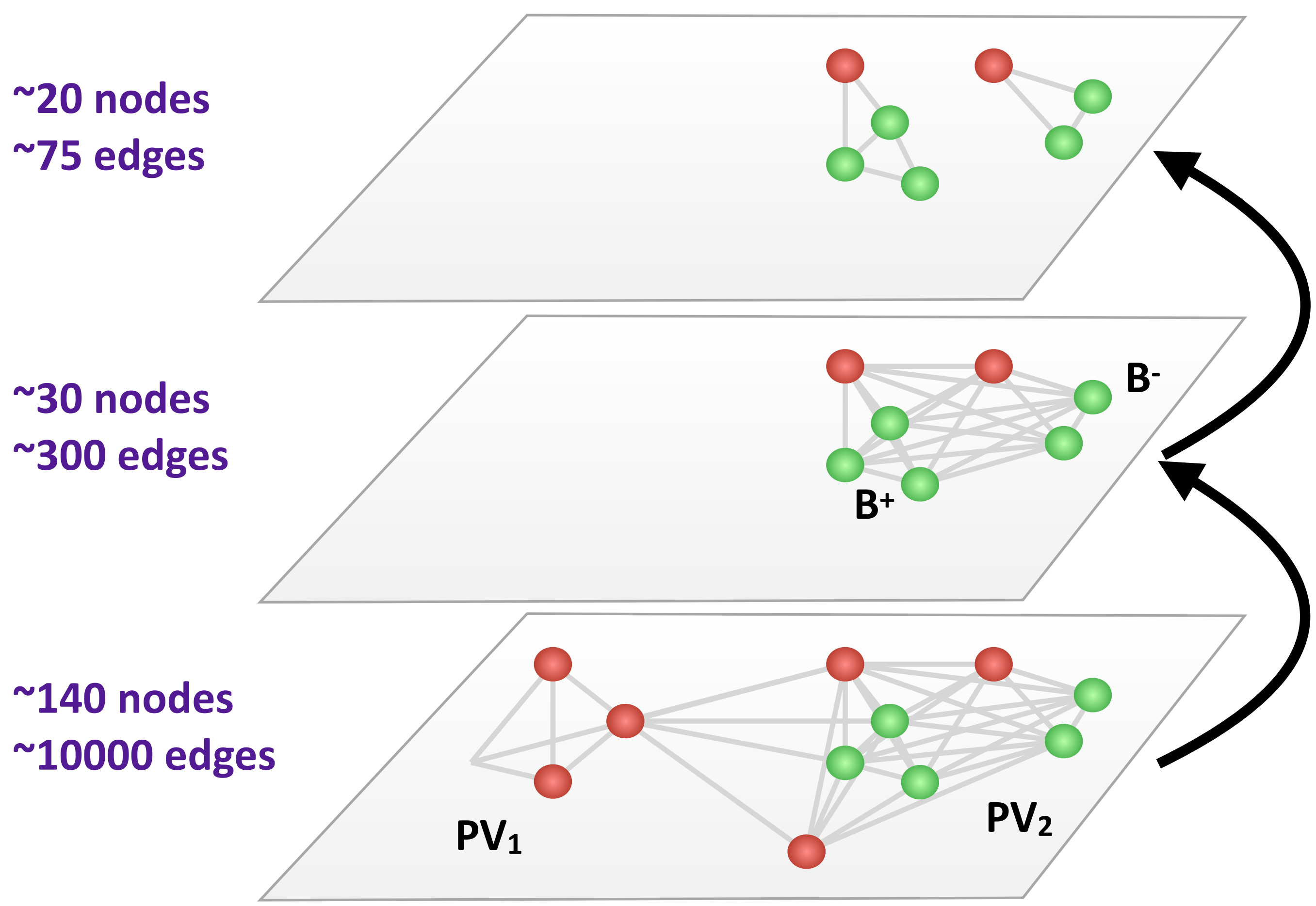


pT: transverse momentum
ETA: pseudorapidity
PV: associated primary vertex
IP: impact parameter with respect to the PV
q: charge

pT, ETA, IP, q.

Opening angle, distance (between origins) along the beam axis, “transverse distance” (see backup), from same PV (boolean).

DFEI algorithm for b-hadrons, charged particles



2. Edge pruning
Kinematics&topology GNN for edge classification.

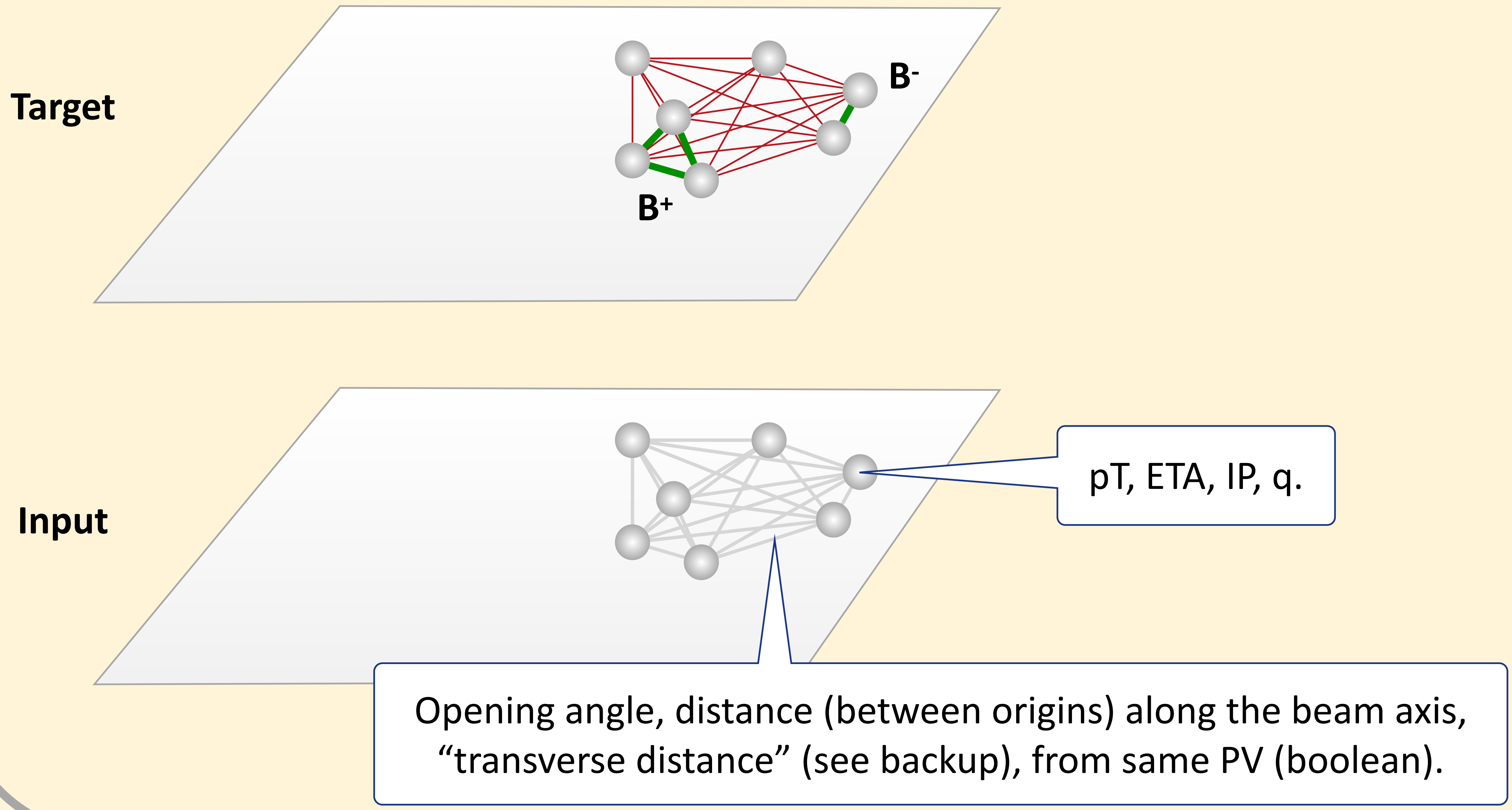
1. Node pruning
Kinematics&topology GNN for node classification.

Prefiltering

Green: particles from a b-hadron
Red: particles from the rest of the event

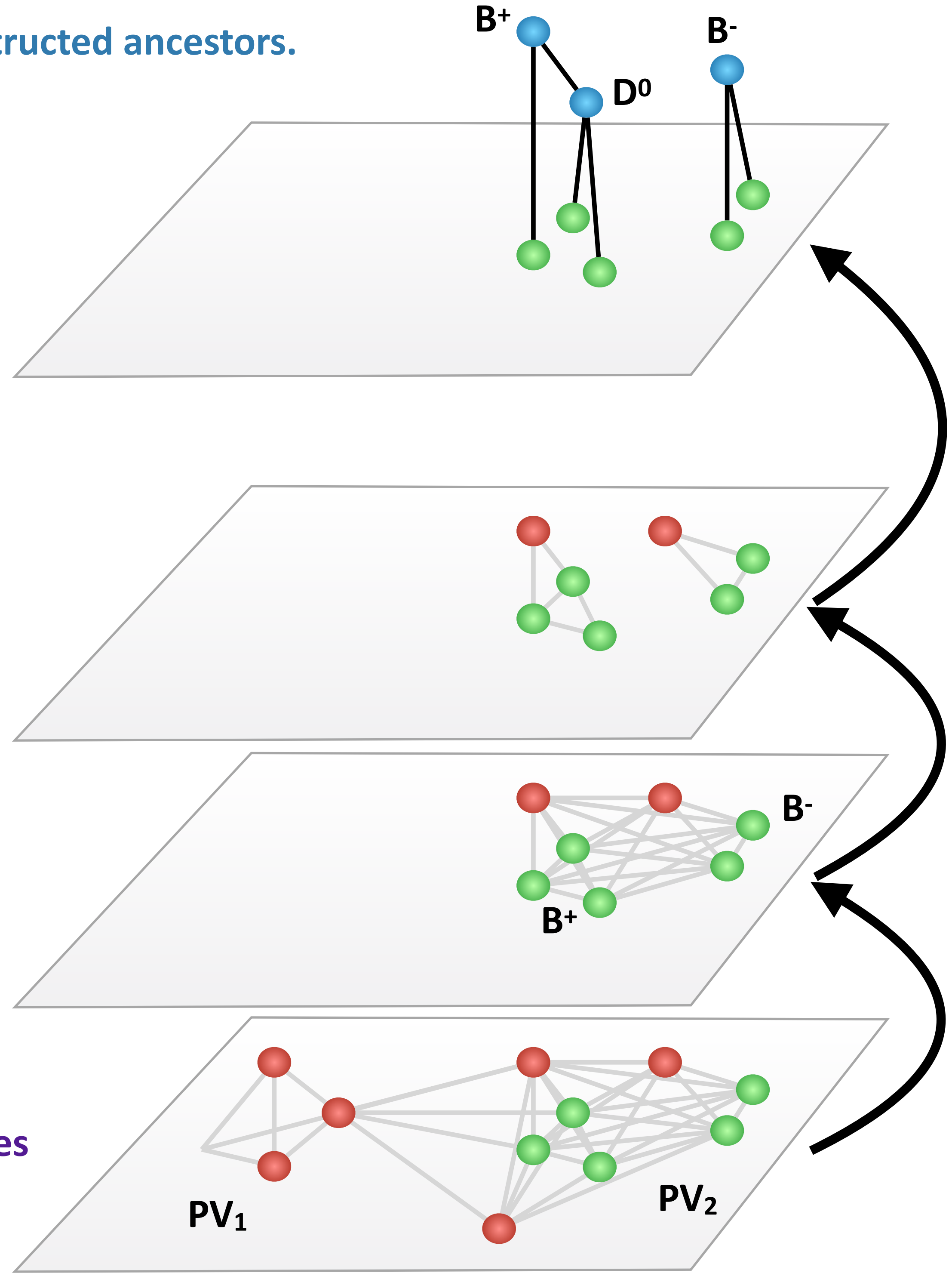
2) Edge pruning

Signal edges: pairs of particles with the same b-hadron ancestor
Background edges: any other pair of particles



DFEI algorithm for b-hadrons, charged particles

Blue: reconstructed ancestors.



~20 nodes
~75 edges

~30 nodes
~300 edges

~140 nodes
~10000 edges

3. "Topological" LCA reconstruction
Kinematics&topology GNN for edge classification ("topological" LCA values).

Prefiltering

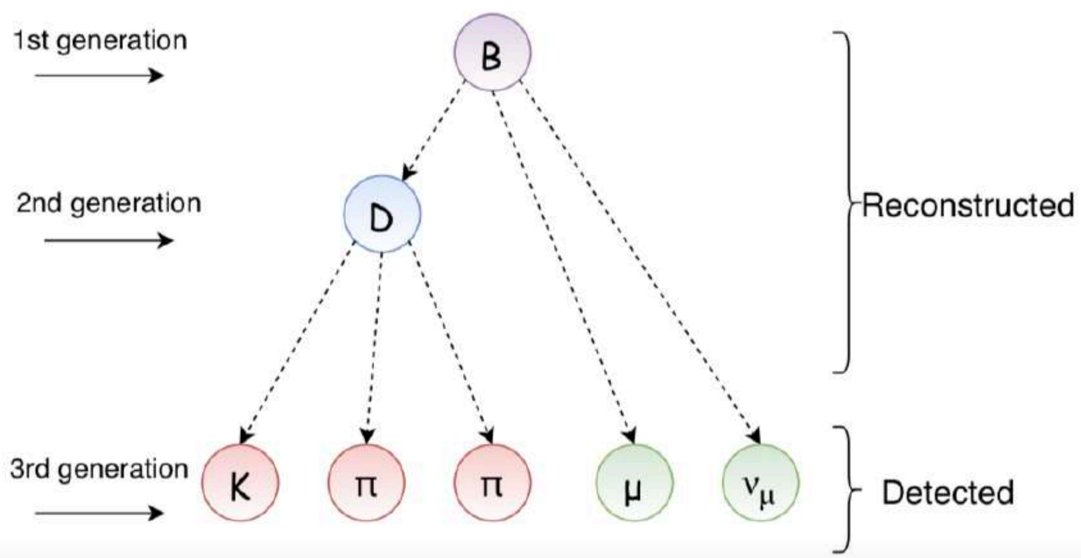
2. Edge pruning
Kinematics&topology GNN for edge classification.

1. Node pruning
Kinematics&topology GNN for node classification.

Green: particles from a b-hadron
Red: particles from the rest of the event

3) “Topological” LCA reconstruction

From [BELLE2-MTHESIS-2020-006]:



Adjacency Matrix

	B	D	K	π	π	μ	ν_μ
B	0	1	0	0	0	1	1
D	1	0	1	1	1	0	0
K	0	1	0	0	0	0	0
π	0	1	0	0	0	0	0
π	0	1	0	0	0	0	0
μ	1	0	0	0	0	0	0
ν_μ	1	0	0	0	0	0	0

LCA Matrix

	K	π	π	μ	ν_μ
K	0	1	1	2	2
π	1	0	1	2	2
π	1	1	0	2	2
μ	2	2	2	0	2
ν_μ	2	2	2	2	0

The information of the **decay tree** can be **fully encoded** using an **integer for each edge.**

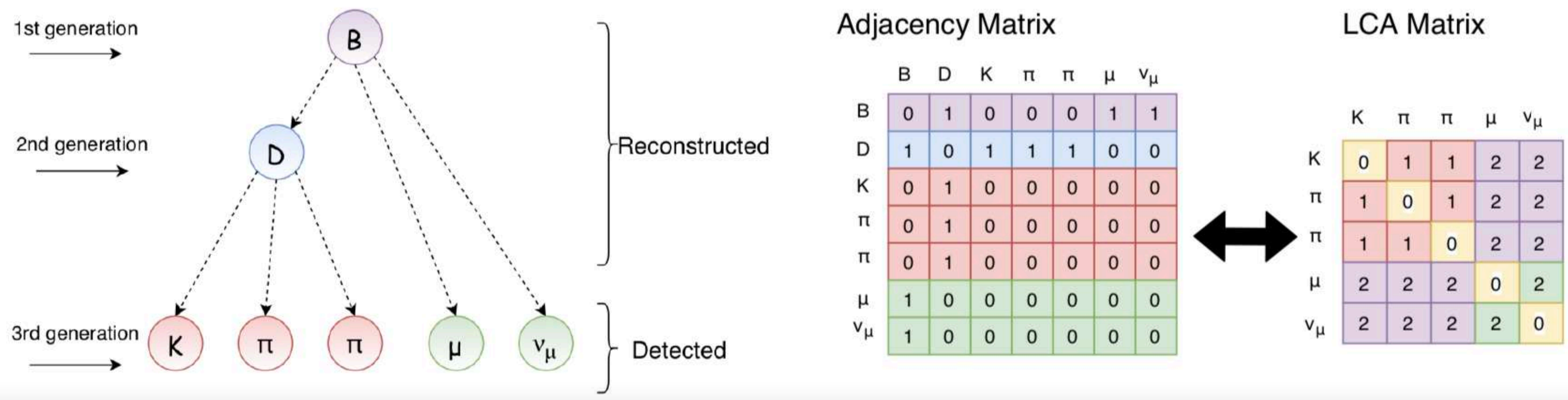
At Belle II, the LCA matrix reconstruction relies on the effect of kinematic constraints (momentum conservation, invariant masses, …).

Since the physics case of LHCb is much richer, but we can profit from a very good vertex resolution, **we are focusing on a simpler “topological” reconstruction,** where the **target decay chain is modified as follows**:

- **Very short lived resonances are “merged” with their ancestor.**
- **Resonances without enough charged descendants to allow a vertex reconstruction are “merged” with their ancestor.**

3) “Topological” LCA reconstruction

From [BELLE2-MTHESIS-2020-006]:



The information of the **decay tree** can be **fully encoded** using an **integer for each edge.**

Example target transformation from a normal ancestor chain into a “topological” ancestor chain:

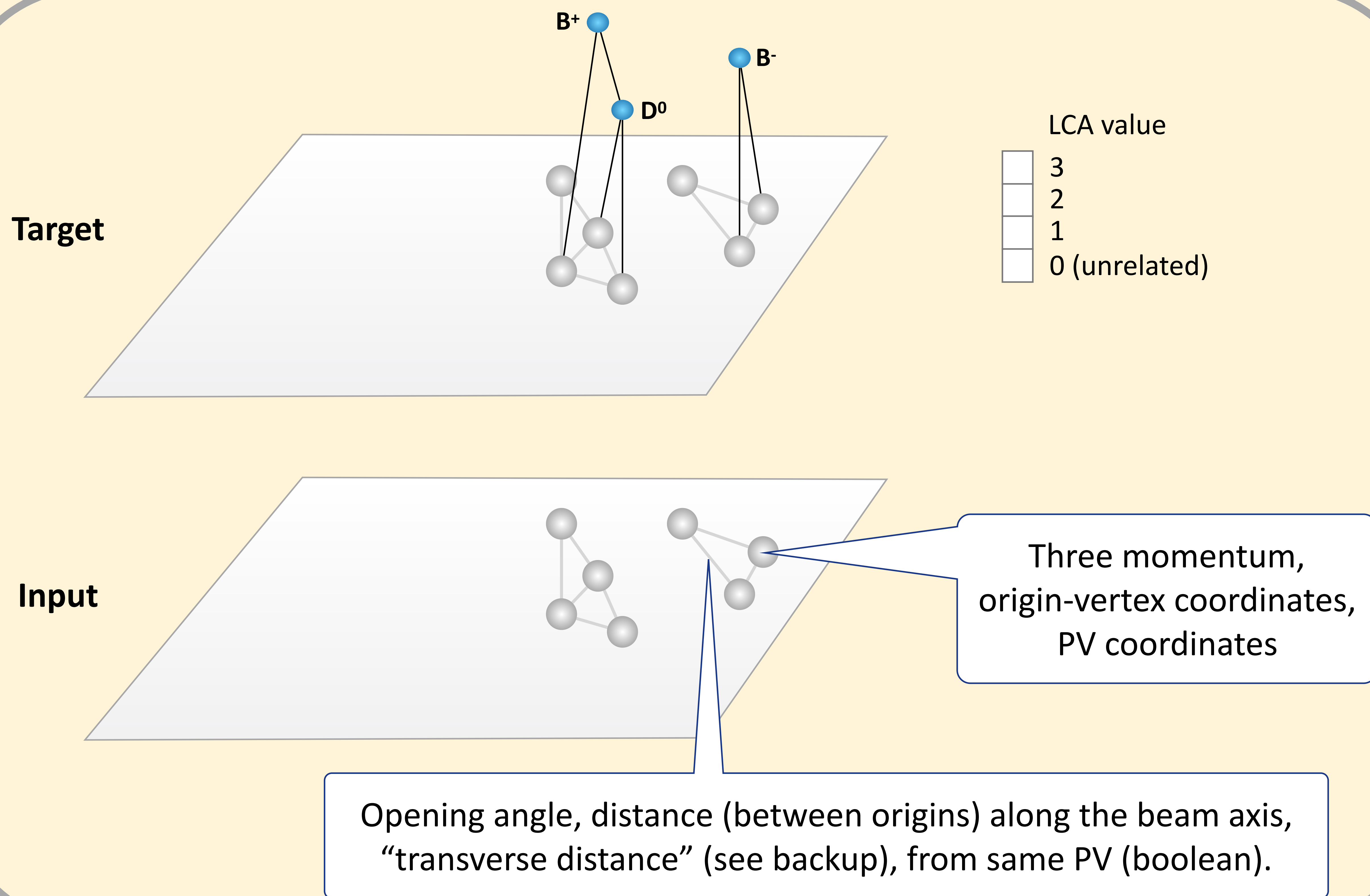
Original ancestor chain ending up in an observed pi+ (5 ancestors):

B*0 → B0 → D+ → phi(1020) → rho(770)0 → pi+

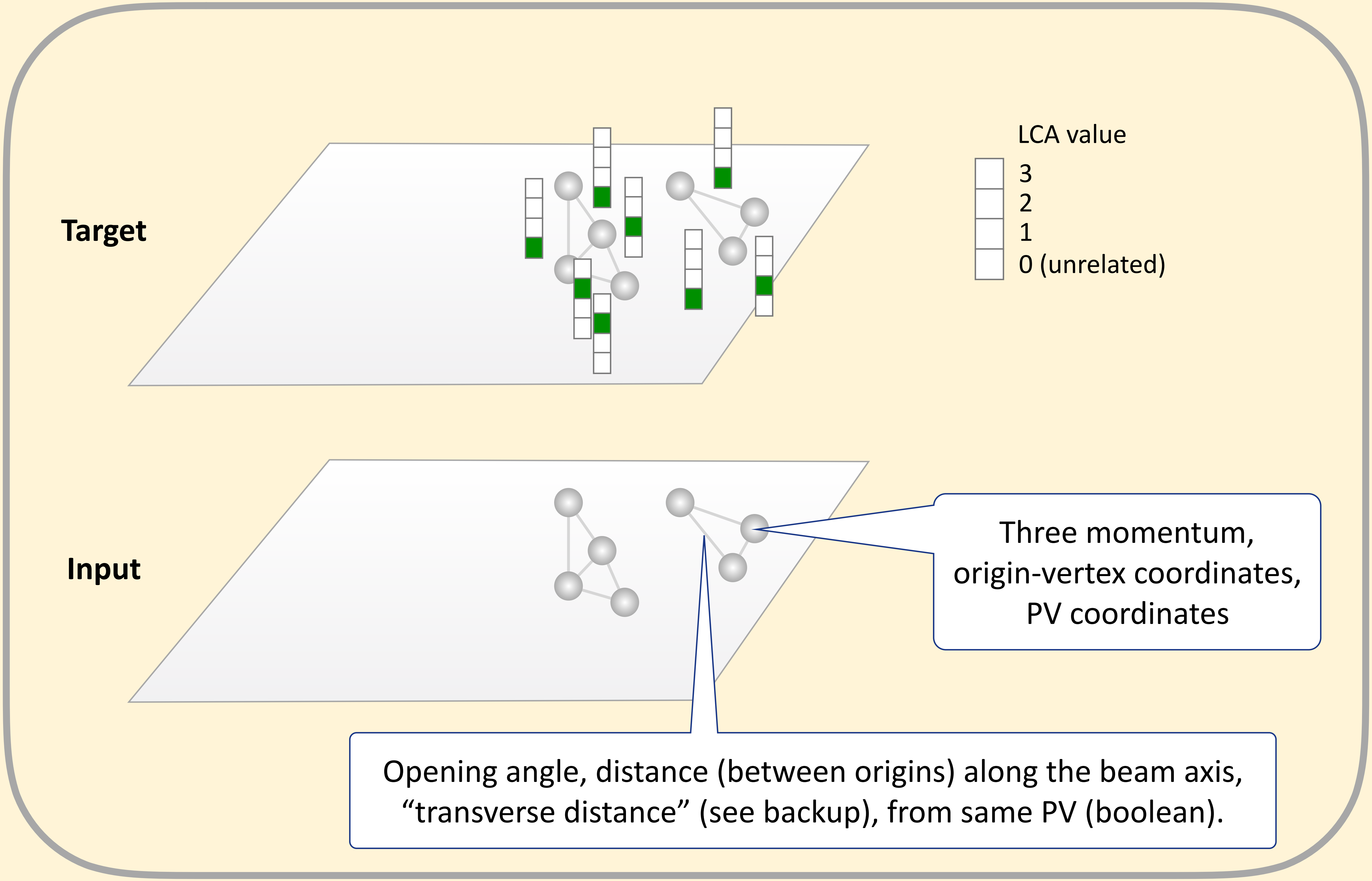
Corresponding “topological” ancestor chain (2 ancestors):

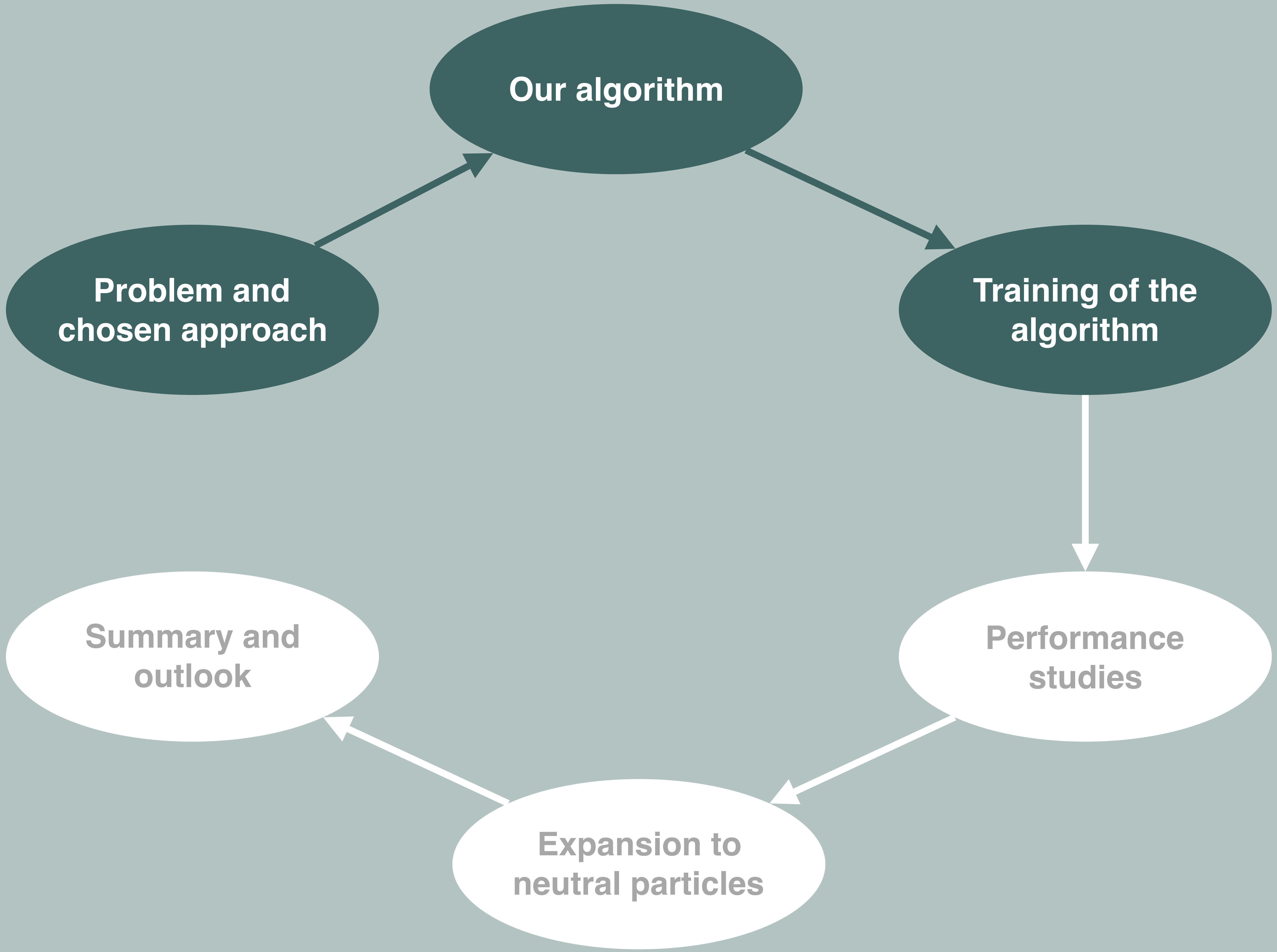
B0 → D+ → pi+

3) “Topological” LCA reconstruction



3) “Topological” LCA reconstruction




Our algorithm
Training of the algorithm
Performance studies
Expansion to neutral particles
Summary and outlook
Problem and chosen approach

Training dataset: emulating Run3 conditions

Particle collision&decay

The training and performance studies are currently done using **PYTHIA**, with the following configuration:

- Proton-proton collisions at 13 TeV.
- Average number of collisions per event: 7.6.
- Selecting **events with at least one b-hadron produced (inclusive decay)**.

"Detection and reconstruction"

We require all the tracks and the b-hadrons to be **inside the LHCb geometrical acceptance**.

In addition, we **emulate the reconstruction of the following quantities**, using publicly available expectations for the LHCb performance in Run3 (see backup):

- **Origin point of the tracks** (first measurement in the Vertex Locator).
- **Three-momentum of the tracks**.
- **Position of the primary vertices**.

Training conditions

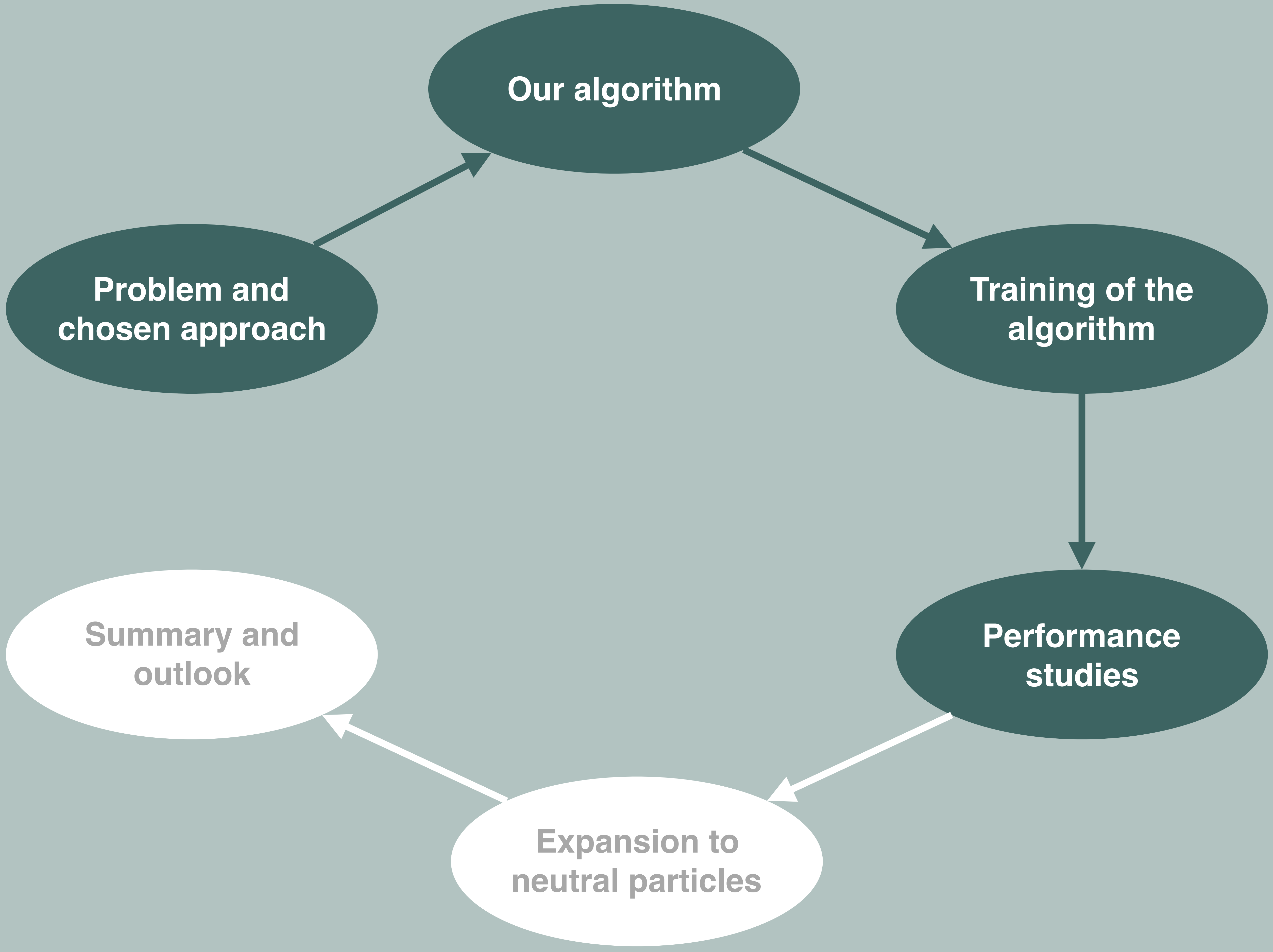
Training performed in series:

1. **Train the node-pruning algorithm**
   Batch size: 32 events. **~1.3x10^4 events** seen by the training process.
2. **Choose a threshold**
   Set to ~95% signal eff.
3. **Filter the data**
4. **Train the edge-pruning algorithm**
   Batch size: 32 events. **~1.3x10^4 events** seen by the training process.
5. **Choose a threshold**
   Set to ~95% signal eff.
6. **Filter the data**
7. **Train the "topological" LCA reconstruction**
   Batch size: 128 events. **~3.1x10^6 events** seen by the training process **(see backup)**.

For more details on the training of the node and edge pruning algorithms, see **our presentation at the IML meeting on 05/10/21**.

In all the cases:
- **Loss:** weighted softmax cross entropy, to balance out the different classes.
- **Minimizer:** Adam.

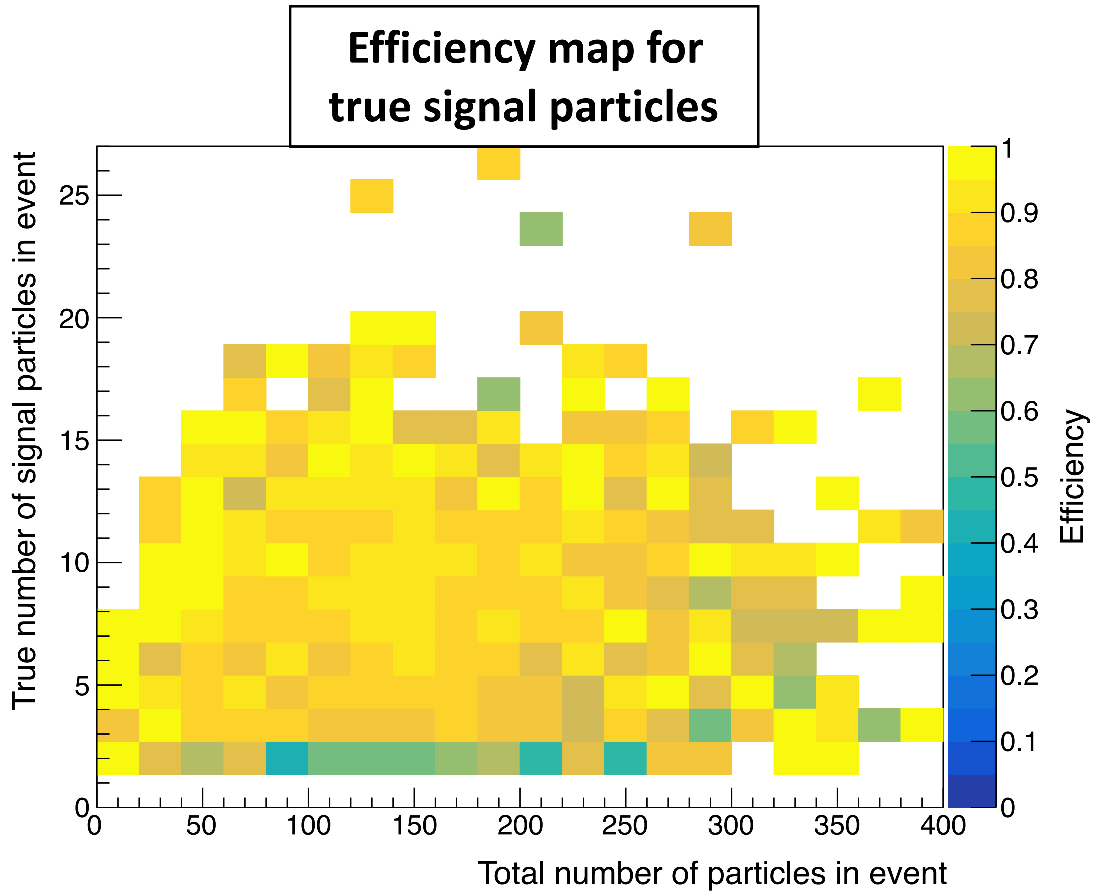
Trained performed in the GPU cluster of the Future SOC Lab.

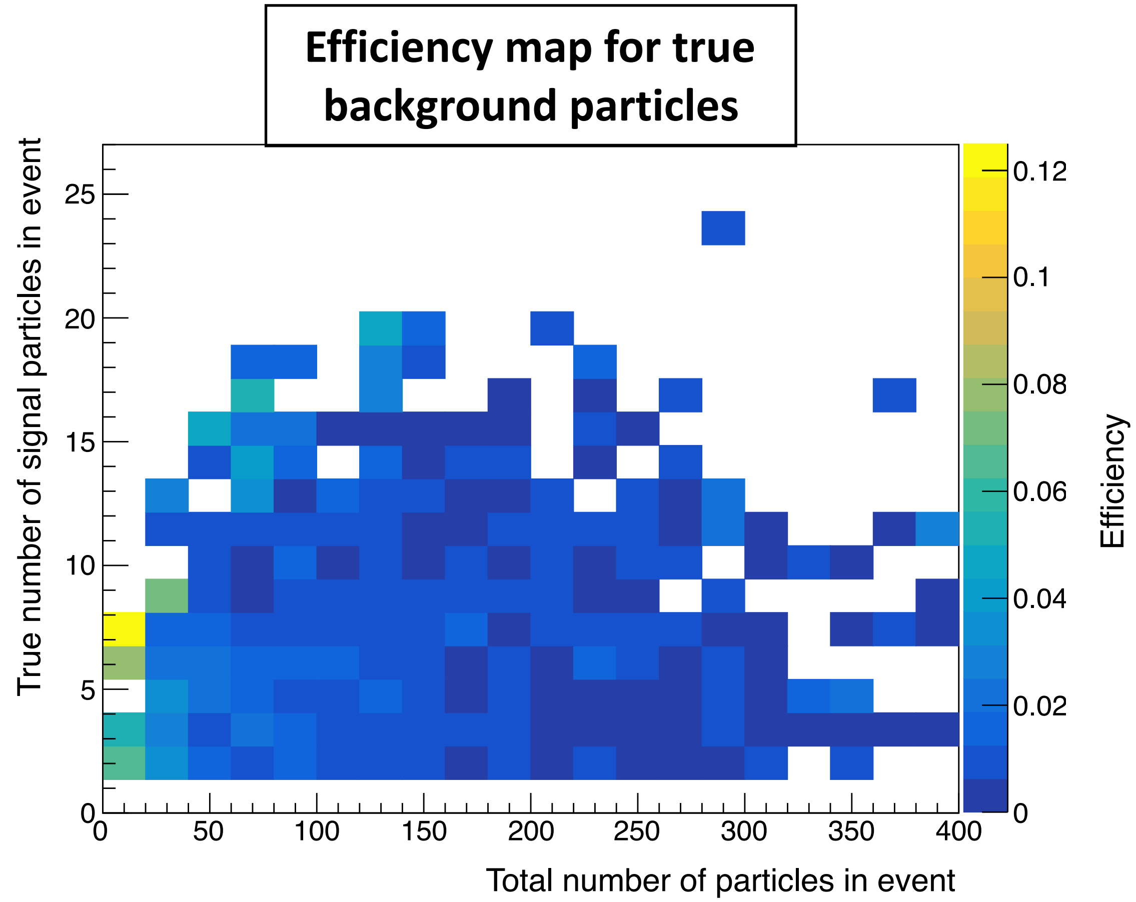

Our algorithm
Problem and chosen approach
Training of the algorithm
Summary and outlook
Performance studies
Expansion to neutral particles

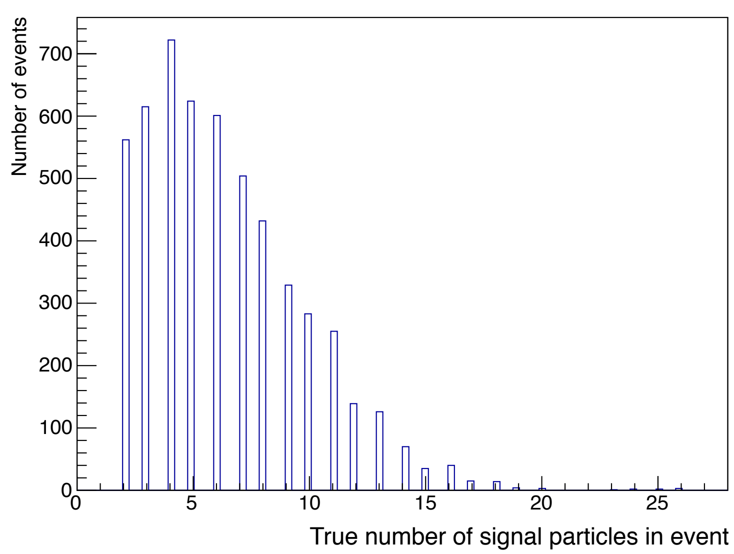
Performance: signal vs. background

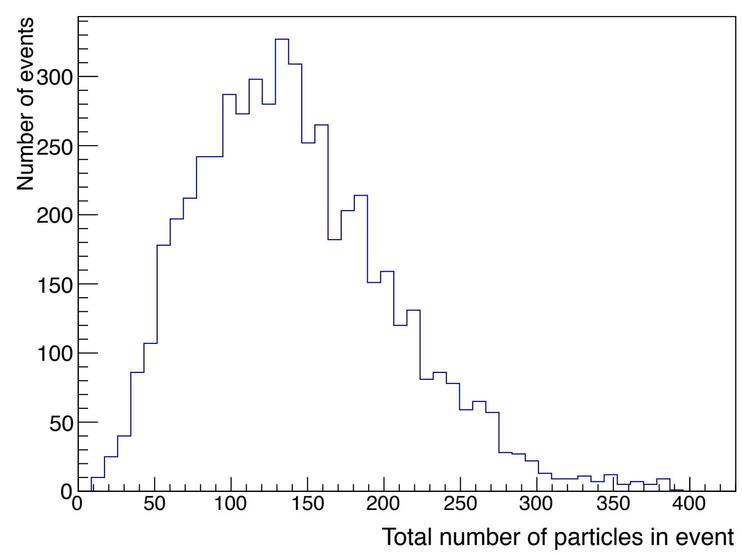
(Signal ≡ having a b-hadron ancestor)

Performance after the "topological" LCA metric reconstruction, evaluated in a batch of ~6000 events.



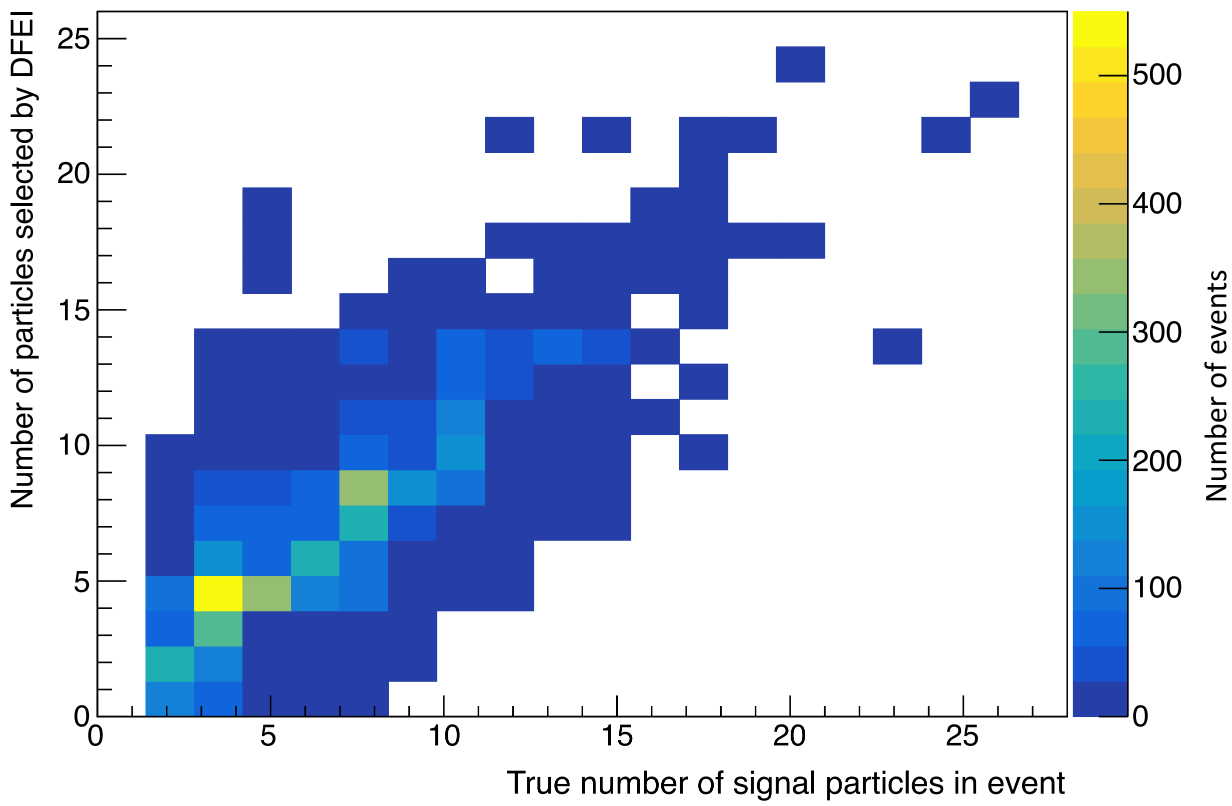


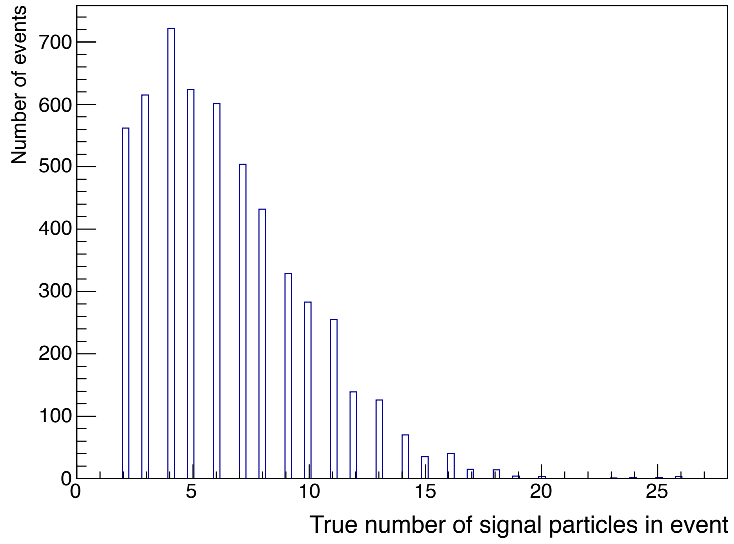


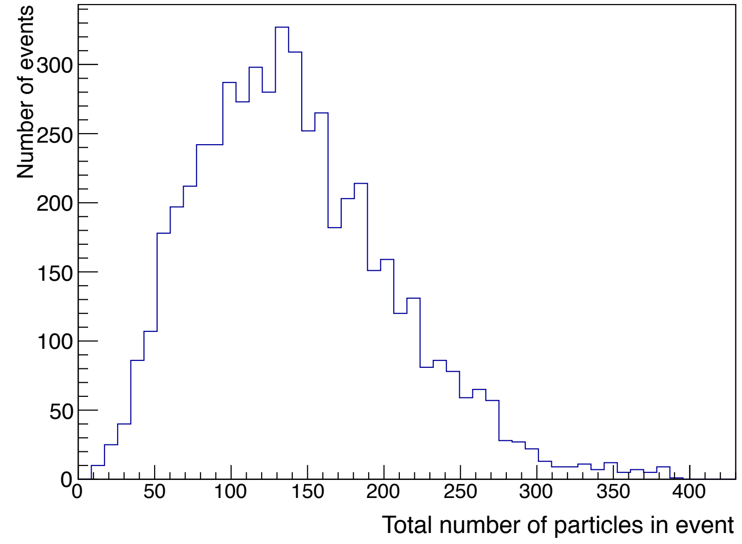


Performance: reduction of event size

Performance after the “topological” LCA metric reconstruction, evaluated in a batch of ~6000 events.



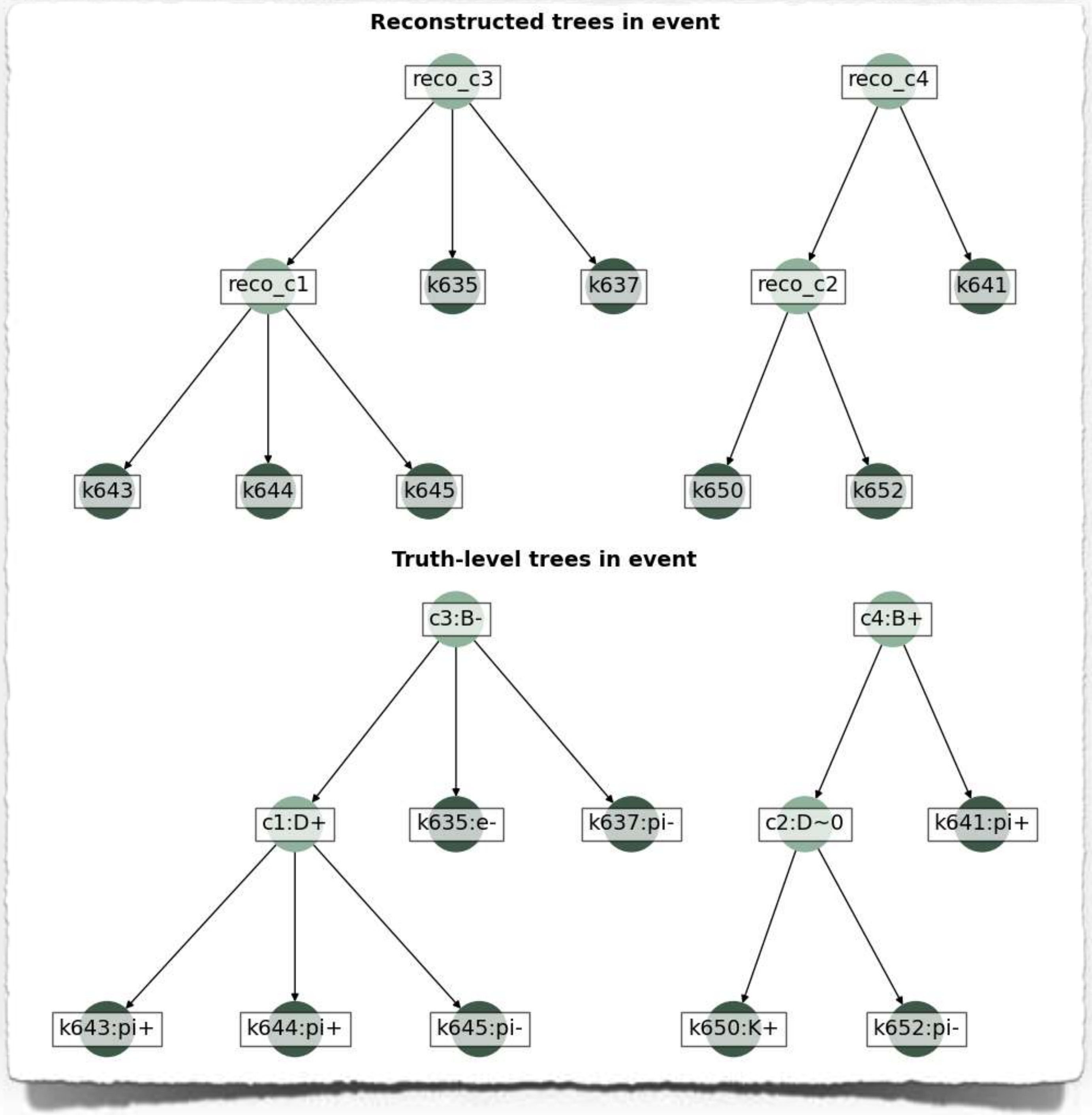




Performance: reconstruction of decay chains

Perfect reconstruction ≡ removal of all the bkg. particles + selection of all the signal particles + inference of all the ancestors + perfect connections of all the final-state particles and ancestors.
→ **Extremely challenging … but possible!**

Example of a perfect reconstruction achieved by our algorithm:



kX (key X): unique identifier for each stable particle in the event.

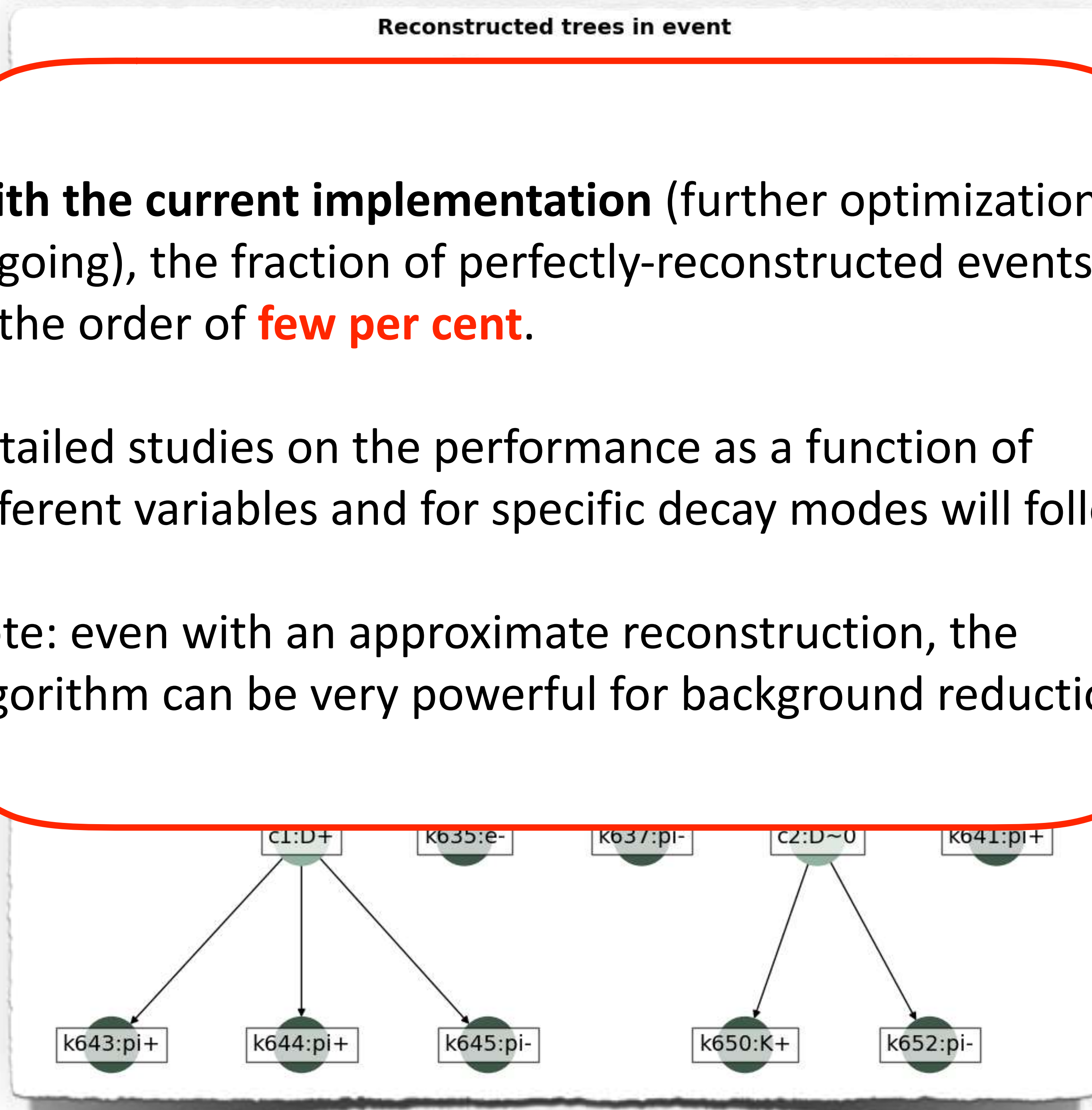
reco_cX: reconstructed ancestor (or cluster, c) X

(NOTE: truth-level background particles not shown, for simplicity).

Performance: reconstruction of decay chains

Perfect reconstruction ≡ removal of all the bkg. particles + selection of all the signal particles + inference of all the ancestors + perfect connections of all the final-state particles and ancestors.
→ **Extremely challenging … but possible!**

Example of a perfect reconstruction achieved by our algorithm:



With the current implementation (further optimizations ongoing), the fraction of perfectly-reconstructed events is of the order of **few per cent**.

Detailed studies on the performance as a function of different variables and for specific decay modes will follow.

Note: even with an approximate reconstruction, the algorithm can be very powerful for background reduction!

kX (key X): unique identifier for each stable particle in the event.

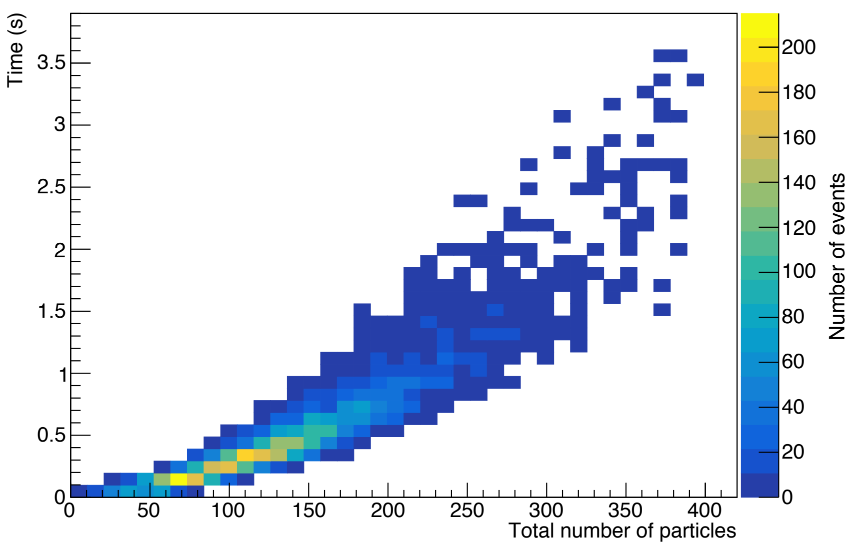
reco_cX: reconstructed ancestor (or cluster, c) X

(NOTE: truth-level background particles not shown, for simplicity).

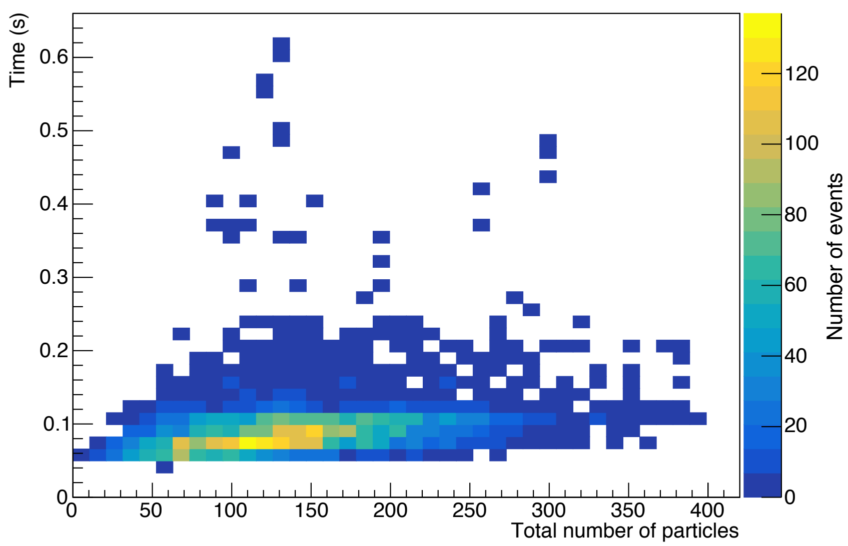
Performance: timing

Simplistic study (no parallelisation, no hardware accelerators*, algorithm to be further optimised), to **understand which are the slowest parts of the algorithm and how they scale with the total number of particles per event.**

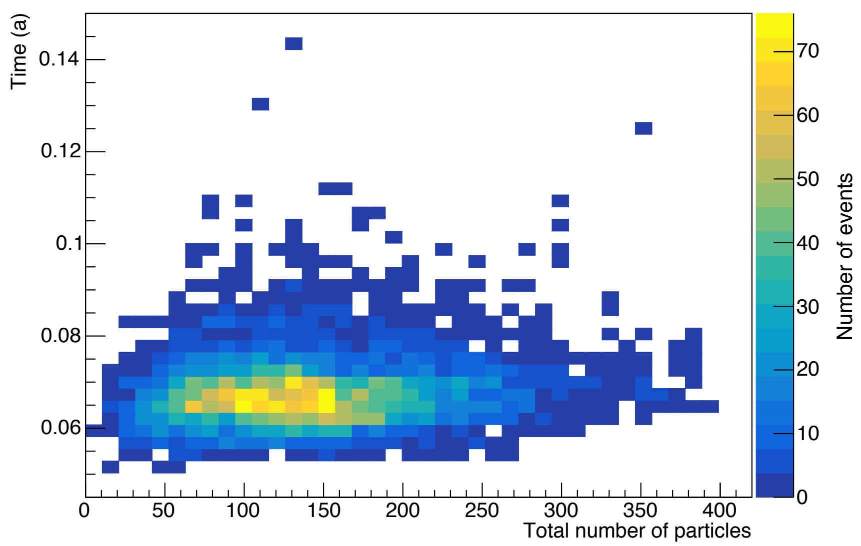
Node pruning



Edge pruning



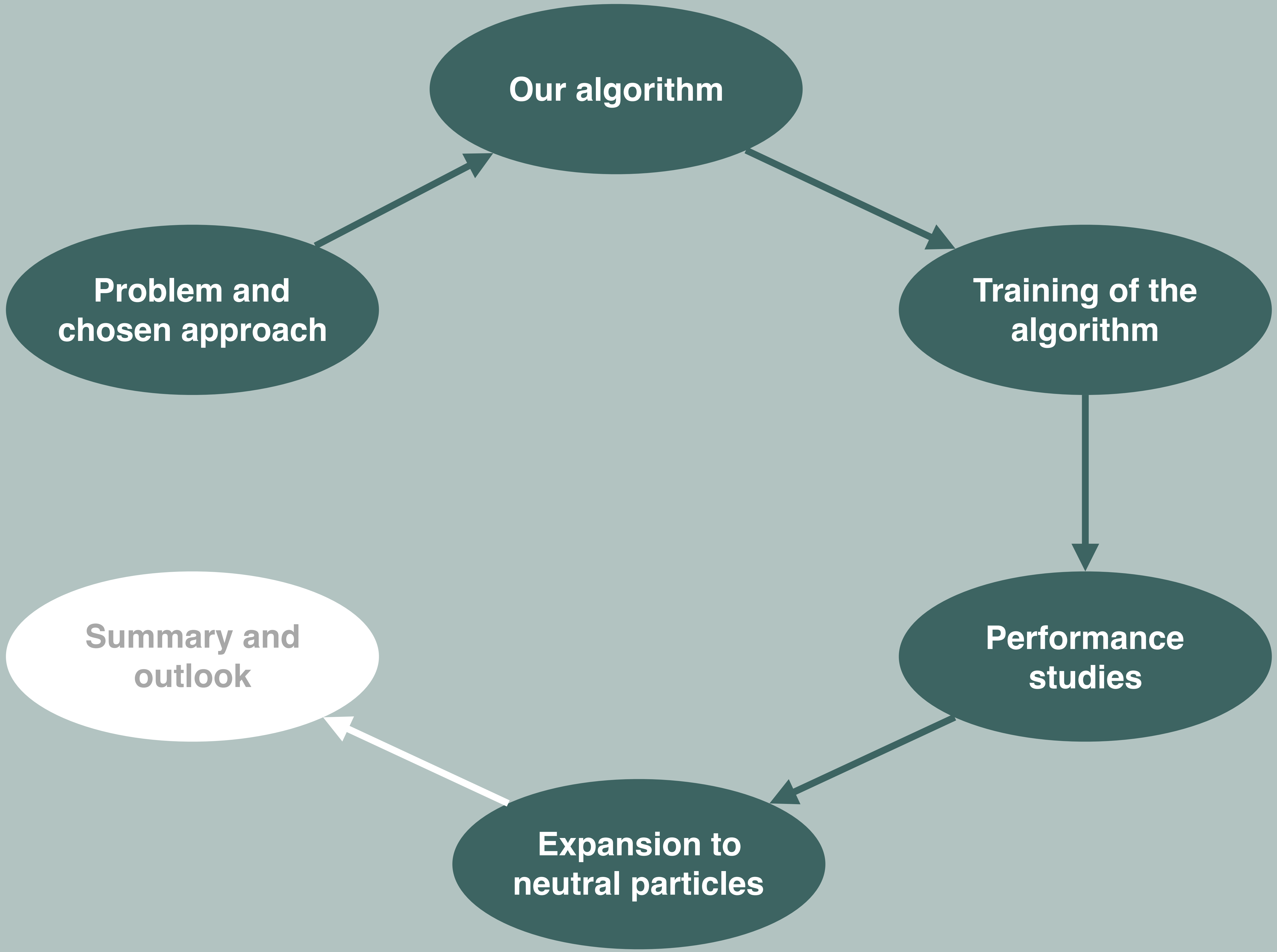
"Topological" LCA reconstruction



The slowest part is the node pruning, which also has the strongest dependency on the number of particles. → Many possible ways of optimisation.

The processing time of the subsequent algorithms is quite stable regarding changes in event complexity.

(*) Study done on a darwin-x86_64 architecture with a 2.8 GHz Intel Core i7 processor.

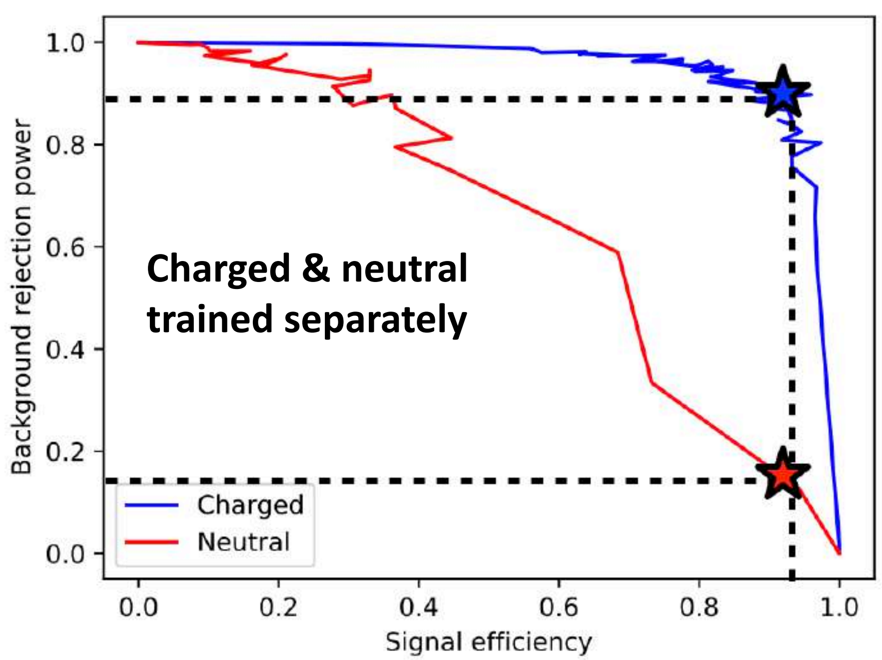

Our algorithm
Training of the algorithm
Performance studies
Expansion to neutral particles
Summary and outlook
Problem and chosen approach

Expansion to neutral particles (very preliminary)

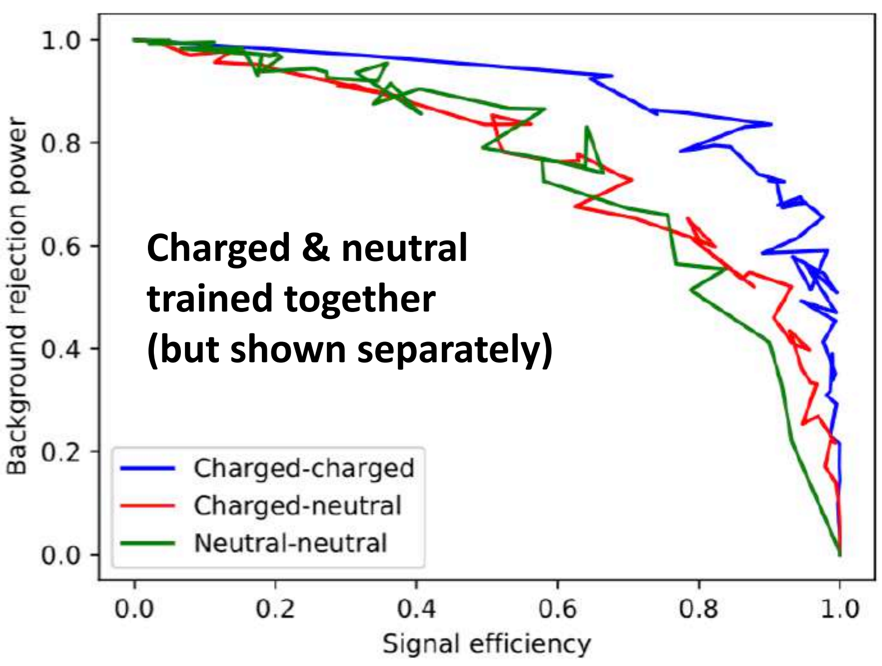
Goal: add the **photons and neutral pions** (more than half of the number of charged particles).

Technical issue #1: much harder to pre-select the edges. → An input charged+neutral graph would be enormous.
➥ **Our approach: do an independent node pruning for charged and neutral, then combine them for the subsequent steps.**

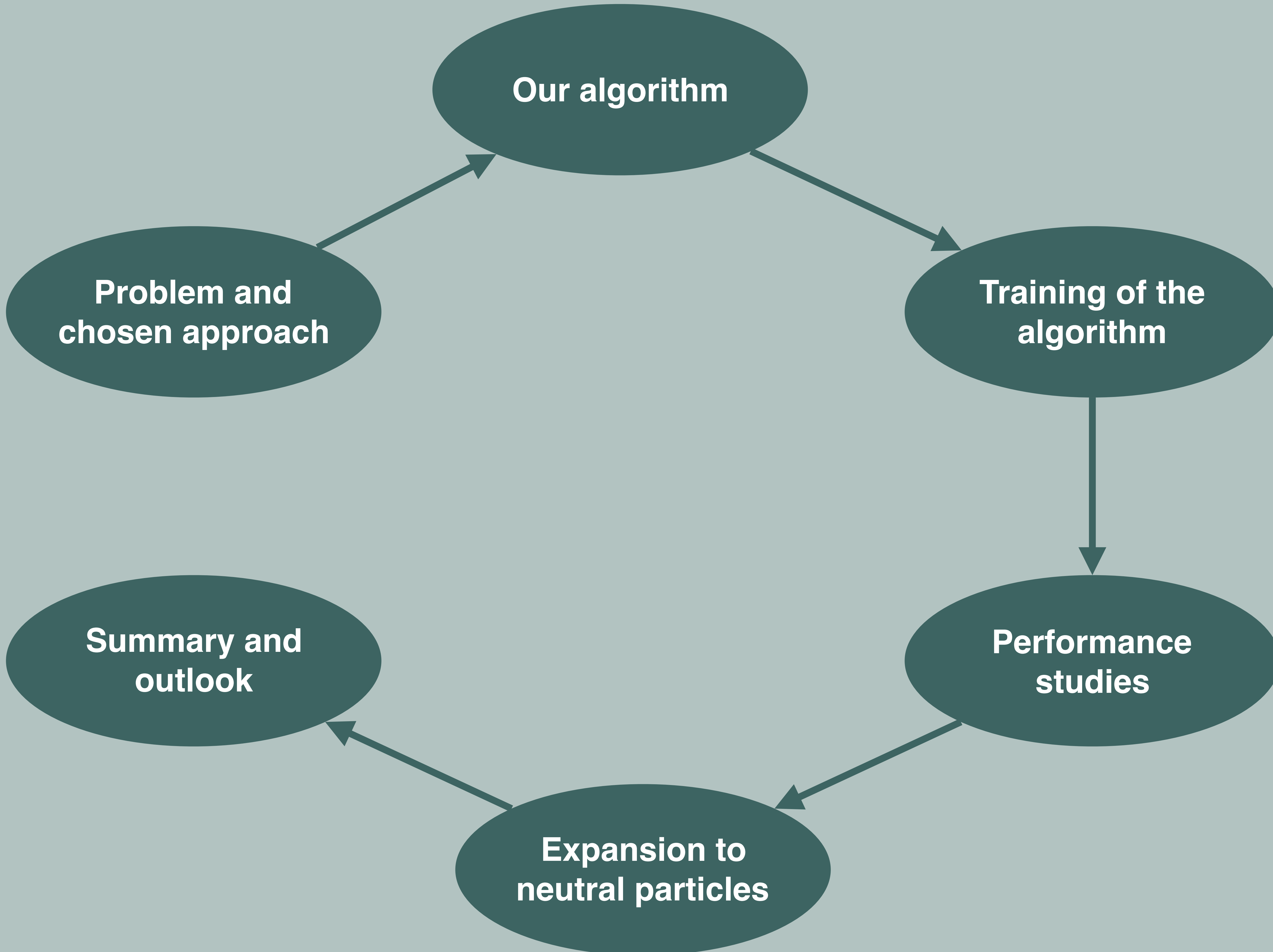
Node pruning - ROC curve



Edge pruning - ROC curve



Technical issue #2: the "topological" LCA reconstruction does not allow to process neutrals (no vertex information).
➥ We will add them in future developments, once our hierarchical reconstruction becomes more general.


Our algorithm
Training of the algorithm
Performance studies
Expansion to neutral particles
Summary and outlook
Problem and chosen approach

Summary

Up to now, LHCb is using a signal-based trigger

➡ Offline-quality reconstruction at trigger level.

➡ Expanded with functionality for a selective persistence of information in the event.

Profiting from the past trigger developments, and aiming at solving the future huge combinatorics challenges from the increased multiplicity, **we propose a new approach**: **a Deep-learning based Full Event Interpretation for LHCb**.

➡ **Goal: automatic and accurate identification and reconstruction of all the heavy-hadron decay chains per event.**

We have developed the first prototype of the algorithm, focused on charged final-state particles and (inclusive) b-hadron decays.

➡ **Very promising performance in realistic conditions!**

Outlook

Many future studies branching out from this work!

- ➥ **Optimise the current algorithm**, both in terms of performance and time.
  - ➥ Explore different ways of **accelerating GNNs**.
- ➥ Study the **performance on specific key channels** for LHCb.
- ➥ Prepare (parts of) the current algorithm to be able to run on the **RTA system of LHCb**.
- ➥ Expand from the “topological” LCA to the full LCA (exploit invariant mass information, etc.).
- ➥ Add the neutrals to the LCA-reconstruction step.
- ➥ Add the final-state-particle PID information, and develop a way to label the PID of the reconstructed ancestors.
  - ➥ Combination of the GNNs with **Reinforcement Learning** algorithms (basing on preliminary studies we have done).
- ➥ Adapt the algorithm to reconstruct c-hadrons, instead of b-hadrons.
- ➥ Study **data/MC differences** in performance.

Backup slides

Further bibliography

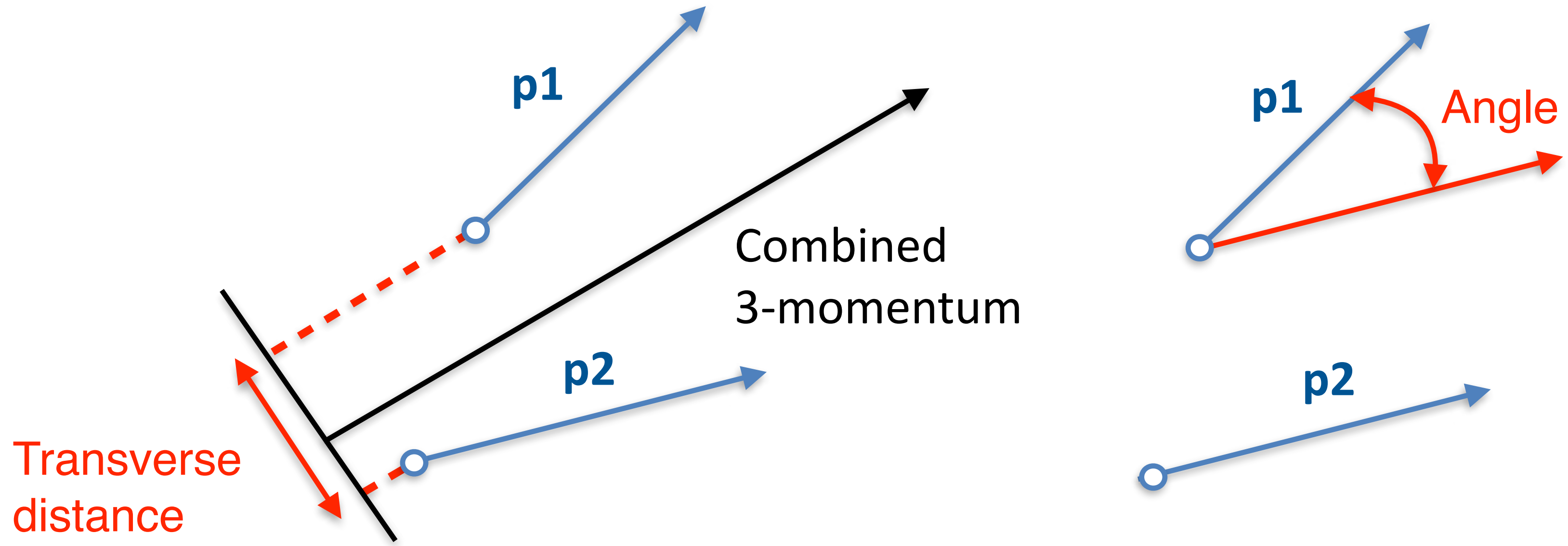
The LHCb Upgrades for Run3 and Run4: https://indico.cern.ch/event/868940/contributions/3813743/attachments/2081057/3495477/200725_ICHEP_LHCbUpgrades_v3.pdf

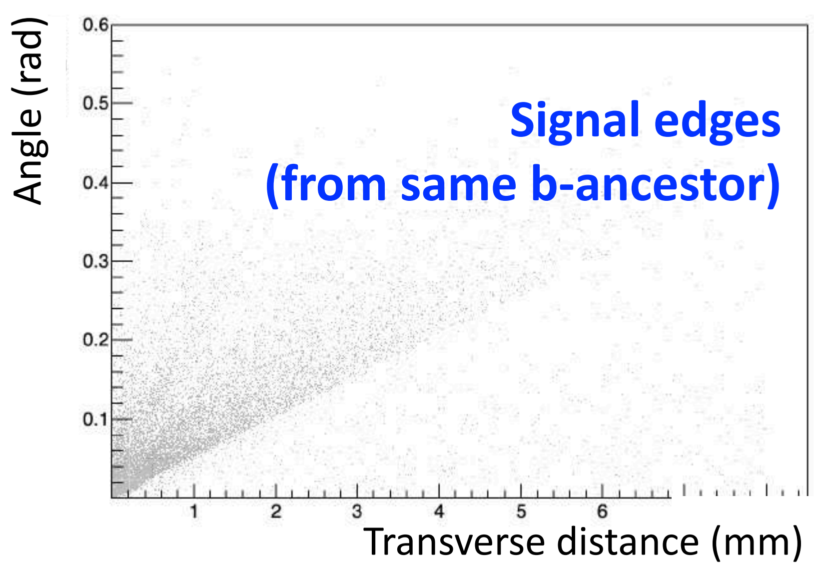
Performance estimates for Run3 conditions, used in our private simulation:

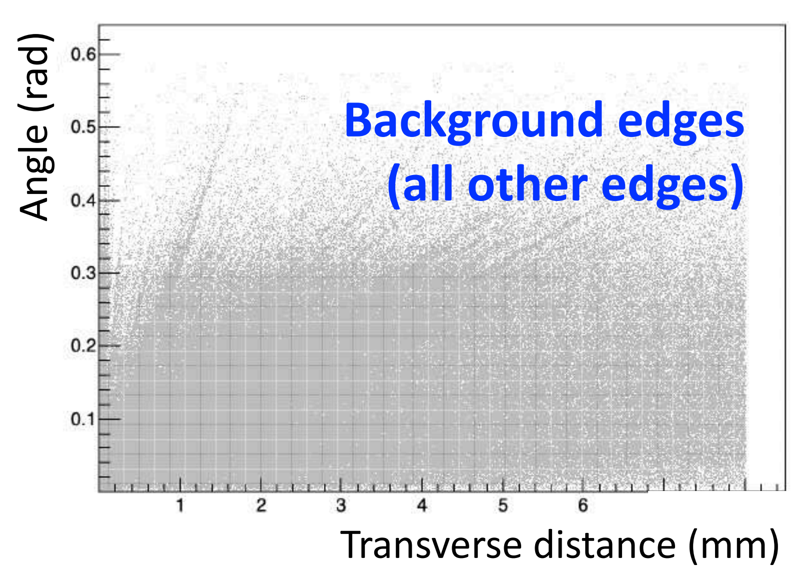
- Smearing of the true PV positions: https://indico.cern.ch/event/831165/contributions/3717129/attachments/2022791/3382986/ctd_2020_freiss.pdf
- Smearing of different reconstructed quantities: https://twiki.cern.ch/twiki/bin/view/LHCb/ConferencePlots and Computer Physics Communications 265, 108026 (2021).
- Geometry of the Vertex Locator: https://cds.cern.ch/record/2147229/files/10.1016_j.nima.2016.04.077.pdf

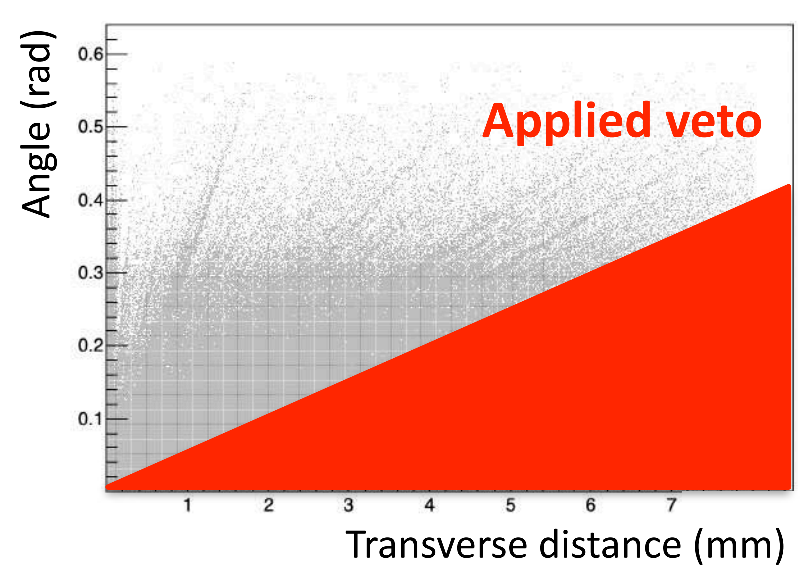
Cut-based edge pruning

Define two adequate topological variables for each edge (pair of particles)







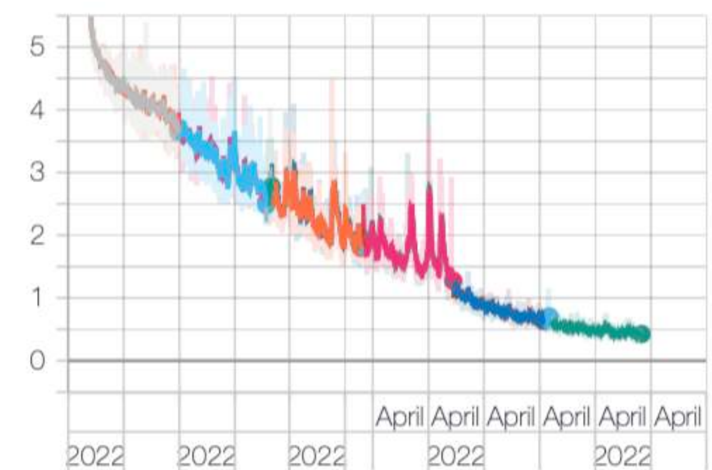


This veto reduces on average 60% of the total number of edges in the graph.
It also reduces connections between signal tracks, but it only leaves ~2% of the signal tracks fully disconnected.

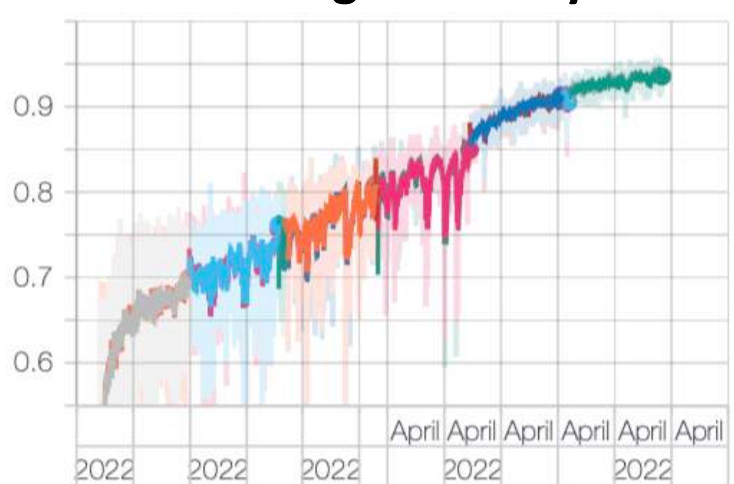
Training of the "topological" LCA reconstruction

Training split in 6 steps, each of them doing 4000 iterations in batches of 128 events.

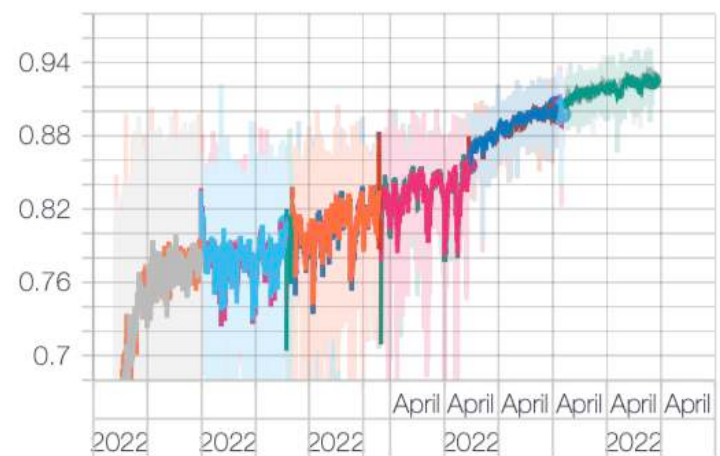
Loss



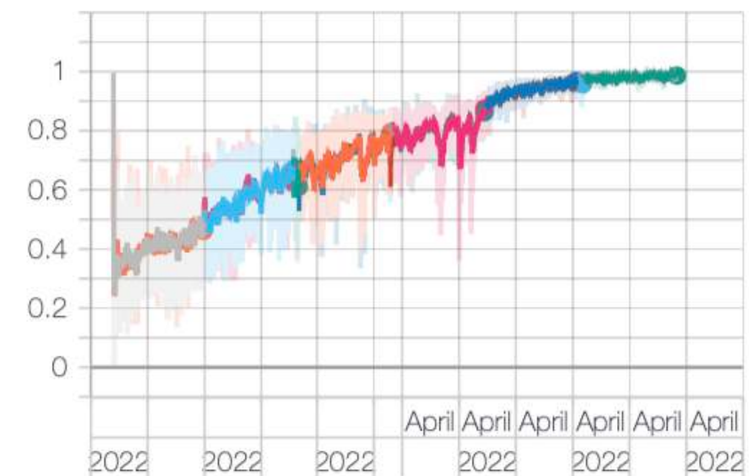
Average accuracy



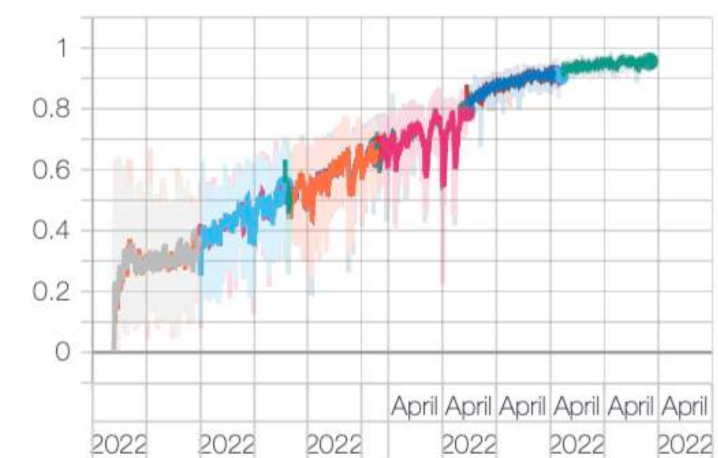
Accuracy (true LCA = 0)



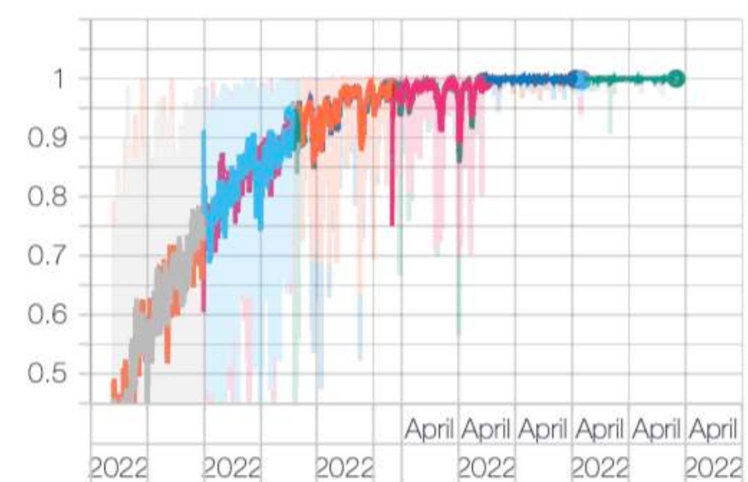
Accuracy (true LCA = 1)



Accuracy (true LCA = 2)



Accuracy (true LCA = 3)



No signs of overtraining (training and test curves always ~overlapping).