JAVIER DUARTE

IRIS-HEP TOPICAL MEETING

APRIL 21, 2021

FAIR4HEP: FINDABLE, ACCESSIBLE, INTEROPERABLE, AND REUSABLE FRAMEWORKS FOR AI IN HEP

UCSD

FAIR4HEP TEAM (FAIR4HEP.GITHUB.IO)


Eliu Huerta
PI, ANL


Daniel S. Katz
Co-PI, NCSA


Volodymyr Kindratenko
Co-PI, NCSA


Mark Neubauer
Co-PI, UIUC


Zhizhen Zhao
Co-PI, UIUC


Priscilla Cushman
Co-PI, UMN


Andrew Furmanski
Co-PI, UMN


Vuk Mandic
Co-PI, UMN

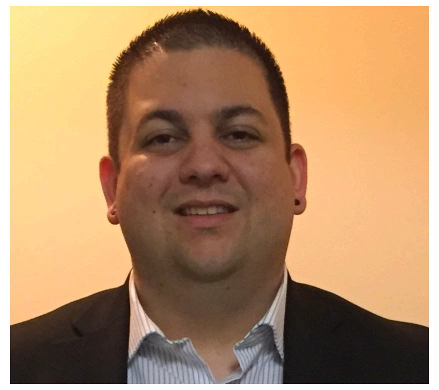

Roger Rusack
Co-PI, UMN


Ju-Sun
Co-PI, UMN


Phil Harris
Co-PI, MIT

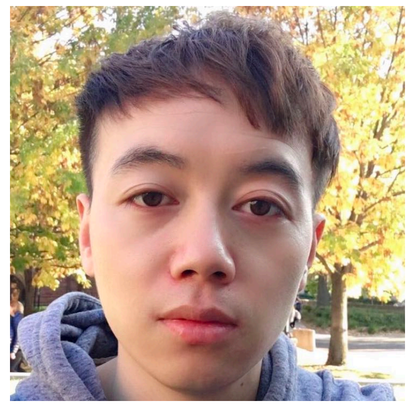

Javier Duarte
Co-PI, UCSD


Andrew Evans
Postdoc, UMN

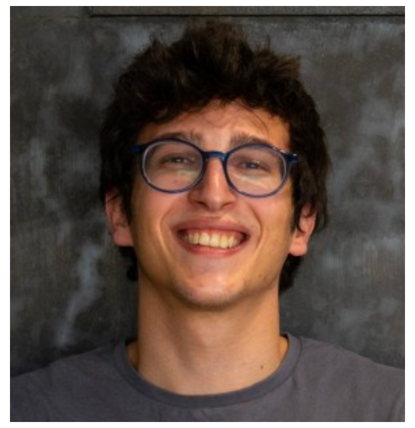

Daniel Diaz
Postdoc, UCSD

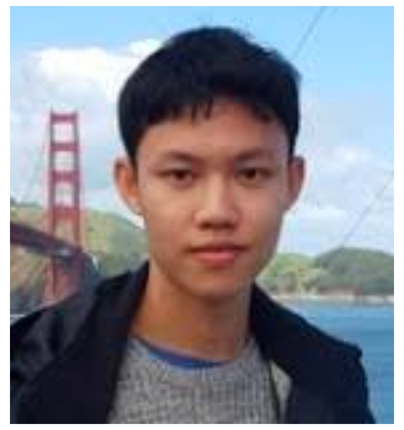

Sangeon Park
PhD Student, MIT


Yifan Chen
MS Student, UIUC


Taihui Li
PhD Student, UMN


Raghav Kansal
PhD Student, UCSD


Farouk Mokhtar
PhD Student, UCSD


Steven Tsan
Undergrad, UCSD

UCSD

U.S. DEPARTMENT OF ENERGY | Office of Science

FAIR4HEP MISSION

- DOE ASCR-funded collaboration (3-year project)
  - To advance our understanding of the relationship between our data and AI models by empowering scientists to explore both through the development of frameworks adhering to the principles of **f**indability, **a**ccessibility, **i**nteroperability, and **r**eusability (FAIR)
  - Using HEP as the science use-case
    - Investigate FAIR ways to share AI models and data
    - Create an environment where novel approaches to AI can be explored and applied to new data
    - Enable new insights for applying AI techniques
- Collaborate with partners: CERN Open Data Portal, Zenodo, DLHub
- Operate within larger community: Australian Research Data Commons (ARDC), Research Data Alliance (RDA)
  - Note: BoF today on Fair for ML

OUTLINE

- Motivation
- FAIR Principles
- FAIR4HEP Projects
  - FAIR standards for data and AI in HEP
  - Develop example FAIR datasets and AI models:
    - H(bb) jet tagging
    - Particle-flow reconstruction
    - Among others
- Vision & Outlook

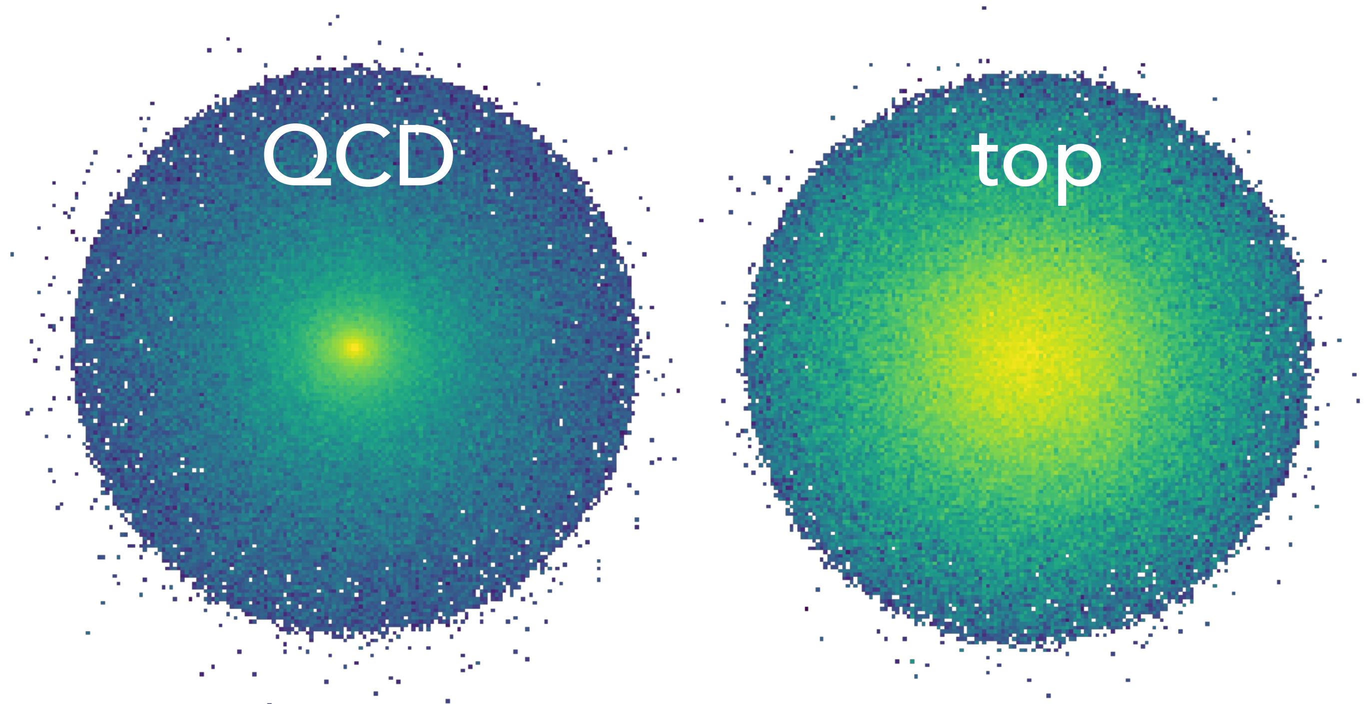
MOTIVATION 1: REFERENCE DATA SETS FOR HEP

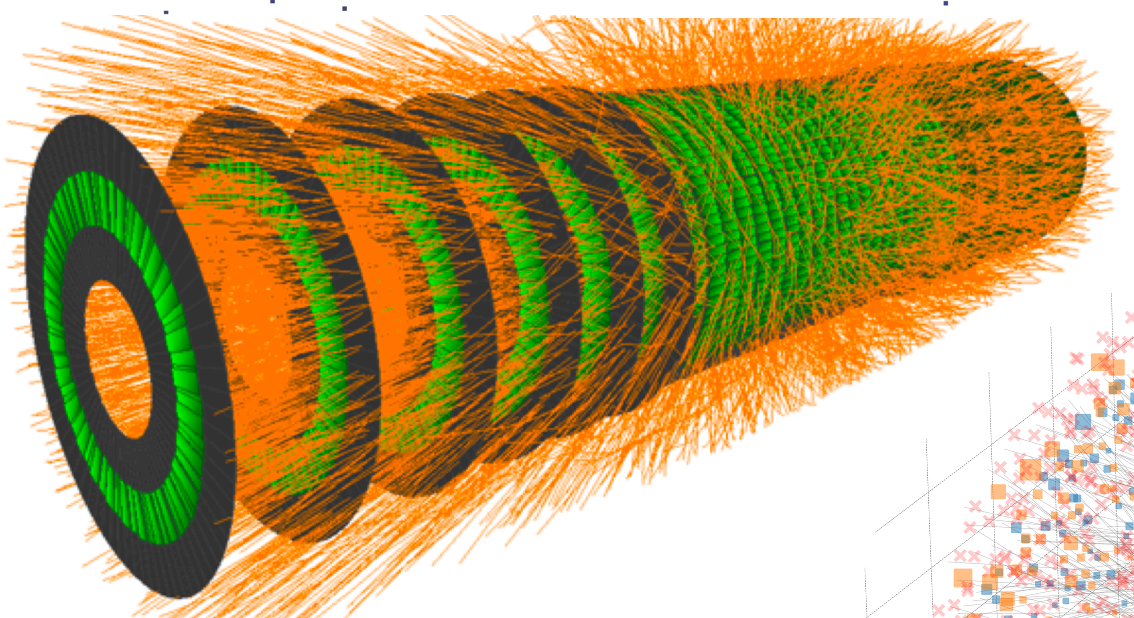
MOTIVATION 1: REFERENCE DATA SETS FOR HEP

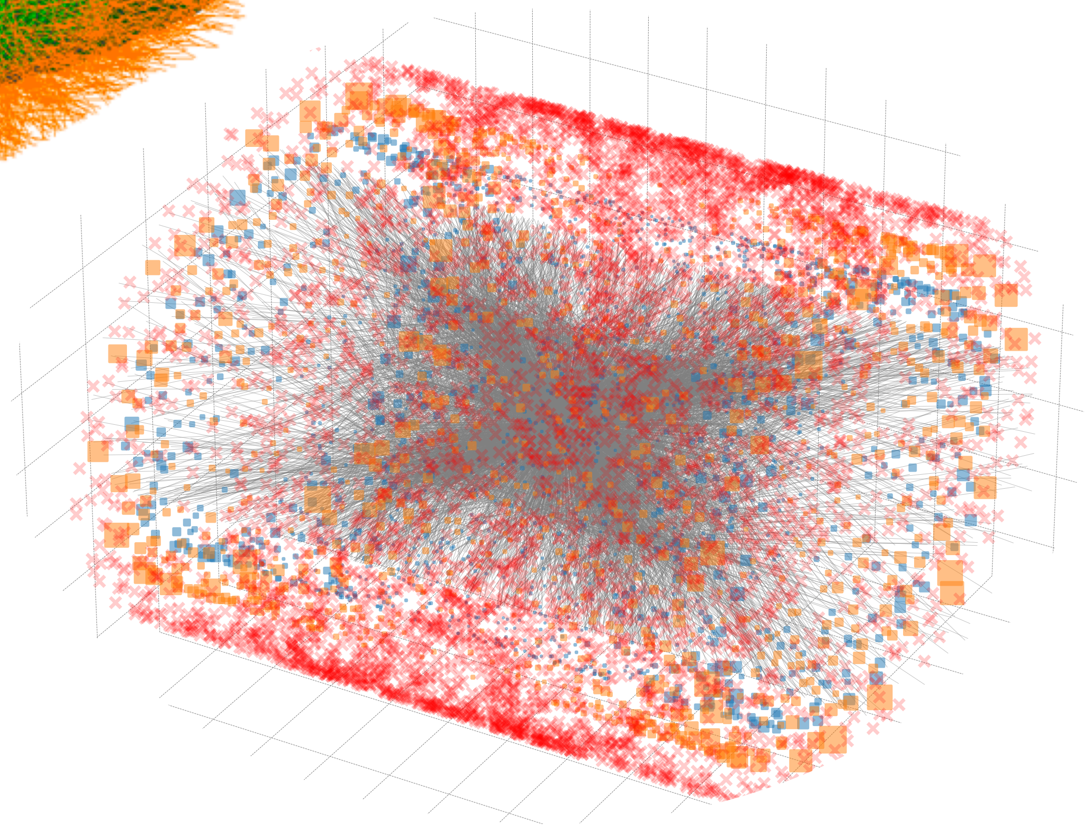
- Engage ML community for interesting, realistic tasks in experimental HEP
  - As ImageNet accelerated advances in computer vision, do the same for HEP

MOTIVATION 1: REFERENCE DATA SETS FOR HEP

- Engage ML community for interesting, realistic tasks in experimental HEP
  - As ImageNet accelerated advances in computer vision, do the same for HEP
- Calls at many workshops for more public HEP data sets with real detector simulation for ML applications
  - Example: dataset for top tagging based on Pythia+Delphes
  - Example: dataset for tracking based on ACTS (kaggle TrackML challenge)
  - Example: dataset for H(bb) tagging based on CMS open simulation
  - Example: dataset for particle-flow based on Pythia+Delphes
  - Example: dataset for ECAL crystal calibration (in preparation)

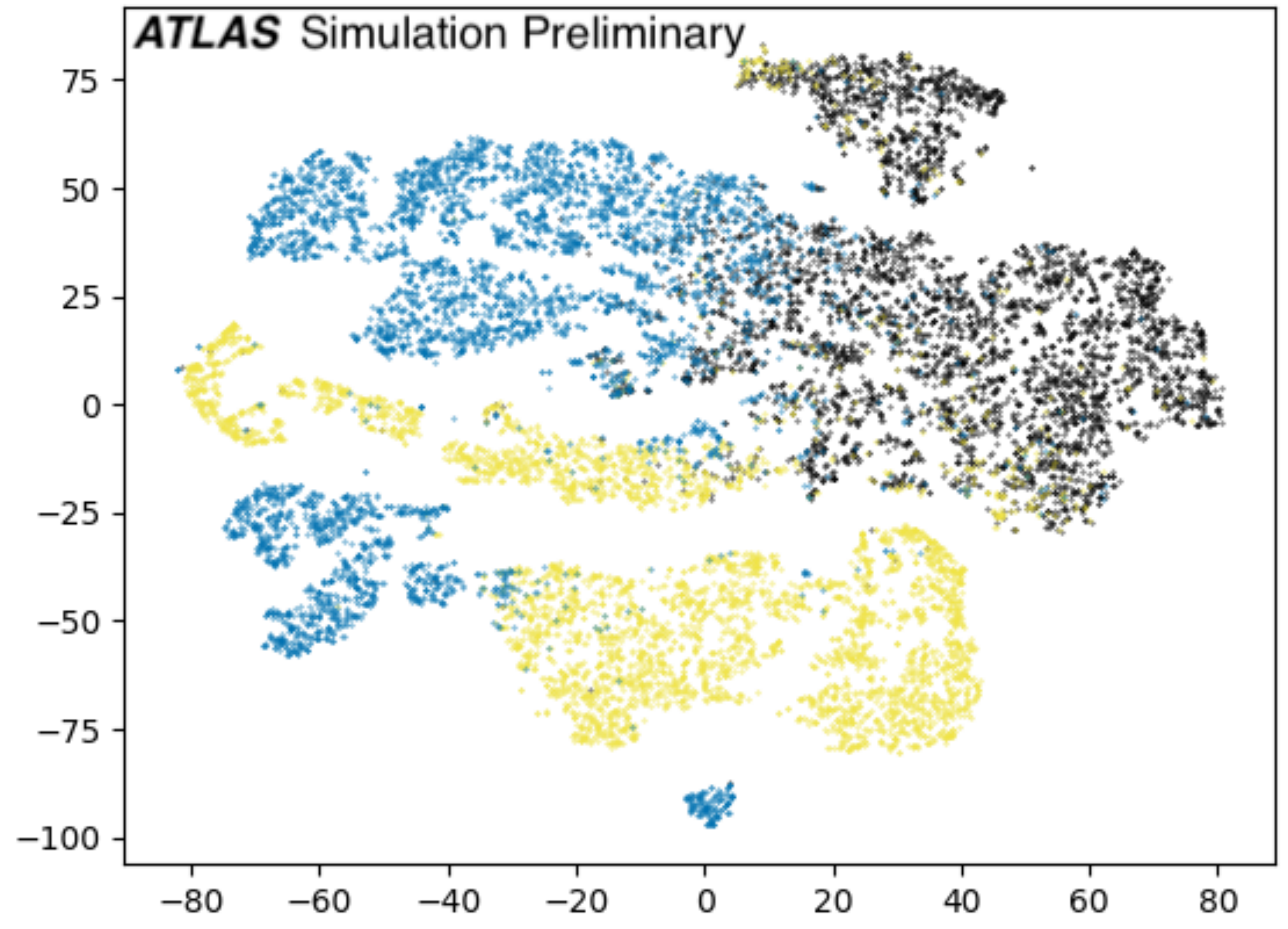






MOTIVATION 2: SHARING/REUSING AI MODELS ACROSS HEP

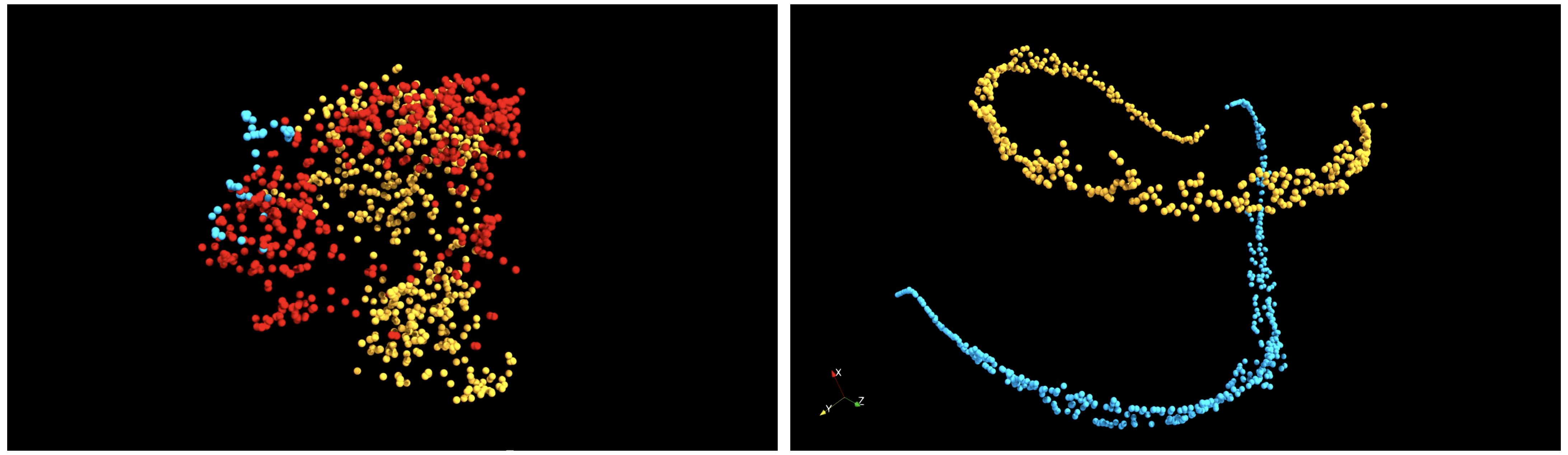
- Allow AI models developed for one experiment to be (re-)trained and (re-)used easily in another experiment
  - Example: ATLAS recently studied GravNet developed by CMS collaborators [https://cds.cern.ch/record/2753414] for physics object localization using point cloud segmentation



MOTIVATION 3: EXPLORING RELATIONSHIPS BETWEEN DATA AND AI MODELS

MOTIVATION 3: EXPLORING RELATIONSHIPS BETWEEN DATA AND AI MODELS

- Easier to build upon existing work (e.g. through transfer learning)

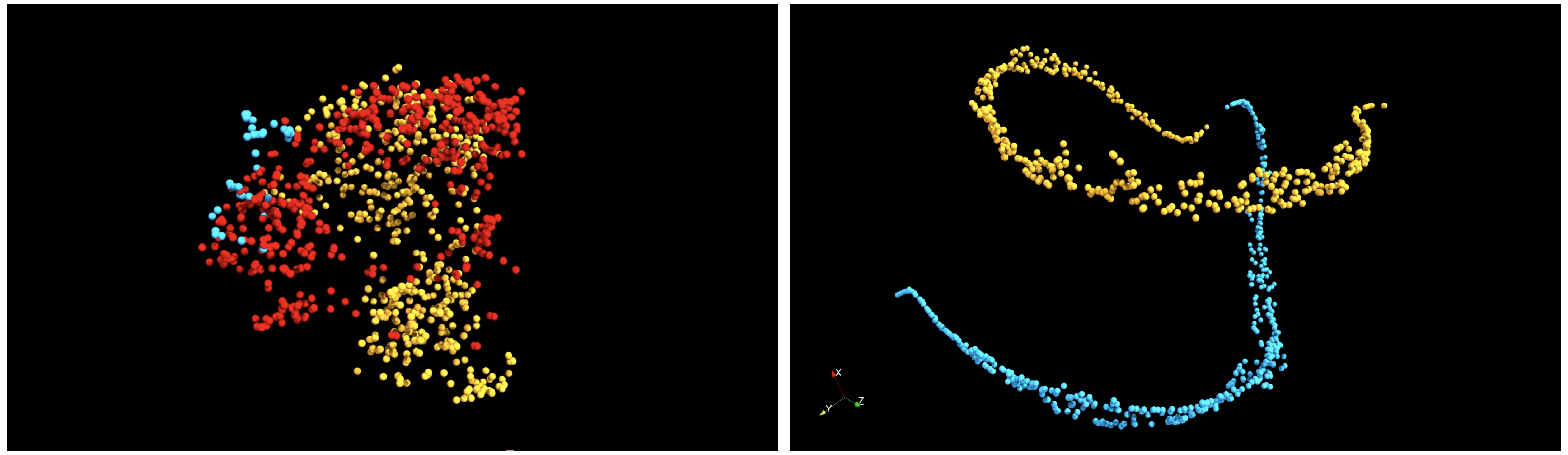


Left: Xception pre-trained on ImageNet applied to galaxies

Right: After fine-tuning on galaxy data, two galaxy clusters can be clearly identified

MOTIVATION 3: EXPLORING RELATIONSHIPS BETWEEN DATA AND AI MODELS

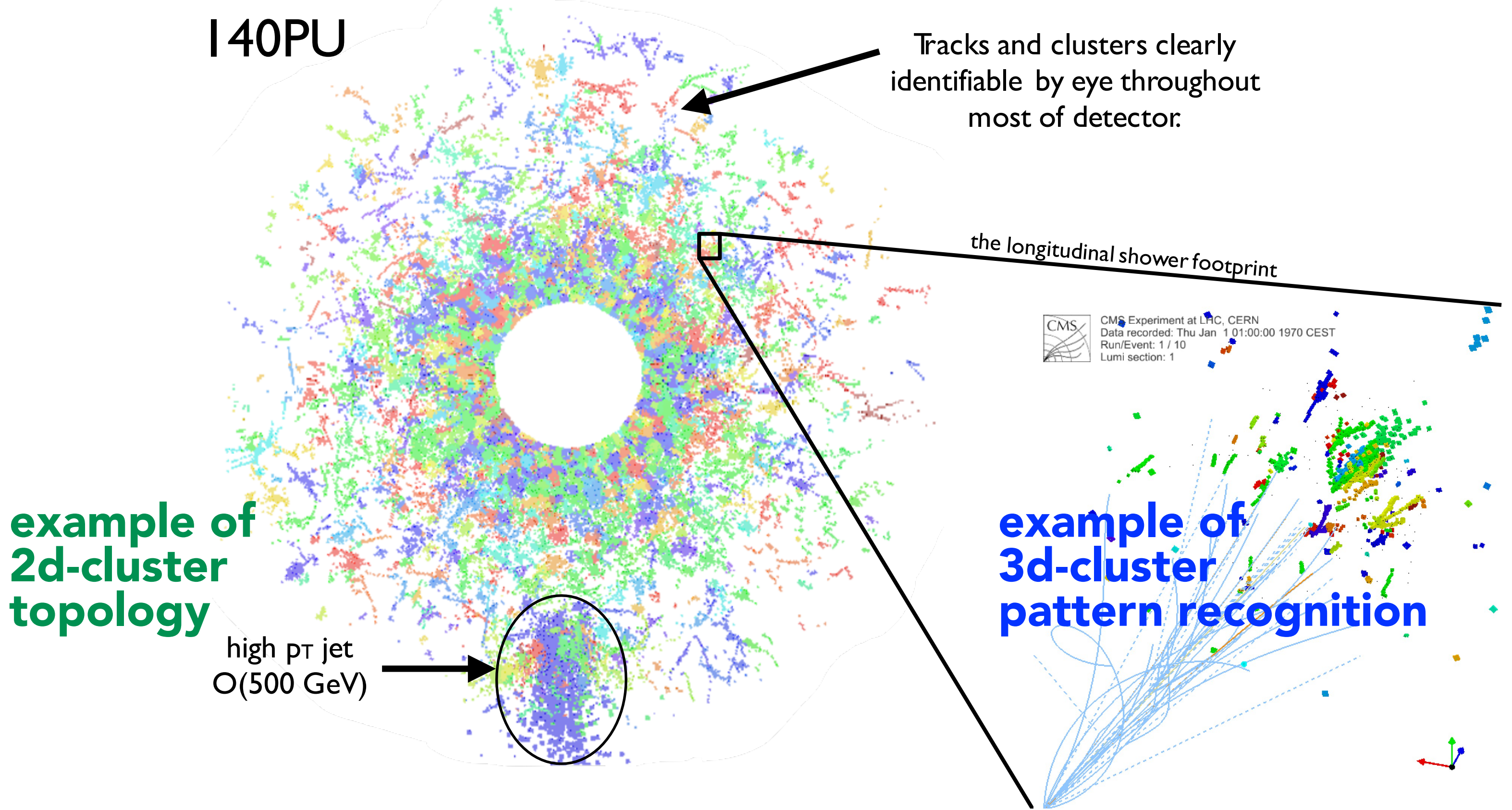
- Easier to build upon existing work (e.g. through transfer learning)

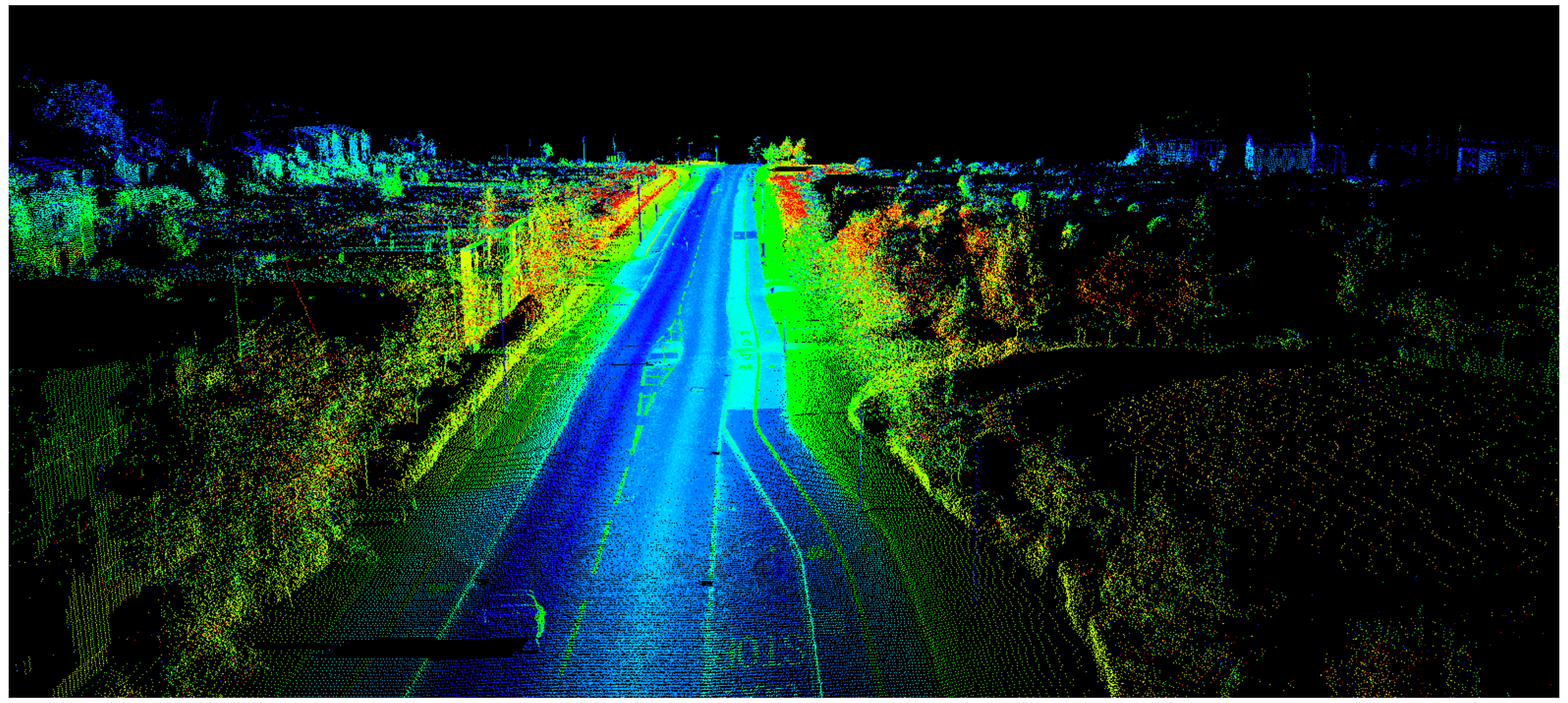


Left: Xception pre-trained on ImageNet applied to galaxies

Right: After fine-tuning on galaxy data, two galaxy clusters can be clearly identified

- Share work beyond HEP
  - AI models developed for HEP-specific tasks may be useful in other domains (e.g. LiDAR point cloud data)





FAIR DATA PRINCIPLES

Image: book.fosteropenscience.eu

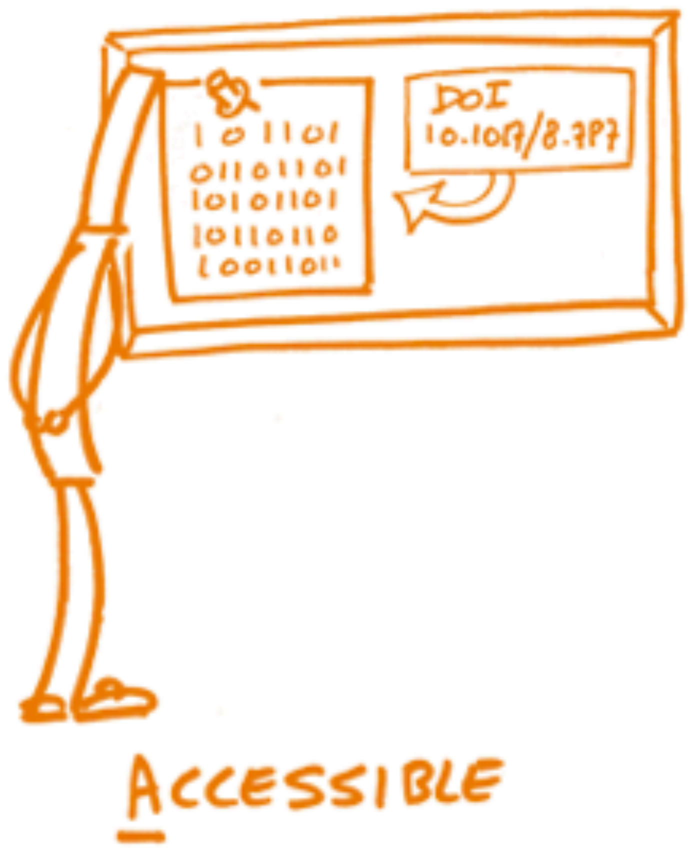
FAIR DATA PRINCIPLES

- F1. (meta)data have **unique** and **persistent** identifier
  F2. data are described with rich metadata
  F3. metadata specify the data identifier
  F4. (meta)data are registered or indexed in a searchable resource

Image: book.fosteropenscience.eu

FAIR DATA PRINCIPLES

Learn more: https://gofair.us/

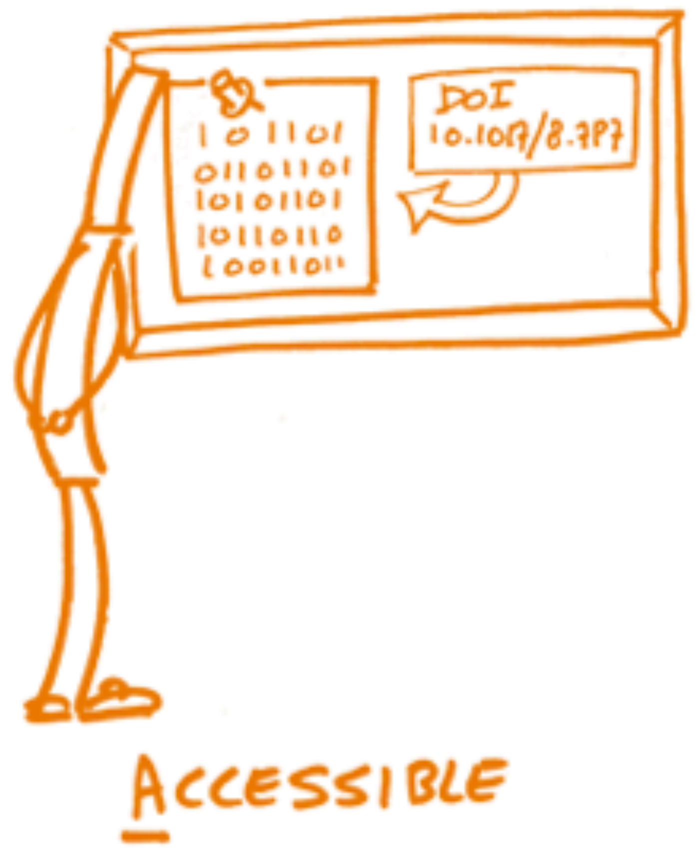


- F1. (meta)data have **unique** and **persistent** identifier
  F2. data are described with rich metadata
  F3. metadata specify the data identifier
  F4. (meta)data are registered or indexed in a searchable resource

Image: book.fosteropenscience.eu

FAIR DATA PRINCIPLES

Learn more: https://gofair.us/

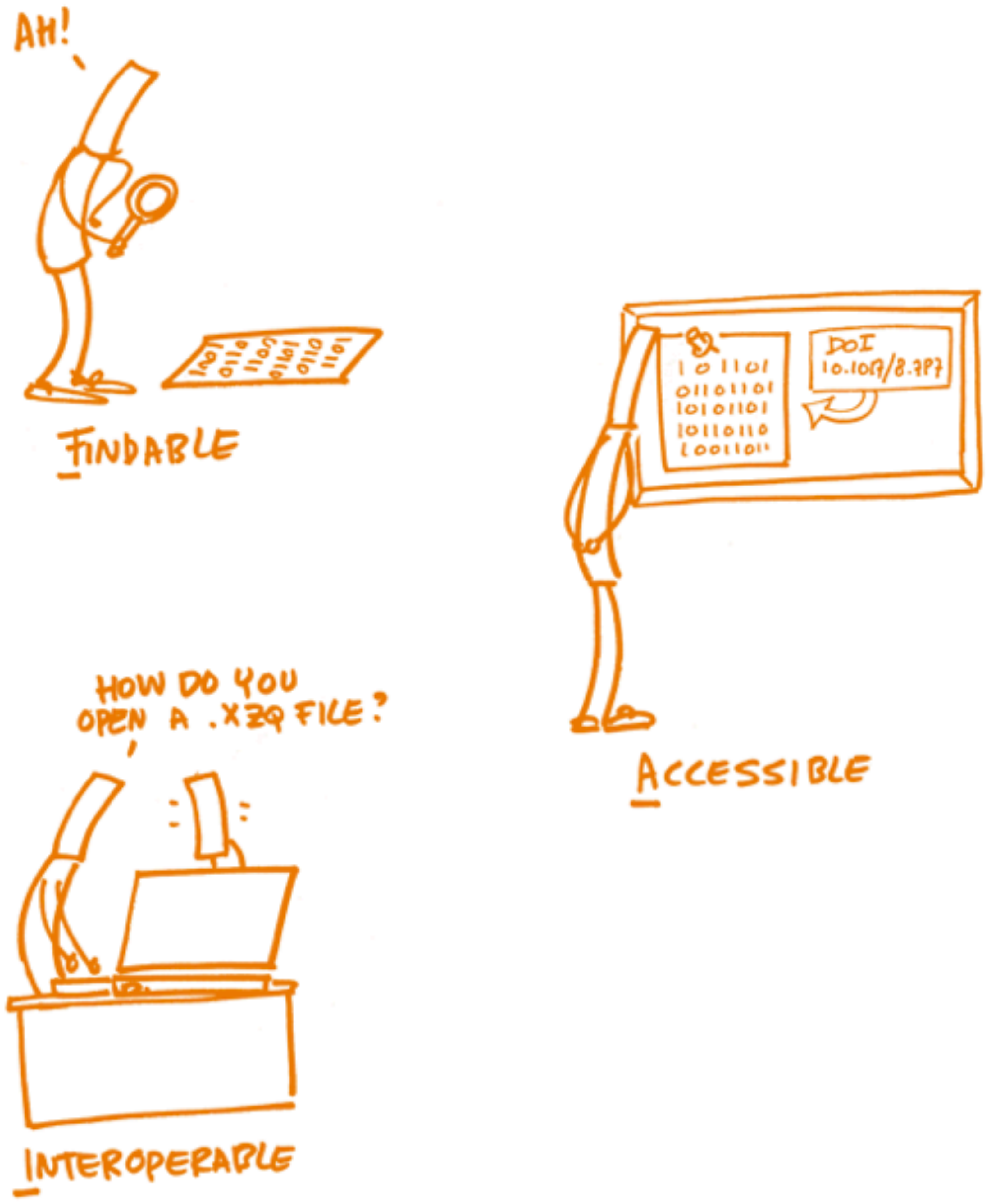


- F1. (meta)data have **unique** and **persistent** identifier
  F2. data are described with rich metadata
  F3. metadata specify the data identifier
  F4. (meta)data are registered or indexed in a searchable resource
- A1. (meta)data are retrievable using standardized protocol
  A1.1 protocol is open, free, and universally implementable
  A1.2 protocol allows for authentication and authorization
  A2. metadata are accessible, even when the data is not

Image: book.fosteropenscience.eu

FAIR DATA PRINCIPLES

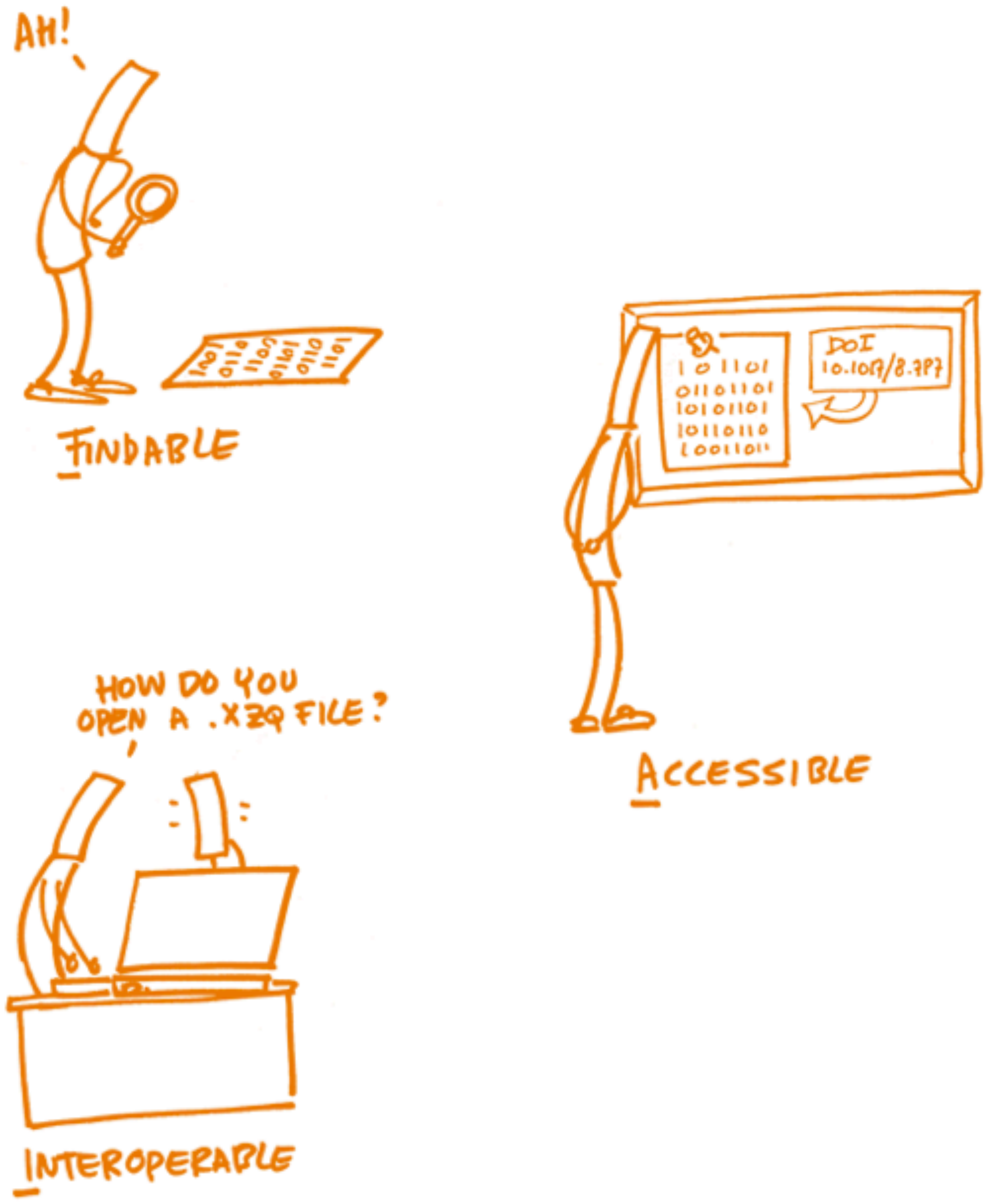
Learn more: https://gofair.us/



- F1. (meta)data have **unique** and **persistent** identifier
  F2. data are described with rich metadata
  F3. metadata specify the data identifier
  F4. (meta)data are registered or indexed in a searchable resource
- A1. (meta)data are retrievable using standardized protocol
  A1.1 protocol is open, free, and universally implementable
  A1.2 protocol allows for authentication and authorization
  A2. metadata are accessible, even when the data is not

Image: book.fosteropenscience.eu

FAIR DATA PRINCIPLES



- F1. (meta)data have **unique** and **persistent** identifier
  F2. data are described with rich metadata
  F3. metadata specify the data identifier
  F4. (meta)data are registered or indexed in a searchable resource
- A1. (meta)data are retrievable using standardized protocol
  A1.1 protocol is open, free, and universally implementable
  A1.2 protocol allows for authentication and authorization
  A2. metadata are accessible, even when the data is not
- I1. (meta)data use a formal, shared, and broadly applicable language for knowledge representation
  I2. (meta)data use **vocabularies** that follow FAIR principles
  I3. (meta)data include qualified references to other (meta)data

Image: book.fosteropenscience.eu

FAIR DATA PRINCIPLES

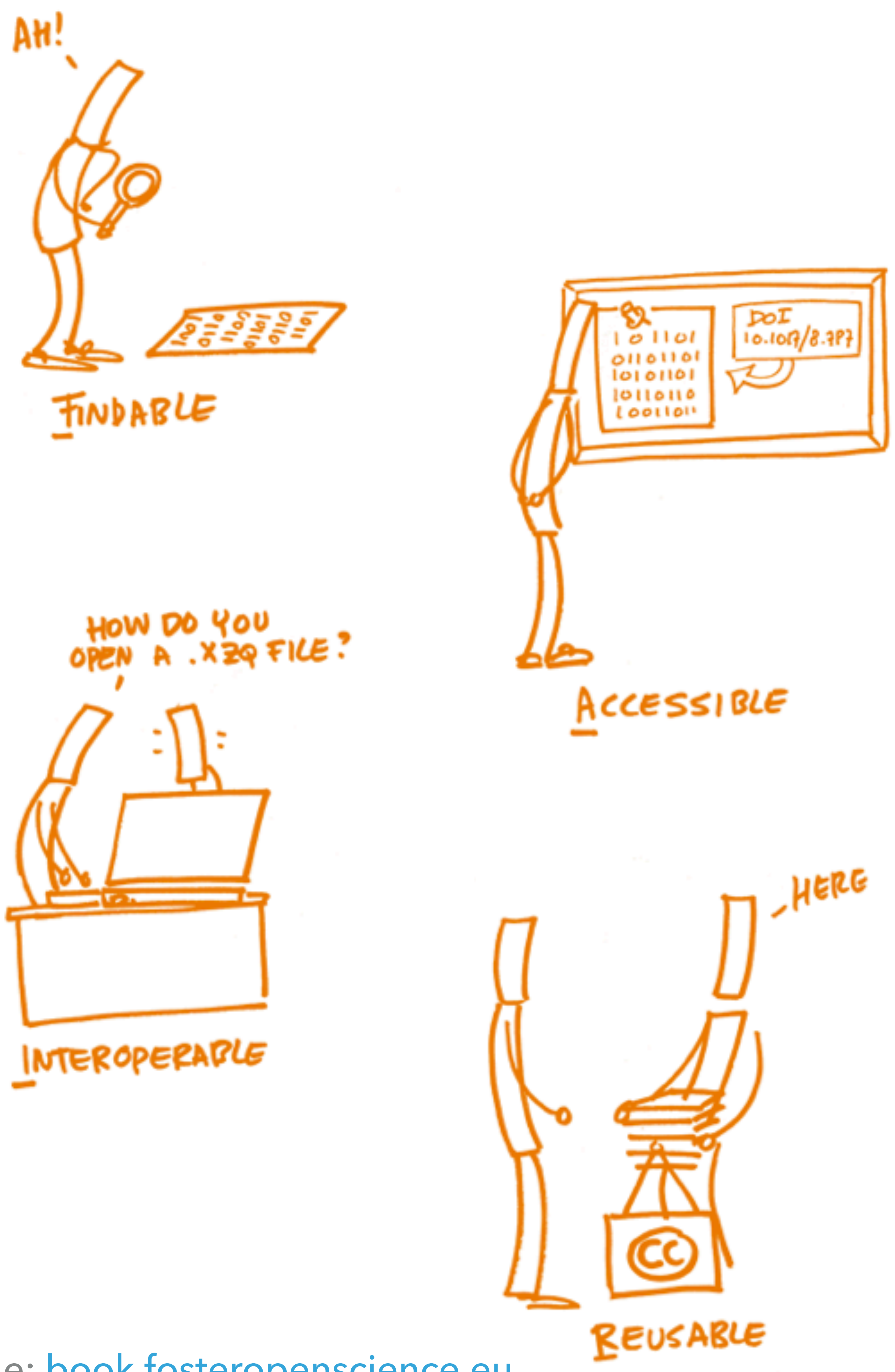
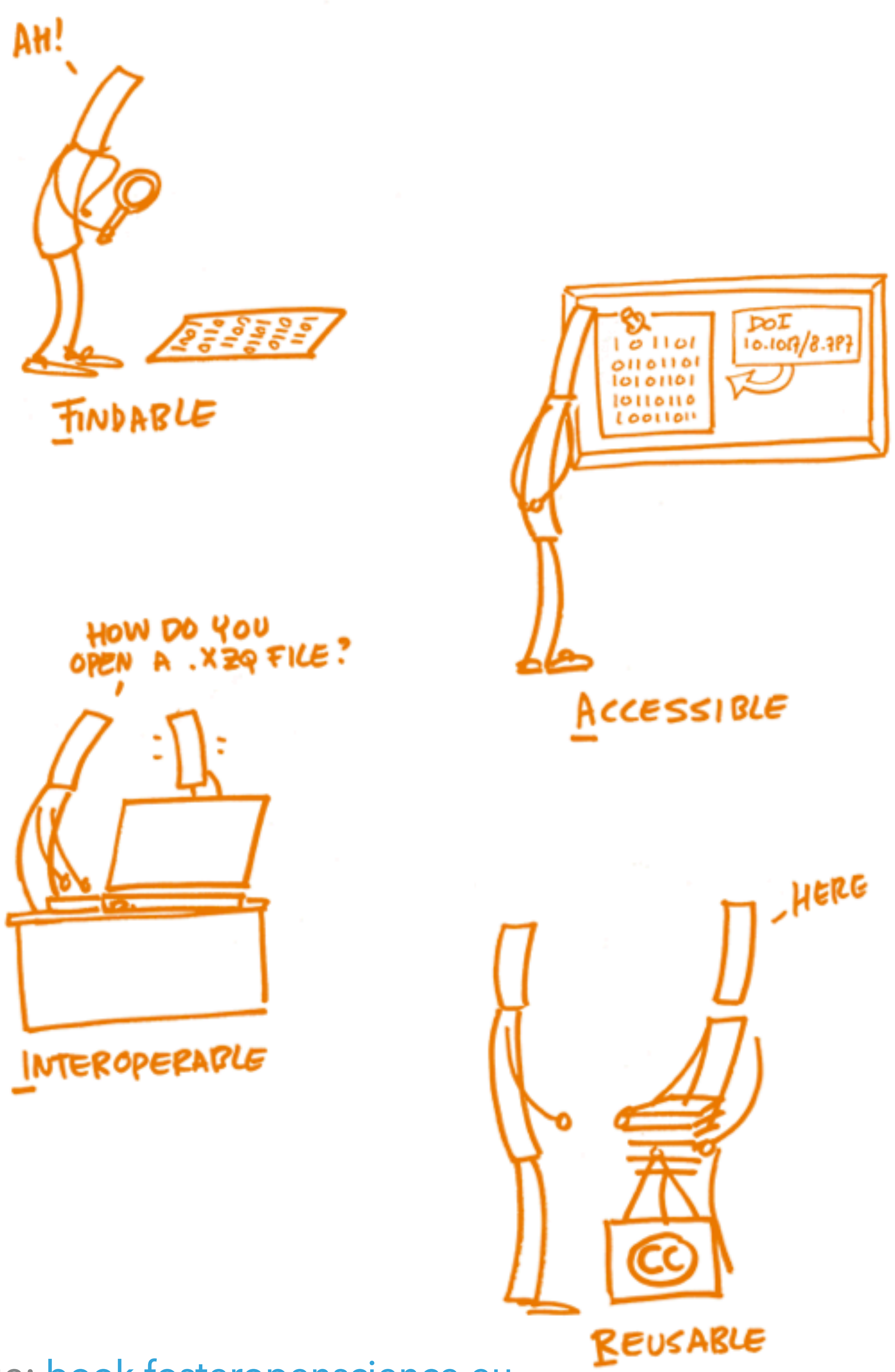


Image: book.fosteropenscience.eu

- F1. (meta)data have **unique** and **persistent** identifier
  F2. data are described with rich metadata
  F3. metadata specify the data identifier
  F4. (meta)data are registered or indexed in a searchable resource
- A1. (meta)data are retrievable using standardized protocol
  A1.1 protocol is open, free, and universally implementable
  A1.2 protocol allows for authentication and authorization
  A2. metadata are accessible, even when the data is not
- I1. (meta)data use a formal, shared, and broadly applicable language for knowledge representation
  I2. (meta)data use **vocabularies** that follow FAIR principles
  I3. (meta)data include qualified references to other (meta)data

FAIR DATA PRINCIPLES

Learn more: https://gofair.us/



- F1. (meta)data have **unique** and **persistent** identifier
  F2. data are described with rich metadata
  F3. metadata specify the data identifier
  F4. (meta)data are registered or indexed in a searchable resource
- A1. (meta)data are retrievable using standardized protocol
  A1.1 protocol is open, free, and universally implementable
  A1.2 protocol allows for authentication and authorization
  A2. metadata are accessible, even when the data is not
- I1. (meta)data use a formal, shared, and broadly applicable language for knowledge representation
  I2. (meta)data use **vocabularies** that follow FAIR principles
  I3. (meta)data include qualified references to other (meta)data
- R1. meta(data) have a plurality of accurate and relevant attributes
  R1.1. (meta)data have clear and accessible data usage license
  R1.2. (meta)data are associated with their provenance
  R1.3. (meta)data meet domain-relevant community standards

Image: book.fosteropenscience.eu

PROJECT: FAIR STANDARDS FOR DATA AND AI IN HEP

UCSD

FAIR STANDARDS FOR DATA AND AI IN HEP

- Are HEP public datasets FAIR?
  - Develop/refine FAIR checklist for HEP datasets
- What does FAIR mean for AI models (or software generally)?
  - Develop a standard protocol to follow to publish FAIR AI models
- How do users contribute their own FAIR data and AI models?
  - Create mechanisms for users to contribute
- As users adopt these FAIR4HEP standards, it will be easier to
  - Publish citable AI models (with credit, etc.)
  - Retrain published AI models on new (also FAIR) datasets
  - Extend published AI models for new tasks
  - Explore the relationships between AI models and data

PROJECT: DEVELOPING EXAMPLE FAIR DATASETS AND AI MODELS

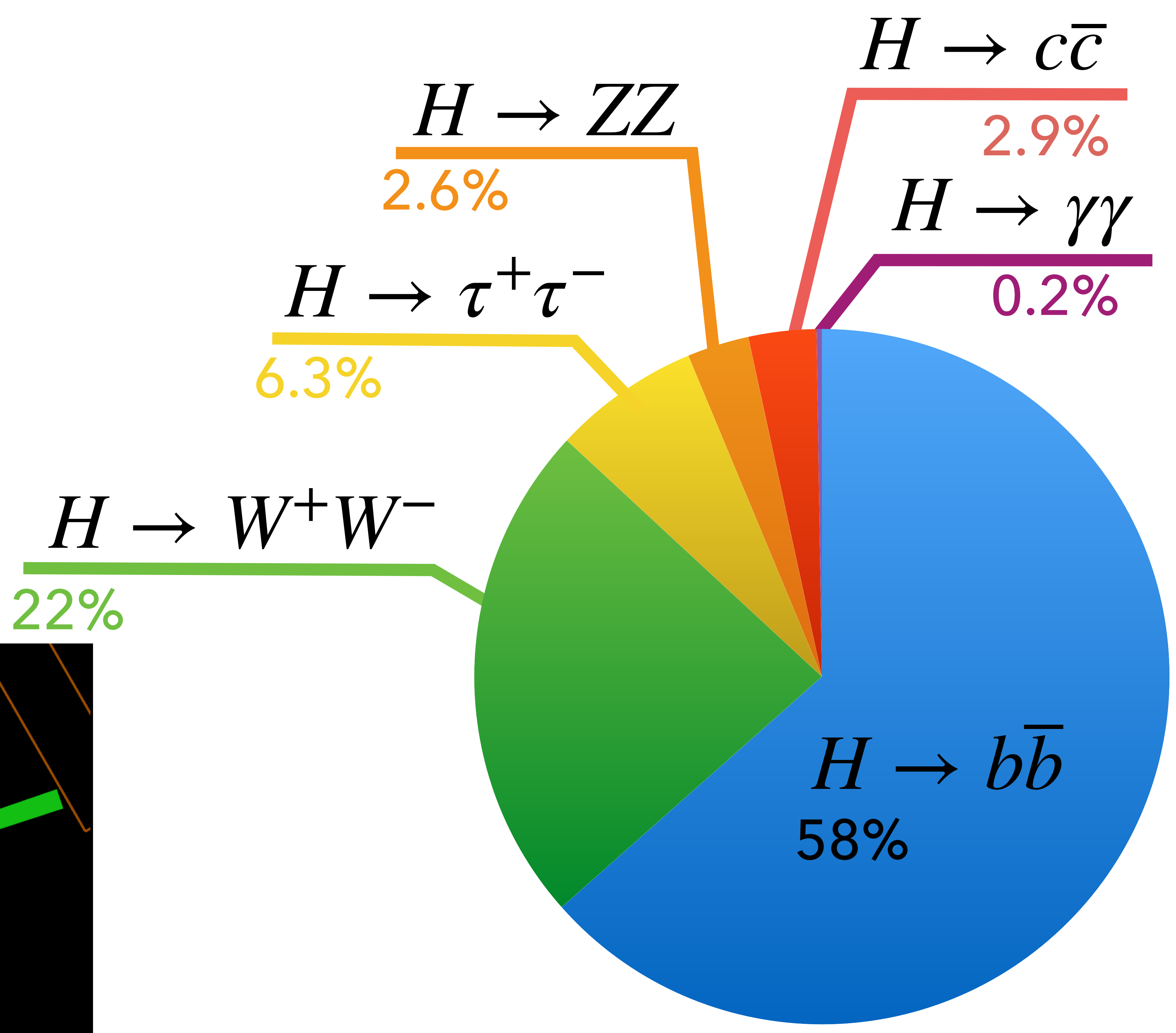
UCSD

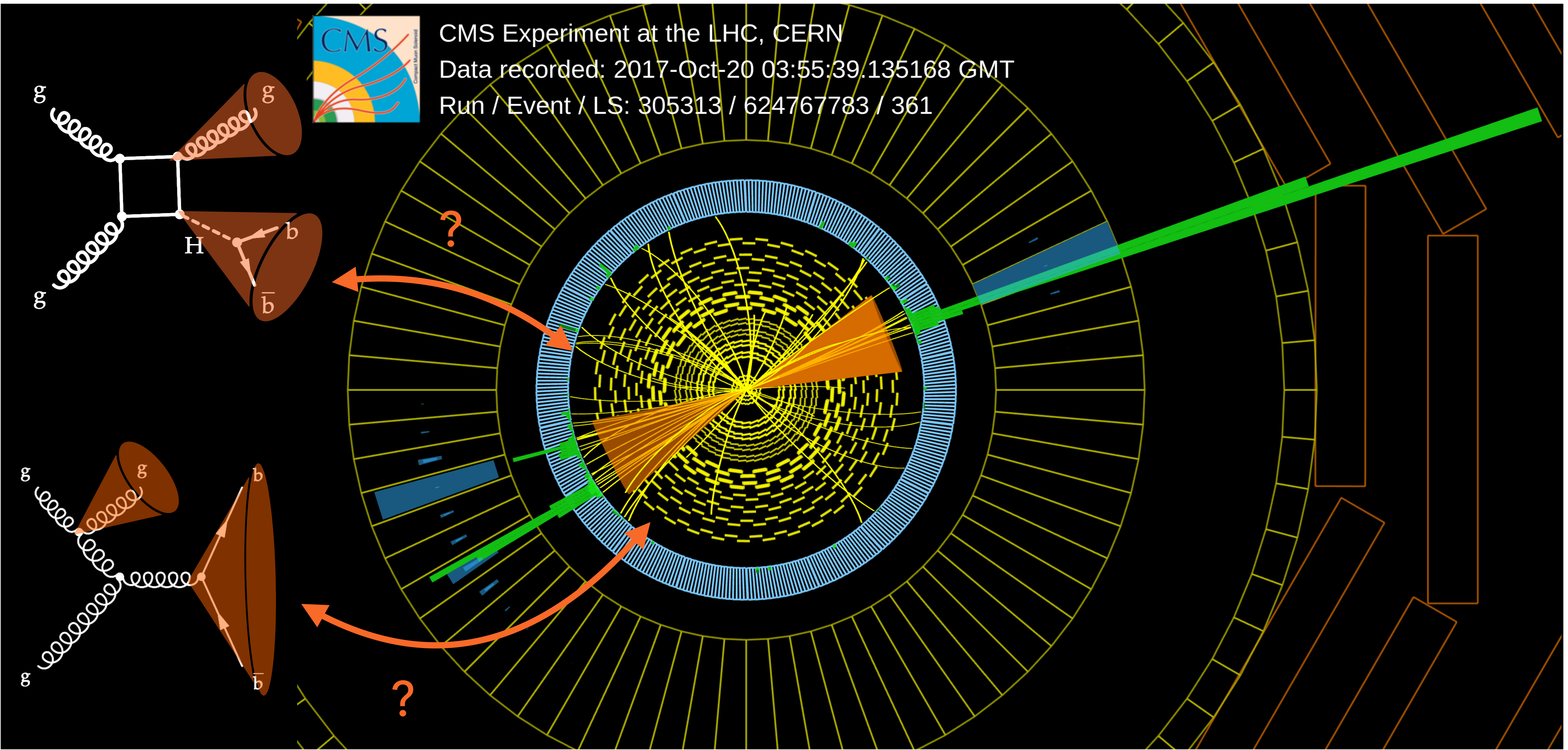
EXAMPLE DATASETS AND AI MODELS

- Advance important tasks in HEP with reference datasets and AI models to explore FAIRness criteria for both
  - **H(bb) jet tagging**
  - **Particle-flow reconstruction**
  - ECAL crystal calibration
  - Level-1 trigger jet reconstruction
  - Charged particle tracking
  - Among others

TASK: TAGGING H(BB) JETS

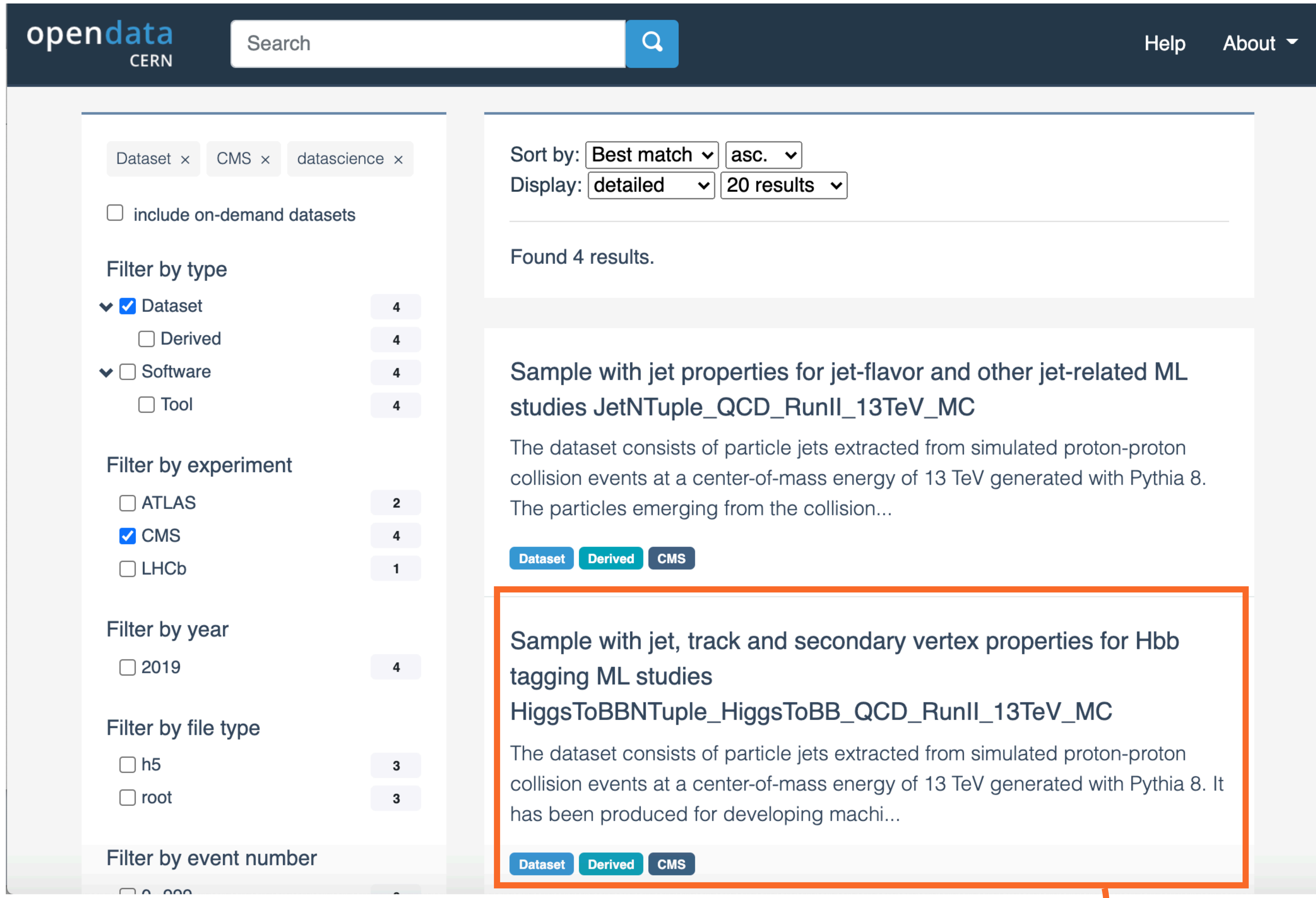
- H→bb is large, but more difficult to measure due to large backgrounds

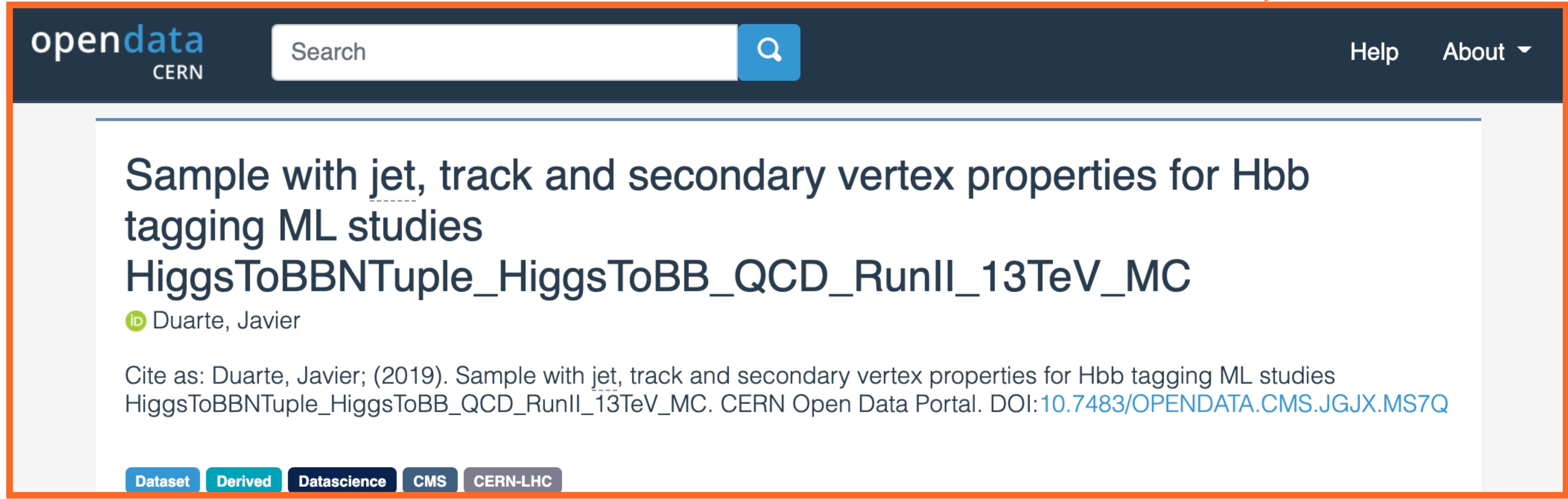




H(BB) JET TAGGING DATASET

- Hosted on CERN Open Data Portal
  - Collaborative effort between CERN IT-CDA and RCS-SIS groups, LHC and OPERA experiments
  - Built with Invenio library management software
  - Products (i.e. data, software, documentation, provenance) shared under open licenses and issued DOIs
  - EOS data storage; access via XRootD, HTTP
- H(bb) dataset [10.7483/OPENDATA.CMS.JGJX.MS7Q]
  - 182 files, 245 GB, 18 million total entries (jets)
    - event features, e.g. MET, ρ (average density)
    - jet features, e.g. mass, p_T, N-subjettiness variables
    - particle candidate features, e.g. p_T, η, ϕ
    - charged particle / track features, e.g. impact parameter
    - secondary vertex features, e.g. flight distance

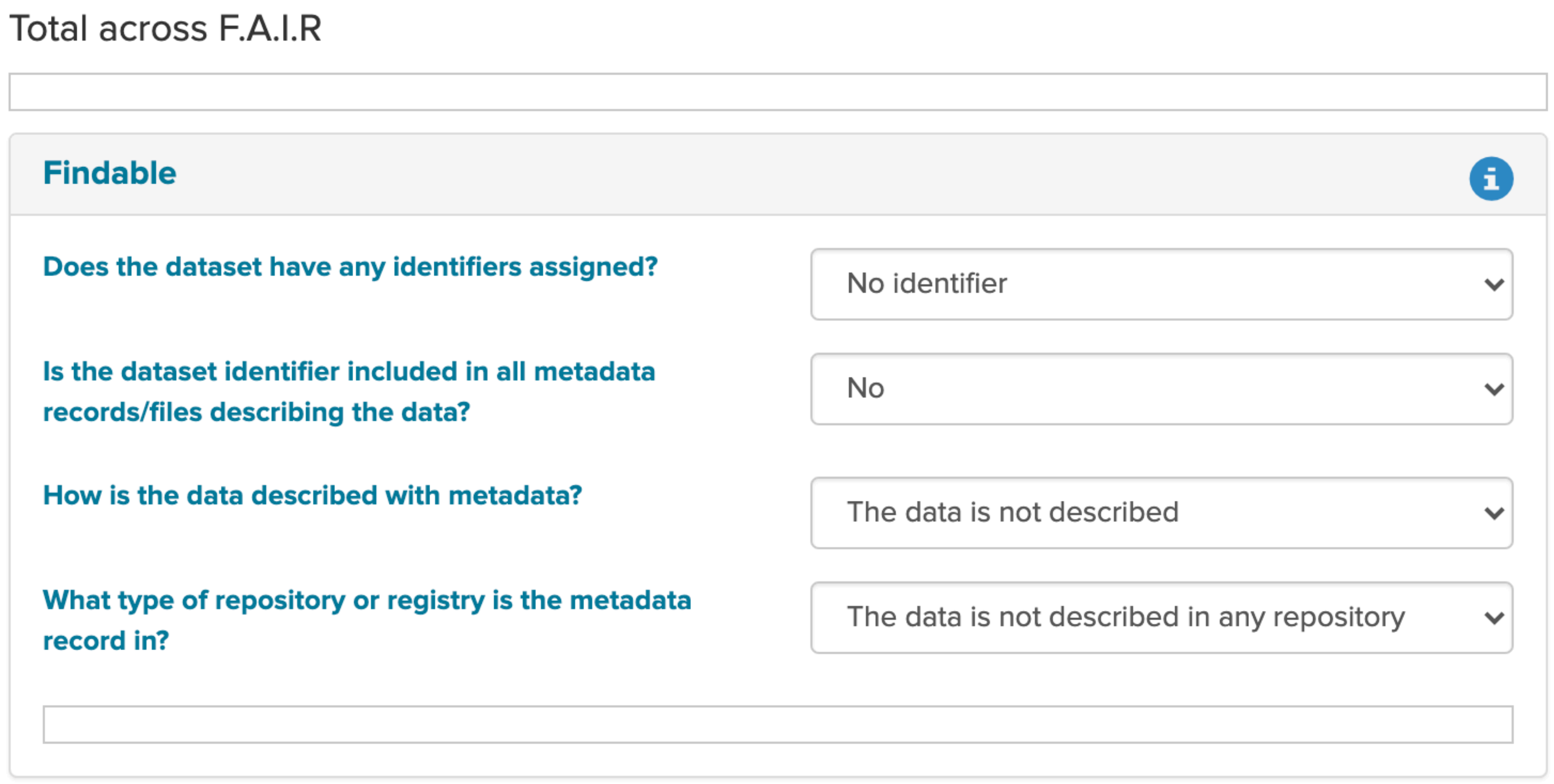




IS THIS H(BB) JET TAGGING DATASET FAIR?

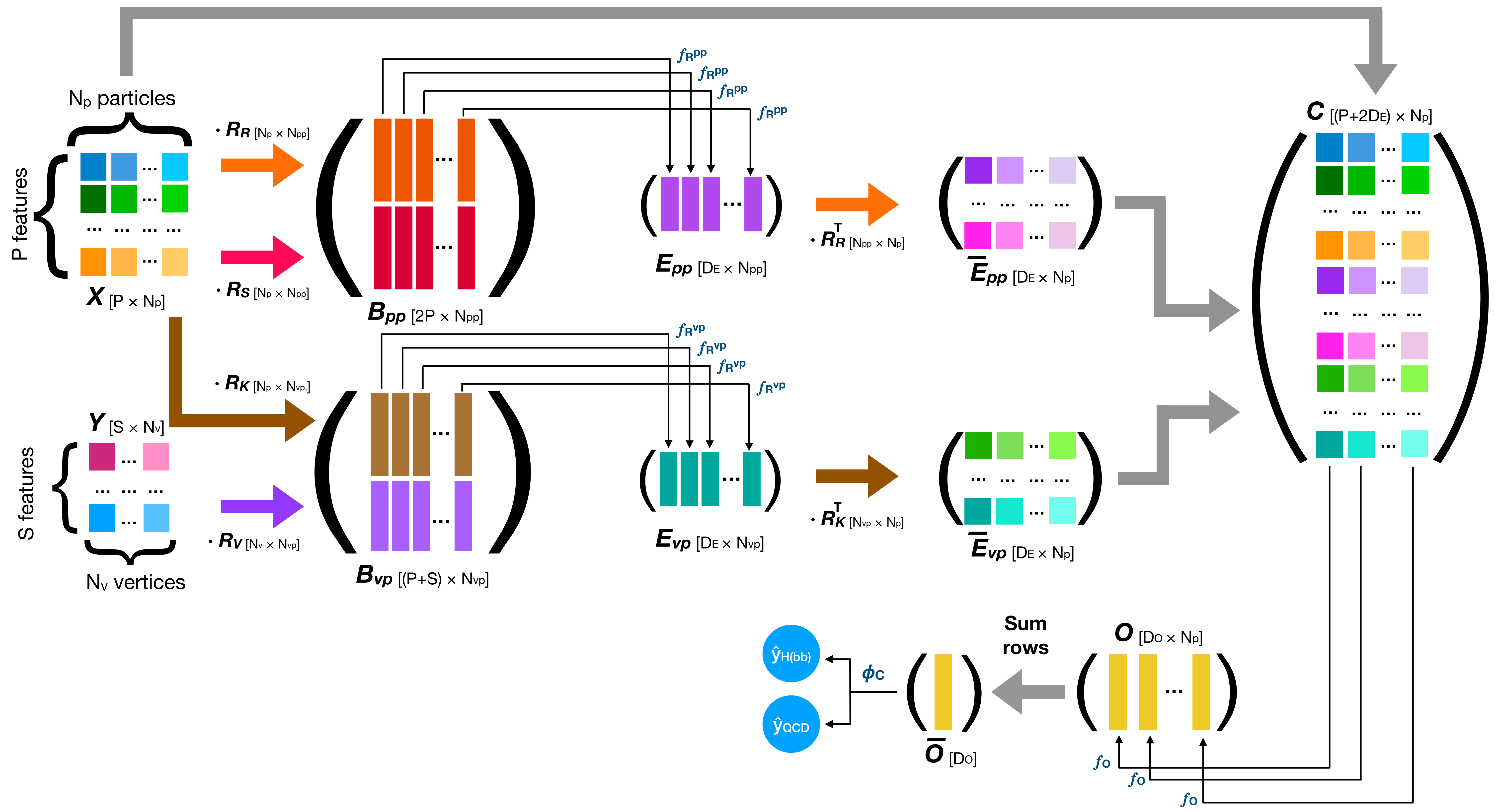
- In the process of evaluating FAIRness and contributing feedback to CERN Open Data Portal
  - Similar to ARDC FAIRness self assessment tool: https://ardc.edu.au/resources/working-with-data/fair-data/fair-self-assessment-tool/
- Take lessons learned and initiate a guide on evaluating FAIRness for the HEP community

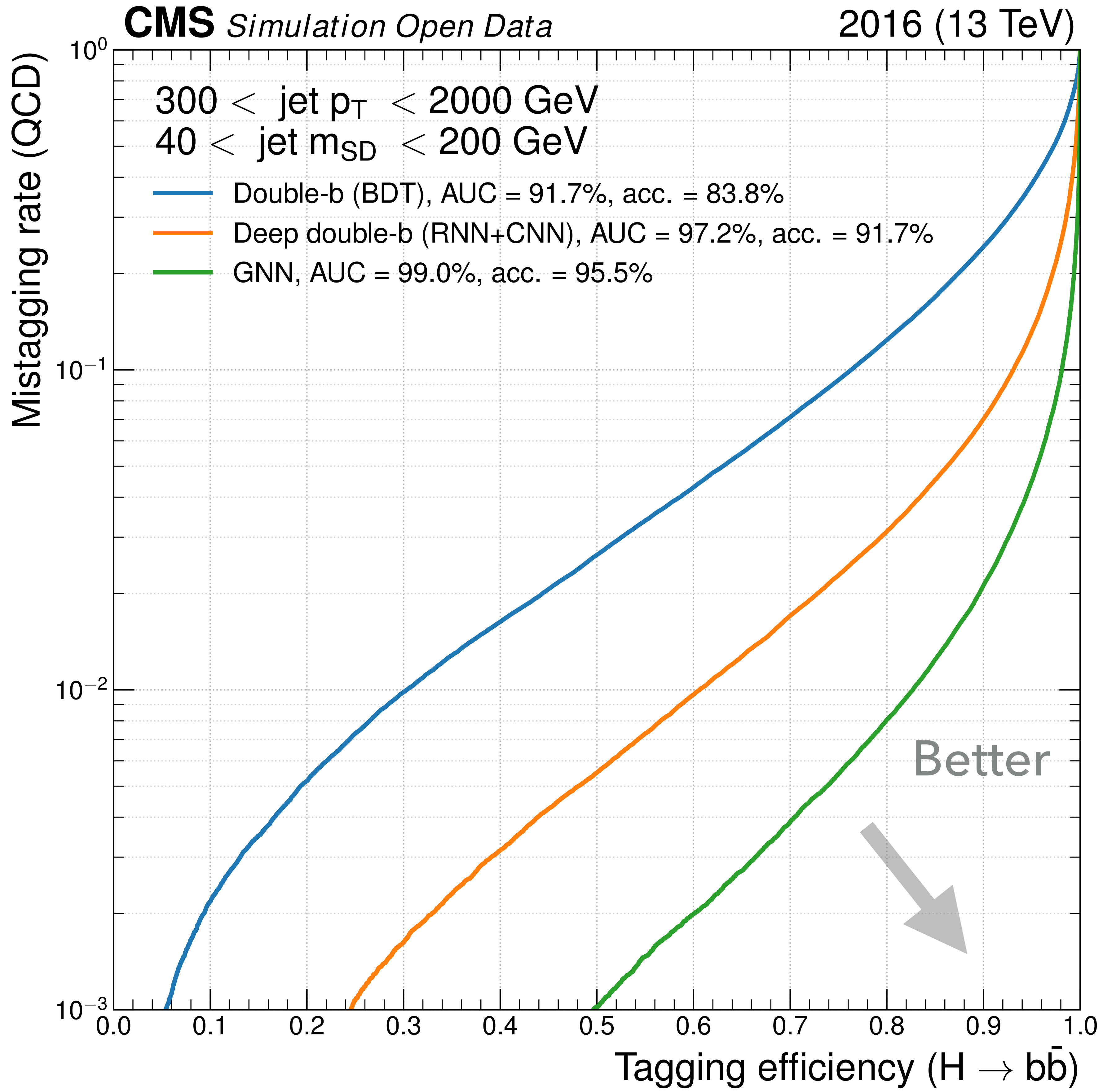
FAIR Principle	F1: (Meta)data are assigned globally unique and persistent identifiers.
METRIC	*Pass/Fail with Comments*
Identifier Uniqueness : this metric measures whether there is a scheme to uniquely identify the digital resource.	**Pass**. There is a unique URL to a registered identifier scheme. The DOI: 10.7483/OPENDATA.CMS.JGJX.MS7Q of the data that resolves to this URL is also available.
Identifier Persistence: it measures whether there is a policy that describes what the provider will do in the event an identifier scheme becomes deprecated.	**Pass**. What the provider will do in the event an identifier scheme becomes deprecated? DOI provide persistent interoperable identifier.
FAIR Principle	F2: Data are described with rich metadata
METRIC	*Pass/Fail with Comments*
Machine-readability of Metadata: a URL to a document containing machine-readable metadata for the digital resource must be provided.	**Pass.** URL for JSON format metadata with REST API: http://opendata.cern.ch/api/records/12102. Also, running the url through https://search.google.com/test/rich-results shows the data page is eligible for rich results and the fields of metadata are machine readable.
Richness of Metadata	**(Newly added) Partially Pass.** The metadata can be improved with richer fields. Reviewing the datacite metadata for the DOI shows a pretty sparse record.
FAIR Principle	F3: Metadata clearly and explicitly include the identifier of the data they describe.
METRIC	*Pass/Fail with Comments*
Resource Identifier in Metadata: it measures whether the metadata document contains the identifier for the digital resource that meets F1 principle.	**Pass** The association between a metadata and the dataset is made explicit that the dataset's globally unique and persistent identifier can be found in the metadata. Specifically, the DOI is a top-level and a mandatory field in the metadata record.
FAIR Principle	F4: (Meta)data are registered or indexed in a searchable resource
METRIC	*Pass/Fail with Comments*
Index in a searchable resource: it measures the degree to which the digital resource can be found using web-based search engines	**Pass.** The dataset is indexed by Google Dataset Search engine.



H(BB) BENCHMARK AI MODEL AND PERFORMANCE

- Edge convolutions for particle-particle and particle-vertex connections update particle features; summed particle features used to predict H(bb) or QCD prob.
- GNN improves on previous methods
- Model and code: https://github.com/eric-moreno/IN [10.5281/zenodo.3891869]

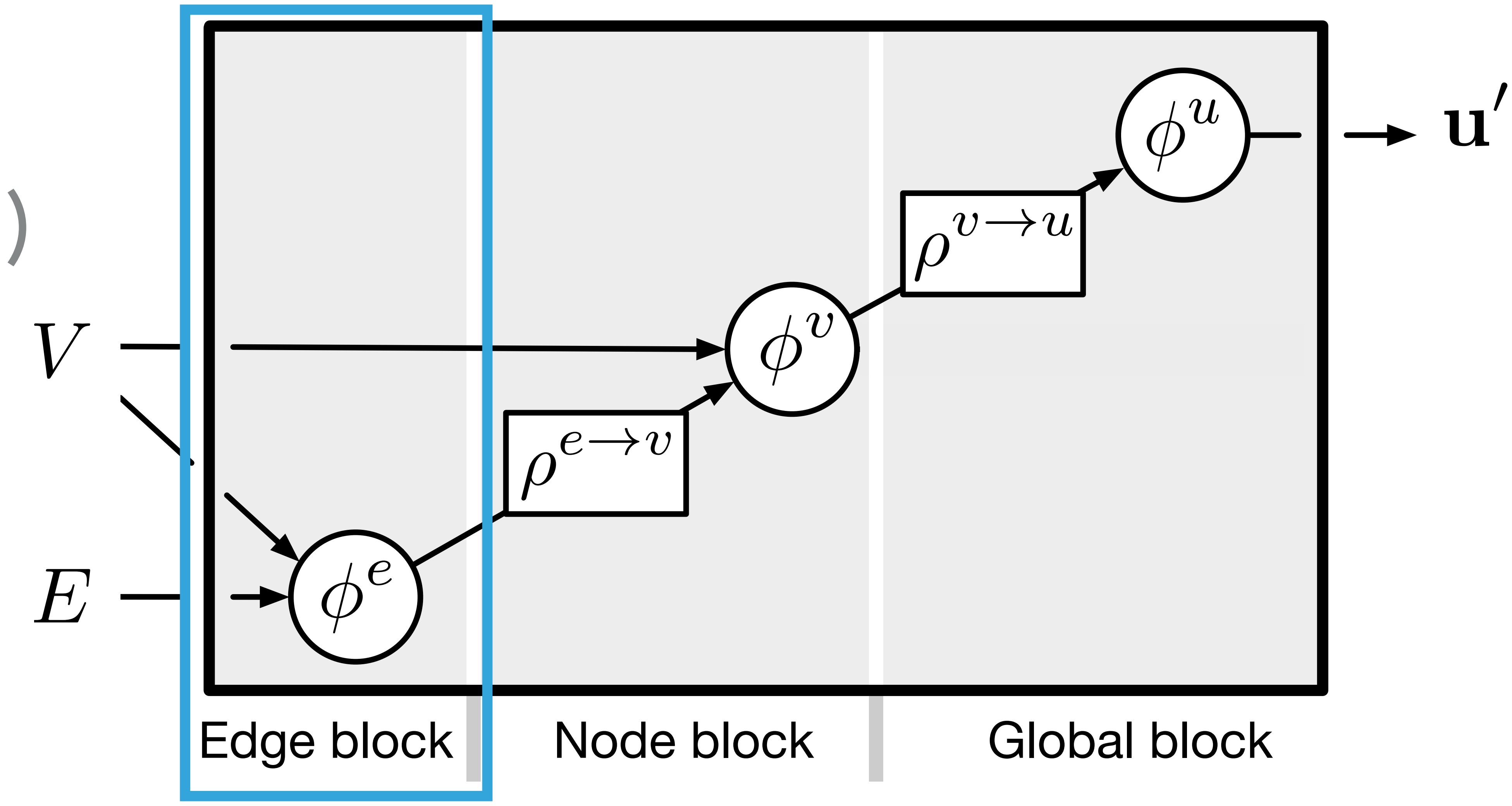




GNN TUTORIAL (UCSD DATA SCIENCE CAPSTONE)

- GNN tutorial with PyTorch Geometric: UCSD Data Science Capstone
- Environment specified with docker and conda
- CI deployed in GitHub Actions
- Expand this example into a FAIR AI model (via e.g. DLHub)

```
class EdgeBlock(torch.nn.Module):
    def __init__(self):
        super(EdgeBlock, self).__init__()
        self.edge_mlp = Seq(Lin(inputs*2, hidden),
                            BatchNorm1d(hidden),
                            ReLU(),
                            Lin(hidden, hidden))

    def forward(self, src, dest, edge_attr, u, batch):
        out = torch.cat([src, dest], 1)
        return self.edge_mlp(out)
```

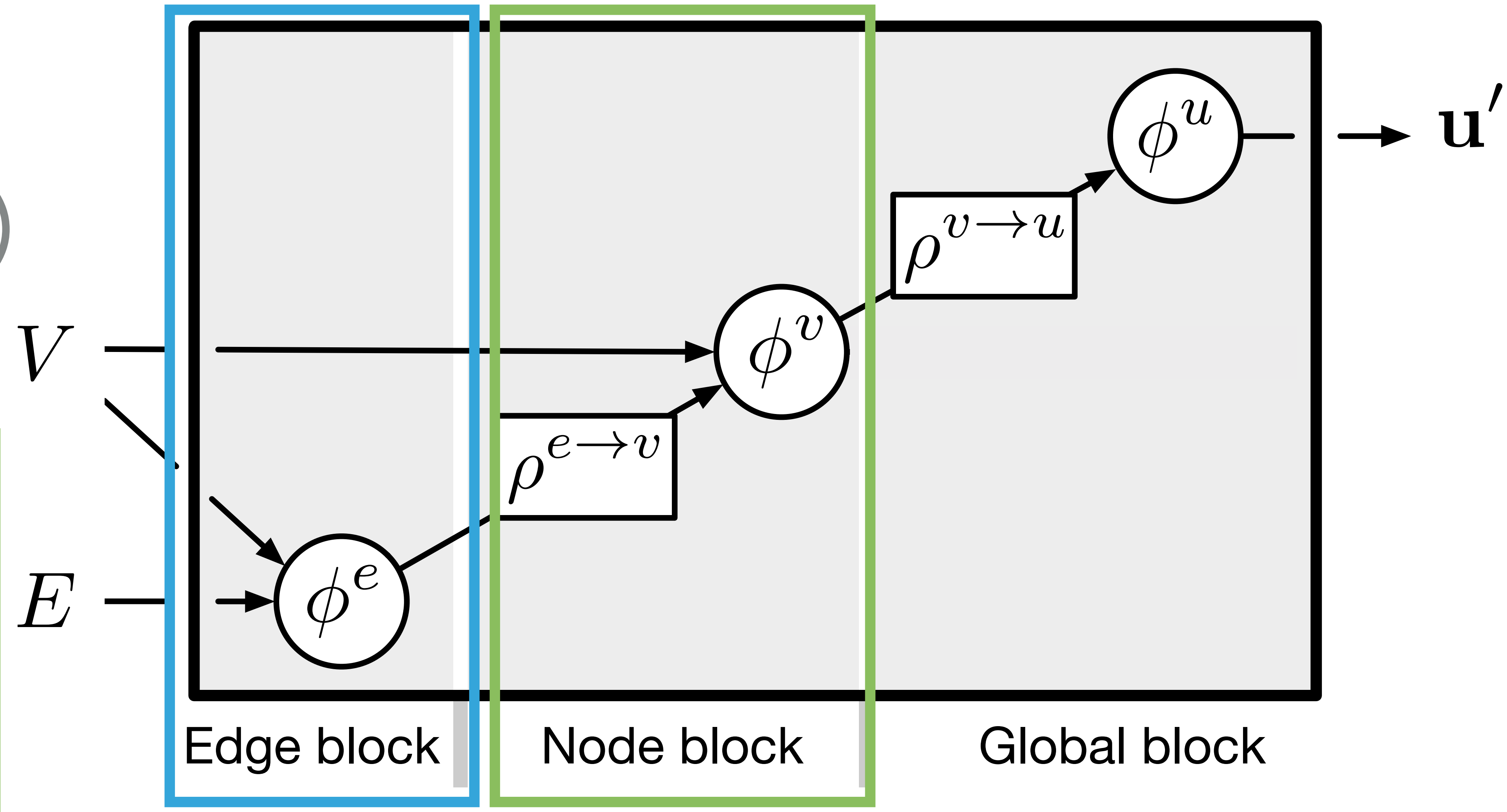


GNN TUTORIAL (UCSD DATA SCIENCE CAPSTONE)

- GNN tutorial with PyTorch Geometric: [UCSD Data Science Capstone]
- Environment specified with docker and conda
- CI deployed in GitHub Actions
- Expand this example into a FAIR AI model (via e.g. DLHub)

```python
class EdgeBlock(torch.nn.Module):
    def __init__(self):
        super(EdgeBlock, self).__init__()
        self.edge_mlp = Seq(Lin(inputs*2, hidden),
                            BatchNorm1d(hidden),
                            ReLU(),
                            Lin(hidden, hidden))

    def forward(self, src, dest, edge_attr, u, batch):
        out = torch.cat([src, dest], 1)
        return self.edge_mlp(out)
```

```python
class NodeBlock(torch.nn.Module):
    def __init__(self):
        super(NodeBlock, self).__init__()
        self.node_mlp_1 = Seq(Lin(inputs+hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, hidden))
        self.node_mlp_2 = Seq(Lin(inputs+hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, hidden))

    def forward(self, x, edge_index, edge_attr, u, batch):
        row, col = edge_index
        out = torch.cat([x[row], edge_attr], dim=1)
        out = self.node_mlp_1(out)
        out = scatter_mean(out, col, dim=0, dim_size=x.size(0))
        out = torch.cat([x, out], dim=1)
        return self.node_mlp_2(out)
```



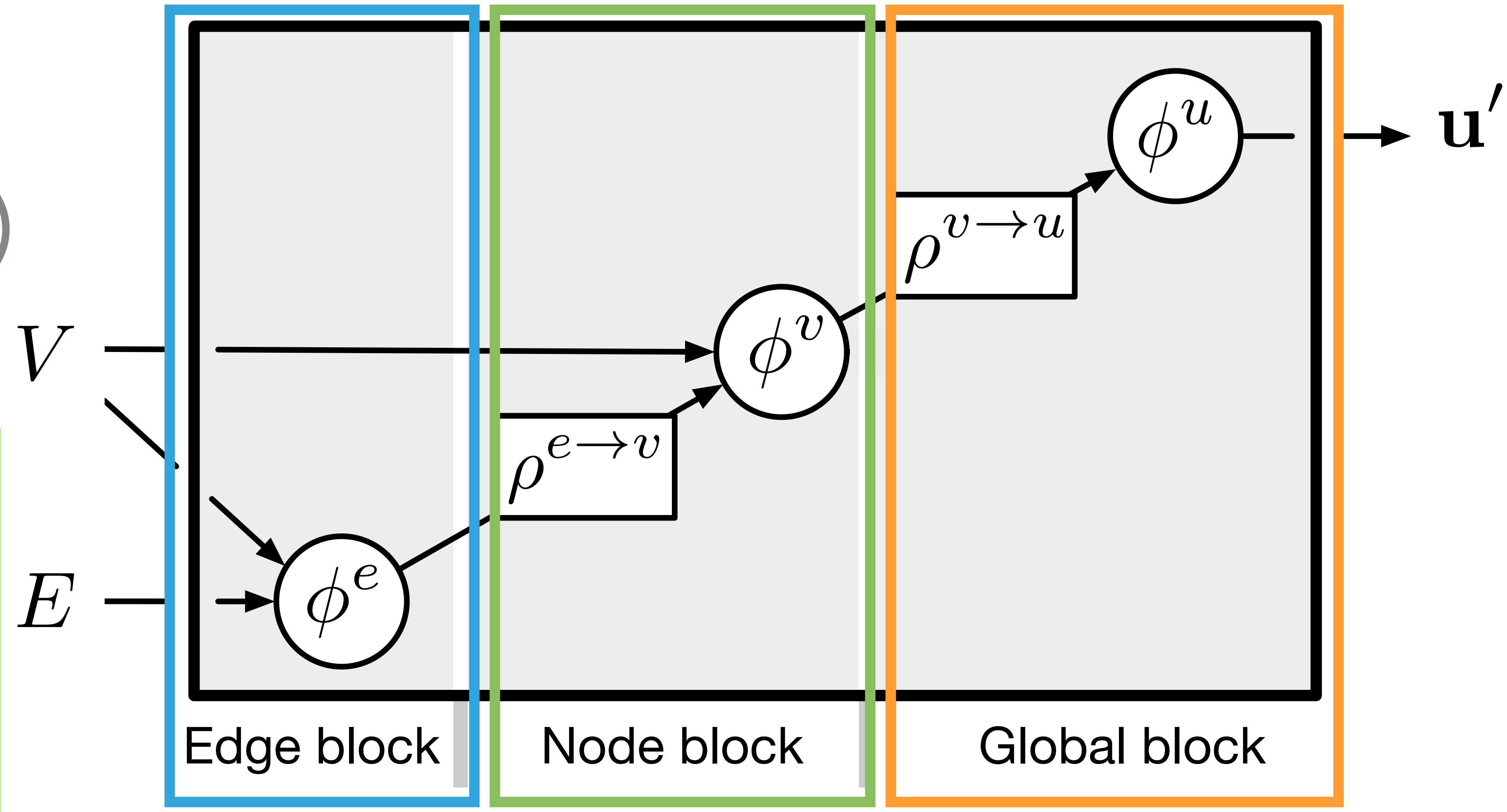
GNN TUTORIAL (UCSD DATA SCIENCE CAPSTONE)

- GNN tutorial with PyTorch Geometric: UCSD Data Science Capstone
- Environment specified with docker and conda
- CI deployed in GitHub Actions
- Expand this example into a FAIR AI model (via e.g. DLHub)

```
class EdgeBlock(torch.nn.Module):
    def __init__(self):
        super(EdgeBlock, self).__init__()
        self.edge_mlp = Seq(Lin(inputs*2, hidden),
                            BatchNorm1d(hidden),
                            ReLU(),
                            Lin(hidden, hidden))

    def forward(self, src, dest, edge_attr, u, batch):
        out = torch.cat([src, dest], 1)
        return self.edge_mlp(out)
```

```
class GlobalBlock(torch.nn.Module):
    def __init__(self):
        super(GlobalBlock, self).__init__()
        self.global_mlp = Seq(Lin(hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, outputs))

    def forward(self, x, edge_index, edge_attr, u, batch):
        out = scatter_mean(x, batch, dim=0)
        return self.global_mlp(out)
```

```
class NodeBlock(torch.nn.Module):
    def __init__(self):
        super(NodeBlock, self).__init__()
        self.node_mlp_1 = Seq(Lin(inputs+hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, hidden))
        self.node_mlp_2 = Seq(Lin(inputs+hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, hidden))

    def forward(self, x, edge_index, edge_attr, u, batch):
        row, col = edge_index
        out = torch.cat([x[row], edge_attr], dim=1)
        out = self.node_mlp_1(out)
        out = scatter_mean(out, col, dim=0, dim_size=x.size(0))
        out = torch.cat([x, out], dim=1)
        return self.node_mlp_2(out)
```



GNN TUTORIAL (UCSD DATA SCIENCE CAPSTONE)

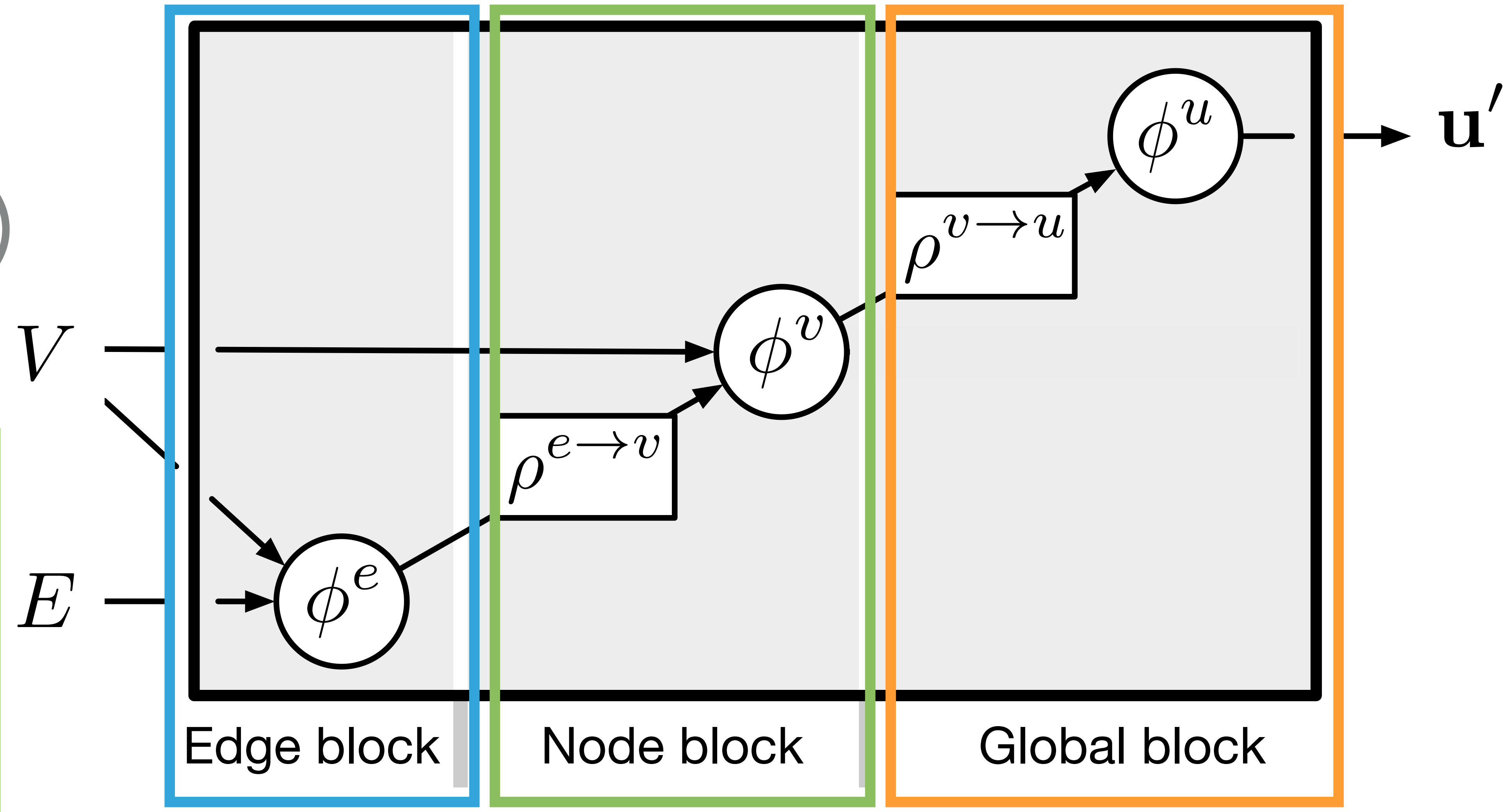
- GNN tutorial with PyTorch Geometric: UCSD Data Science Capstone
- Environment specified with docker and conda
- CI deployed in GitHub Actions
- Expand this example into a FAIR AI model (via e.g. DLHub)

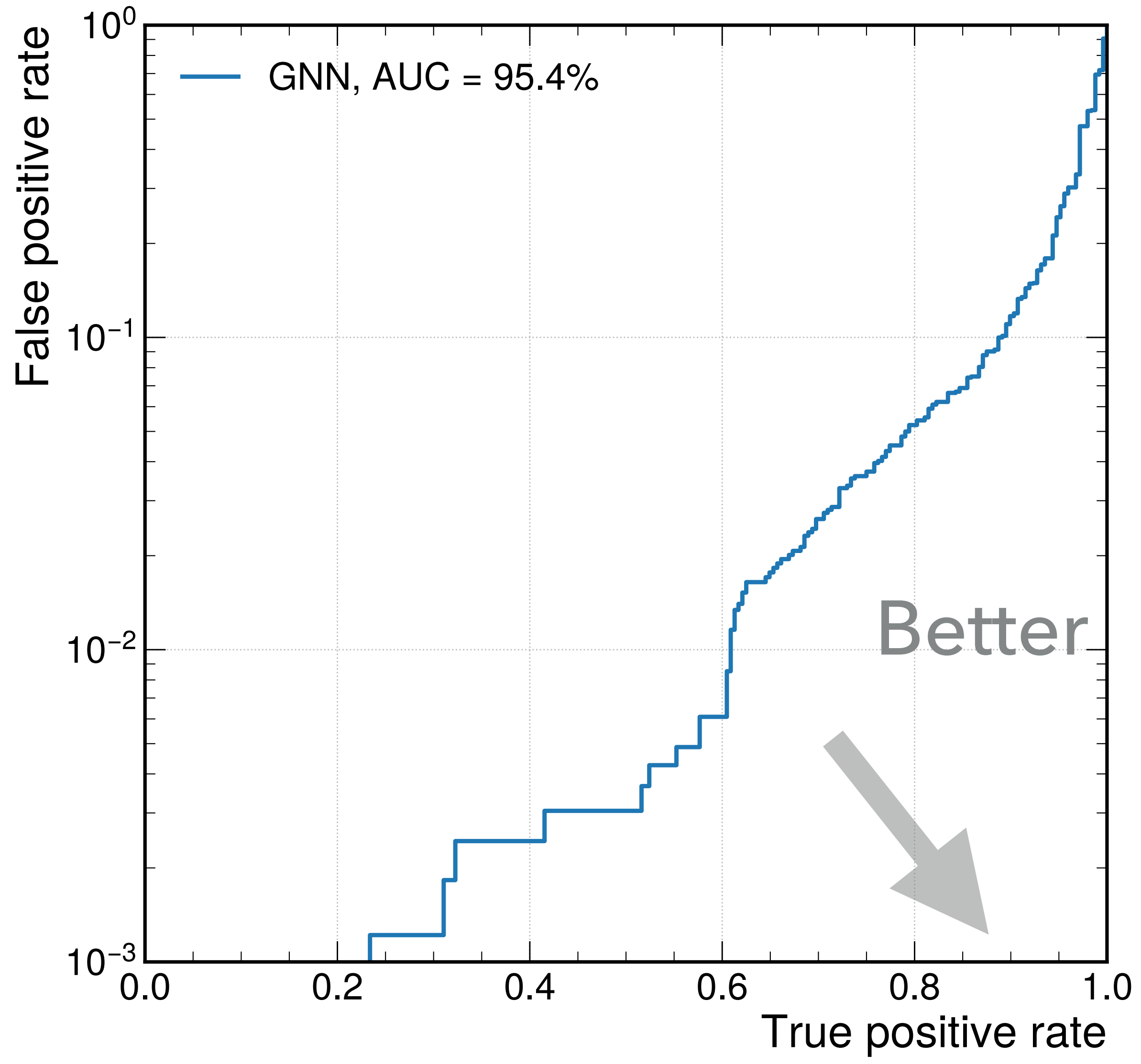
```
class EdgeBlock(torch.nn.Module):
    def __init__(self):
        super(EdgeBlock, self).__init__()
        self.edge_mlp = Seq(Lin(inputs*2, hidden),
                            BatchNorm1d(hidden),
                            ReLU(),
                            Lin(hidden, hidden))

    def forward(self, src, dest, edge_attr, u, batch):
        out = torch.cat([src, dest], 1)
        return self.edge_mlp(out)
```

```
class GlobalBlock(torch.nn.Module):
    def __init__(self):
        super(GlobalBlock, self).__init__()
        self.global_mlp = Seq(Lin(hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, outputs))

    def forward(self, x, edge_index, edge_attr, u, batch):
        out = scatter_mean(x, batch, dim=0)
        return self.global_mlp(out)
```

```
class NodeBlock(torch.nn.Module):
    def __init__(self):
        super(NodeBlock, self).__init__()
        self.node_mlp_1 = Seq(Lin(inputs+hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, hidden))
        self.node_mlp_2 = Seq(Lin(inputs+hidden, hidden),
                              BatchNorm1d(hidden),
                              ReLU(),
                              Lin(hidden, hidden))

    def forward(self, x, edge_index, edge_attr, u, batch):
        row, col = edge_index
        out = torch.cat([x[row], edge_attr], dim=1)
        out = self.node_mlp_1(out)
        out = scatter_mean(out, col, dim=0, dim_size=x.size(0))
        out = torch.cat([x, out], dim=1)
        return self.node_mlp_2(out)
```

```
class InteractionNetwork(torch.nn.Module):
    def __init__(self):
        super(InteractionNetwork, self).__init__()
        self.interactionnetwork = MetaLayer(EdgeBlock(), NodeBlock(), GlobalBlock())
        self.bn = BatchNorm1d(inputs)

    def forward(self, x, edge_index, batch):
        x = self.bn(x)
        x, edge_attr, u = self.interactionnetwork(x, edge_index, None, None, batch)
        return u
```

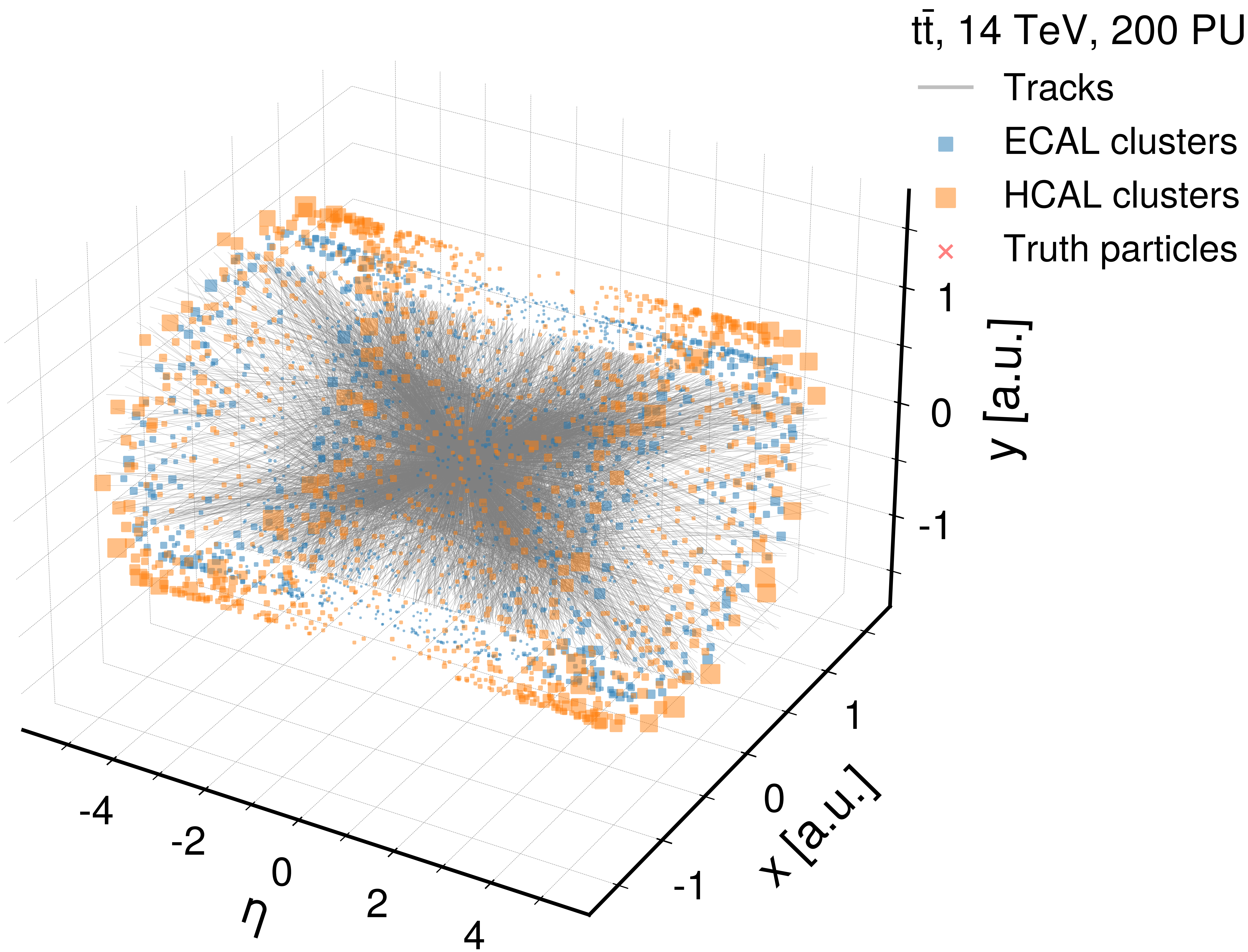




TASK: PARTICLE-FLOW RECONSTRUCTION

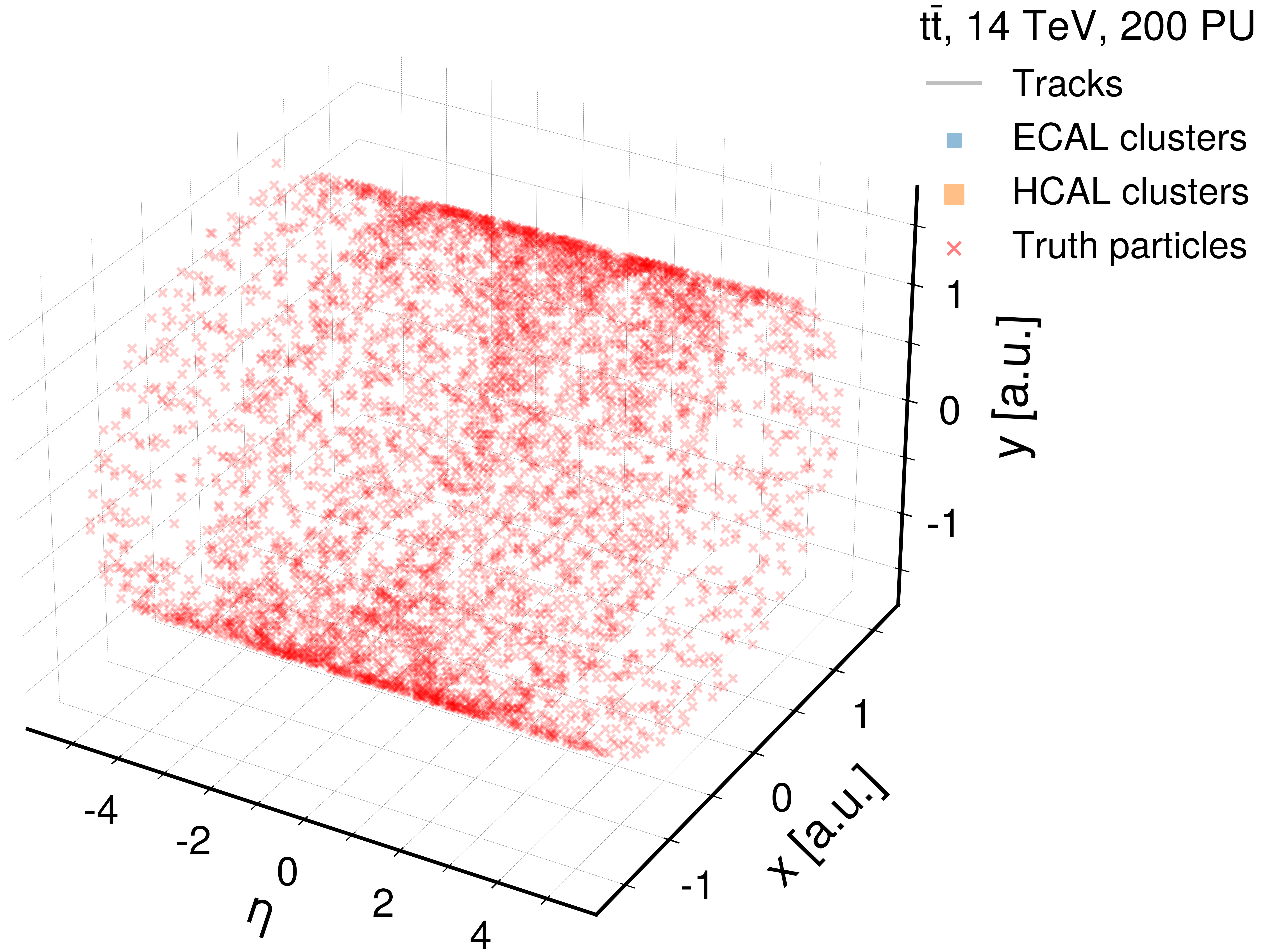
TASK: PARTICLE-FLOW RECONSTRUCTION

- Particles interact with detector, leaving energy deposits (ECAL clusters, HCAL clusters) and tracks



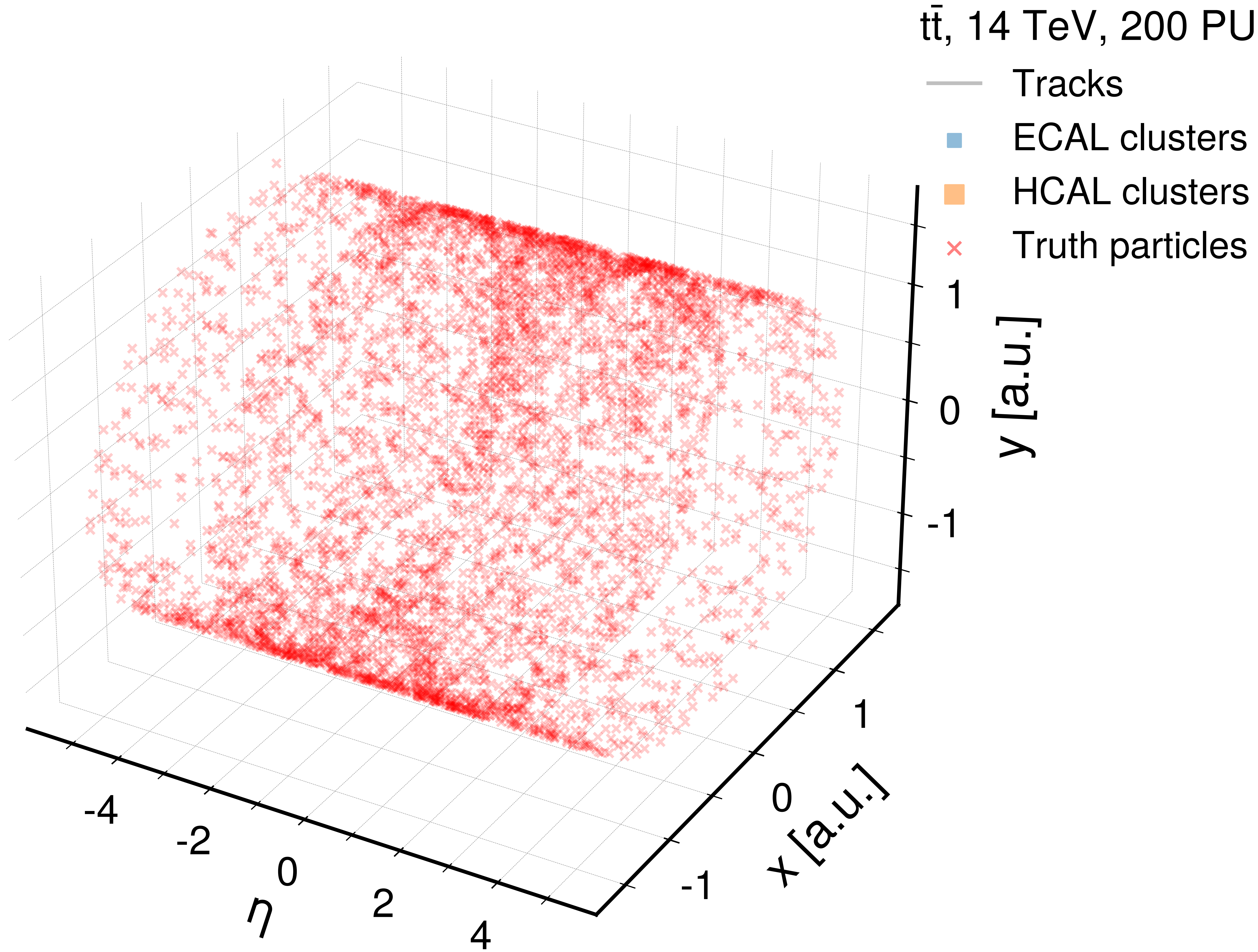
TASK: PARTICLE-FLOW RECONSTRUCTION

- Particles interact with detector, leaving energy deposits (ECAL clusters, HCAL clusters) and tracks
- PF: combine info from complementary detector subsystems to produce a holistic, particle interpretation of the event (truth particles)



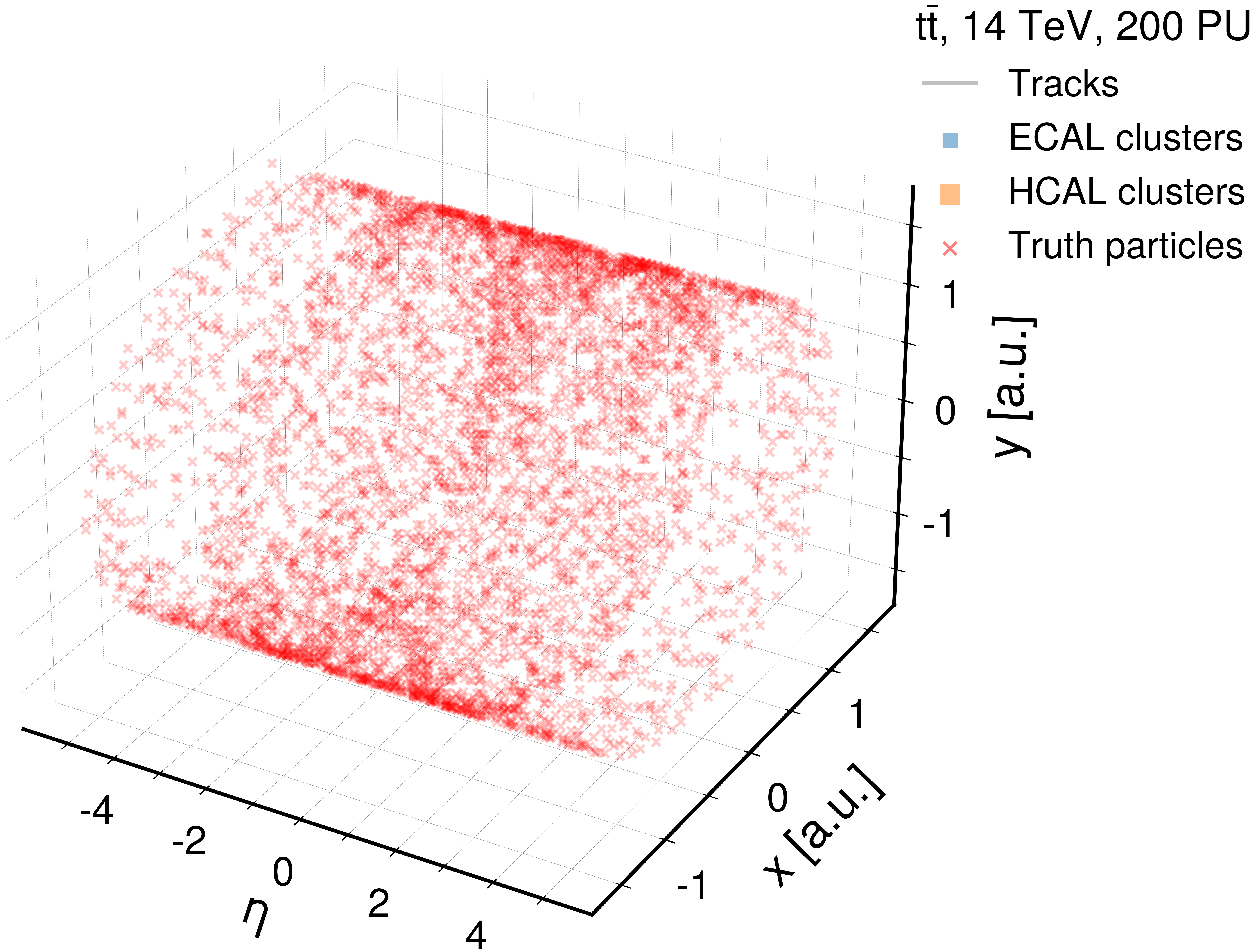
TASK: PARTICLE-FLOW RECONSTRUCTION

- Particles interact with detector, leaving energy deposits (ECAL clusters, HCAL clusters) and tracks
- PF: combine info from complementary detector subsystems to produce a holistic, particle interpretation of the event (truth particles)
- Goal: construct a mapping that minimizes some distance between truth particles and reconstructed particles



TASK: PARTICLE-FLOW RECONSTRUCTION

- Particles interact with detector, leaving energy deposits (ECAL clusters, HCAL clusters) and tracks
- PF: combine info from complementary detector subsystems to produce a holistic, particle interpretation of the event (truth particles)
- Goal: construct a mapping that minimizes some distance between truth particles and reconstructed particles
- Make public dataset: [10.5281/zenodo.4559324] and AI model [10.5281/zenodo.4559587] **FAIR**



VISION AND OUTLOOK

UCSD

Data repositories

VISION

Data repositories

opendata

zenodo

Deployable AI models

docker

GitHub

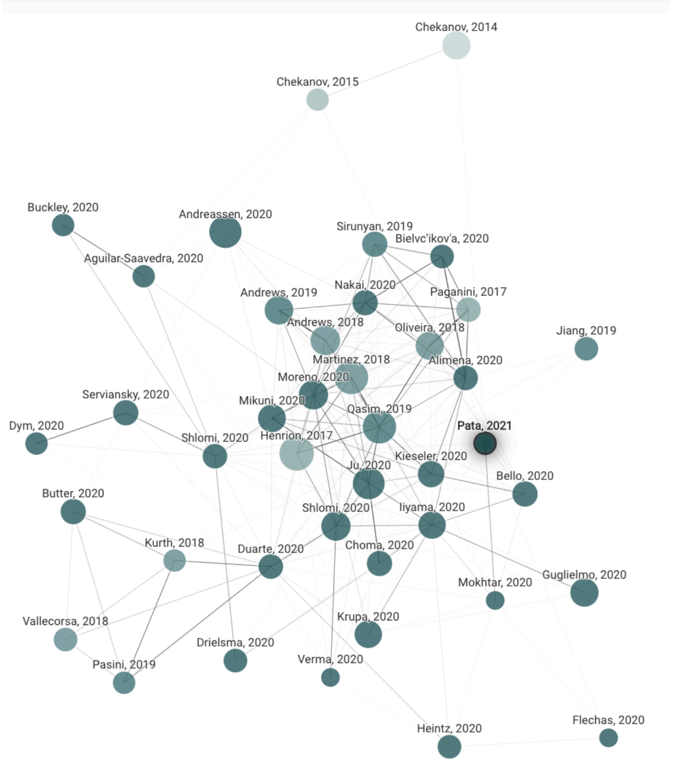
binder

VISION

Data repositories

Deployable AI models

Papers / indexing / search / discovery



VISION

Data repositories

opendata

zenodo

Deployable AI models

docker

GitHub

binder

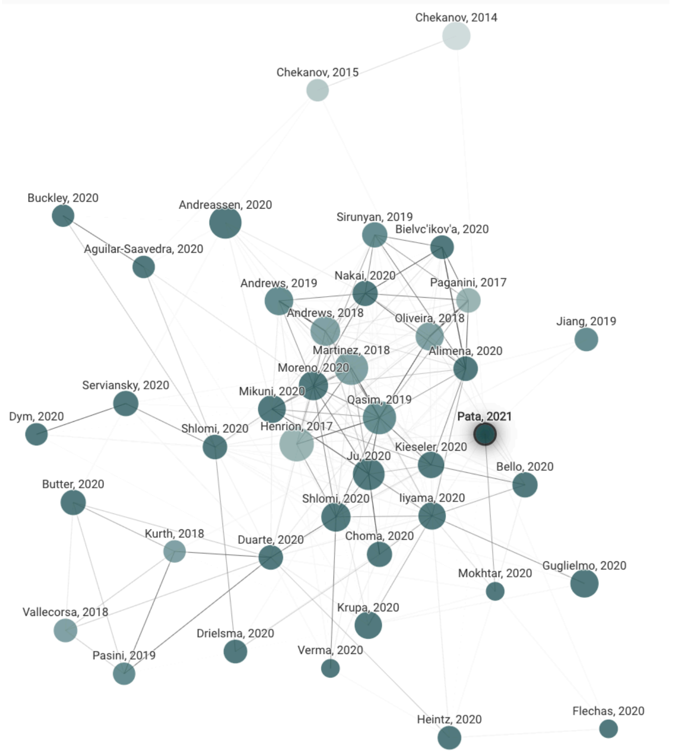
Papers / indexing / search / discovery

Cornell University

arXiv.org

INSPIRE HEP

CONNECTED PAPERS



Papers With Code

VISION

Data repositories

opendata

zenodo

Competitions?

CodaLab kaggle

Deployable AI models

docker GitHub

binder

Papers With Code

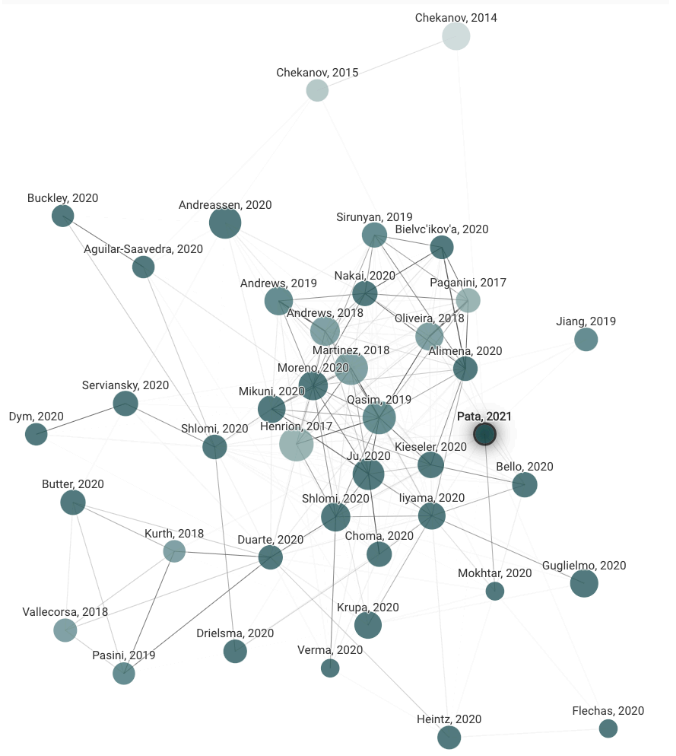
Papers / indexing / search / discovery

Cornell University

arXiv.org

INSPIRE HEP

CONNECTED PAPERS



VISION

Data repositories

opendata

zenodo

Competitions?

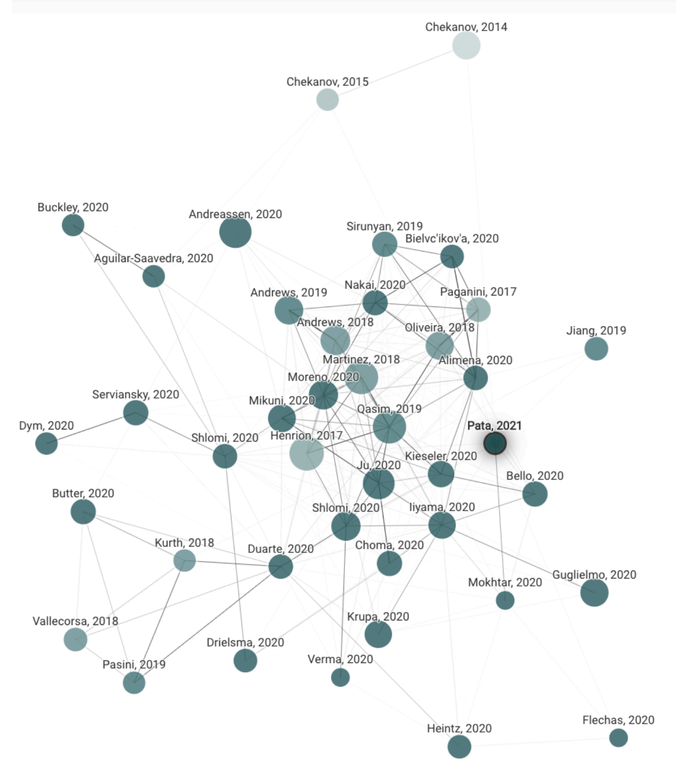
CodaLab kaggle

Papers / indexing / search / discovery

Cornell University arXiv.org

INSPIRE HEP

CONNECTED PAPERS



Deployable AI models

docker

GitHub

binder

Papers With Code

Automated ML (retraining) flows

reana

OpenML

mlflow

DLHub

OUTLOOK

- Goal of FAIR4HEP is to interpret and refine what FAIR means for HEP data/models
  - Enable "plug and play" datasets: allow for combinations of different computing resources
- Vision: connected services linking datasets, benchmark models (code), deployment servers, and publications to make everything more FAIR
  - Simpler discovery of new datasets and models
- Projects
  - Evaluate FAIRness of existing public datasets
  - Standardize FAIR publication of AI models in HEP
  - Create example FAIR datasets and AI models
  - Enhance existing services to make them more FAIR
- Welcome feedback!

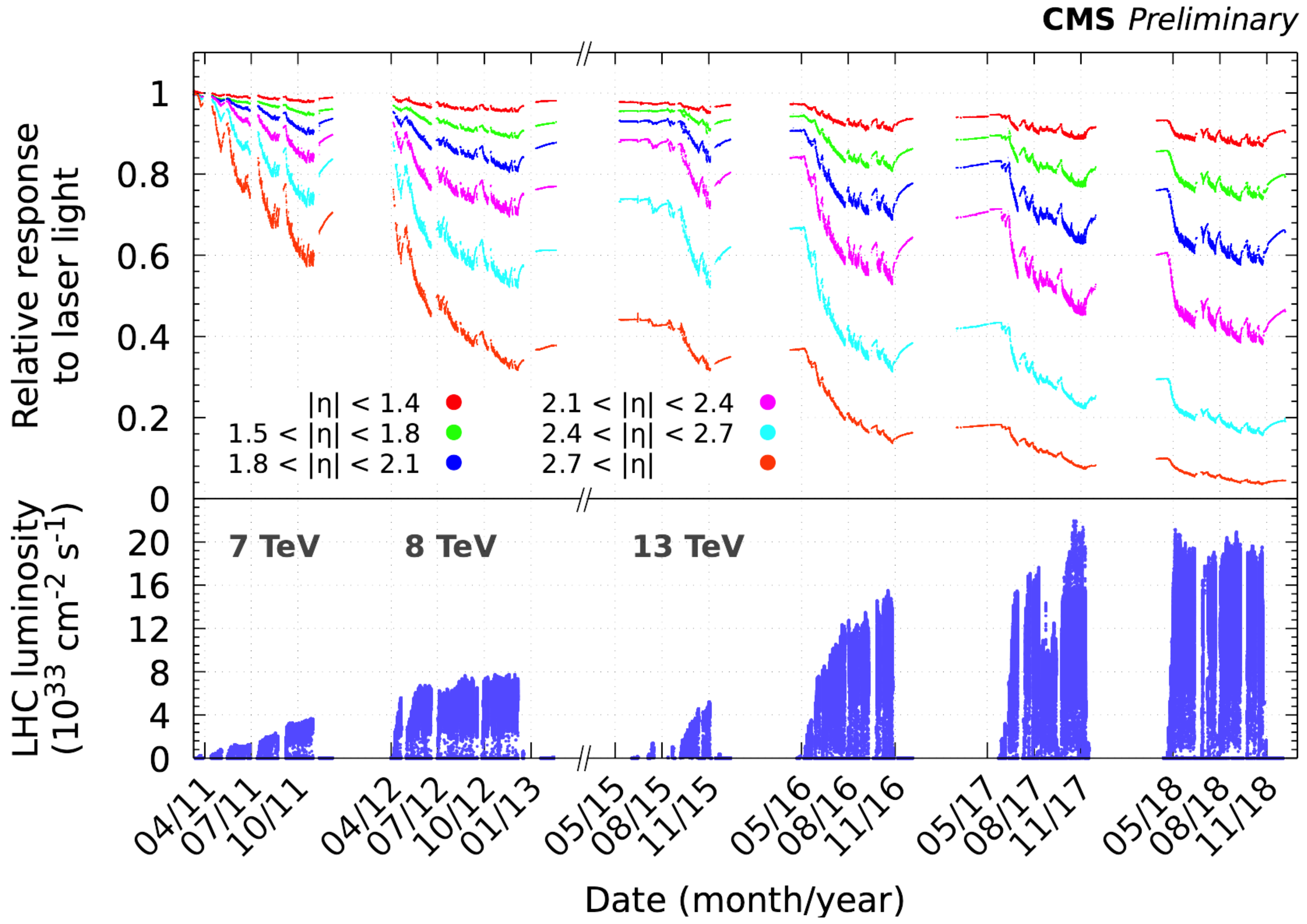
JAVIER DUARTE
IRIS-HEP TOPICAL MEETING
APRIL 21, 2021

BACKUP

UCSD

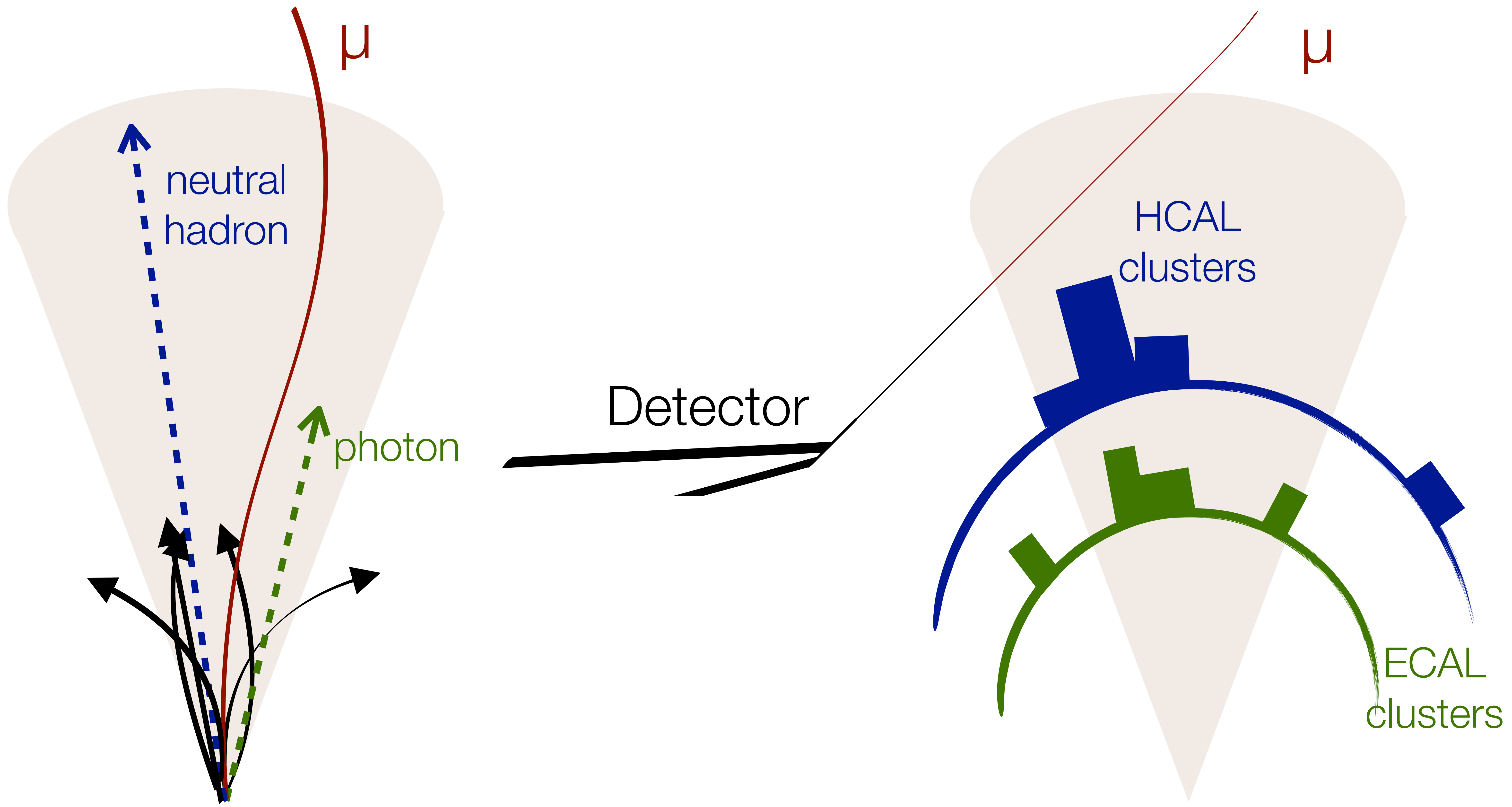
TASK: ECAL CRYSTAL CALIBRATION

- High radiation dose causes transparency loss in ECAL crystals
  - Crystals recover over time
  - ~70,000 crystals × ~10,000 calibrations per year = ~700,000,000 learnable parameters



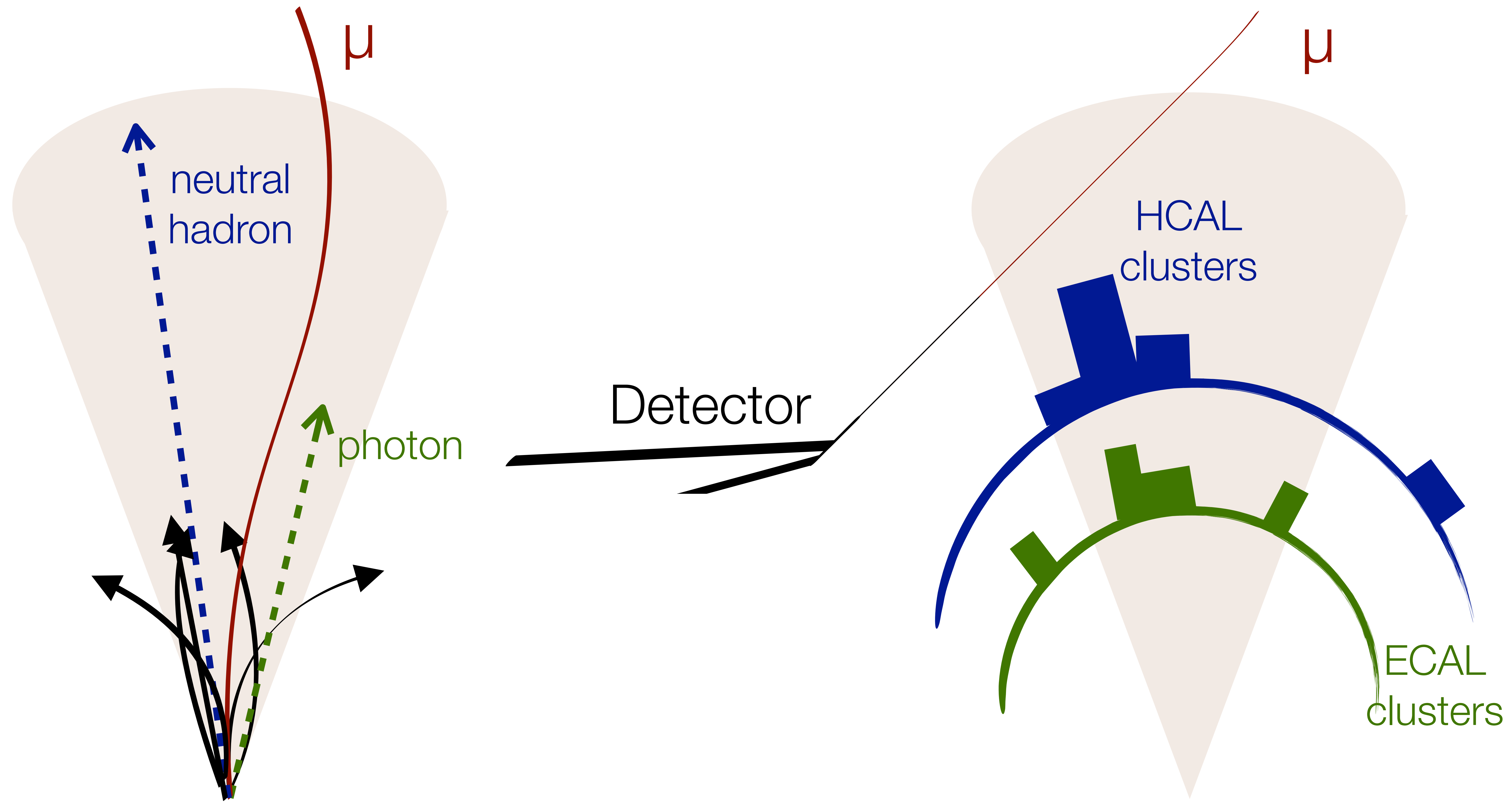
- Developing a public FAIR dataset (and AI model) to study (and predict) time dependence of transparency loss

TASK: PARTICLE-FLOW RECONSTRUCTION



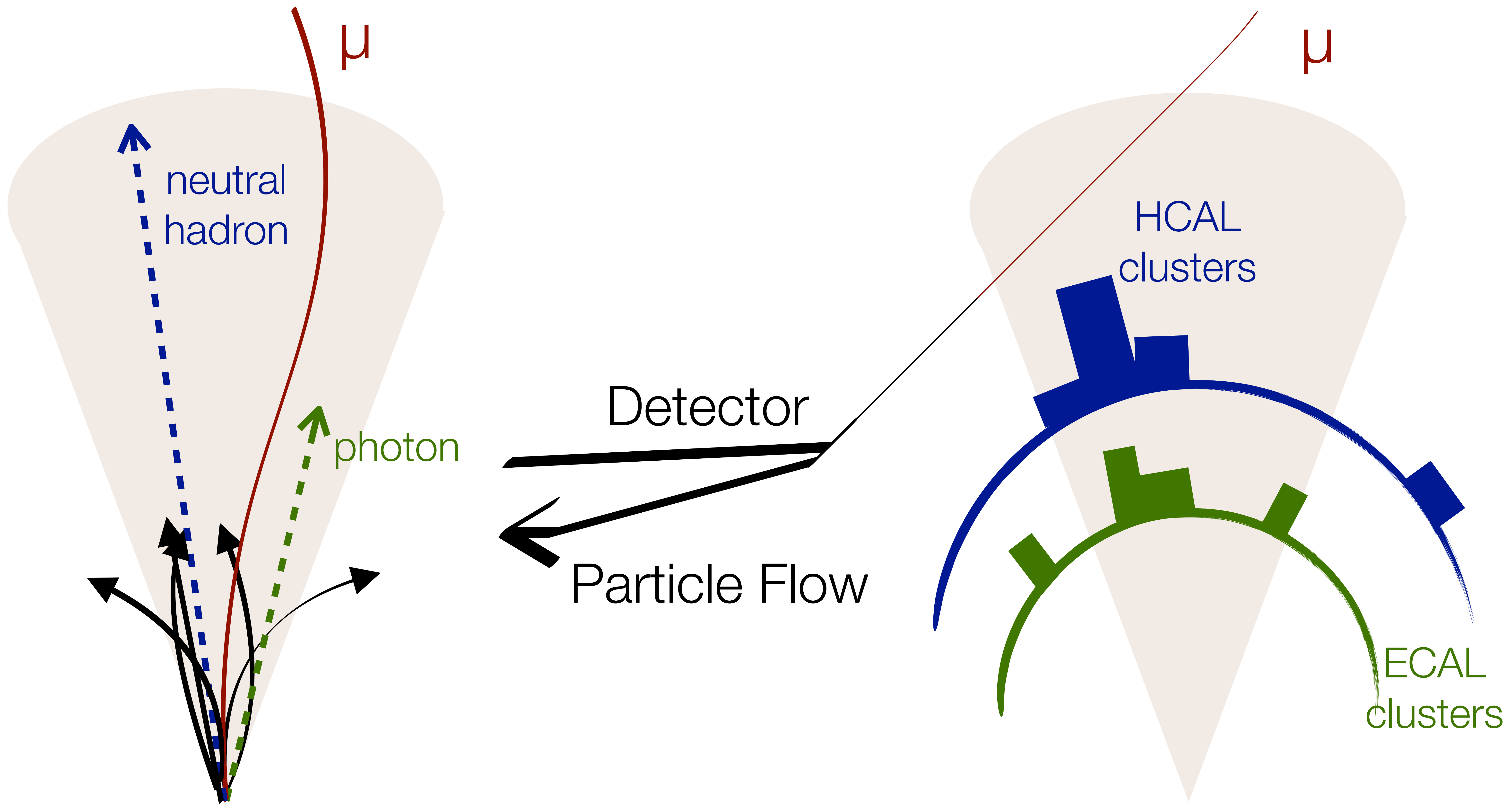
TASK: PARTICLE-FLOW RECONSTRUCTION

- Particles interact with detector, leaving energy deposits and tracks



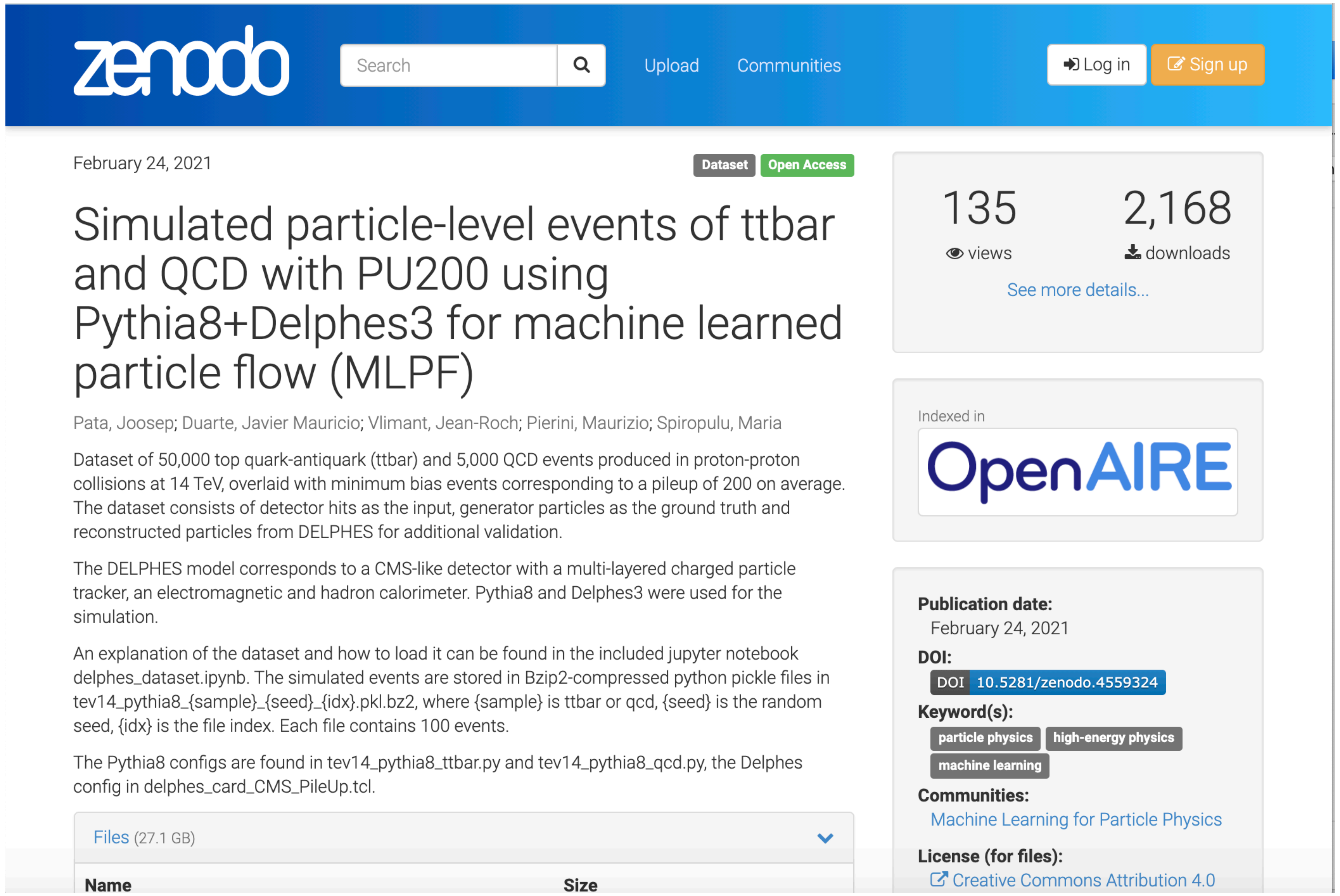
TASK: PARTICLE-FLOW RECONSTRUCTION

- Particles interact with detector, leaving energy deposits and tracks
- Efficient combination of info. from complementary detector subsystems to produce a holistic, particle interpretation of the event (that improves on any individual subsystem)



PARTICLE-FLOW DELPHES DATASET

- Pythia8+Delphes3 50,000 $t\bar{t}$+jets and 5,000 QCD events produced in pp collisions at 14 TeV with 200 pileup
- Partial provenance (Pythia+Delphes configuration, versions, etc.)
- Detector calorimeter towers and tracks as input and generator particles as ground truth
  - bzip2-compressed python pickle including

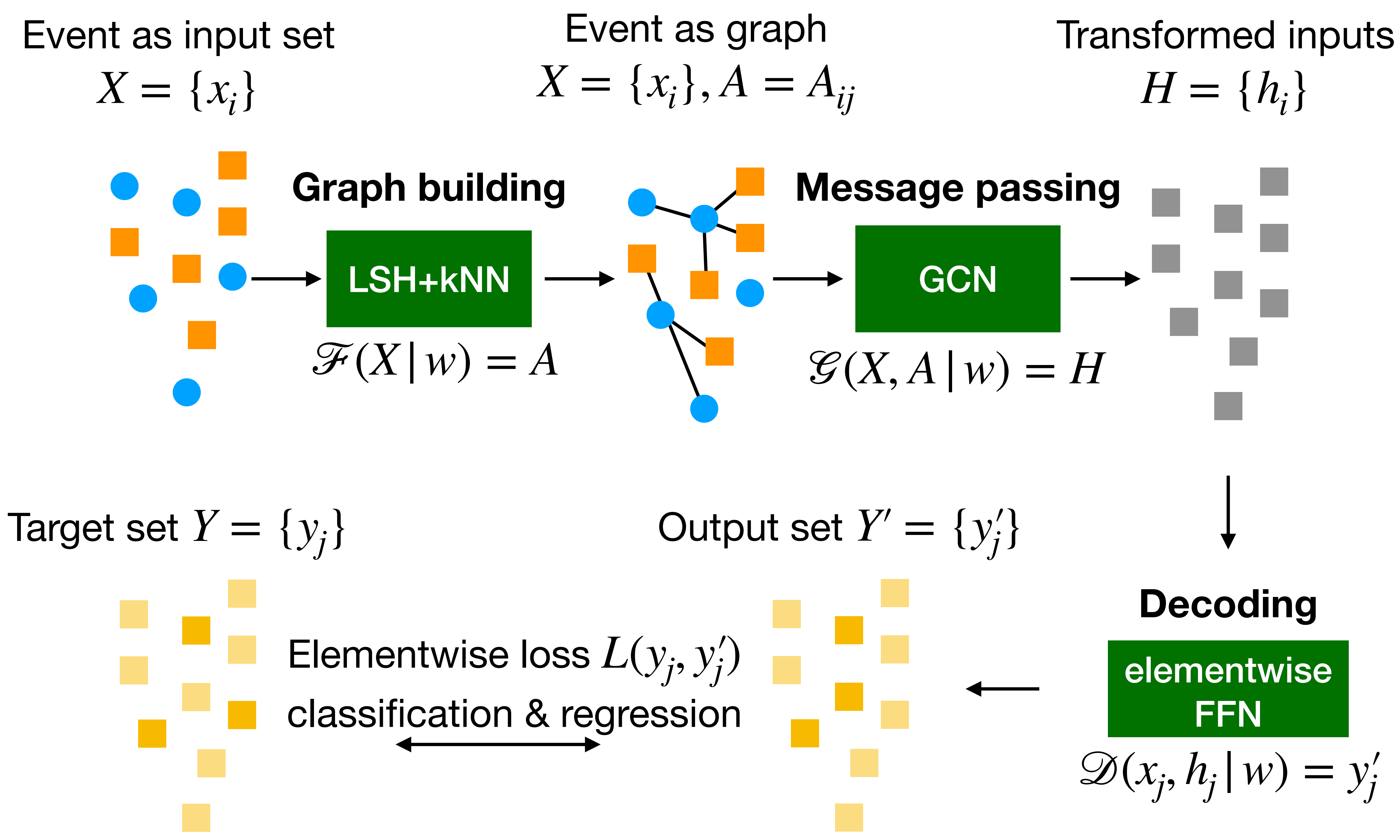


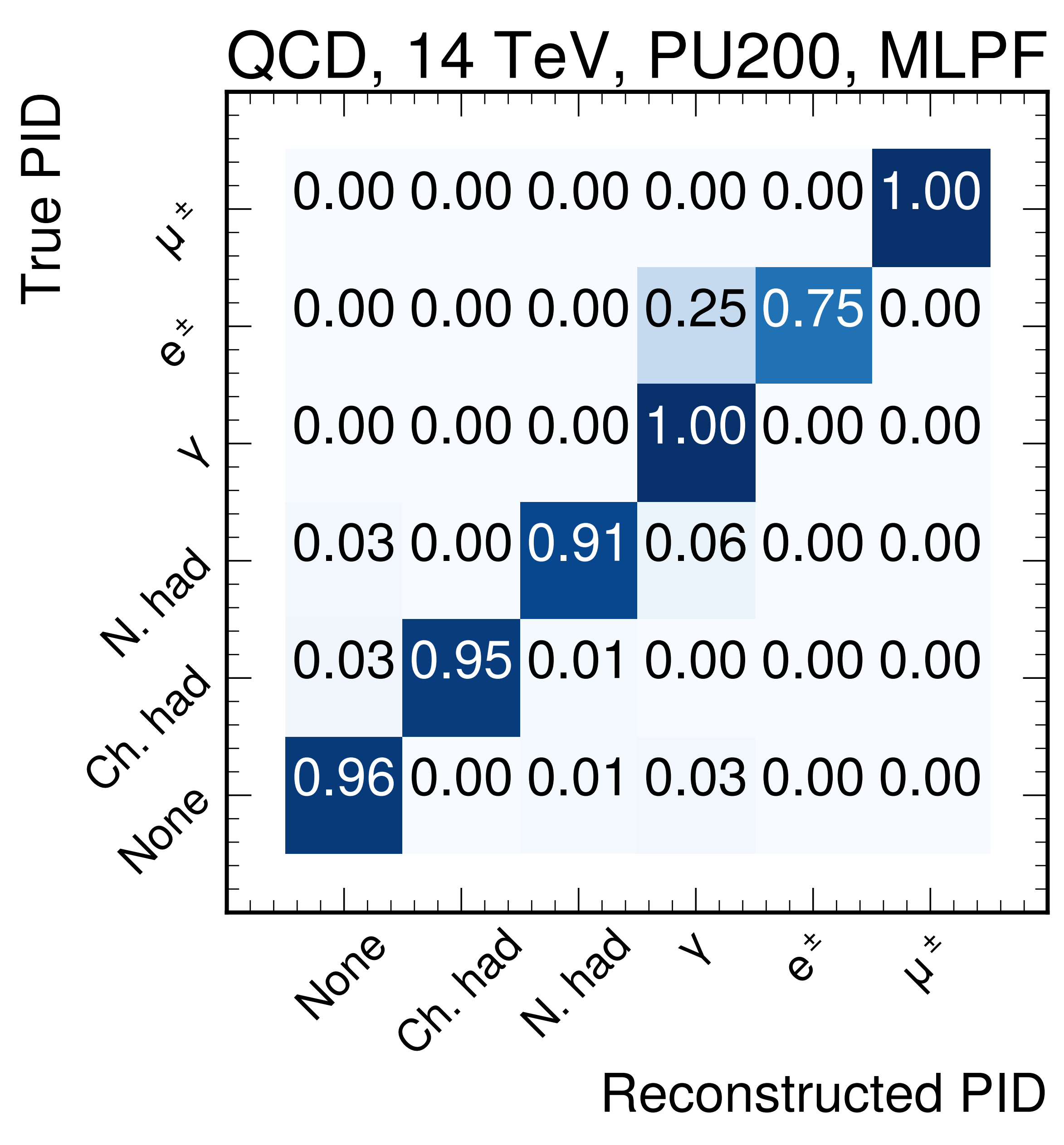
$$x_i = [\text{type}, p_\mathrm{T}, E_\mathrm{ECAL}, E_\mathrm{HCAL}, \eta, \phi, \eta_\mathrm{outer}, \phi_\mathrm{outer}, q, \ldots], \quad \text{type} \in \{\text{track}, \text{cluster}\}$$

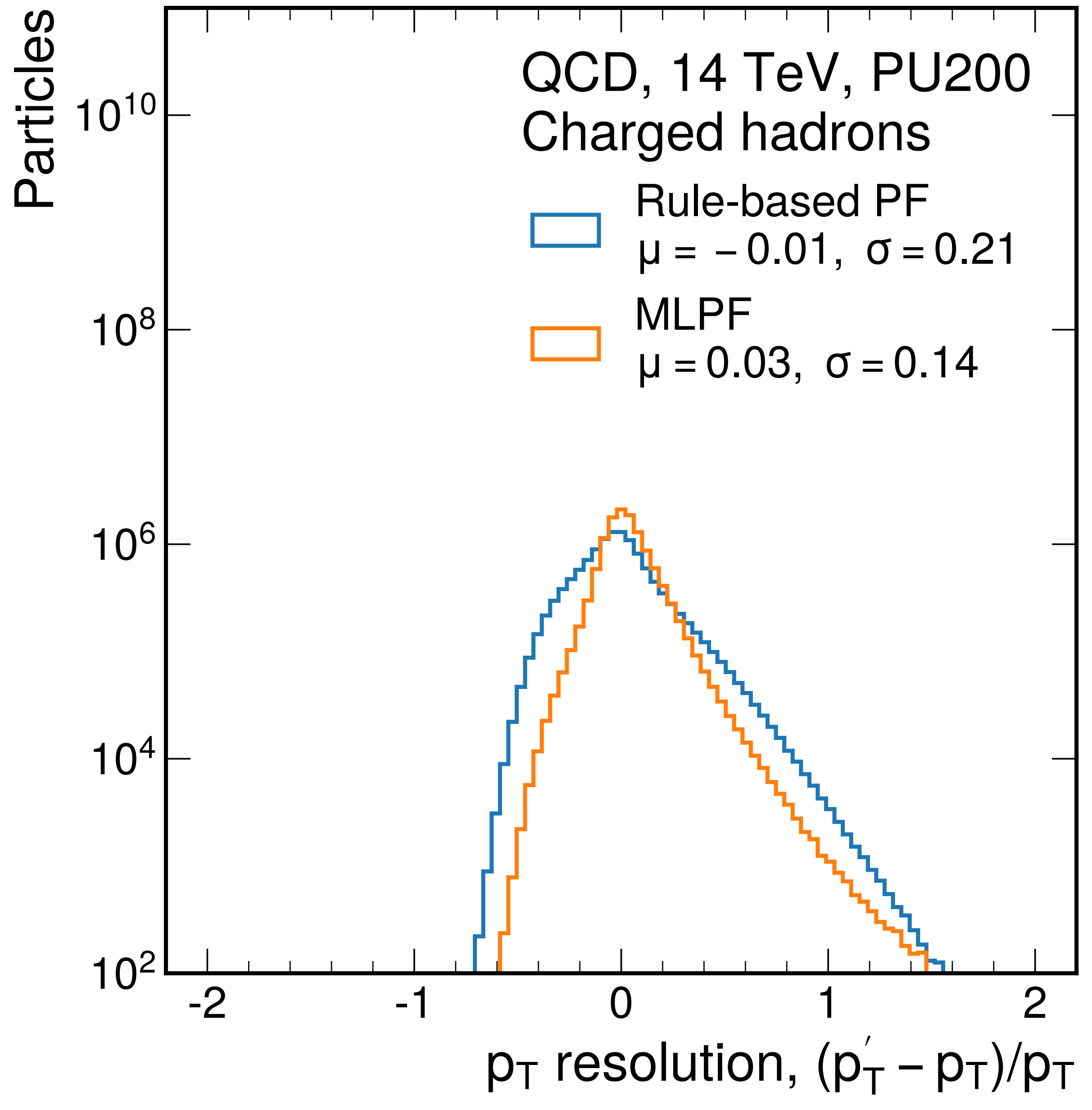
$$y_j = [\text{PID}, p_\mathrm{T}, E, \eta, \phi, q, \ldots], \quad \text{PID} \in \{\text{none}, \text{charged hadron}, \text{neutral hadron}, \gamma, e^\pm, \mu^\pm\}$$

BENCHMARK MLPF MODEL

arXiv:2101.08578







- Classification and regression performance: MLPF same or better than rule-based PF algorithm
- Models and training code available:
  - https://github.com/jpata/particleflow [10.5281/zenodo.4559587]
- Environment specified with singularity
- CI deployed through GitHub Actions