# b-hive:
# a modular training framework for state-of-the-art object-tagging within the Python ecosystem at the CMS experiment

Niclas Eich
On behalf of
the CMS Collaboration

III. Physikalisches Institut A

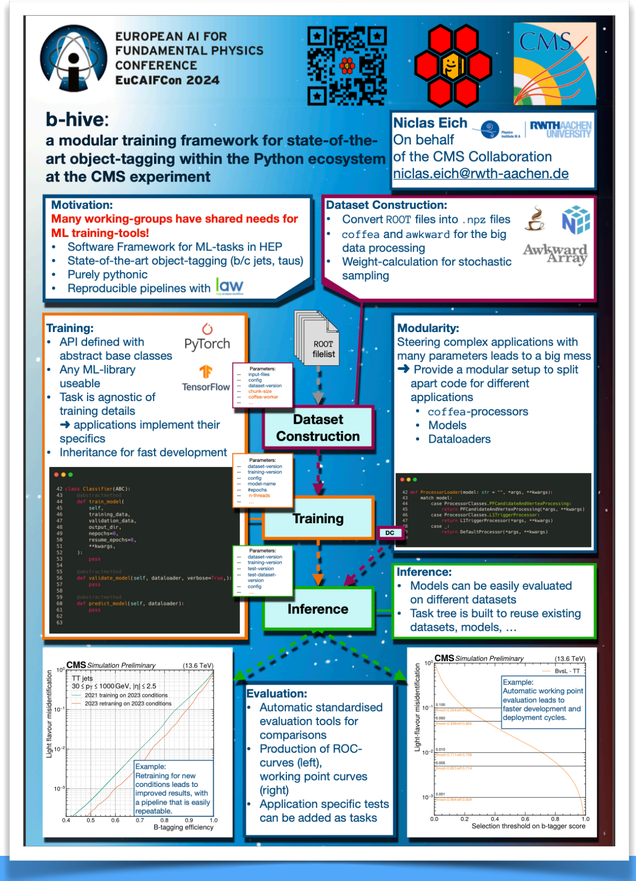
RWTH AACHEN UNIVERSITY

## Motivation:

- Everybody wants to do Machine-Learning trainings
- Full end-to-end pipeline is way harder than an example Notebook
  - Big data processing (`ROOT` files)
  - Conversion into a ML-friendly format (`.npy`/`.npz`)
  - Deploy state-of-the-art models
    - ParticleNet (graph-convolutions)
    - Transformer models

## Also:

- Have clearly defined workflows (not your 7 bash scripts!)
- Make trainings repeatable
- Standardized evaluation tools



# b-hive attacks these problems

- Pythonic training framework
- Workflow management with `law`
- `coffea`, `awkward`, `numpy` for the heavy data lifting
- No ML-framework lock in
  - `TensorFlow` and `PyTorch` can be used

➜ Modular Setup

- Easy configuration for different working-groups
- New applications are embedded in the pipeline

➜ Knowledge-sharing by code-sharing

Have a look: CERN-CMS-DP-2024-020

```yaml
bins_pt:
 - 30
 - 100
 - 500
 - 2000

bins_eta:
 - -2.5
 - -2.0
 - -0.5
 - 0.5
 - 2.0
 - 2.5

treename:
  "producer/custom_tree"

 global_features:
  - "jet_pt"
  - "jet_eta"
  - "n_Cpfcand"
  - "n_Npfcand"
  - "nsv"
  - "npv"

pf_candidate_features:
 - "pfcand_pt"
 - "pfcand_eta"
 - "pfcand_trackPt"
 - "pfcand_trackEta"
 - "pfcand_trackDeltaR"

truths:
 - "isB"
 - "isUDS"
 - "isG"
```