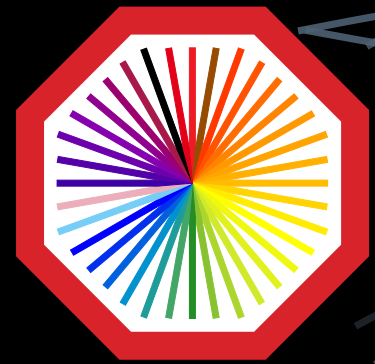




ALICE



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

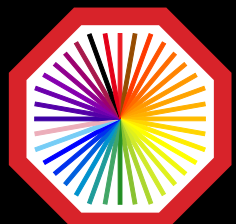
O2 Analysis Tutorial 5.0

Introduction to the ML workflow in DQ framework

Jinjoo Seo

Physikalisches Institut, Universität Heidelberg

12.11.2025



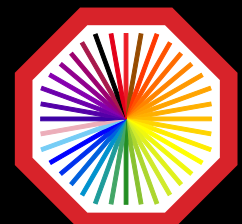
ALICE

Overview



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

- **ML workflow in DQ framework**
 - Dielectron analysis [[Link](#)]
 - MFT-Muon matching [[Link](#)]
- **ML workflow of dielectron analysis (focus of this talk)**
 - BDT configuration is fully handled through JSON [[example](#)]
 - BDT class, centrality & p_T binning, BDT selection, input features, onnx models...
 - Support Binary (signal/background) and MultiClass (prompt/nonprompt/background)
 - Support Centrality and p_T dependent BDT selections
 - Follows a similar JSON structure to the cut/histogram
 - Input features are defined based on DielectronsAll [[See](#)]
 - Welcome to extend the feature list as needed!

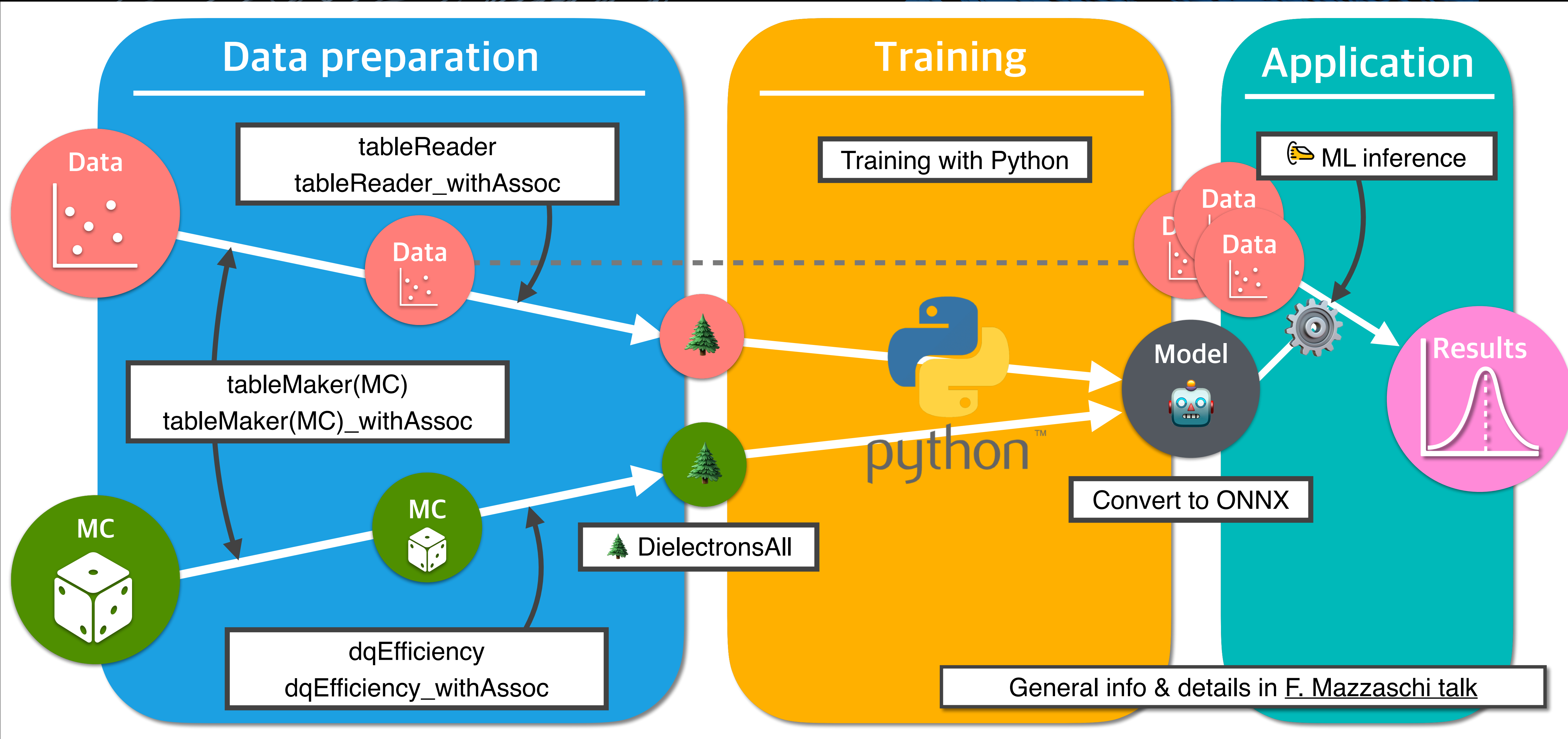


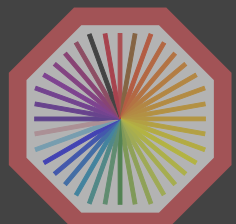
ALICE

ML workflow in DQ framework



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386



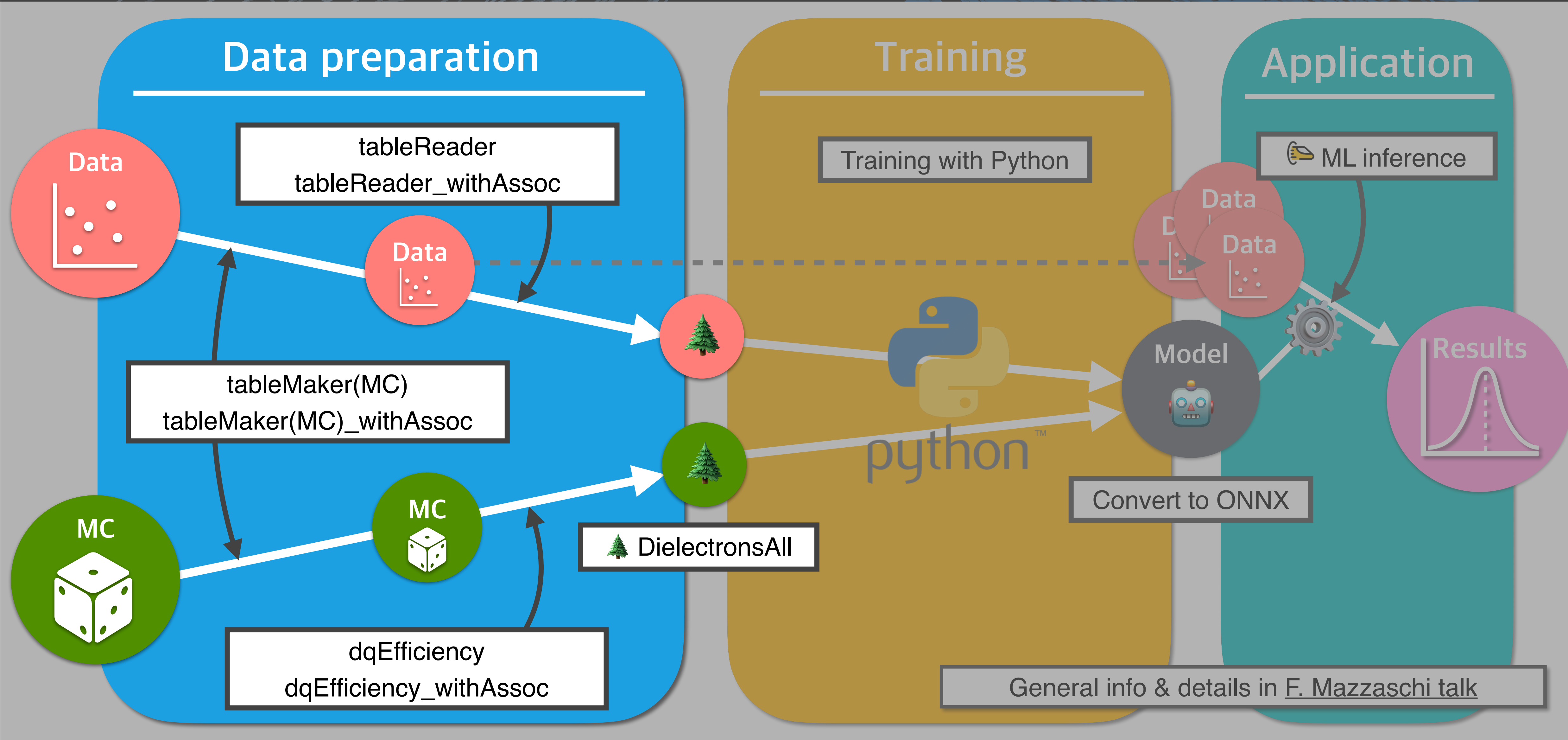


ALICE

ML workflow in DQ framework



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386





ALICE

Data preparation

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

- Do the same work as your normal DQ analysis

dq-pp-table-reader-with-assoc

analysis-same-event-pairing

processAllSkimmed

processBarrelOnlySkimmed

cfgBarrelMCRecSignals

cfgCorrFullGeo

cfgPrefilterTrackCut

cfgTrackCuts

electronSelection1_idstoreh

electronSelection1_idstoreh

Define MC signals

Apply looser selections than your selections

cfgTrackCuts

electronSelection1_idstoreh

cfgUseAbsDCA

0

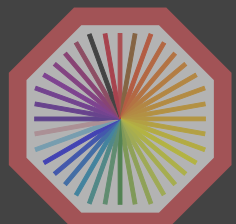
Enable the DielectronsAll table

Latest change by jseo at 11/11/25, 12:10 CET

Sync Max DF size: 100000000 Max derived file size: 0 Ready for slim derived data

Only enable tables which should be saved into an AO2D.root output file. This requires a derived data train which, unless 'Ready for slim' is checked, does not submit automatically and may need additional approval (click ? for more details). If you just need the information in these tables in a subsequent wagon in the same train, there is no need to enable the tables.

Store	Origin	Binding	Description	Version
☐	AOD	Dielectrons	RTDIELECTRON	0
☒	AOD	DielectronsAll	RTDIELECTRONALL	0
☐	AOD	DielectronsExtra	RTDIELEEXTRA	0

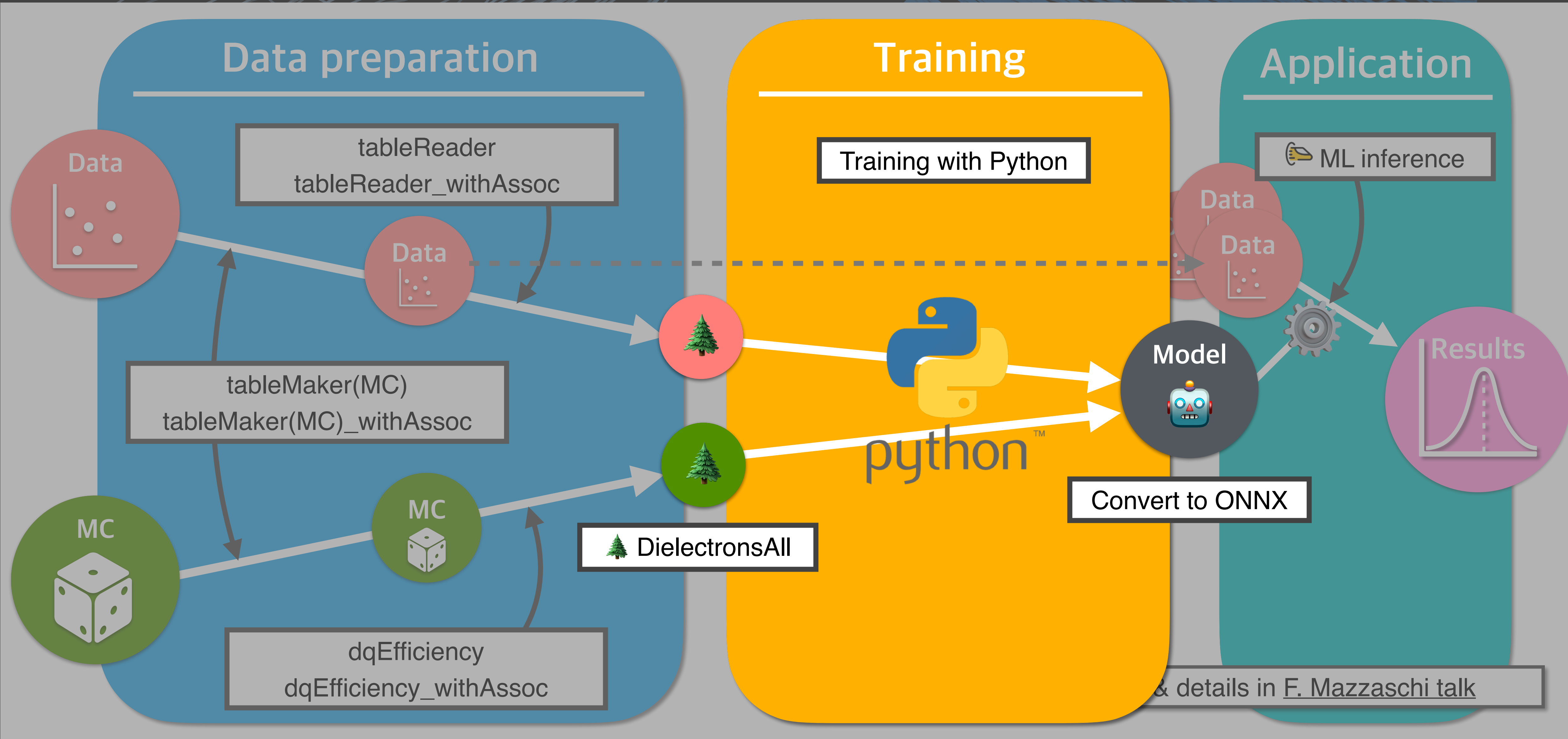


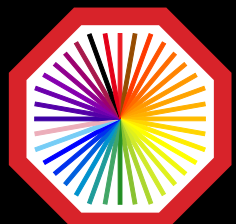
ALICE

ML workflow in DQ framework



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386





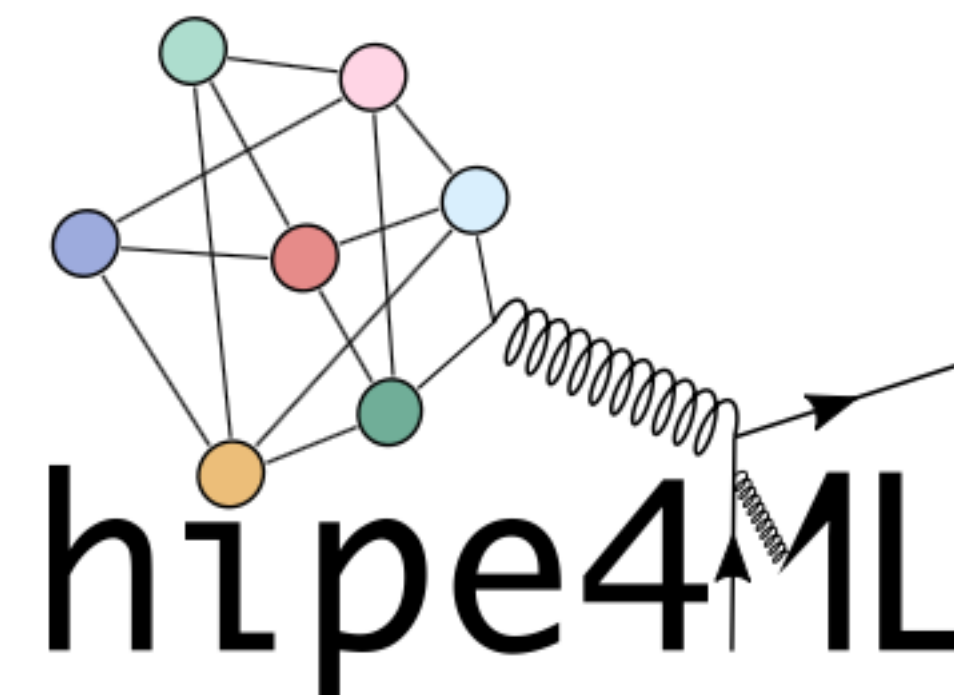
ALICE

Model training



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

- ✨ **Machine Learning: ML Tutorial (14 Nov)**
 - For details on using hiPe4ML, follow the ML tutorial on Friday
- If you don't have a sketch, start from this [\[Link\]](#)
 - Optimised for DQ analysis
 - Train one model per centrality and pT bin

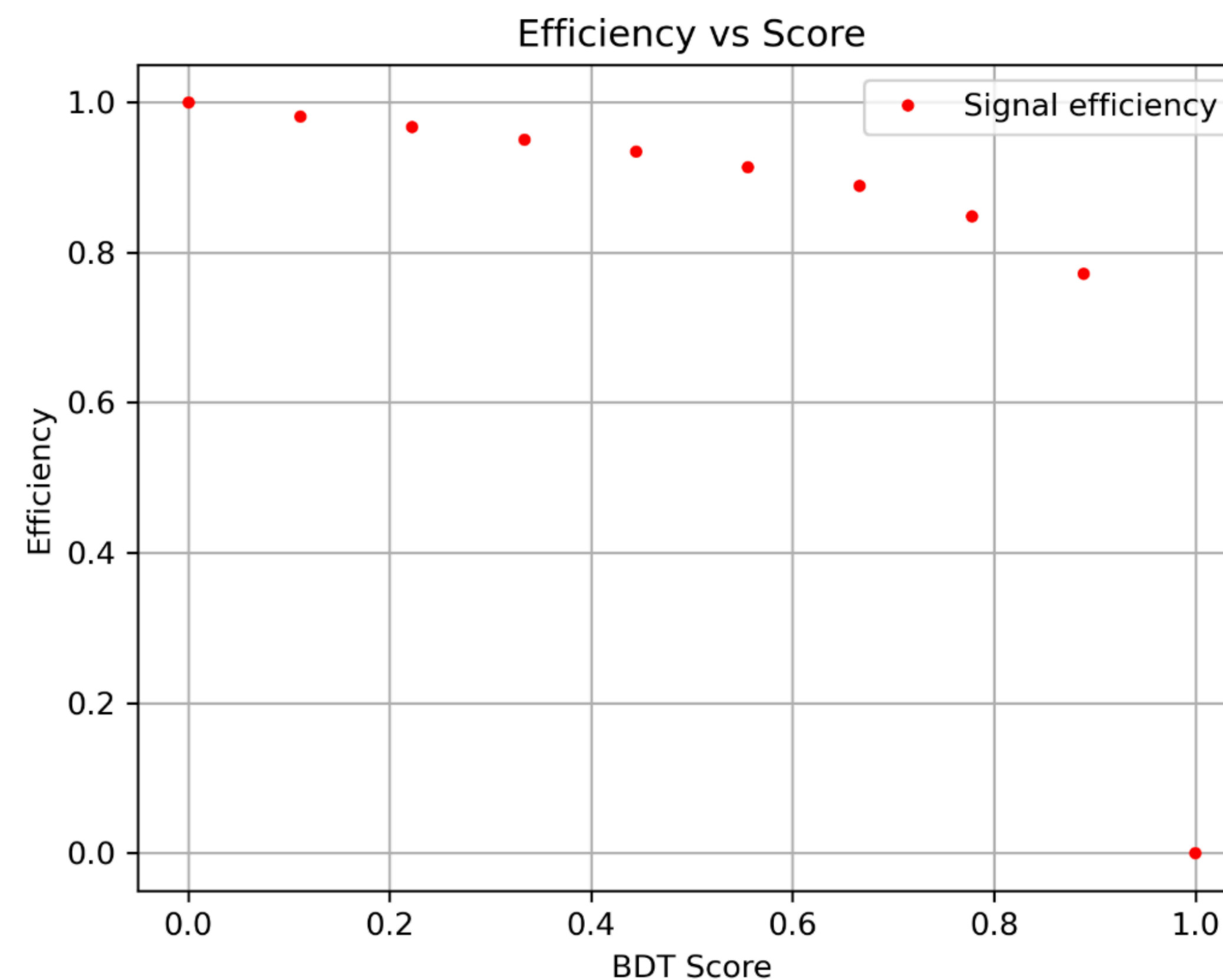
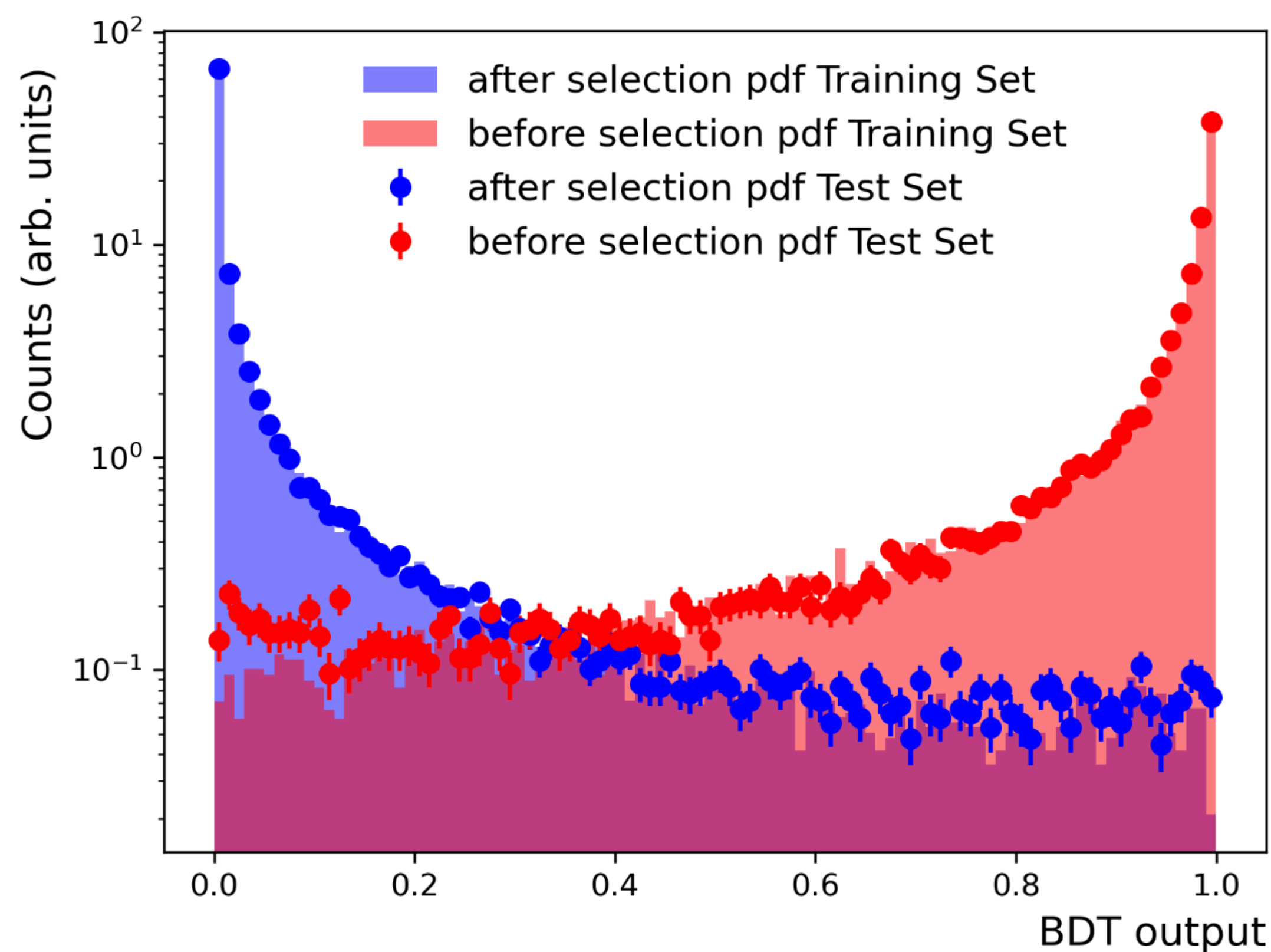


```
# pT and centrality bins and type for cut and JSON
PT_BINS = [(0, 2), (2, 8), (8, 20)] # pT bins
CENT_BINS = [(0, 30), (30, 60), (60, 90)] # centrality bins
CENT_TYPE = "kCentFT0C" # kCentFT0C, kCentFT0A, kCentFT0M
```

```
for centmin, centmax in CENT_BINS:
    for i, (ptmin, ptmax) in enumerate(PT_BINS):
        print(f"Processing cent {centmin}-{centmax}%, pT bin {ptmin} - {ptmax} GeV/c")
```

- Determine BDT threshold using pseudo-significance (binary_classification_example.py)
- Export model to ONNX
- Generate the JSON configuration for the hyperloop and local run

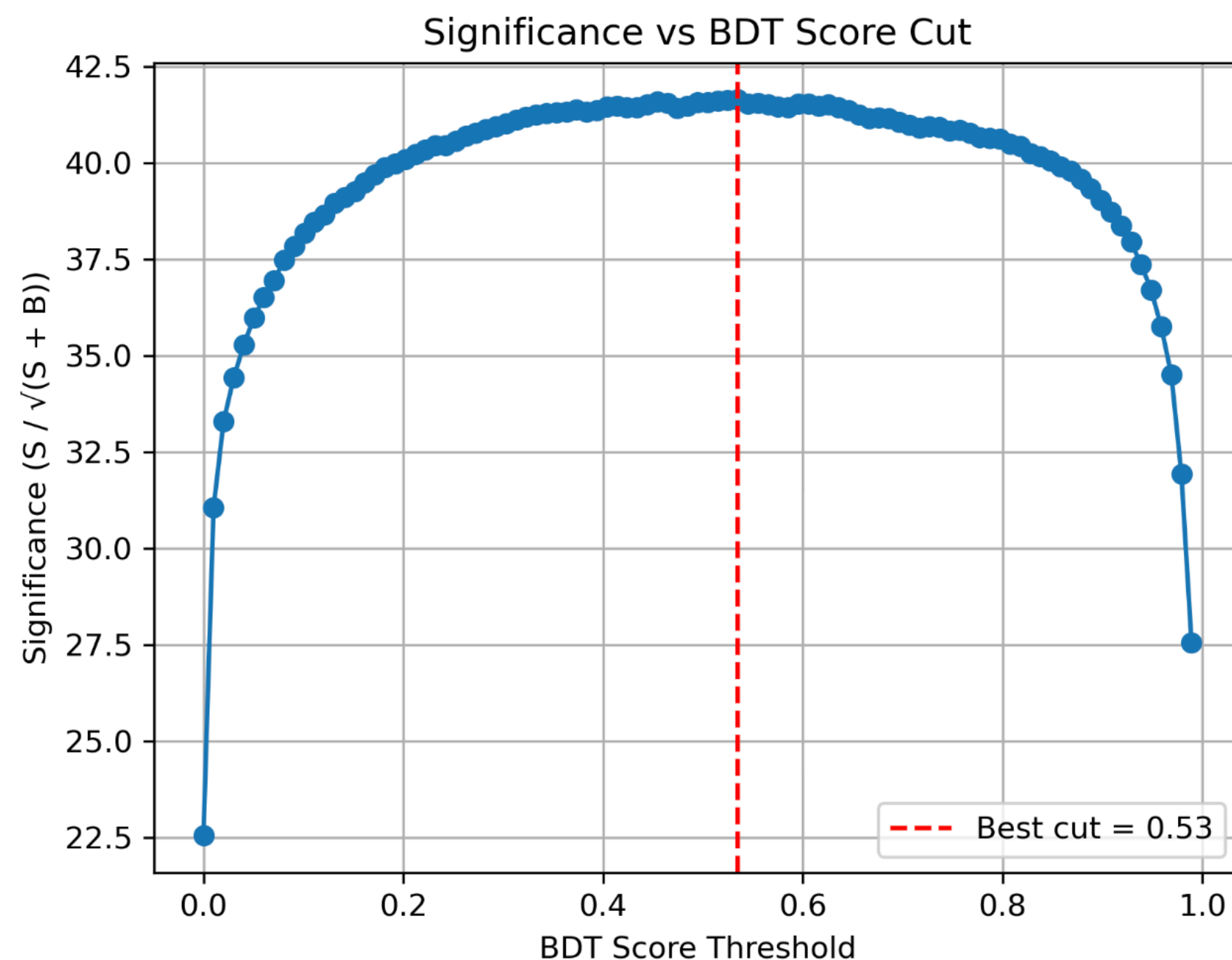
- How do we determine the BDT threshold?
- Reject the background, keep the signal, not too low efficiency...



- How do we determine the BDT cut?
- Reject the background, keep the signal, not too low efficiency...

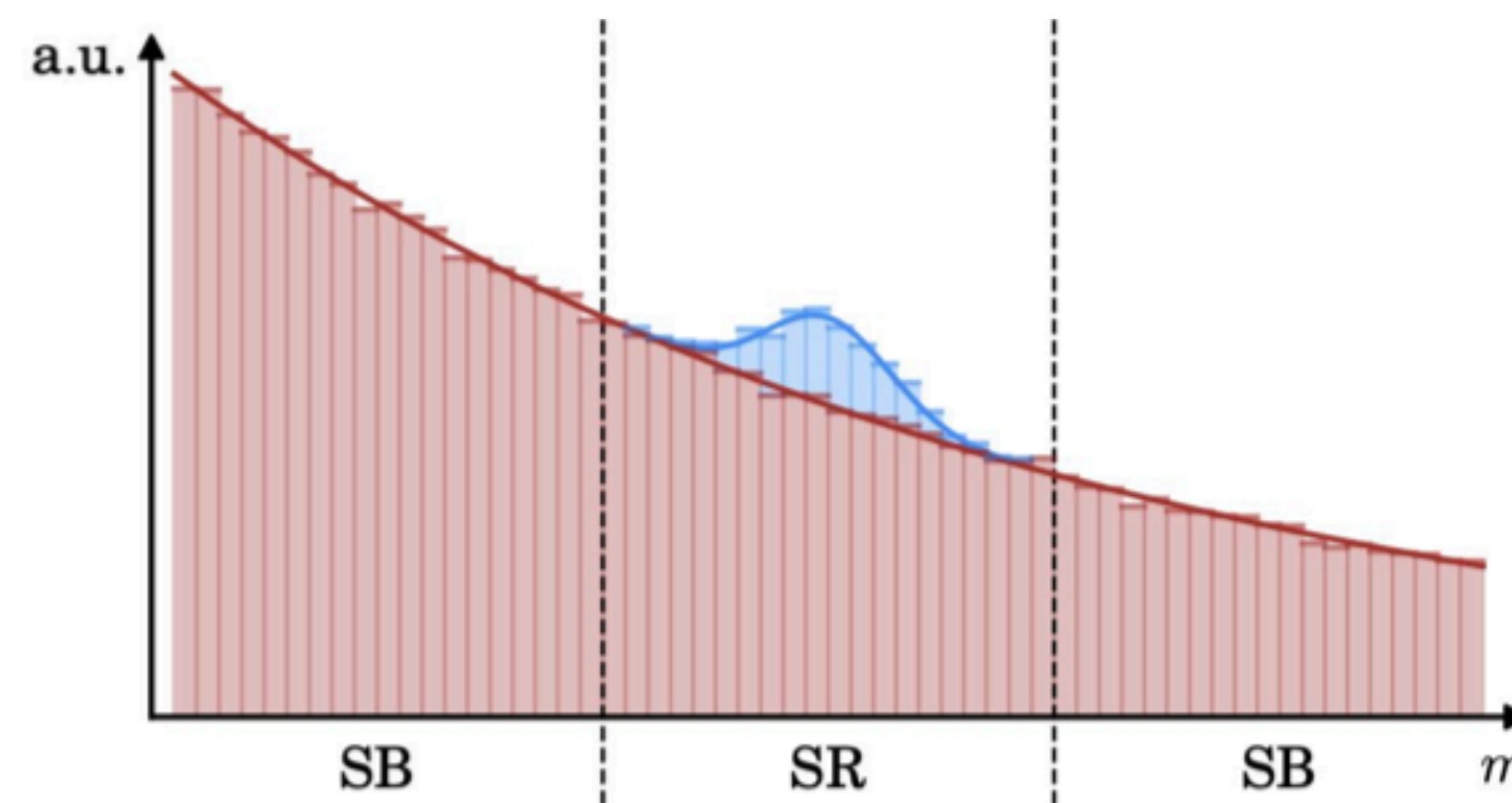
➔ Pseudo-significance

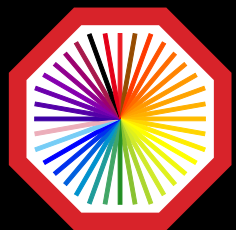
```
sig_array, thresholds = significance_array(bdt_scores, mass_after_cut, efficiency_array,
LUMI, XSEC_MUB, ACC_EFF, DELTAY, DELTAPT, BR_JPSI, SIG_WINDOW, SB_LEFT, SB_RIGHT, n_points=100)
```



$$PseudoSignificance = \frac{S_{expected}}{\sqrt{S_{expected} + B_{SR}}}$$

$$S_{expected} = \frac{d^2\sigma}{dp_T dy} \cdot (Acc \times \epsilon) \cdot \Delta p_T \cdot \Delta y \cdot BR \cdot L_{int}$$





ALICE

Export model to ONNX

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

- Export model to ONNX
- Feature names need to be converted ($f \rightarrow k$) to match DQ input features

```
# Export to ONNX
features_onnx = [f.replace("f", "k", 1) if f.startswith("f") else f for f in features_for_train]
export_to_onnx(model_hdl, prefix=f"{OUT_DIR}/modelBDT_{tag}", feature_names=features_onnx, input_table=parquet_path)
```



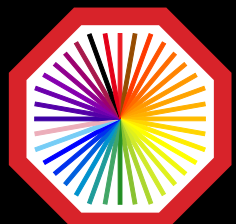
DielectronsAll

```
root [2] O2rtdielectronall->Show(1)
=====> EVENT:1
  fMass      = 3.11433
  fPt        = 1.62024
  fEta       = 1.27266
  fPhi       = 2.47833
  fSign      = 0
  fFilterMap = 1
  fMcDecision = 4
  fPt1       = 2.10735
```



DQMIResonse.h

```
enum class InputFeatures : uint8_t
{
    kMass = 0,
    kPt,
    kEta,
    kPhi,
    kPt1,
    kITSChi2NC11,
    kTPCNCLsCR1,
    kTPCNCLsFound1,
    kTPCChi2NC11,
    kDcaXY1,
    kDcaZ1,
```

```

"TestCut": {
  "type": "Binary",
  "title": "MyBDTModel",
  "inputFeatures": [
    "kMass",
    "kPt",
    "kEta",
    "kPhi",
    "kPt1",
    "kITSChi2NCl1",
    "kTPCNClSCl1",
    "kTPCNClSClFound1",
    "kTPCChi2NCl1",
    "kDcaXY1",
    "kDcaZ1"
  ],
  "modelFiles": [
    "cent_10_30_pt0_2_onnx.onnx",
    "cent_10_30_pt2_20_onnx.onnx",
    "cent_30_50_pt0_2_onnx.onnx",
    "cent_30_50_pt2_20_onnx.onnx"
  ]
},

```

- **type:** **Binary / MultiClass**
- **title:** title of ML configuration
- **inputFeatures:** input features
- **modelFiles:** onnx file name

```

"cent": "kCentFT0C",
"AddCentCut-Cent1030": {
  "centMin": 10,
  "centMax": 30,
  "AddPtCut-pTBin1": {
    "pTMin": 0,
    "pTMax": 2,
    "AddMLCut-background": {
      "var": "kBdtBackground",
      "cut": 0.5,
      "exclude": false
    }
  },
  "AddPtCut-pTBin2": {
    "pTMin": 2,
    "pTMax": 20,
    "AddMLCut-background": {
      "var": "kBdtBackground",
      "cut": 0.5,
      "exclude": false
    }
  }
},

```

```

"modelPathsCCDB": {
  "values": [
    "Users/j/jseo/ML/test/binary/bin0",
    "Users/j/jseo/ML/test/binary/bin0",
    "Users/j/jseo/ML/test/binary/bin0",
    "Users/j/jseo/ML/test/binary/bin0"
  ]
}

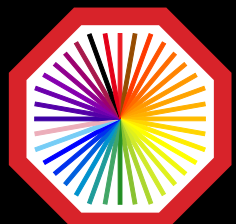
```

Model files and cut arrays must be in 1:1 and in the same order

- **cent:** centrality estimator determination
 - **kCentFT0C / kCentFT0A / kCentFT0M**
- **AddCentCut:** centrality cut
 - **centMin:** minimum centrality
 - **centMax:** maximum centrality
- **AddPtCut:** Pt cut

defined inside each centrality block

 - **pTMin:** minimum Pt
 - **pTMax:** maximum Pt
 - **var:**
 - **kBdtBackground / kBdtPrompt / kBdtNonprompt**
 - **cut:** bdt threshold
 - **exclude:** **false/true**
 - false: keep candidates with score > cut
 - true: keep candidates with score < cut



- Generate the JSON for the hyperloop and local run

```
for centmin, centmax in CENT_BINS:
    cent_key = f"AddCentCut-Cent{centmin:02d}{centmax:02d}"
    json_dict["TestCut"][cent_key] = {
        "centMin": centmin,
        "centMax": centmax
    }
    for i, (ptmin, ptmax) in enumerate(PT_BINS):
        pt_key = f"AddPtCut-pTBin{i+1}"
        model_name = f"modelBDT_cent{centmin}_{centmax}_pt{ptmin}_{ptmax}_onnx.onnx"
        cut_value = best_cuts_dict[f"{centmin}_{centmax}_{ptmin}_{ptmax}"]

        json_dict["TestCut"]["modelFiles"].append(model_name)
        json_dict["TestCut"][cent_key][pt_key] = {
            "pTMin": ptmin,
            "pTMax": ptmax,
            "AddMLCut-background": {
                "var": "kBdtBackground",
                "cut": cut_value,
                "exclude": False
            }
        }
```

JSON file

JSON string

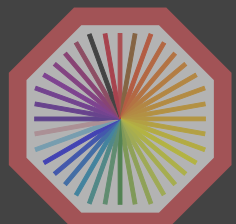
```
{
  "TestCut": {
    "type": "Binary",
    "title": "MyBDTModel",
    "inputFeatures": [
      "kITSChi2NCL1",
      "kTPCNCLsCR1",
      "kTPCChi2NCL1",
      "kTPCNSigmaEl1",
      "kITSChi2NCL2",
      "kTPCNCLsCR2",
      "kTPCChi2NCL2",
      "kTPCNSigmaEl2"
    ],
    "modelFiles": [
      "cent_0_100_pt0_2_onnx.onnx",
      "cent_0_100_pt2_100_onnx.onnx"
    ],
    "cent": "kCentFT0C",
    "AddCentCut-Cent00100": {
      "centMin": 0,
      "centMax": 100,
      "AddPtCut-pTBin1": {
        "pTMin": 0,
        "pTMax": 2,
        "AddMLCut-background": {
          "var": "kBdtBackground",
          "cut": 0.643113145059189,
          "exclude": false
        }
      },
      "AddPtCut-pTBin2": {
        "pTMin": 2,
        "pTMax": 100,
        "AddMLCut-background": {
          "var": "kBdtBackground",
          "cut": 0.5751400298420191,
          "exclude": false
        }
      }
    }
  }
}
```

Hyperloop JSON string:

```
{"TestCut":{"type":"Binary","title":"MyBDTModel","inputFeatures":["kITSChi2NCL1","kTPCNCLsCR1","kTPCChi2NCL1","kTPCNSigmaEl1","kITSChi2NCL2","kTPCNCLsCR2","kTPCChi2NCL2","kTPCNSigmaEl2"],"modelFiles":["cent_0_100_pt0_2_onnx.onnx","cent_0_100_pt2_100_onnx.onnx"],"cent":"kCentFT0C","AddCentCut-Cent00100":{"centMin":-1,"centMax":100,"AddPtCut-pTBin1":{"pTMin":0,"pTMax":2,"AddMLCut-background":{"var":"kBdtBackground","cut":0.643113145059189,"exclude":false},"AddPtCut-pTBin2":{"pTMin":2,"pTMax":100,"AddMLCut-background":{"var":"kBdtBackground","cut":0.5751400298420191,"exclude":false}}}}}
```

Local JSON string:

```
"{"TestCut":{"type":"Binary","title":"MyBDTModel","inputFeatures":["kITSChi2NCL1","kTPCNCLsCR1","kTPCChi2NCL1","kTPCNSigmaEl1","kITSChi2NCL2","kTPCNCLsCR2","kTPCChi2NCL2","kTPCNSigmaEl2"],"modelFiles":["cent_0_100_pt0_2_onnx.onnx","cent_0_100_pt2_100_onnx.onnx"],"cent":"kCentFT0C","AddCentCut-Cent00100":{"centMin":0,"centMax":100,"AddPtCut-pTBin1":{"pTMin":0,"pTMax":2,"AddMLCut-background":{"var":"kBdtBackground","cut":0.643113145059189,"exclude":false},"AddPtCut-pTBin2":{"pTMin":2,"pTMax":100,"AddMLCut-background":{"var":"kBdtBackground","cut":0.5751400298420191,"exclude":false}}}}}"
```

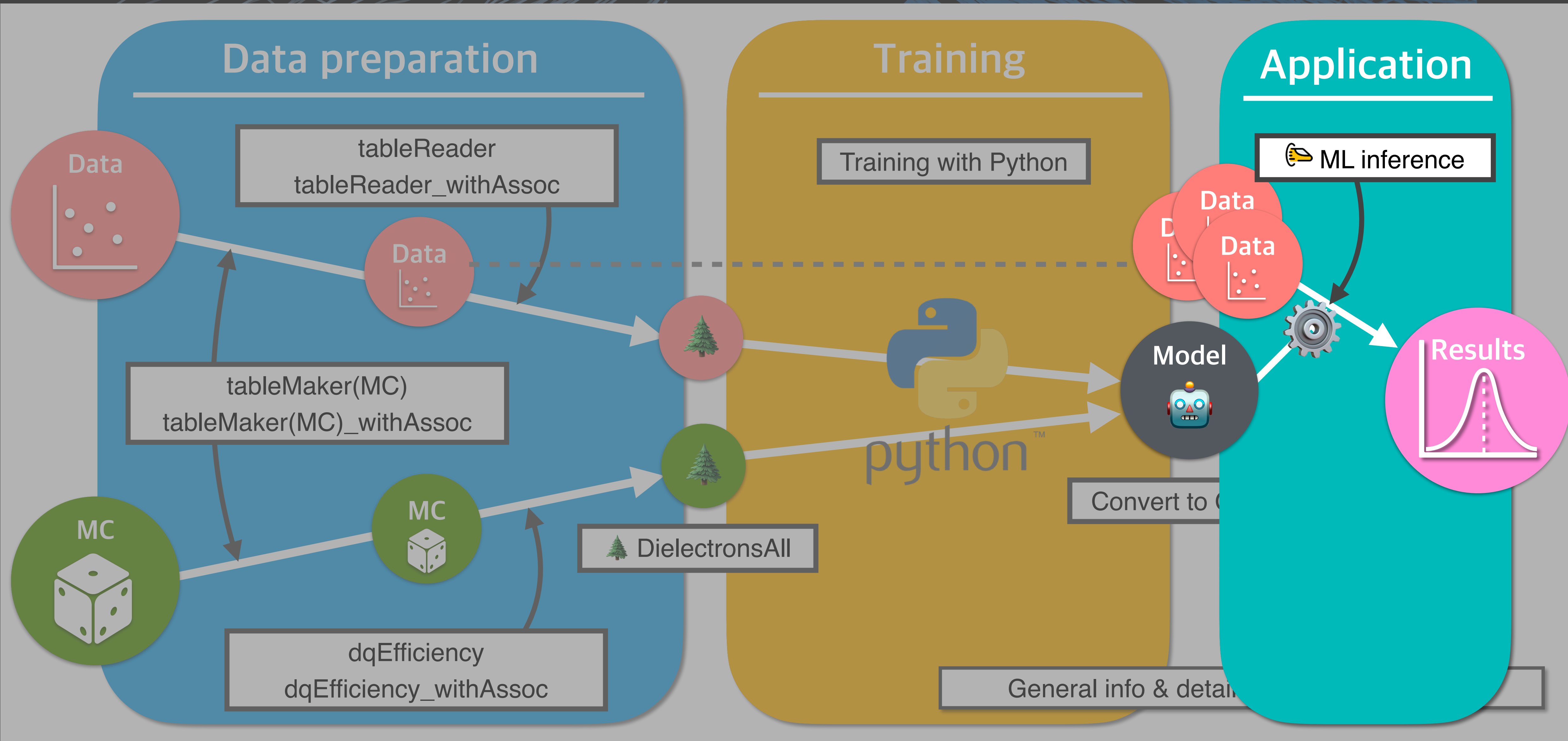



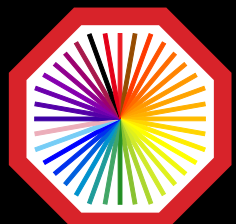
ALICE

ML workflow in DQ framework



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386





ALICE

Configure BDT analysis

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

- Upload ONNX to CCDB [[Link](#)]
- Adjust the configuration of the wagon

applyBDT		1
ccdb-url		http://alice-ccdb.cern.ch
fConfigBdtCutsJSON		{"TestCut":{"type":"Binary","title":"MyBD
geoPath		GLO/Config/GeometryAligned
grplhcif		GLO/Config/GRPLHCIF
grpmagPath		GLO/Config/GRPMagField
loadModelsFromCCDB		1
lutPath		GLO/Param/MatLUT
modelPathsCCDB		Users/j/jseo/pp/tutorial/bin0 Users/j/jseo/pp/tutorial/bin1

Path:

File:

Validity ☒ Infinite

New CCDB object upload

Appended to your user space (alice-ccdb.cern.ch/Users/j/jseo/)

Binary blob to upload, max 500MB

Global object, valid from 1 to ∞

to

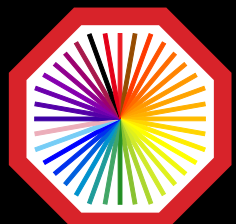
to run no.

Absolute epoch timestamp (in milliseconds)

Run numbers, like 245146 (both ends fully included)

path and ONNX in 1:1 and in the same order

```
"modelFiles": [
  "modelBDT_cent0_100_pt0_2_onnx.onnx",
  "modelBDT_cent0_100_pt2_100_onnx.onnx"
],
```

tableReader(_withAssoc).cxx



- ML inference can be configurable

```
// ML inference
Configurable<bool> applyBDT{"applyBDT", false, "Flag to apply ML selections"};
Configurable<std::string> fConfigBdtCutsJSON{"fConfigBdtCutsJSON", "", "Additional list of BDT cuts in JSON format"};
Configurable<std::vector<std::string>> modelPathsCCDB{"modelPathsCCDB", std::vector<std::string>{"Users/j/jseo/ML/PbPbPsi/default/"}, "Paths of models on CCDB"};
Configurable<int64_t> timestampCCDB{"timestampCCDB", -1, "timestamp of the ONNX file for ML model used to query in CCDB"};
Configurable<bool> loadModelsFromCCDB{"loadModelsFromCCDB", false, "Flag to enable or disable the loading of models from CCDB"};
```

- Load BDT configurations from the JSON file

```
if (applyBDT) {
    // BDT cuts via JSON
    std::vector<double> binsML;
    o2::framework::LabeledArray<double> cutsML;
    std::vector<int> cutDirML;
    int nClassesML = 1;
    std::vector<std::string> namesInputFeatures;
    std::vector<std::string> onnxFileNames;

    auto config = o2::aod::dqmlcuts::GetBdtScoreCutsAndConfigFromJSON(fConfigBdtCutsJSON.value.c_str());

    if (std::holds_alternative<dqmlcuts::BinaryBdtScoreConfig>(config)) {
        auto& cfg = std::get<dqmlcuts::BinaryBdtScoreConfig>(config);
        binsML = cfg.binsML;
        nClassesML = 1;
        cutsML = cfg.cutsML;
        cutDirML = cfg.cutDirs;
        namesInputFeatures = cfg.inputFeatures;
        onnxFileNames = cfg.onnxFiles;
        fDQMLResponse.setBinsCent(cfg.binsCent);
        fDQMLResponse.setBinsPt(cfg.binsPt);
        fDQMLResponse.setCentType(cfg.centType);
        LOG(info) << "Using BDT cuts for binary classification";
    } else {
```

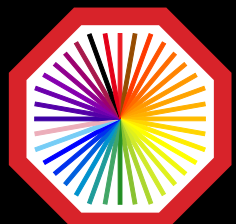
Load the ONNX model and
BDT threshold from the JSON

Binary classification

```
} else {
    auto& cfg = std::get<dqmlcuts::MultiClassBdtScoreConfig>(config);
    binsML = cfg.binsML;
    nClassesML = 3;
    cutsML = cfg.cutsML;
    cutDirML = cfg.cutDirs;
    namesInputFeatures = cfg.inputFeatures;
    onnxFileNames = cfg.onnxFiles;
    fDQMLResponse.setBinsCent(cfg.binsCent);
    fDQMLResponse.setBinsPt(cfg.binsPt);
    fDQMLResponse.setCentType(cfg.centType);
    LOG(info) << "Using BDT cuts for multiclass classification";
}

fDQMLResponse.configure(binsML, cutsML, cutDirML, nClassesML);
if (loadModelsFromCCDB) {
    ccdbApi.init(ccdburl);
    fDQMLResponse.setModelPathsCCDB(onnxFileNames, ccdbApi, modelPathsCCDB, timestampCCDB);
} else {
    fDQMLResponse.setModelPathsLocal(onnxFileNames);
}
fDQMLResponse.cacheInputFeaturesIndices(namesInputFeatures);
fDQMLResponse.init(false, 1); // 1 thread
}
```

Multi classification



tableReader(_withAssoc).cxx



```

if (applyBDT) {
    std::vector<float> dqInputFeatures = fDQMLResponse.getInputFeatures(t1, t2, VarManager::fgValues);

    if (dqInputFeatures.empty()) {
        LOG(fatal) << "Input features for ML selection are empty! Please check your configuration.";
        return;
    }

    int modelIndex = -1;
    const auto& binsCent = fDQMLResponse.getBinsCent();
    const auto& binsPt = fDQMLResponse.getBinsPt();
    const std::string& centType = fDQMLResponse.getCentType();

    if ("kCentFT0C" == centType) {
        modelIndex = o2::aod::dqmlcuts::getMlBinIndex(VarManager::fgValues[VarManager::kCentFT0C], VarManager::
    } else if ("kCentFT0A" == centType) {
        modelIndex = o2::aod::dqmlcuts::getMlBinIndex(VarManager::fgValues[VarManager::kCentFT0A], VarManager::
    } else if ("kCentFT0M" == centType) {
        modelIndex = o2::aod::dqmlcuts::getMlBinIndex(VarManager::fgValues[VarManager::kCentFT0M], VarManager::
    } else {
        LOG(fatal) << "Unknown centrality estimation type: " << centType;
        return;
    }

    if (modelIndex < 0) {
        LOG(debug) << "ML index is negative! This means that the centrality/pt is not in the range of the model";
        continue;
    }

    LOG(debug) << "Model index: " << modelIndex << ", pT: " << VarManager::fgValues[VarManager::
    isSelectedBDT = fDQMLResponse.isSelectedML(dqInputFeatures, modelIndex, fOutputMLPsi2ee);
    VarManager::FillBdtScore(fOutputMLPsi2ee); // TODO: check if this is needed or not
}

```

Feature extraction

ML inference

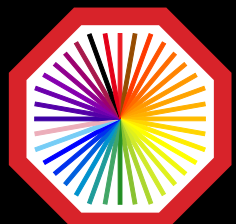
Retrieve output scores

```

void VarManager::FillBdtScore(T1 const& bdtScore, float* values)
{
    if (!values) {
        values = fgValues;
    }

    if (bdtScore.size() == 1) {
        values[kBdtBackground] = bdtScore[0];
    } else if (bdtScore.size() == 3) {
        values[kBdtBackground] = bdtScore[0];
        values[kBdtPrompt] = bdtScore[1];
        values[kBdtNonprompt] = bdtScore[2];
    } else {
        LOG(warning) << "Unexpected number of BDT outputs: " <<
            bdtScore.size();
    }
}

```

ALICE

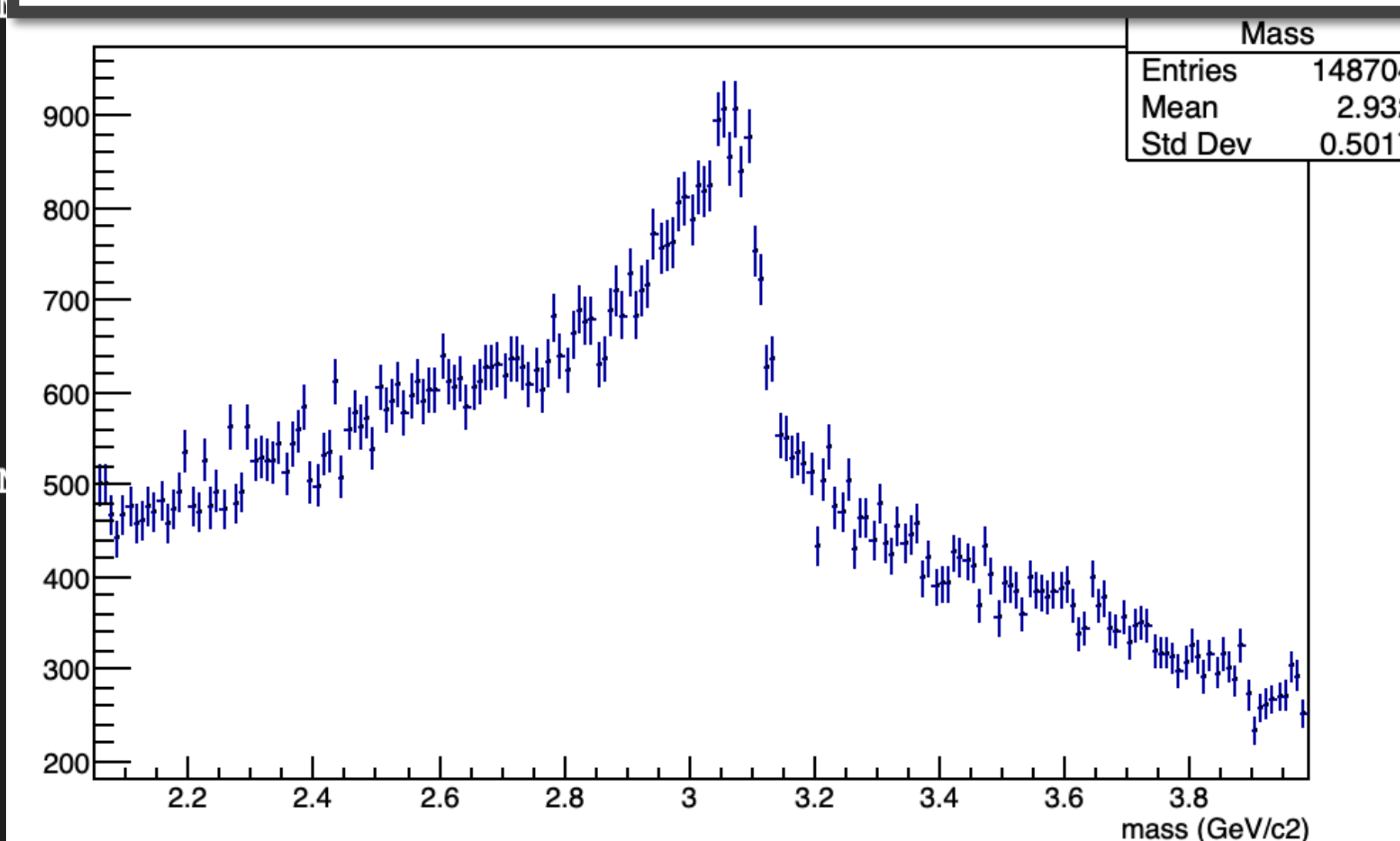
AnalysisResults.root

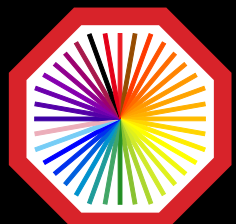
UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

```
[784164:analysis-same-event-pairing]: [20:02:23][INFO] ===== interpreting JSON for
ML analysis cuts
[784164:analysis-same-event-pairing]: [20:02:23][INFO] JSON string: {"TestCut":
{"type":"Binary","title":"MyBDTModel","inputFeatures":
["kITSChi2NC11","kTPCNClCR1","kTPCChi2NC11","kTPCNSigmaEl1","kITSChi2NC12","kTPCNClCR2","kTPCChi2NC12","kTPCNSigmaEl2"]
,"modelFiles":["cent_0_100_pt0_2_onnx.onnx","cent_0_100_pt2_100_onnx.onnx"],"cent":"kCentFT0C","AddCentCut-Cent00100":
{"centMin":-1,"centMax":100,"AddPtCut-pTBin1":{"pTMin":0,"pTMax":2,"AddMLCut-background":
{"var":"kBdtBackground","cut":0.643113145059189,"exclude":false}},"AddPtCut-pTBin2":{"pTMin":2,"pTMax":100,"AddMLCut-
background":{"var":"kBdtBackground","cut":0.5751400298420191,"exclude":false}}}}
[784164:analysis-same-event-pairing]: [20:02:23][INFO] Added cut for kCentFT0C_cent-1_100_pt0.0_2.0 with cuts: [
[784164:analysis-same-event-pairing]: 0.643113] and direction: CutSmaller
[784164:analysis-same-event-pairing]: [20:02:23][INFO] Added cut for kCentFT0C_cent-1_100_pt2.0_100.0 with cuts: [
[784164:analysis-same-event-pairing]: 0.57514] and direction: CutSmaller
[784164:analysis-same-event-pairing]: [20:02:23][INFO] Using BDT cuts for binary classification
```

```
[784164:analysis-same-event-pairing]: [20:02:25][INFO] --- ONNX-ML model ---
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Hyperloop test/Grid job detected! 1
threads anyway to 1.
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Input Nodes:
[784164:analysis-same-event-pairing]: [20:02:25][INFO] input : 1x8
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Output Nodes:
[784164:analysis-same-event-pairing]: [20:02:25][INFO] label : 1
[784164:analysis-same-event-pairing]: [20:02:25][INFO] probabilities : 1x2
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Model validity - From: 0, Until: 0
[784164:analysis-same-event-pairing]: [20:02:25][INFO] --- Model initialized! ---
[784164:analysis-same-event-pairing]: [20:02:25][INFO] --- ONNX-ML model ---
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Hyperloop test/Grid job detected! 1
threads anyway to 1.
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Input Nodes:
[784164:analysis-same-event-pairing]: [20:02:25][INFO] input : 1x8
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Output Nodes:
[784164:analysis-same-event-pairing]: [20:02:25][INFO] label : 1
[784164:analysis-same-event-pairing]: [20:02:25][INFO] probabilities : 1x2
[784164:analysis-same-event-pairing]: [20:02:25][INFO] Model validity - From: 0, Until: 0
[784164:analysis-same-event-pairing]: [20:02:25][INFO] --- Model initialized! ---
```

✨ Get histograms after the BDT selections!





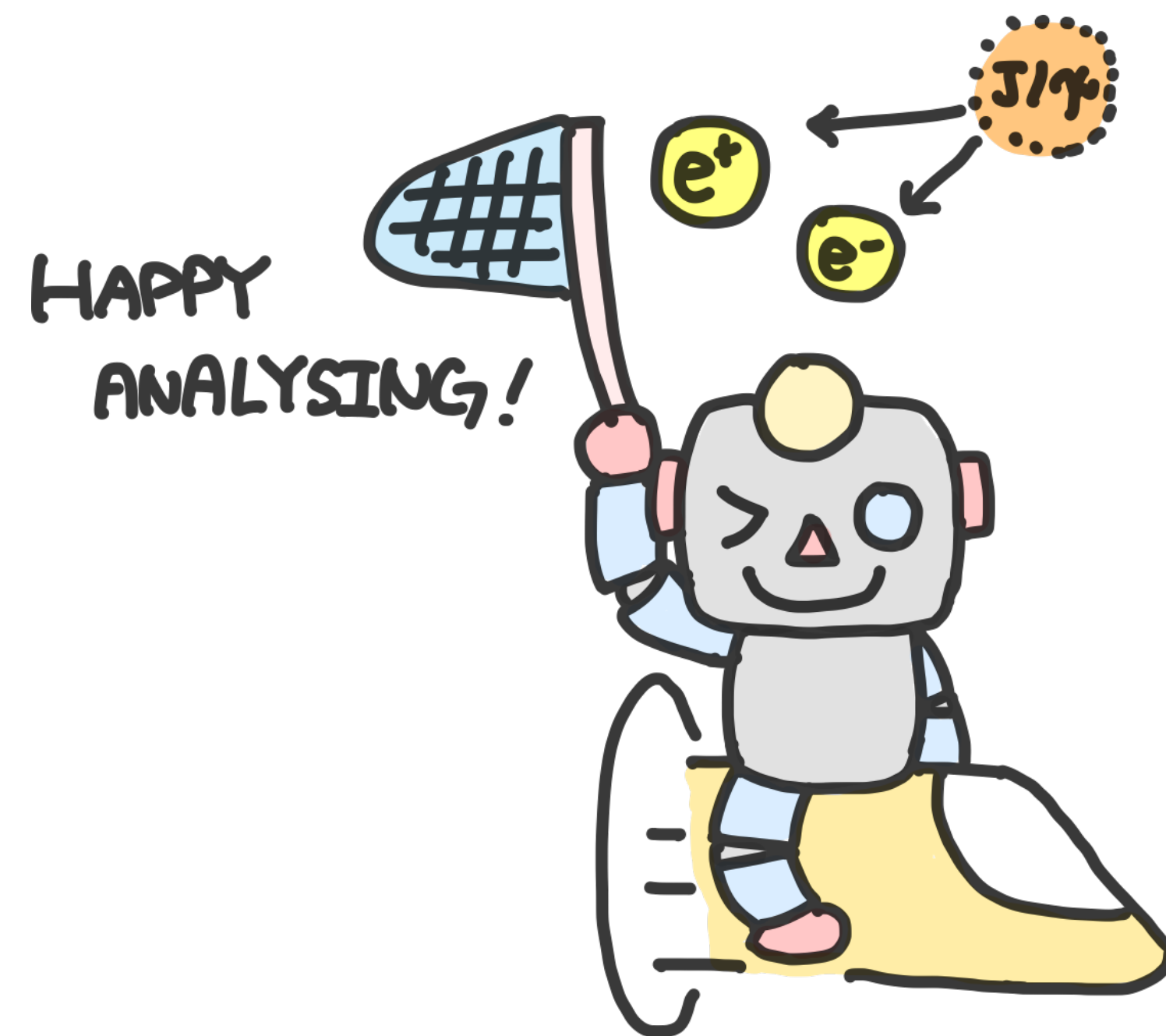
ALICE

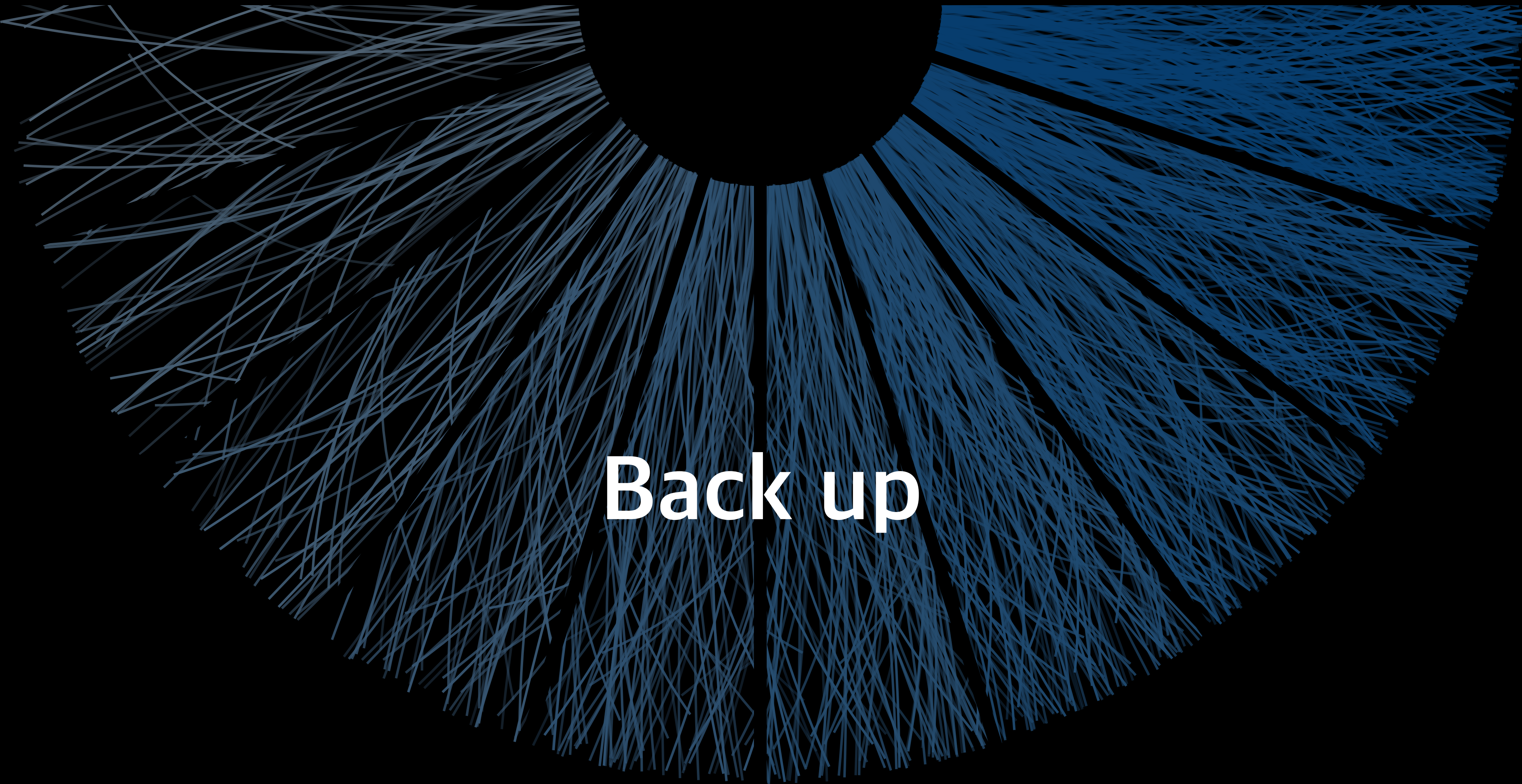
Summary



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

- ML in the DQ framework is ready to be used!
 - BDT configuration is fully done via JSON
 - Support **Binary** (signal/background) and **MultiClass** (prompt/nonprompt/background)
 - Support **Centrality** and p_T dependent BDT selections
- Try it out and give feedback!





Back up