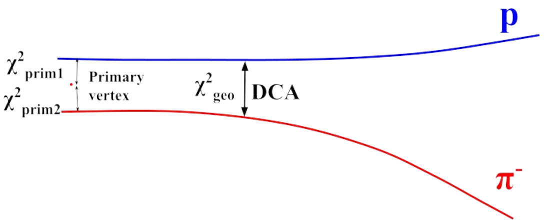
CBM Performance for Λ Yield Analysis for CFV and Day1 Setup

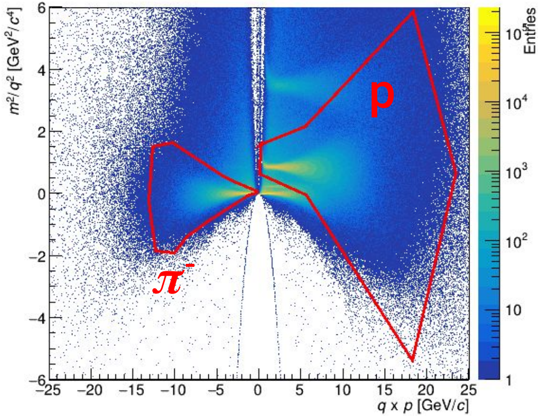
46th CBM Collaboration Meeting
October 20, 2025, Lanzhou, China

Axel Puntke

Reconstruction using PFSimple

- Default PID from PID-Framework
- Very loose post cuts (signal selection will be done with XGBoost later)
- For Day1 and CFV:
    - 4M QA events





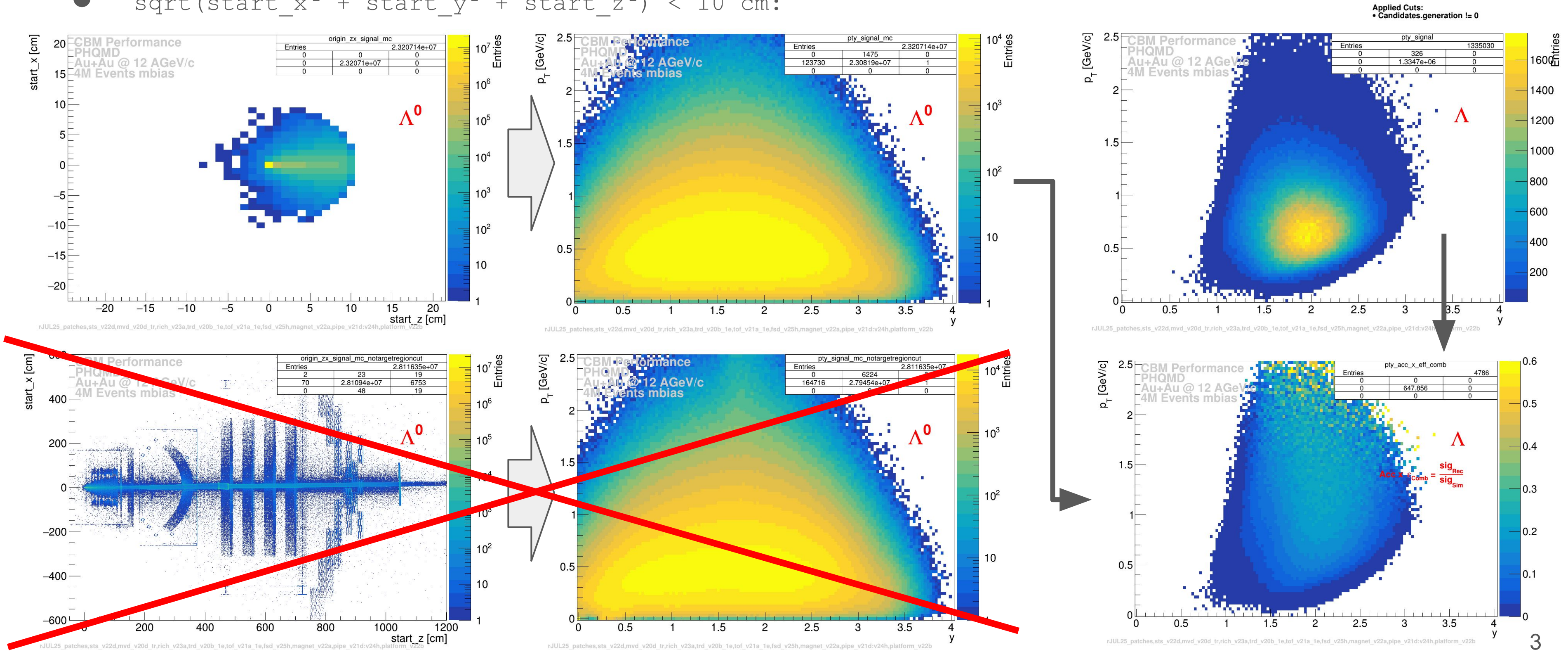
Input TOF m^2 plot for Day1 Setup

```
{
    "io" : {
        "input_treename" : "aTree",
        "rectracks_branchname" : "RecParticles",
        "n_events" : -1,
        "save_options" : []
    },
    "pid_mode" : 2, // PID-Framework PID (default)
    "decays" : [
        {
            "mother" : {
                "name" : "Lambda",
                "pdg_code" : 3122,
                "cuts2" : {"dist" : 100.0, "chi2geo" : 1000.0}
            },
            "daughters" : [
                {
                    "pdg_code" : [-211]
                },
                {
                    "pdg_code" : [2212]
                }
            ]
        }
    ],
    "output_cuts" : [
        {"var" : "mass", "from" : 1.07, "to" : 1.2},
        {"var" : "x", "from" : -50, "to" : 50},
        {"var" : "y", "from" : -50, "to" : 50},
        {"var" : "distance", "from" : 0, "to" : 100},
        {"var" : "eta", "from" : 1, "to" : 6.5},
        {"var" : "chi2_topo", "from" : 0, "to" : 100000},
        {"var" : "chi2_geo", "from" : 0, "to" : 1000}
    ]
}
```

used JSON Config

Definition of Acc x ε_{Comb}

- sqrt(start_x^2 + start_y^2 + start_z^2) < 10 cm:

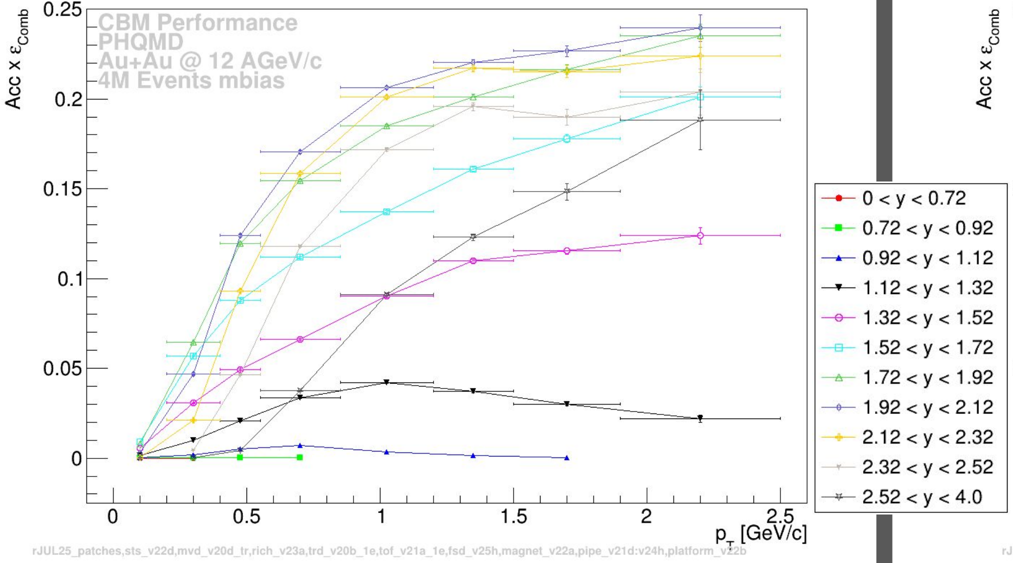


Taking all MC-true Λs leads to unphysical y distributions

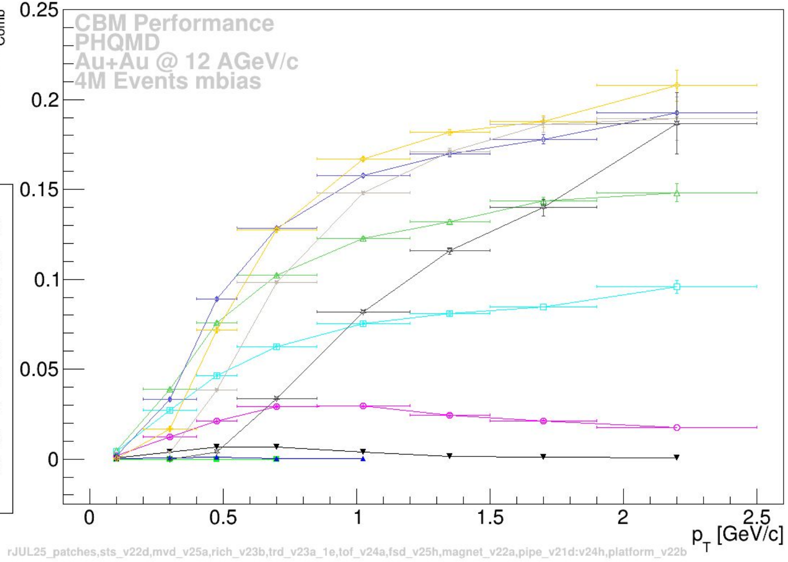
Determine Acc x ε_{Comb} for each p_T-y Bin

- Includes detector acceptance & efficiency for branching ratio and reconstruction

Day1



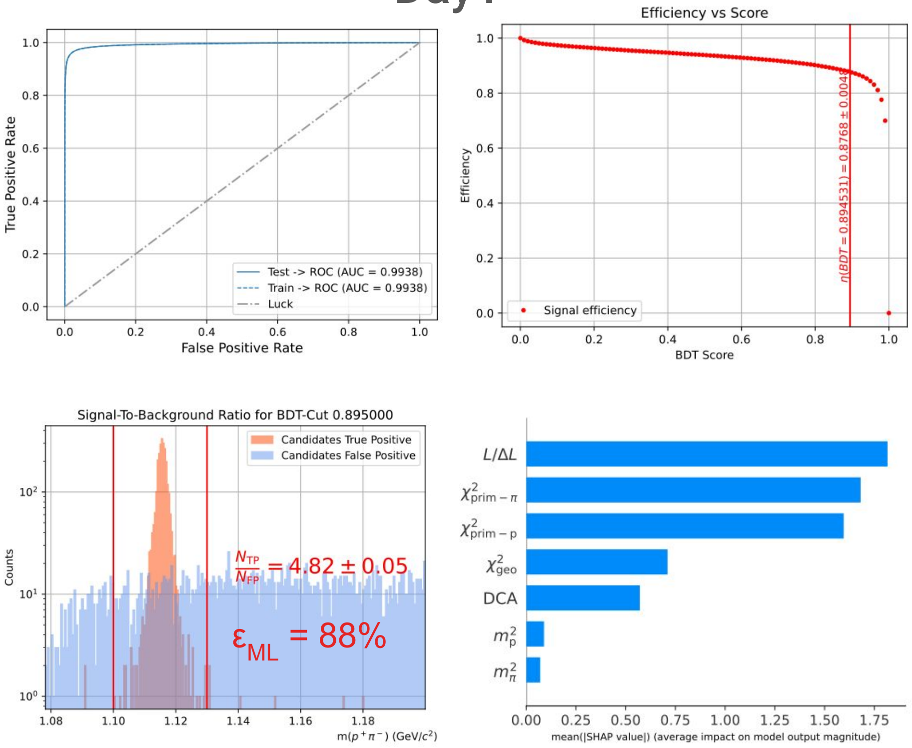
CFV



XGBoost Model Comparison

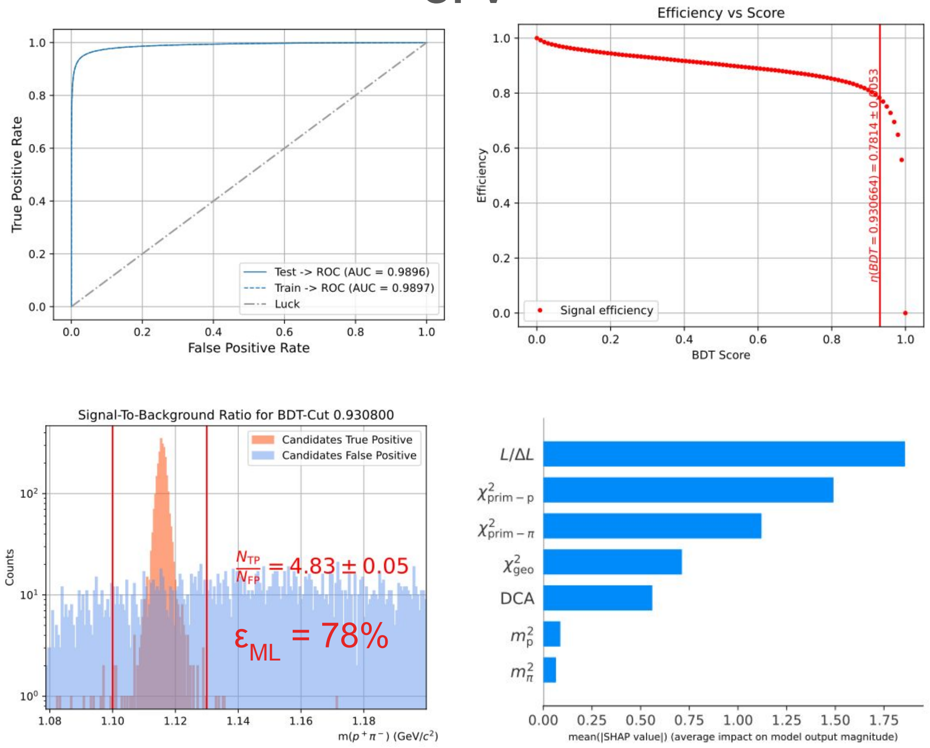
This are still models trained with JUL24 common productions data because statistics of JUL25 is not sufficient yet

Day1



Model 0028

CFV

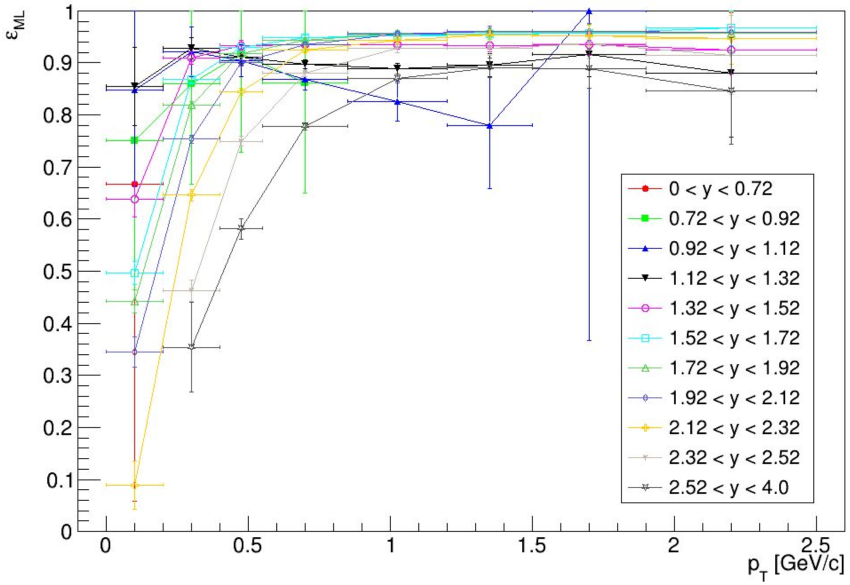


Model 0027

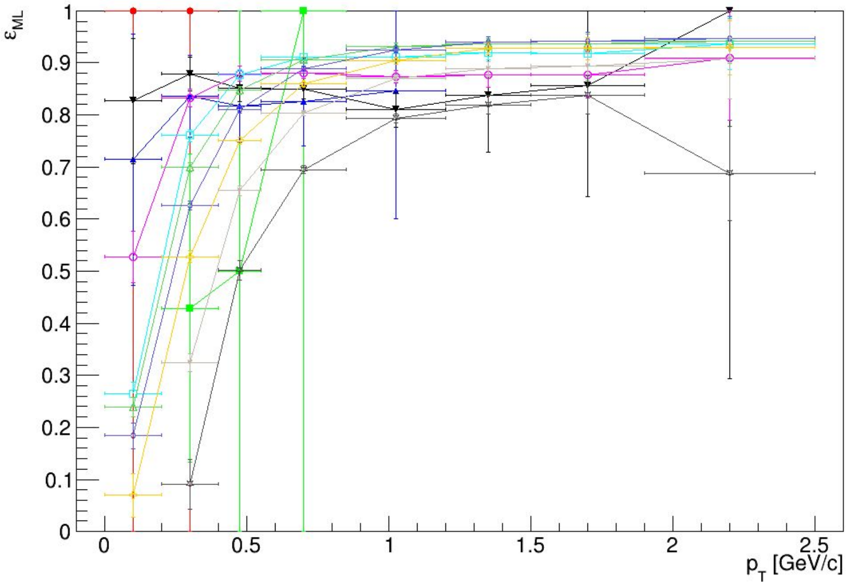
Efficiency of XGBoost Model per p_T-y Bin

- Shows larger stat. uncertainties at p_T acceptance border
  - Issue will be investigated with new models soon

Day1



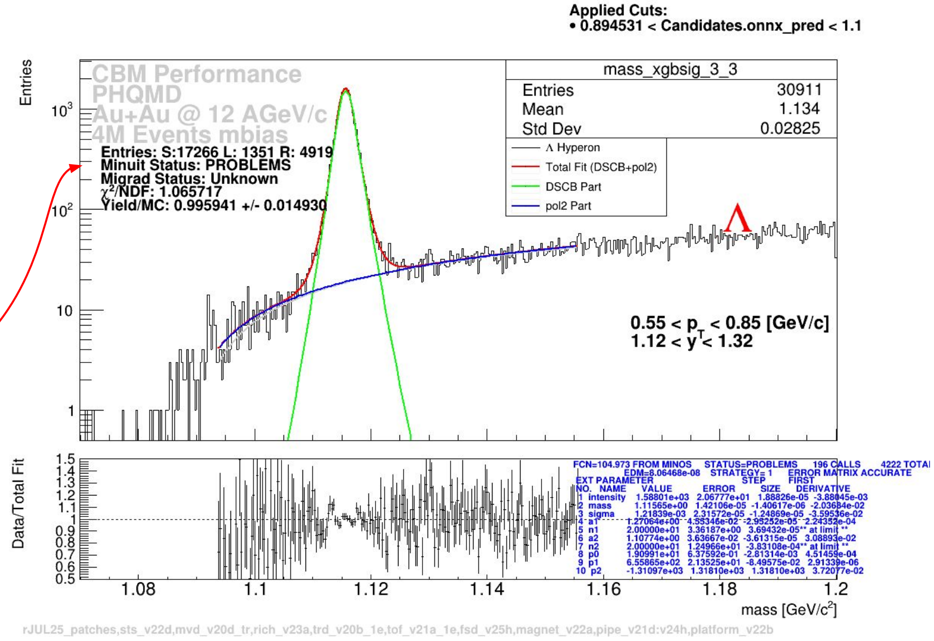
CFV



Yield Extraction Performance

- Yield is extracted by fitting the inv. mass distribution using DSCB function + 2nd order polynomial (downward-opening)
  - Bin exclusion criteria softened since last report in Physics Forum due to finer binning and therefore lower statistics (e.g. accepting fits with PROBLEMS status)

Example Fit:

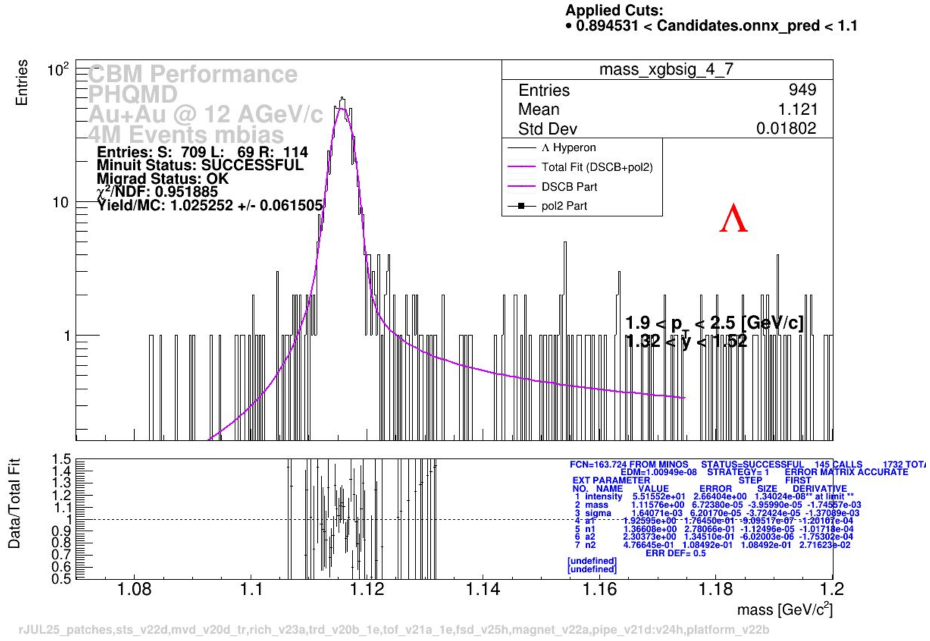


(Day1 Setup)

Yield Extraction Performance

Example Fit:

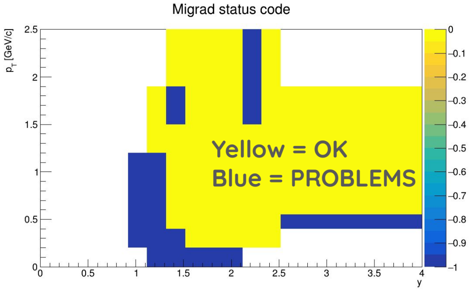
- For bins with very low statistics I also now try to fit DSCB only without polynomial background
  - Fits are successful
  - Yield/MC acceptable

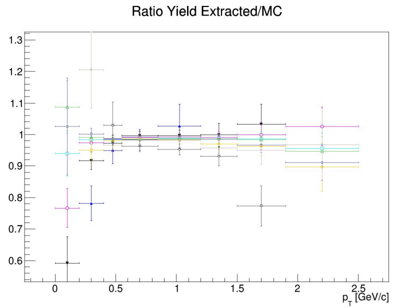


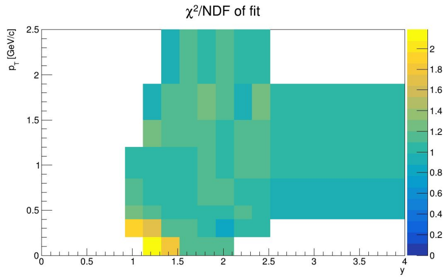
(Day1 Setup)

Yield Extraction Performance

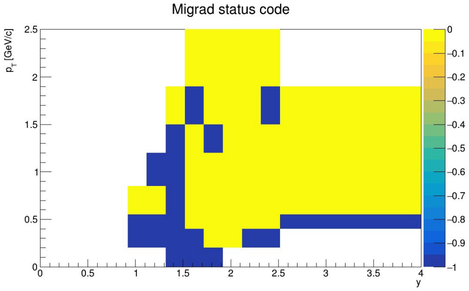
Day1

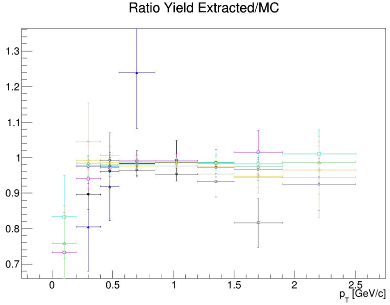


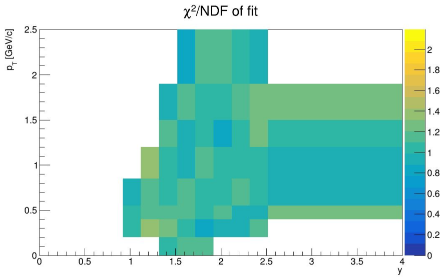




CFV



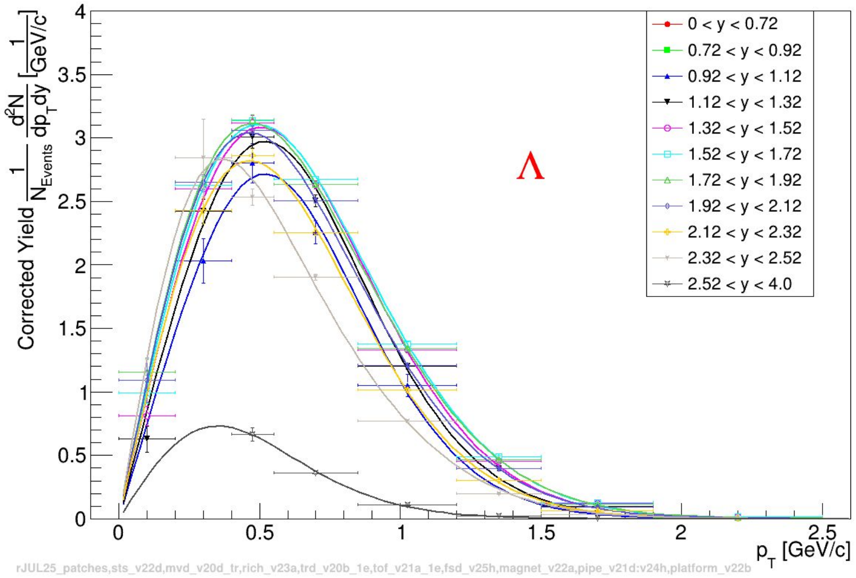




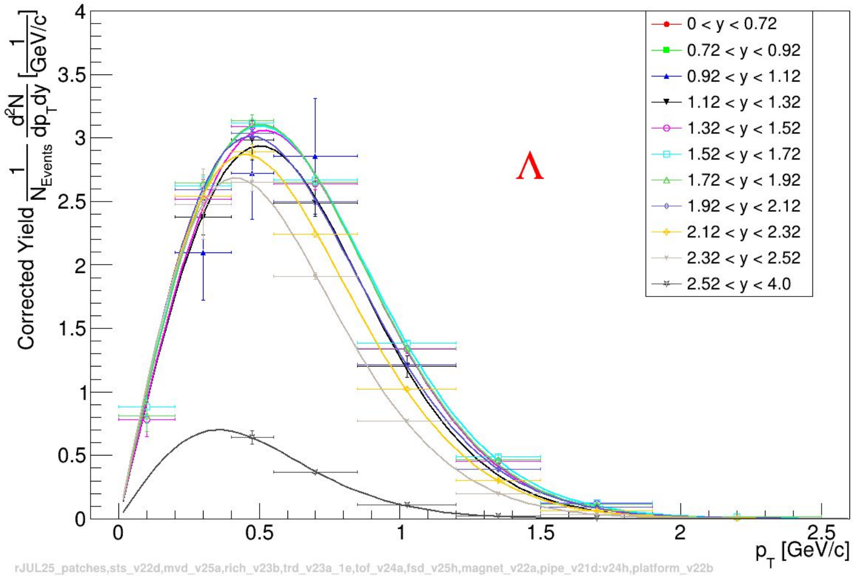
Corrected Yield

- Raw yield corrected by Acc x ε_{Comb} x ε_{ML} and normalized by bin area
- Will be fitted with blastwave function later

Day1



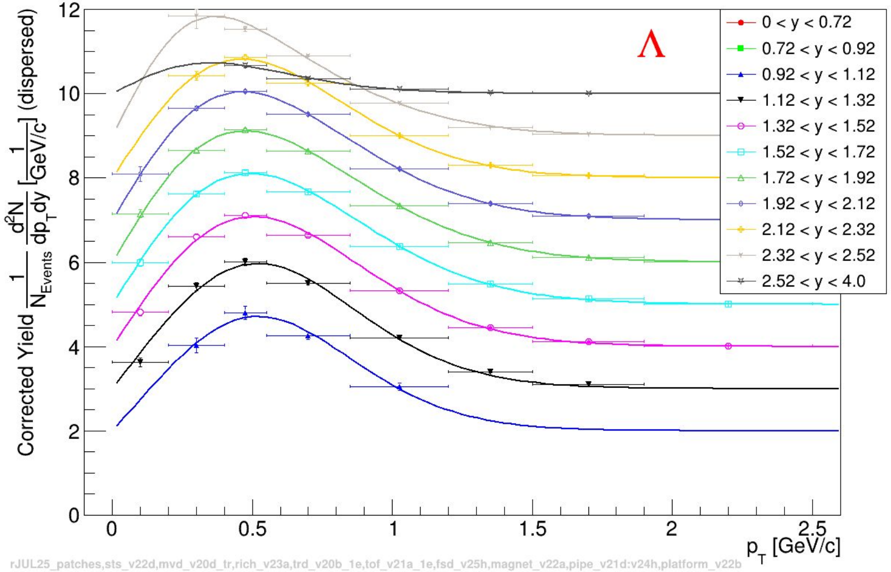
CFV



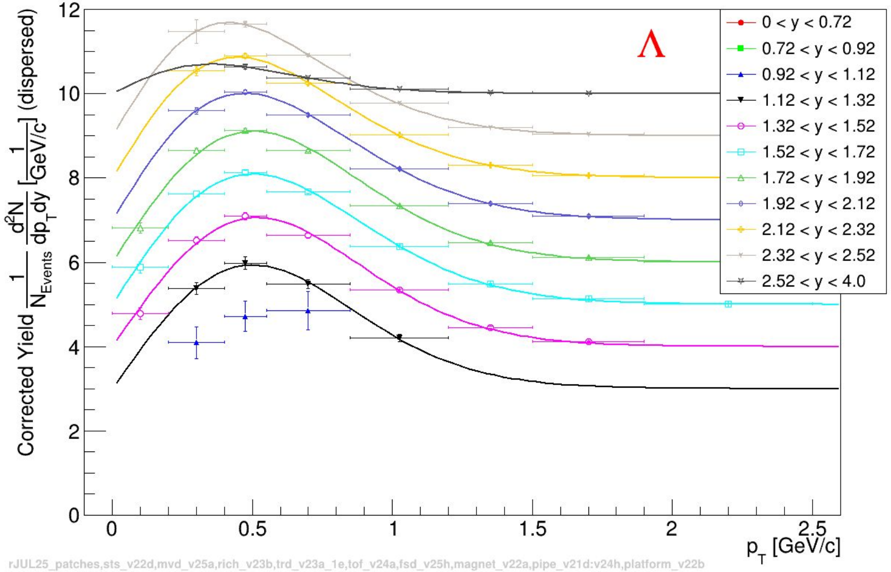
Corrected Yield (Dispersed)

- Raw yield corrected by Acc x ε_{Comb} x ε_{ML} and normalized by bin area
- Fitted with blastwave function

Day1



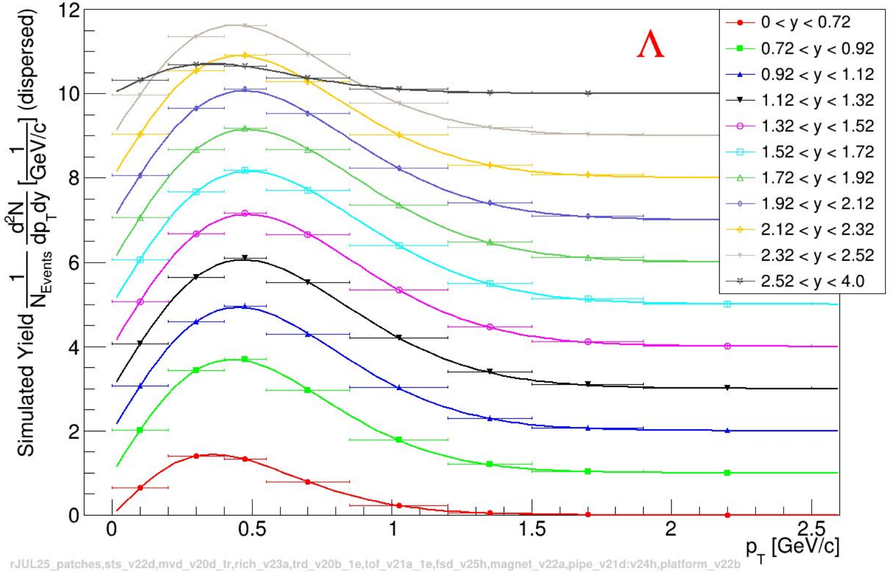
CFV



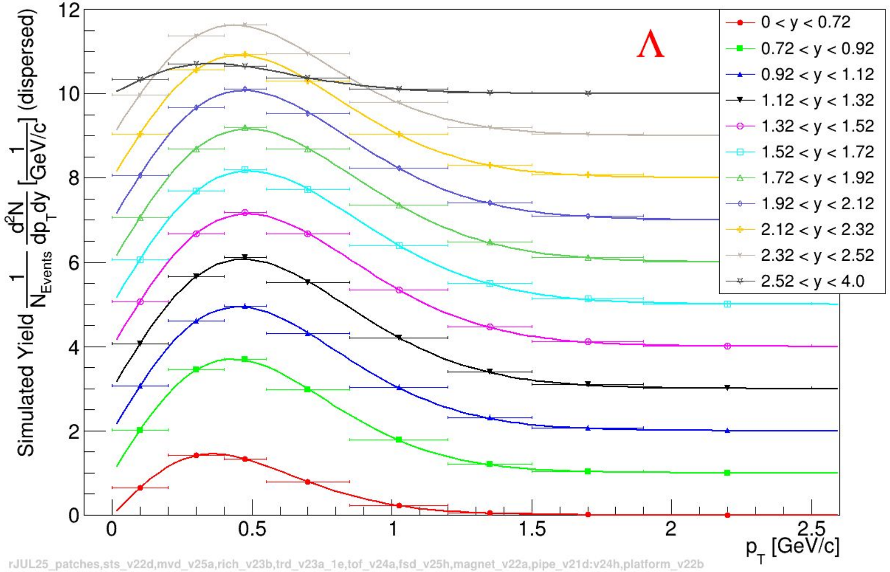
Simulated Yield

- All Λs from SimParticles branch, normalized by bin area

Day1



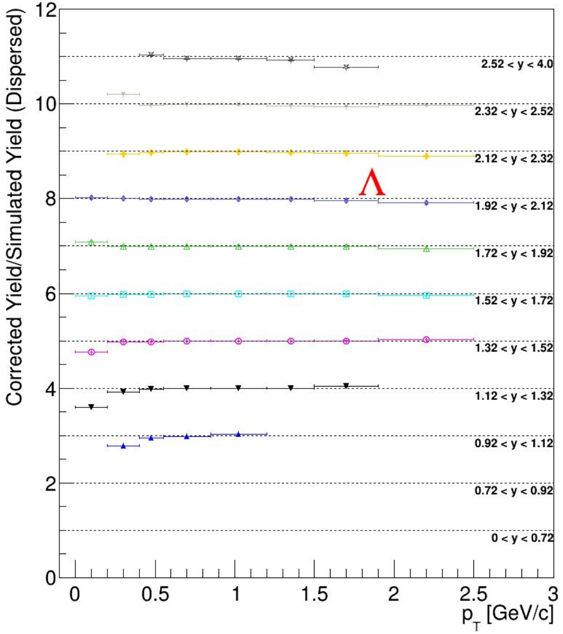
CFV



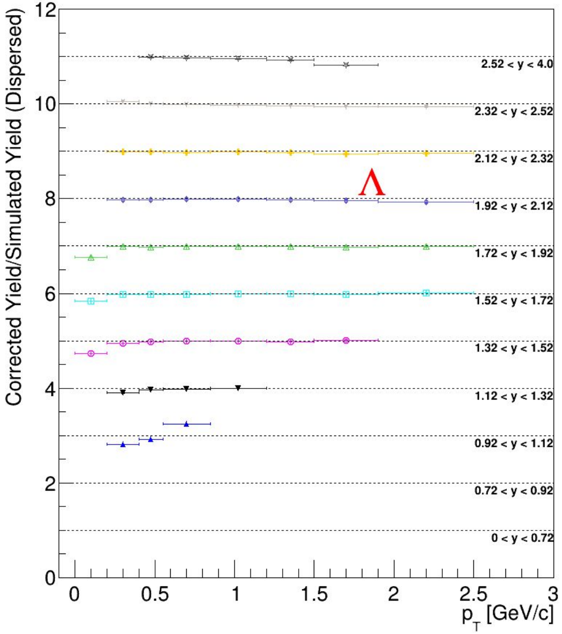
Corrected Yield/Simulated Yield

- Similar performance for both setups, differences probably originate from yield extraction method

Day1



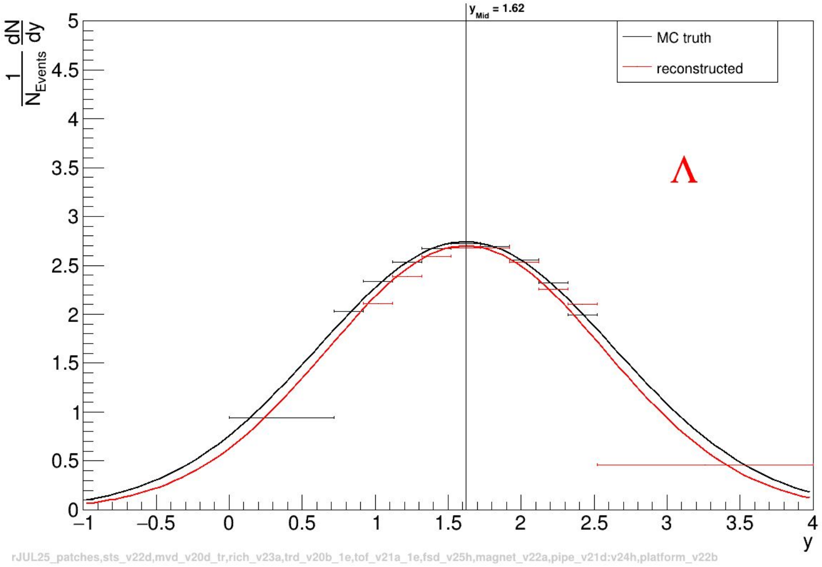
CFV



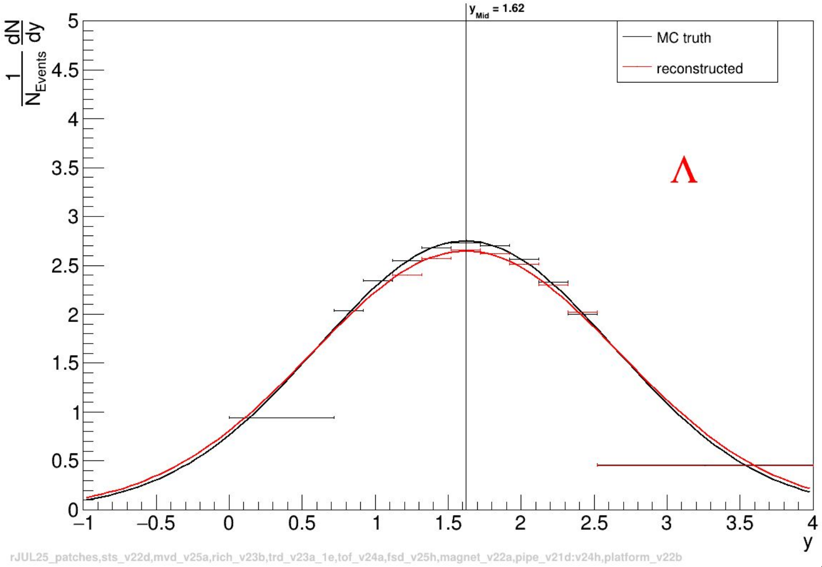
Integrated Yield

- Integrated blastwave fits, fitted with gaussian with fixed mean y_{Mid} = 1.62

Day1



CFV



QA Framework

- AnalysisTreeQA based
- Used to produce the QA plots on large data sets (runs fully on GSI cluster)
- Includes merging and postprocessing (binwise operations), also on virgo
- Tool is ready-to-use, download at https://git.cbm.gsi.de/apuntke/basic-geo-qa
- Documentation is also available
- Is already used by F. Fidorra for K^0_S

Name	Last commit	Last update
resources	initial commit	3 days ago
sh	initial commit	3 days ago
Readme.md	update Readme.md	2 days ago
config.json	initial commit	3 days ago
geo_qa.py	initial commit	3 days ago
plot_list.json	initial commit	3 days ago

Readme.md

Basic Geo QA

Introduction

This script was developed to perform the computation of QA plots of various CBM geometries in parallel on the virgo cluster using the AnalysisTreeQA package and automatically merge the resulting root files. The plots are also renamed to a user-defined name specified in the config file and ordered by their appearance in the plot list. Key features are the definition of complex variables which are automatically converted to lambda functions for AnalysisTreeQA as well as the counting of particles/geant process ids/etc. and automatical generation of a counting histograms.

Initial Settings

In the config.json it is possible to define some basic parameters:

Key	Description
`rootinstall`	Path to the CbmRootConfig.sh of the root installation to be used to compute the plots
`outdirbase`	Base directory in which the intermediate and final files and subdirectories of a geometry to be QAed are stored
`slurm`	JSON objects containing information for the slurm jobs to be spawned, separate for the jobs for the AnalysisTreeQA jobs and the final merge job

Example Definitions

For some plots shown in this presentation

General Analysis Definition

```json
{
    "name" : "lambdaanalysis_phqmd12agev_day1",
    "inputpath" : "/lustre/cbm/users/apuntke/machinelearning/data/pfsimple-various-particles/phqmd_day1/12agev/mbias",
    "infile_format_string" : "${inputpath}/filelist${SLURM_ARRAY_TASK_ID}/PFSimpleOutput.root",
    "friend_filelist_format_string" : "${inputpath}/filelist${SLURM_ARRAY_TASK_ID}/filelist_a.txt",
    "tree_name" : "pTree",
    "custom_tasks" : ["AddCustomVariablesTask", "AddTofMass2InfoToCandidatesTask", "AddOnnxModelPredictionTask"],
    "onnx_model" : "/lustre/cbm/users/apuntke/onnxtestmodel/0028.onnx",
    "mdbins" : {
        "pT" : [0, 0.2, 0.4, 0.55, 0.85, 1.2, 1.5, 1.9, 2.5],
        "rapidity" : [0, 0.72, 0.92, 1.12, 1.32, 1.52, 1.62, 1.72, 1.92, 2.12, 2.32, 2.52, 4.0],
        "centrality" : [0, 100]
    },
    "constants" : {
        "C_BDT_CUT" : 0.894531,
        "C_N_EVENTS" : 3921000
    },
    "array_ids" : "1201-2000",
    "plot_list" : "plot_list_lambdaanalysisqa.json",
    "data_info" : ["CBM Performance", "PHQMD", "Au+Au @ 12 AGeV/c", "4M Events mbias"],
    "setup_info" : "rJUL24_patches,sts_v22d,mvd_v20d_tr,rich_v23a,trd_v20b_1e,tof_v21a_1e,fsd_v25h,magnet_v22a,pipe_v21d:v24h,platform_v22b"
},
```

DSCB Fit Plot

```json
#foreach multidiffbin(rapidity,pT)
    {
        "name": "mass_xgbsig_${BINID}",
        "type": "H1",
        "atqa_cuts": [
            {
                "type": "rangecut",
                "cuton": "Candidates.onnx_pred",
                "from": ${C_BDT_CUT},
                "to": 1.1
            },
            {
                "type": "rangecut",
                "cuton": "Candidates.pT",
                "from": ${BIN_PT_FROM},
                "to": ${BIN_PT_TO}
            },
            {
                "type": "rangecut",
                "cuton": "Candidates.rapidity",
                "from": ${BIN_RAPIDITY_FROM},
                "to": ${BIN_RAPIDITY_TO}
            }
        ],
        "xaxis": {
          "variable": "Candidates.mass",
          "from": 1.07,
          "to": 1.2,
          "binwidth": 0.0001,
          "title": "mass [GeV/c^{2}]"
        },
        "processing_flags" : ["fit_signal_bg"],
        "canvas_style" : {
            "keywords" : [],
            "size" : {"height" : "880", "width" : "1250"},
            "labels" : [
                    {"text" : ["#Lambda"], "color" : "kRed", "x" : 0.77, "y" : 0.66, "fontsize" : 0.07},
                    {"text" : ["Applied Cuts:", "#bullet ${C_BDT_CUT} < Candidates.onnx_pred < 1.1"], "color" : "kBlack", "x" : 0.6, "y" : 0.98,
                    {"text" : ["${BIN_PT_FROM} < p_{T} < ${BIN_PT_TO} [GeV/c]", "${BIN_RAPIDITY_FROM} < y < ${BIN_RAPIDITY_TO}"], "color" : "kBla
            ],
            "legend" : {"x1": 0.55, "x2" : 0.72, "y1" : 0.66, "y2" : 0.78, "entries" : [
                    {"object" : "", "label" : "#Lambda Hyperon"},
                    {"object" : "totalfit", "label" : "Total Fit (DSCB+pol2)"},
                    {"object" : "signalfunc", "label" : "DSCB Part"},
                    {"object" : "bgfunc", "label" : "pol2 Part"}
            ]}
        }
    }
#endfor
```

2D Acc x ε_{Comb} Plot

```json
{
    "name": "pty_acc_x_eff_comb",
    "type": "H2",
    "content" : {
        "type" : "binwise_operation",
        "value_expression" : "pty_signal/pty_signal_mc",
        "error_expression" : "(pty_signal/pty_signal_mc) * sqrt(1.0/pty_si
    },
    "xaxis": {
      "mdbins": "rapidity",
      "title": "y"
    },
    "yaxis": {
      "mdbins": "pT",
      "title": "p_{T} [GeV/c]"
    },
    "canvas_style" : {
        "keywords" : [],
        "size" : {"height" : "880", "width" : "1250"},
        "labels" : [
                {"text" : ["#Lambda"], "color" : "kRed", "x" : 0.77, "y" :
                {"text" : ["Acc x #varepsilon_{Comb} = #frac{sig_{Rec}}{si
        ]
    }
},
```

1D Acc x ε_{Comb} Plot

```json
        {
            "name": "pt_acc_x_eff_comb",
            "type": "MultiGraph",
            "content" : {
                "type" : "sliced_from_2dhist",
                "source_2dhist" : "pty_acc_x_eff_comb"
            },
            "xaxis": {
              "from": -0.1,
              "to": 2.6,
              "title": "p_{T} [GeV/c]"
            },
            "yaxis": {
              "from": 0,
              "#to": 0.44,
              "to": 0.5,
              "title": "Acc x #varepsilon_{Comb}"
            },
            "canvas_style" : {
                "keywords" : [],
                "size" : {"height" : "880", "width" : "1250"},
                "labels" : [
                        {"text" : ["#Lambda"], "color" : "kRed", "x" : 0.77, "y" : 0.66, "fontsize" : 0.
                ],
                "legend" : {"x1": 0.72, "x2" : 0.9, "y1" : 0.7, "y2" : 0.9, "entries" : [
#foreach multidiffbin(rapidity)
                        {"object" : "xslice_${BINID}", "label" : "${BIN_RAPIDITY_FROM} < y < ${BIN_RAPID
#endfor
                ]}
            }
        },
```

Summary

- Efficiencies (Acc x ε_{Comb}, ε_{ML}) slightly lowered in CFV
- Multi-differential Λ reconstruction chain in CBM is well established
  - MC closure achieved
  - Minor problems need to be fixed
- Analysis is flexible to configure, can easily be adapted for other particles

Outlook

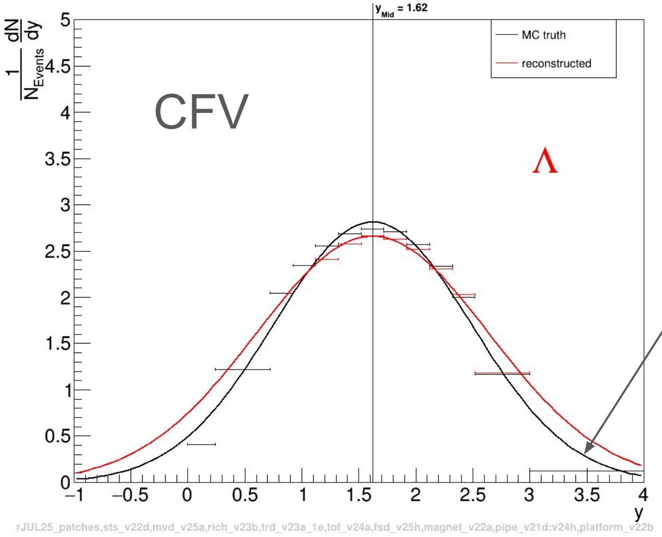
- Train new models when enough statistics is available (9M events minimum)
- Systematic uncertainties will be estimated soon
  - Track selection/matching
  - BDT cut
  - Blast wave fits
- Planned CBM common productions (100M events) will allow finer binning and centrality-dependent analysis

Thank you for your Attention

Backup

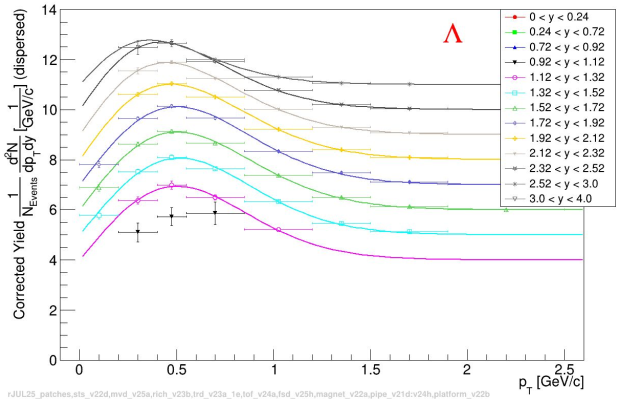
Trying different Binning to improve Integrated Yield Fit

Split outermost bins in two → 2 additional bins:

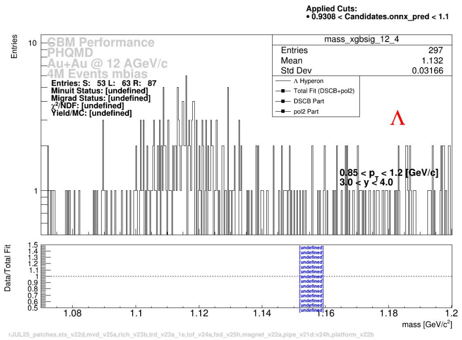


No reconstructed yield data point

No data points for blastwave fit for 3.0 < y < 4.0:



Statistics in inv. mass plot not sufficient (other p_T bins with even less statistics):



Analysis Outline

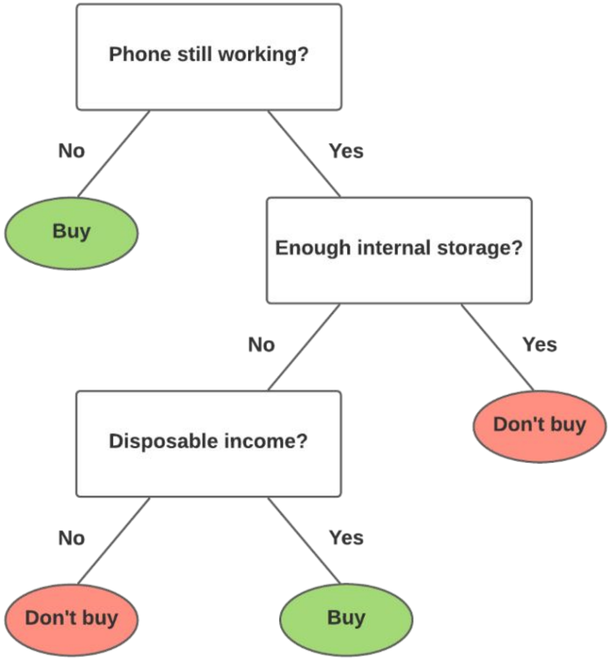
1. Reconstruction of Λ candidates via PFSimple
2. Selection of candidates using XGBoost model
3. Multi-Differential yield extraction from invariant mass spectrum
4. Computation of efficiencies of reconstruction and ML selection
5. Correction of measured Λ yields
6. Extrapolation of yields to low and high p_T regions using blastwave fit
7. Comparison of integrated yields dN/dy with MC-truth (MC-closure)

Steps 2-7 are done using a newly developed analysis framework which allows to easily change parameters and run the same analysis for other particles than Λ

How to classify candidates with XGBoost

Decision tree

- Predictive model to go from observations of an item to conclusions about an item's target value
- Target value can have either continuous values (= regression trees) or discrete values (=classification trees)
- Goal: predict target value based on several input variables
  - e.g. on the right: decision to buy or not buy a new phone using 3 input variables

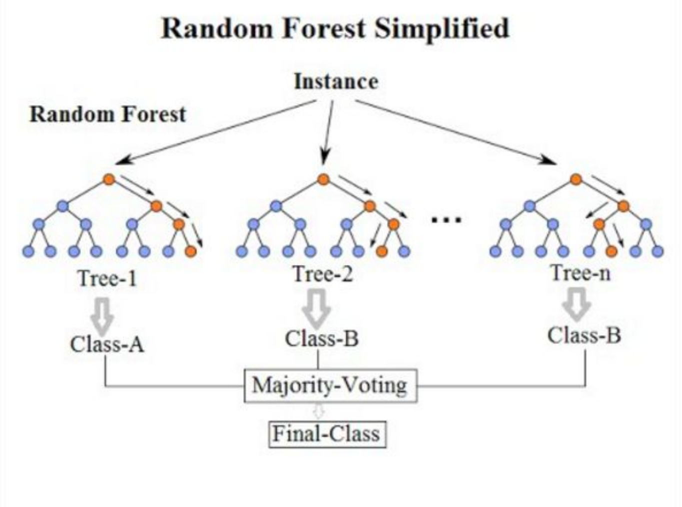


https://towardsdatascience.com/decision-trees-random-forests-and-gradient-boosting-whats-the-difference-ae435cbb67ad

How to classify candidates with XGBoost

Random Forest

- Ensemble of decision trees which predict the items target value independently
- Each tree is independently trained with a random selection of the training data set as a weak learner
    - parallelization possible
- Majority-voting decides the final class of the target item (or in case of a regression task, the average prediction of target value is returned)



https://en.wikipedia.org/wiki/Random_forest#/media/File:Random_forest_diagram_complete.png

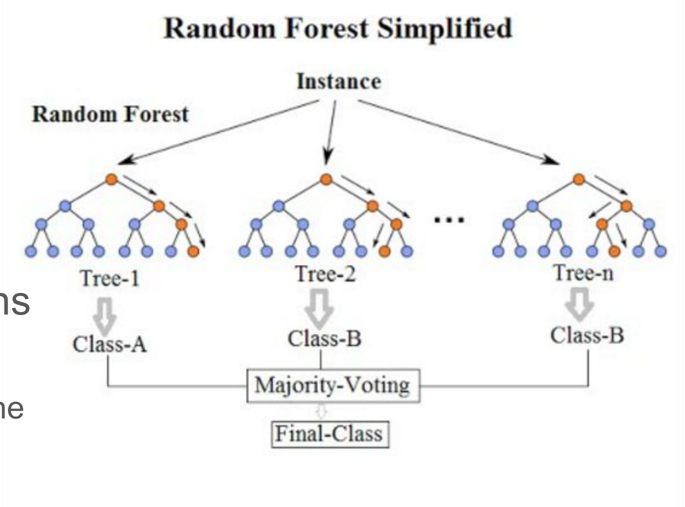
How to classify candidates with XGBoost

Gradient Boosting

- also an ensemble of decision trees (like random forest)

Difference to a random forest lies in the training:

- at each iteration, more weight is added to the observations with the worst prediction of the previous iteration
  - Idea: Improve the results of the previous iteration by focusing on the observations that were far away from the truth
  - computation cannot be parallelized (apart from parallelism in each tree)



https://en.wikipedia.org/wiki/Random_forest#/media/File:Random_forest_diagram_complete.png

➔ Can improve the bias (better accuracy) compared to a random forest (usually outperforms RF)

How to classify candidates with XGBoost

XGBoost

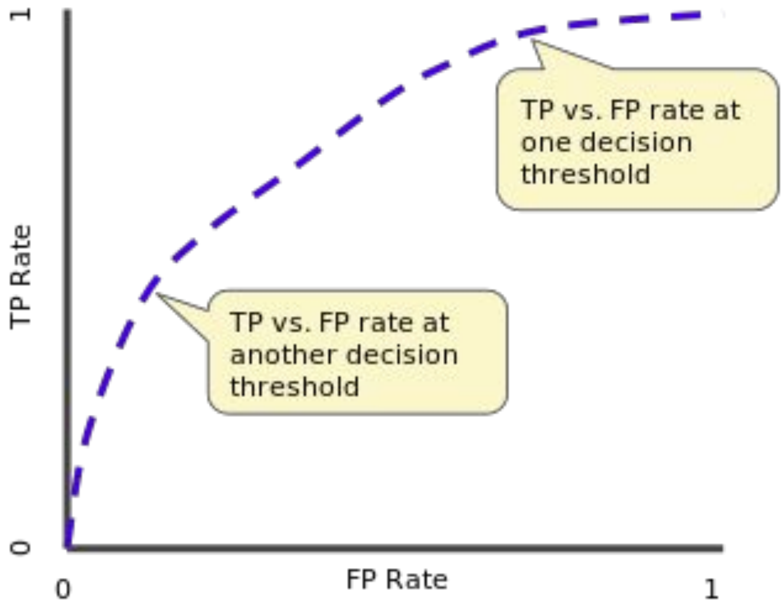
- optimized gradient boosting library
- highly efficient, flexible and portable
- provides parallel tree boosting
    - can run on the major distributed environments like MPI
- hipe4ml uses it through its ModelHandler class:

```
import xgboost as xgb
from hipe4ml.model_handler import ModelHandler

model_clf = xgb.XGBClassifier()
model_hdl = ModelHandler(model_clf, features_for_train)
```

Receiver Operating Characteristic (ROC) curve

- Graph showing the performance of a classification model at all classification thresholds
- Plots true positive rate vs. false positive rate
- In the optimal case, TPR should increase without an FPR increase



$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

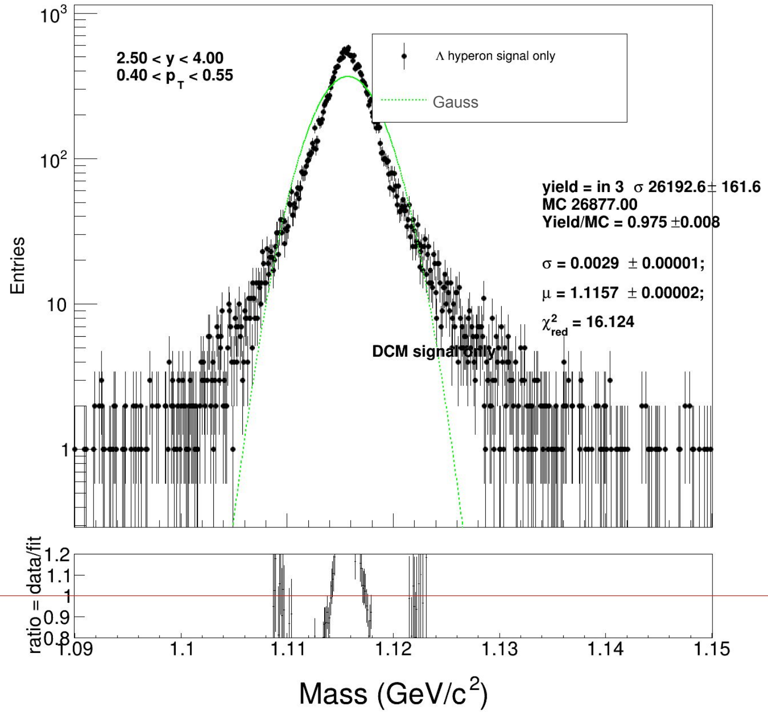
https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Fit with Gauss

- Fitted in 3 σ around Λ mass with σ=RMS
- Boundary conditions:

```
fitter_mc_signal_gauss->SetBoundGaussianMean(1.11567,1.113,1.119);
fitter_mc_signal_gauss->SetBoundGaussianSigma(0.0012,2);
```

Bin 5,2



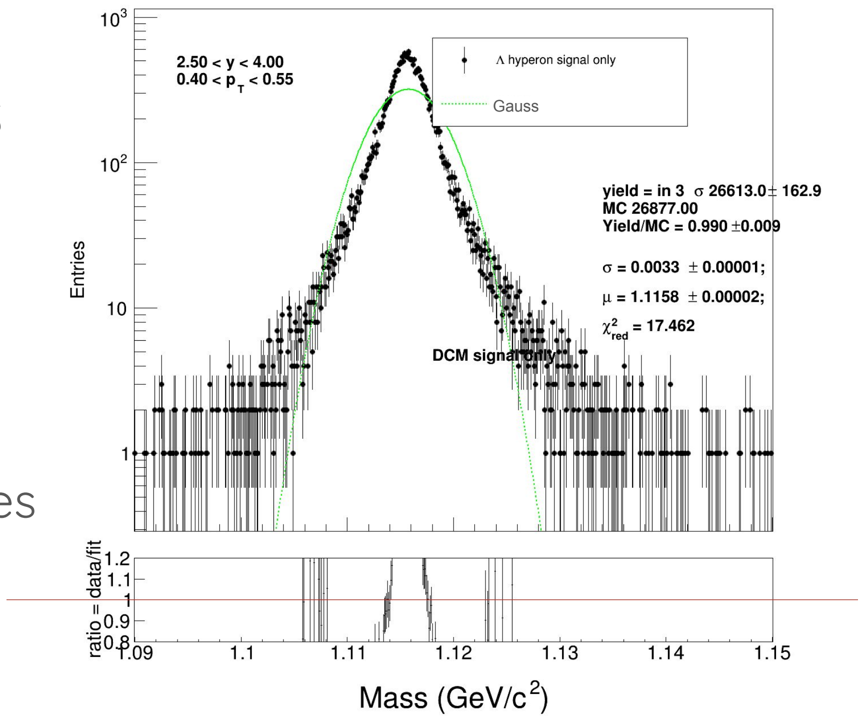
Fit with Gauss

- Fitted in 5 σ around Λ mass with σ=RMS
- Boundary conditions:

```
fitter_mc_signal_gauss->SetBoundGaussianMean(1.11567,1.113,1.119);
fitter_mc_signal_gauss->SetBoundGaussianSigma(0.0012,2);
```

- Yield/MC improves, but function still does not match the distribution optically

Bin 5,2



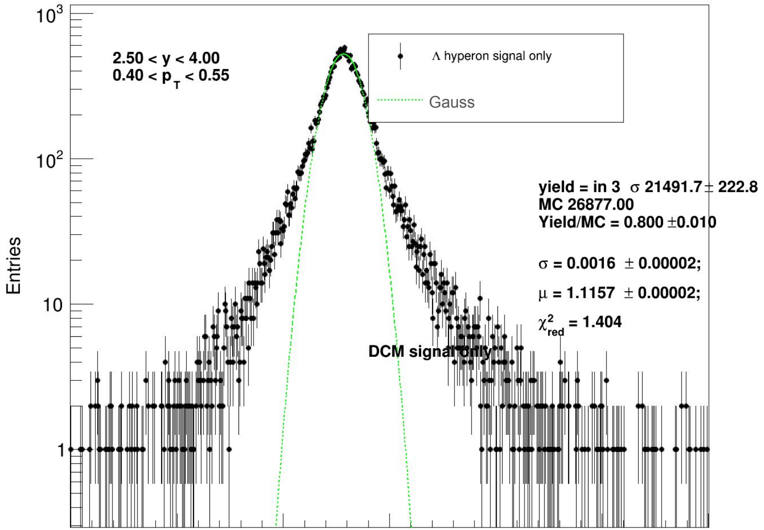
Fit with Gauss

- Fitted from 1.11331 to 1.11789 GeV/c^2
  - Manually found using ROOTs fit panel
- Boundary conditions:

```
fitter_mc_signal_gauss->SetBoundGaussianMean(1.11567,1.113,1.119);
fitter_mc_signal_gauss->SetBoundGaussianSigma(0.0012,2);
```

- Function optically matches better, but yield/MC is worst

Bin 5,2



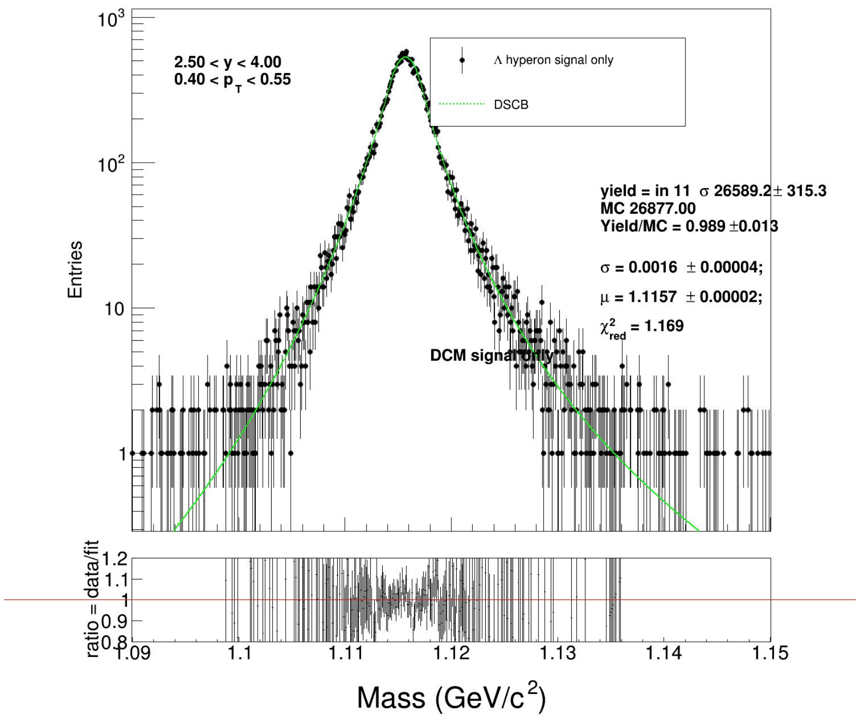
Fit with DSCB

- Fitted in 3 σ around Λ mass with σ=RMS
- Boundary conditions:

```
fitter_mc_signal_DSCB->SetBoundGaussianMean(1.11567,1.113,1.119);
fitter_mc_signal_DSCB->SetBoundGaussianSigma(0.0012,2);
fitter_mc_signal_DSCB->SetBoundDSCBa1(1,0,10);
fitter_mc_signal_DSCB->SetBoundDSCBn1(1,0,100);
fitter_mc_signal_DSCB->SetBoundDSCBa2(1,0,10);
fitter_mc_signal_DSCB->SetBoundDSCBn2(1,0,100);
```

initial value min max

Bin 5,2



Blast-Wave Function

- p_T spectras in heavy-ion collisions can be described by a superposition of thermal sources which move with a transverse velocity

$$\beta(\hat{r}) = \underline{\beta_s}\,\hat{r}^{\underline{n}}$$

□$_s$: Velocity at the surface of the fireball
$\hat{r}$: Radial coordinate ($\hat{r}$ = r/R)
R: Maximum radial distance

- Functional form is given by

$$\frac{1}{2\pi p_T}\frac{dN}{dp_T dy} \underline{\propto} m_T \int_0^1 \hat{r}\,d\hat{r}\, I_0\left(\frac{p_T \sinh\rho(\hat{r})}{\underline{T}}\right) K_1\left(\frac{m_T \cosh\rho(\hat{r})}{\underline{T}}\right)$$

m_T: Transverse mass $m_T = \sqrt{p_T^2 + m^2}$
ρ: Transverse rapidity $\rho = \text{arctanh}\,\beta$
T: Kinetic freeze-out temperature

I_0, K_1: Modified Bessel Functions

➔ 4 fit parameters: T, □$_s$, n and normalization parameter