Machine Learning application for Λ hyperon reconstruction in CBM at FAIR

Shahid Khan*, Olha Lavoryk, Oleksii Lubynets, Viktor Klochkov, Andrea Dubla, Ilya Selyuzhenkov

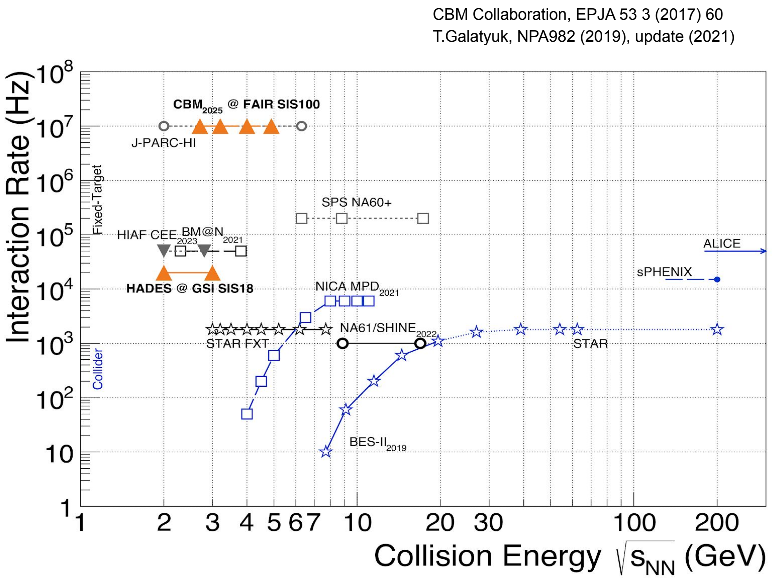
for the CBM Collaboration

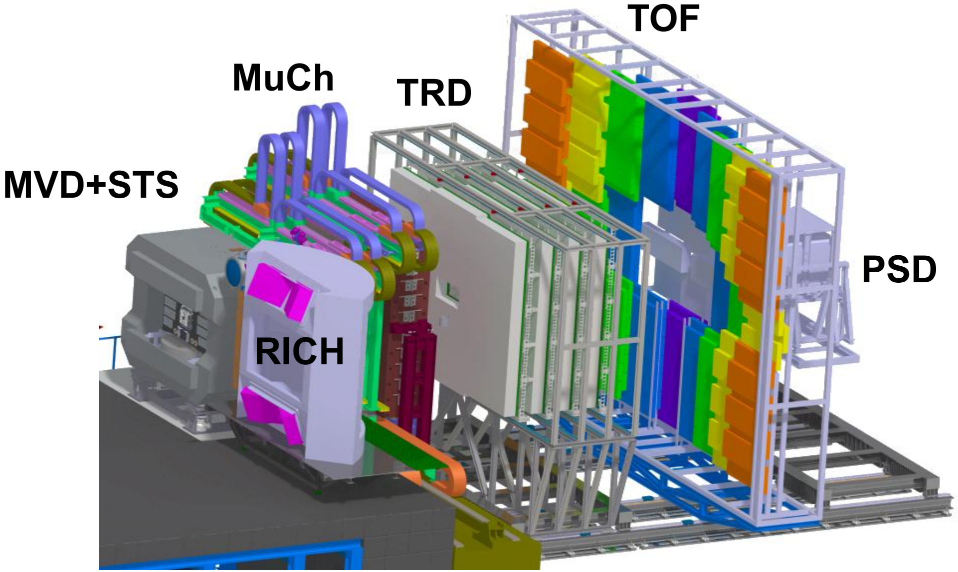
*shahid.khan@uni-tuebingen.de

MPD at NICA
25 May 2021

EBERHARD KARLS UNIVERSITÄT TÜBINGEN
GSI
GOETHE UNIVERSITÄT FRANKFURT AM MAIN
MEPhI National Research Nuclear University

CBM physics goals and experimental challenges





Main CBM physics cases:

- QCD matter equation-of-state at large baryon densities
- The production of strange quarks is sensitive to the properties of created matter in high energy nuclear collisions
  - (Multi)-Strange particles
- Extend nuclei chart with hypernuclei measurements

- Tracking: Micro-Vertex Detector (MVD), Silicon Tracking System (STS)
- Particle identification: Muon Chamber (MuCh), Ring Imaging Cherenkov (RICH), Transition Radiation Detector (TRD), Time of Flight (TOF)
- Collision geometry: Projectile Spectator Detector (PSD)

To study rare probes CBM will operate at an unprecedented interaction rate, up to 10 MHz!

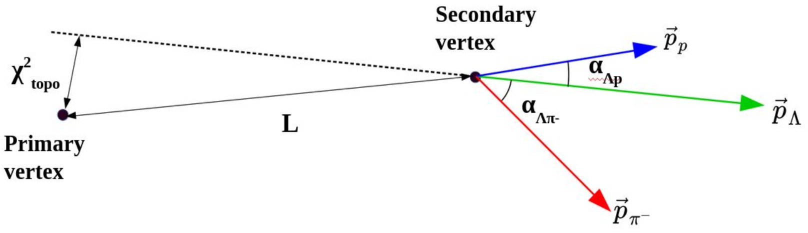

FAIR
CBM
cave
Full video

(Multi)-Strange reconstruction via weak decays

- Λ hyperons are the most abundant strange baryons produced at FAIR energies
- Collisions generated by URQMD and DCM-QGSM-SMM with Au+Au collisions at p_{beam} = 12A GeV/c ($\sqrt{s_{NN}}$ = 4.93), mbias, 100k
- Using GEANT4 simulation, CA tracking, KFParticle within CbmRoot framework

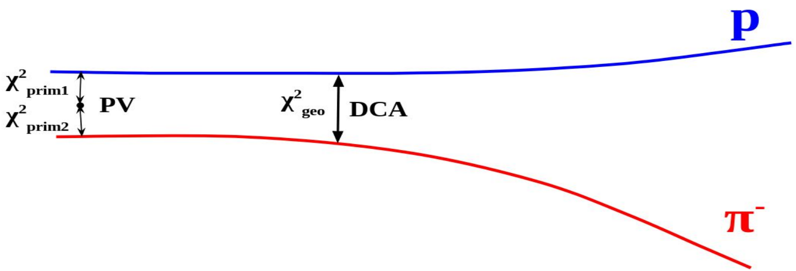
Λ candidates reconstruction: $\Lambda^0 \rightarrow p^+\pi^-$

- Combine all proton and pion tracks
- Signal from a lambda decay
- Combinatorial background



Variables :

- χ^2_{prim} - squared distance between the daughter track and the primary vertex divided by its Covariance Matrix (CV)
- **DCA** - distance of closest approach between proton & pion tracks
- χ^2_{geo} - squared distance between daughter tracks divided by CV
- **L/ΔL** - distance between primary and secondary vertex divided by CV
- **cos** $\alpha_{p\Lambda}$, **cos** $\alpha_{\Lambda\pi}$, χ^2_{topo} (future investigation)



Selection criteria are optimized multi-dimensionally, non-linearly and in an automatized way with Machine Learning algorithms

Two implementations of KFParticle based reconstruction

KFParticleFinder

Software design driven by requirement of very fast online particle selection to fit CBM operation at the 10^7 interaction rate

Implementation:

- vectorized libraries (SIMD technic)
- Fixed set of decays selection criteria
- all-in-one approach:
  >150 decay channels
  different decay topologies, etc.

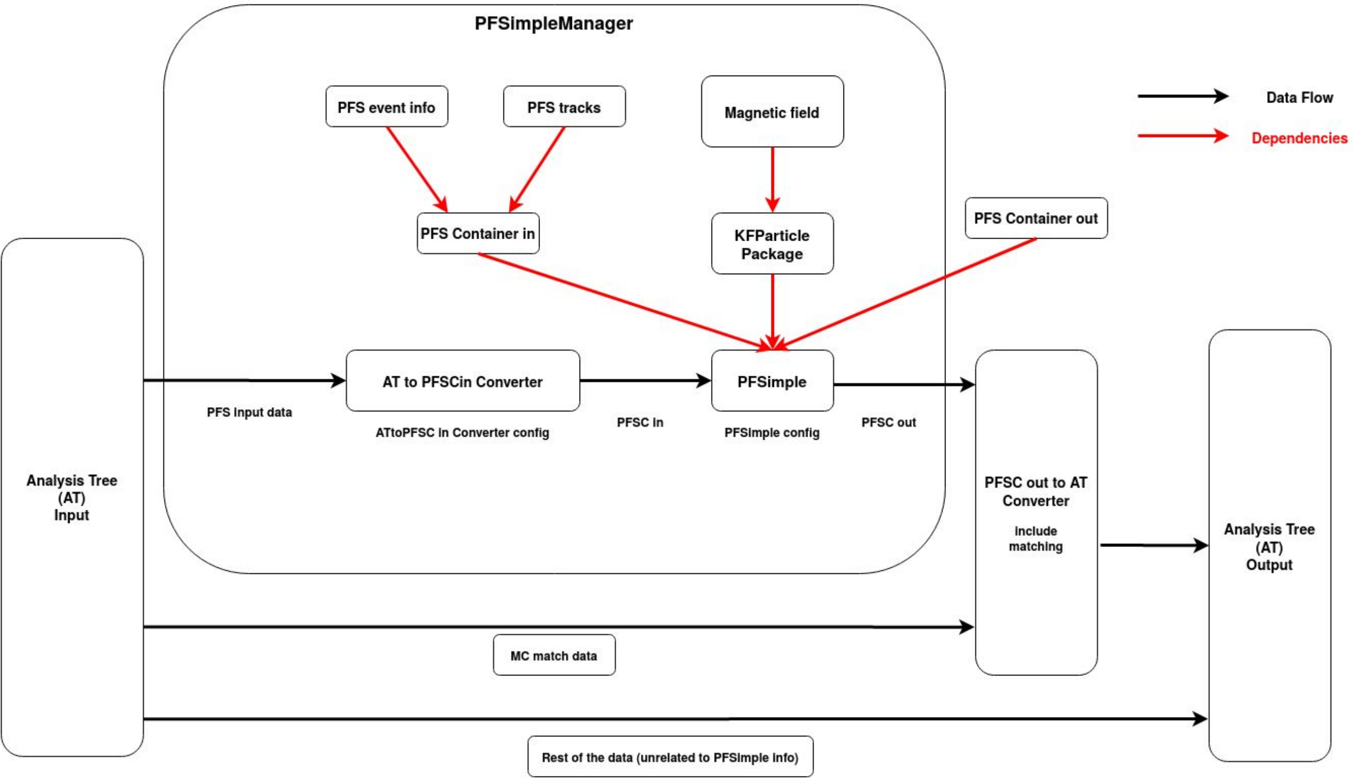
PFSimple

Software design oriented on flexibility and modularity for systematic performance studies and physics analysis

Implementation:

- User controlled reconstruction algorithm
- Stream reconstruction information and parameters from/to other analysis tools
- Manual selection criteria optimization
- Offers the possibility of flat trees for python
- Optimize selection criteria using Machine Learning techniques for individual decays,
  or/and phase space (e.g. p_T - rapidity - centrality)

PFSimple

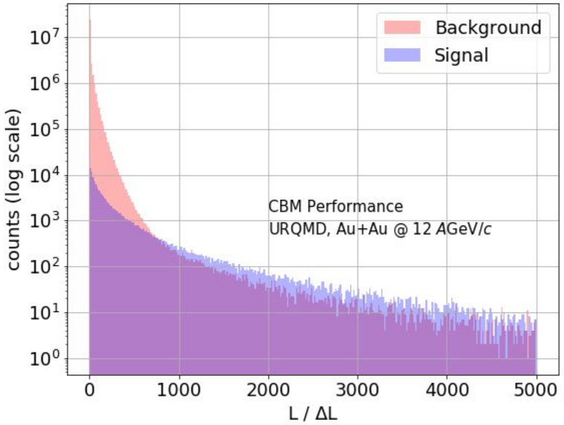


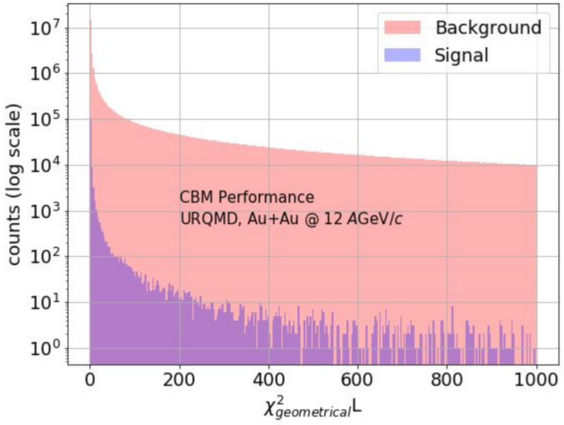
parameter name	#	source
Track parameters (X, Y, Z, T_x, T_y, Q/P)	6	from CBM L1 tracking
Track charge	1	
Covariance matrix of track parameters	21	
Particle type hypotheses (PDG code)	1	TOF, MC PID, no PID
Magnetic field (MF): $\textit{B} = (B_x, B_y, B_z)$ parameterized with parabolic function: $B_i = (a_i+b_i(r_i-r_{0,i})+c_i[r_i-r_{0,i}]^2)$	9	using CBM L1 functionality
Reference position for MF calculation: $r_0 = (0,0,z_0)$	1	position of the 1st hit
Primary vertex coordinates	3	from CBM tracking
In total: 42 parameters		

More details on Git: Analysis tree, KFParticle, **PFSimple**

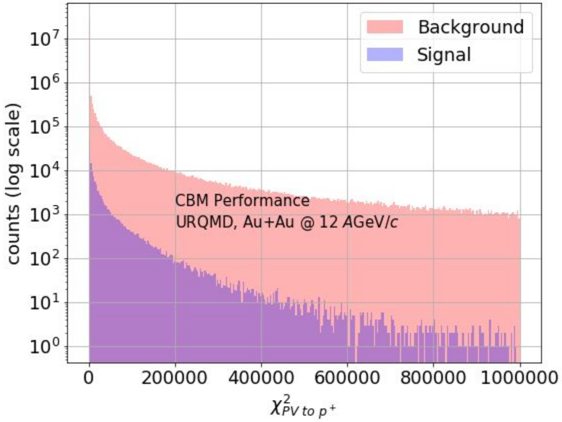
X,Y,Z (track position), $T_x=dx/dz$, $T_y=dy/dz$

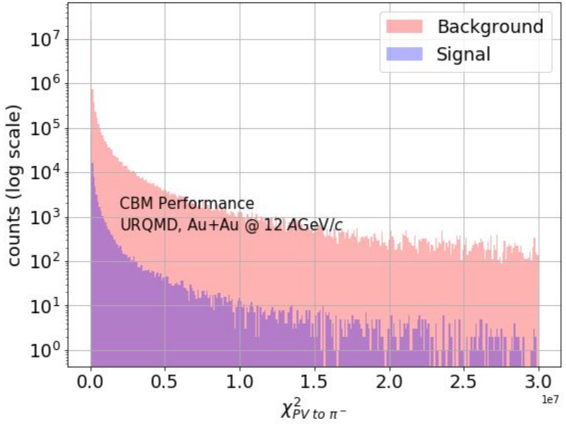
Distribution of signal & background in the variables

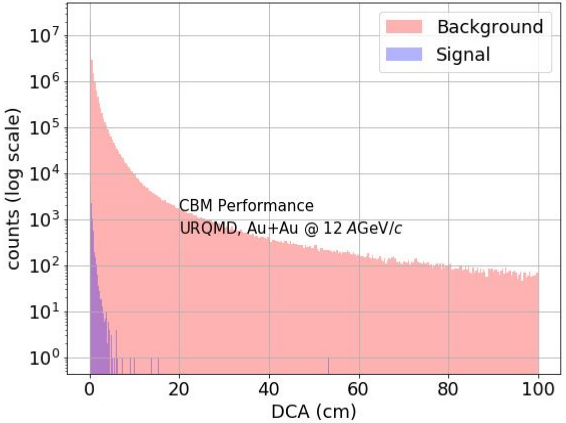




Distribution of MC signal (pure signal) and background plotted for the variables used in this study

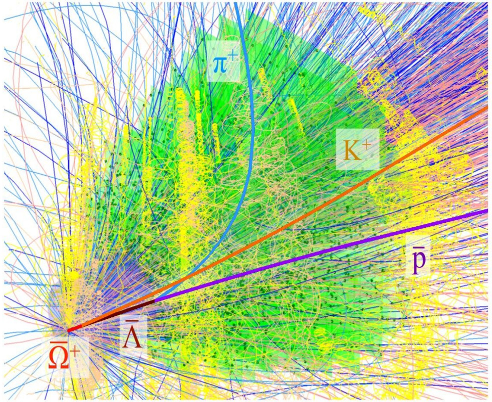






Machine Learning (ML)

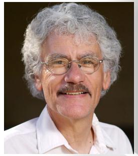
- ML algorithms can perform a specific task by analyzing examples and can learn from data
- Variables associated with decay tracks are analyzed by the algorithm to classify Λ candidates
- Various ML algorithms tested: (SVM, Regression, MLP, Decision Trees, Gradient Boosting (GB), Extreme GB (XGB))
    - XGB works better in terms of performance



CBM Au+Au collisions @ 12*A* GeV/*c*

Gradient Boosting (GB)

Jerome Friedman:



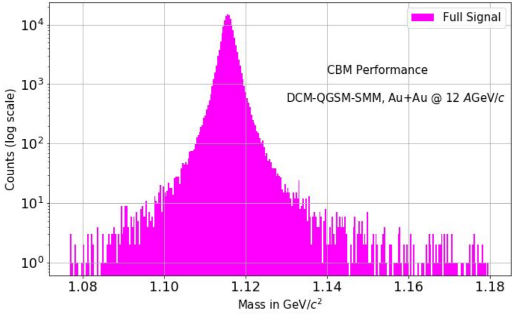
> empirical evidence shows that taking lots of small steps in the right direction results in better predictions with the Testing Data

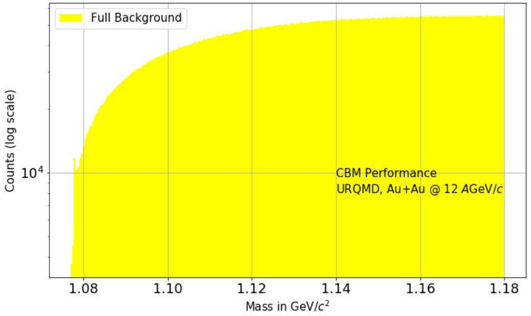
- Boosting combines weak learners (error rate <50%) to make a strong learner (error rate <25%)
- Decision trees (weak learners) are combined together to make a GB algorithm
- In each step a new tree is used to improve the previous prediction
- XGB is an extension of GB with:
    - better control over overfitting
    - parallel processing
    - additional features

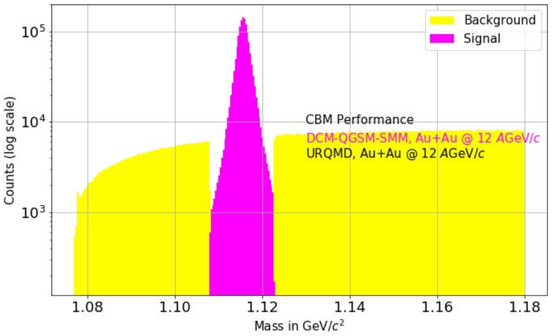
Following the implementation of the ALICE Collaboration ML package

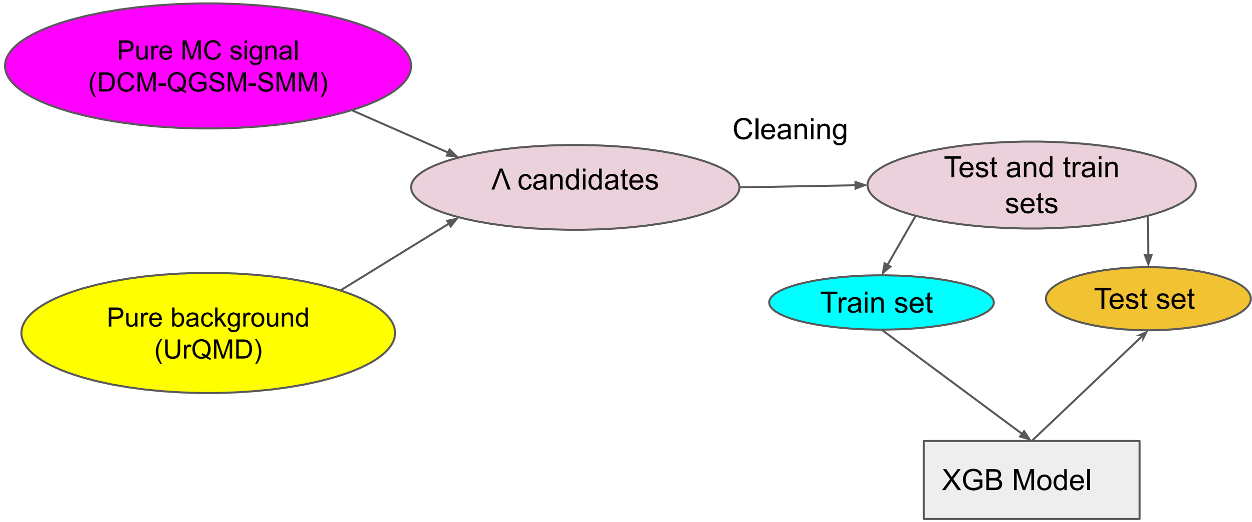
XGB implementation for Λ

- UrQMD sample is taken as experimental data (pure background)
- DCM-QGSM-SMM sample as simulated data (pure signal)
- Λ candidates are cleaned by removing nonphysical values
- Λ candidates are divided into train and test samples





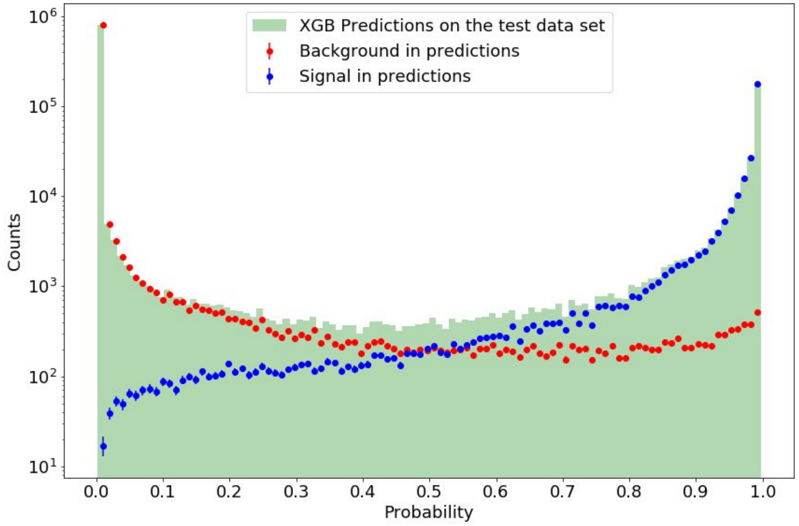




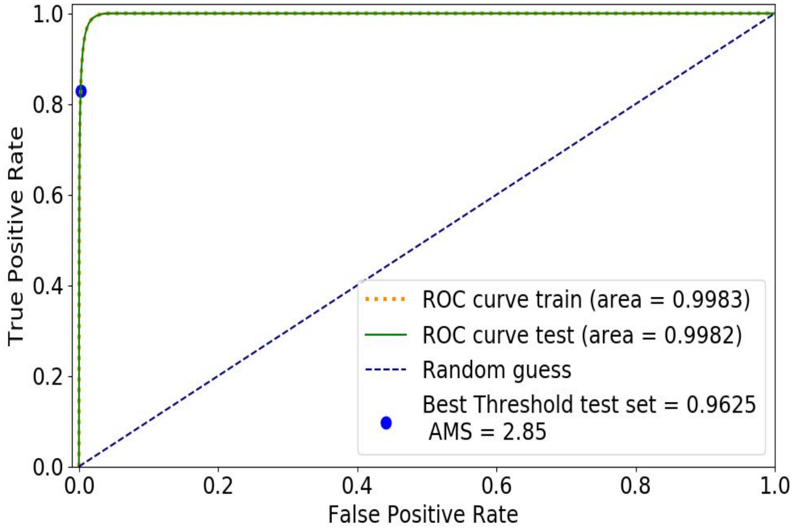
Background is selected ± 5σ away from the Λ peak mean

XGB Model evaluation

Model trained on the train sample is applied to the test sample



Optimize Λ candidates selection for significance



True positive rate = tpr; Signal = S ; Background = B

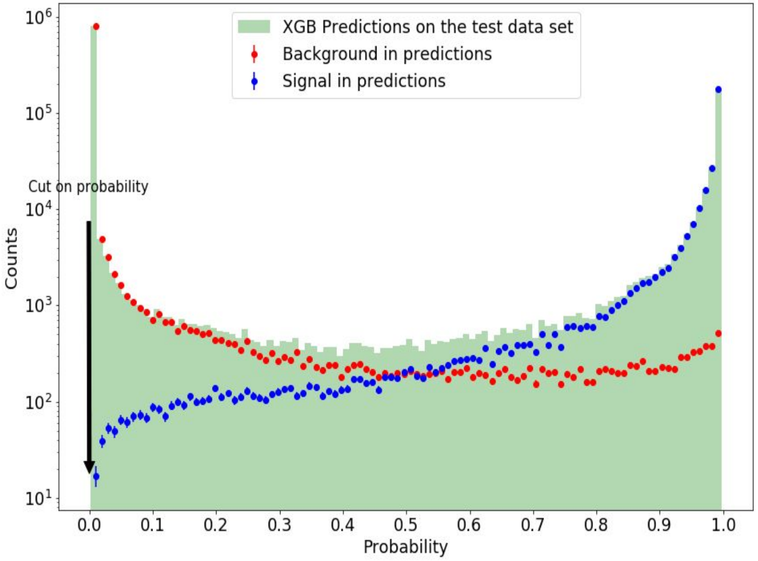
$$tpr = \frac{S\ classified\ as\ S}{S\ classified\ as\ S + S\ classified\ as\ B}$$

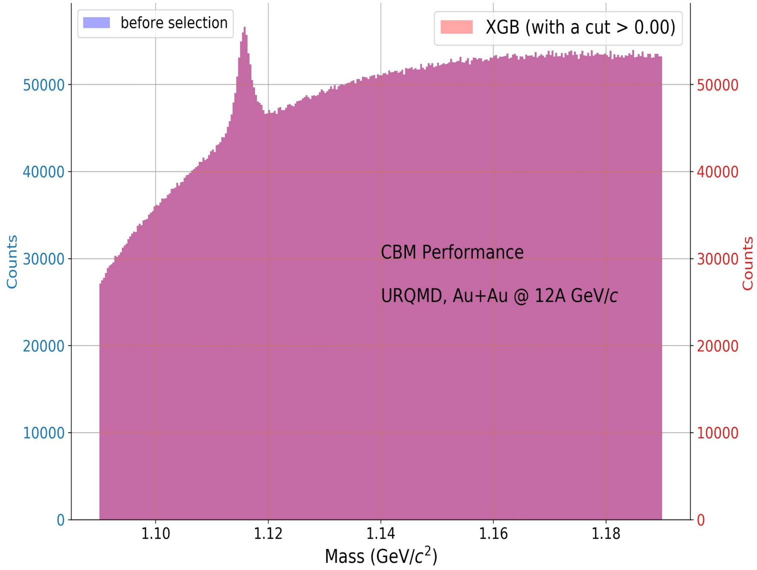
$$fpr = \frac{B\ classified\ as\ S}{B\ classified\ as\ B + B\ classified\ as\ S}$$

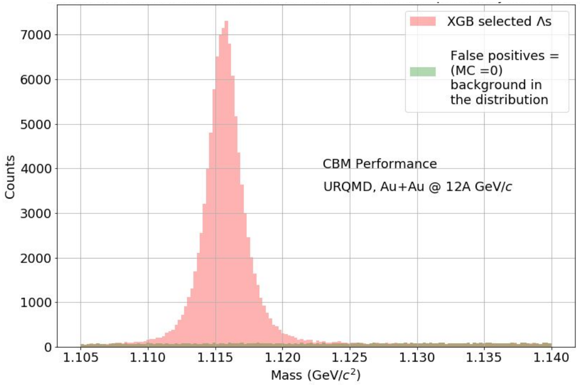
Threshold on the ROC (Receiver Operating Characteristic) curve which maximizes Approximate Median Significance (AMS) on the test sample is our Best Threshold

$$AMS = \sqrt{2\left[(tpr + fpr)\log(1 + tpr/fpr) - tpr\right]}$$

XGB performance for Λ candidates selection

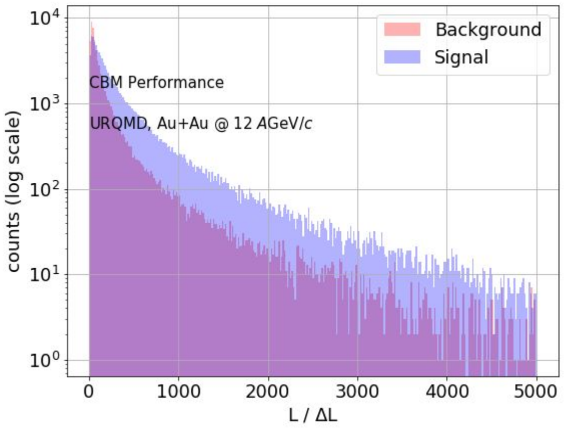


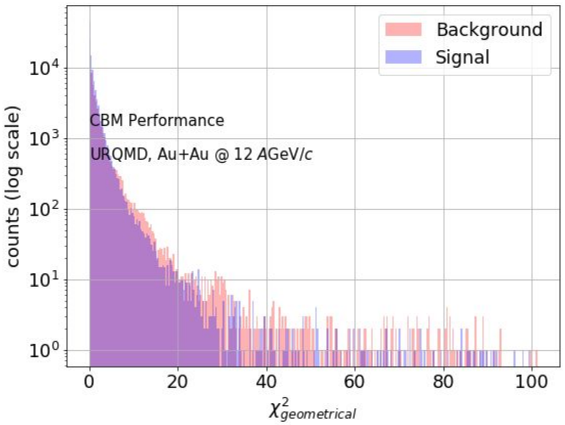


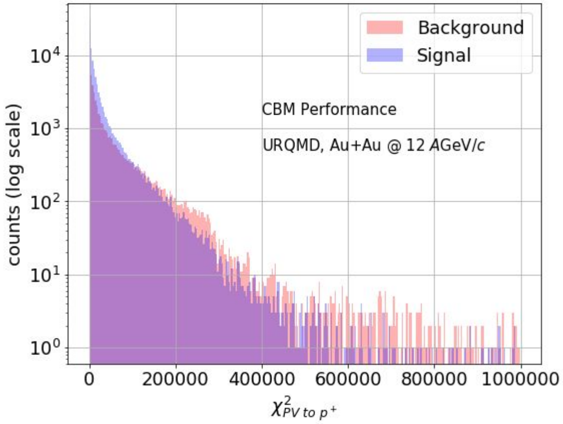


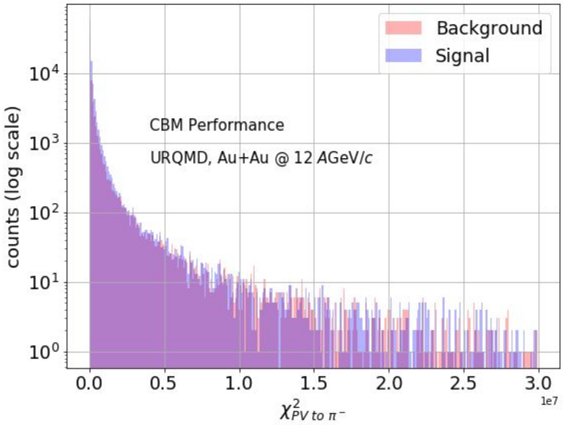
- Preserve smooth background shape after XGB selection
- Optimal XGB probability (0.96) is applied

Distribution of True signal and background in XGB Selected Signal

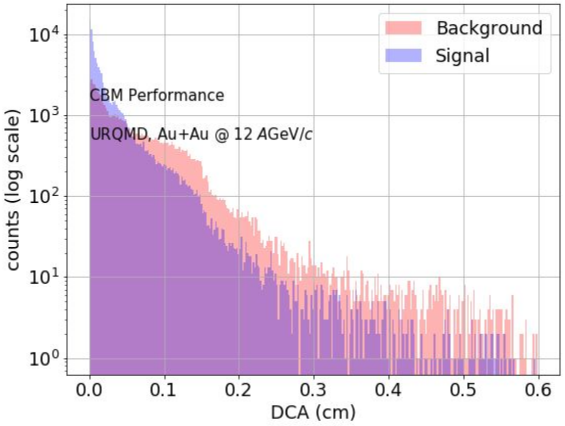






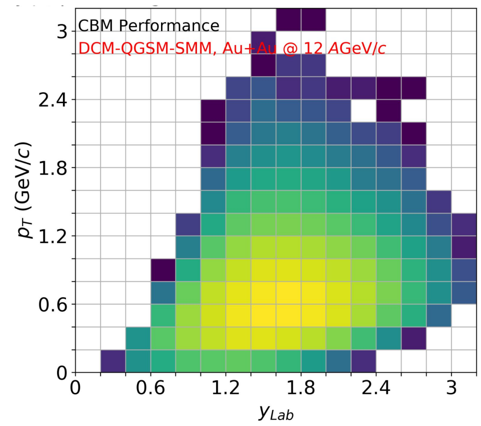


- The distribution of MC signal (pure signal) and background in the XGB selected signal



Yield Extraction: fitting procedure

Divide (p_T,y) phase space into 15x15 bins

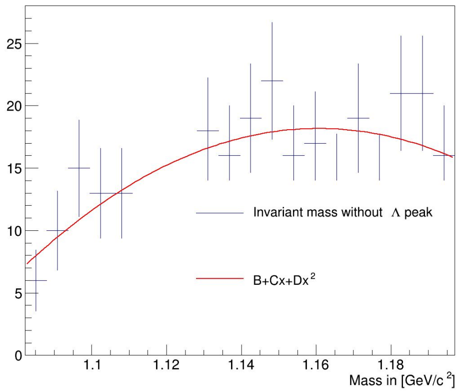


Lorentzian function is used for signal and 2nd order polynomial for background:

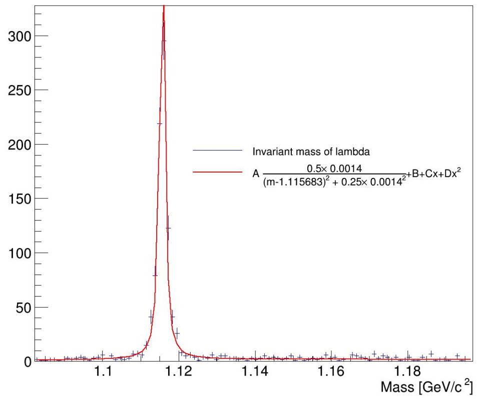
$$Fit(m) = A\frac{(1/2)\Gamma}{(m-m_0)^2+(\Gamma/2)^2} + pol2(m)$$

1. Exclude signal region ($m<1.108$ & $m>1.13$) and fit background with $pol2(m)$
2. Use background fit parameters as initial values for next iteration, where signal (Lorentzian) fit function has fixed $m_0 = 1.1156\,\mathrm{GeV/c^2}$ and width $\Gamma=0.0014$ GeV
3. Use fit parameters as initial values for unconstrained fit to the whole inv. mass range

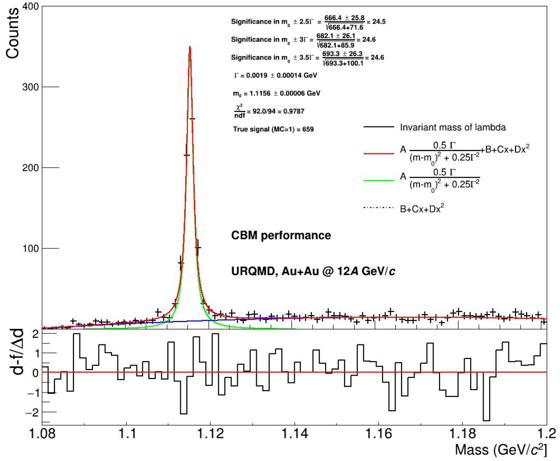
Step 1



Step 2



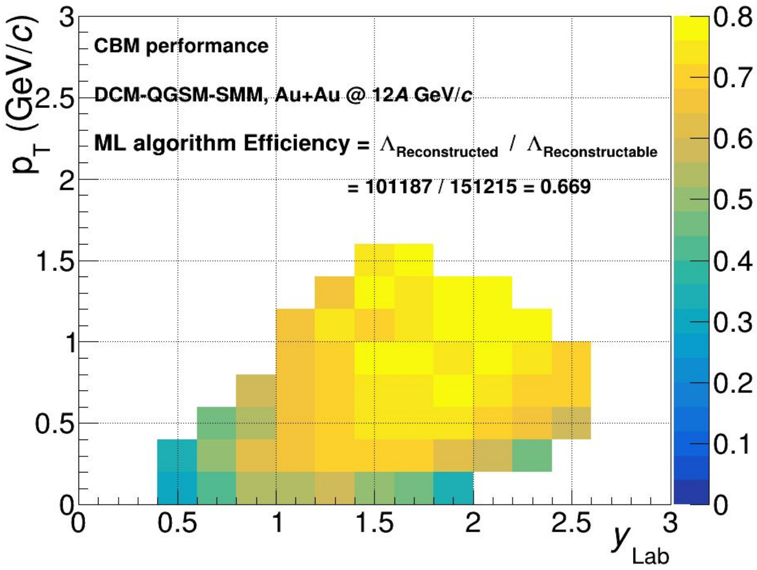
Step 3



Fits for all (p_T,y) bins available here

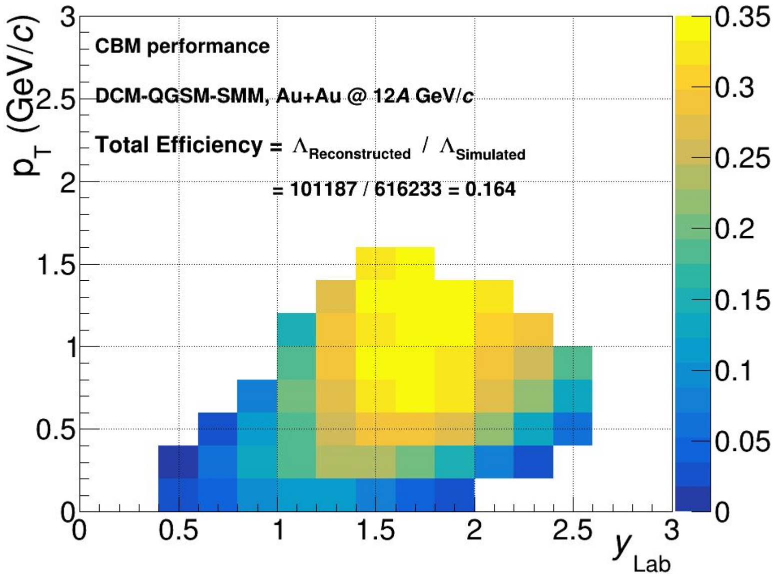
Results: acceptance and efficiency of Λ reconstruction

XGB algorithm efficiency



XGB algorithm shows high efficiency ~ 80%

Acceptance and efficiency

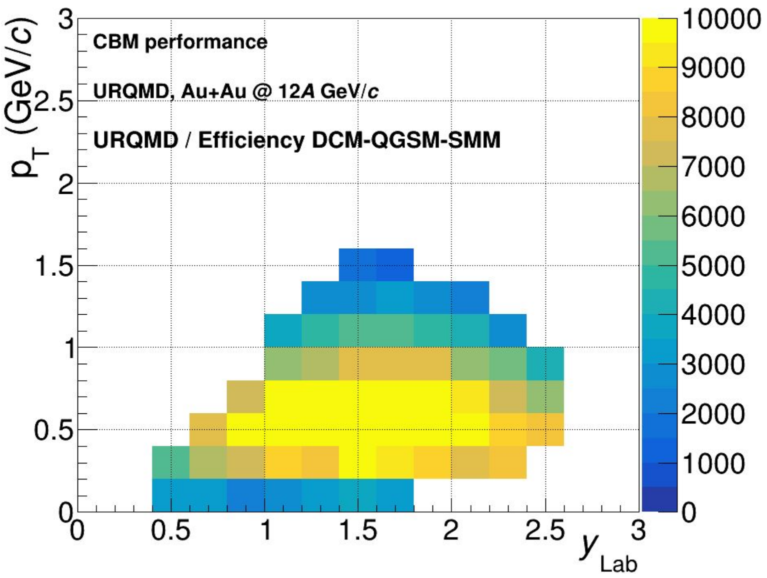


Total reconstruction (acc x efficiency) ~ 35%

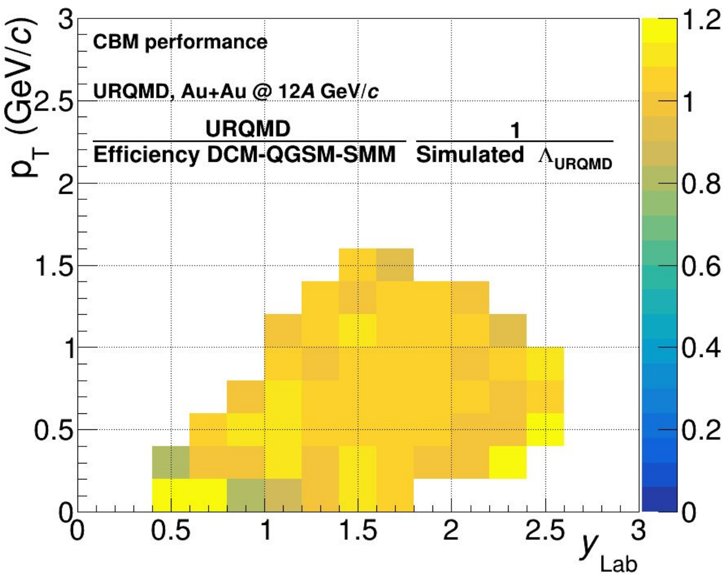
- Reconstructed = reconstructed + selected Λ
- Reconstructable = both daughters are reconstructed

Results: efficiency and acceptance corrected Λ yield

Corrected Λ yield

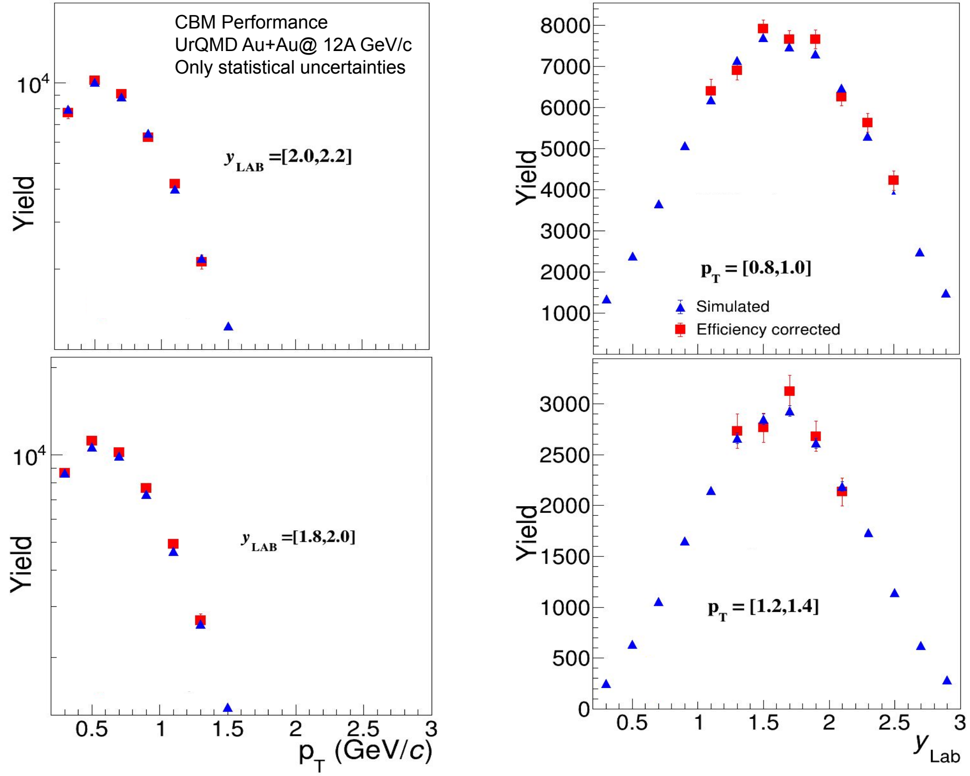


(Corrected Λ yield) / true Λ

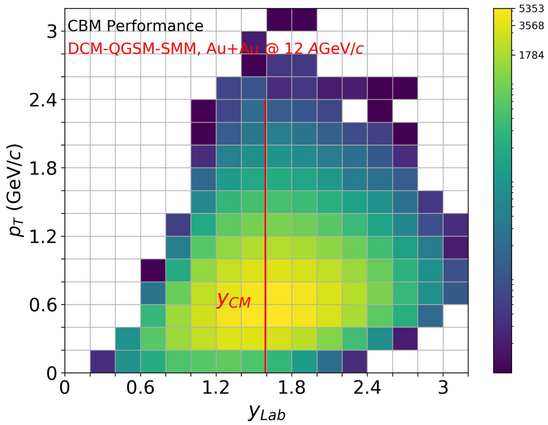


XGB selection, yield extraction procedure, and efficiency correction allow to recover true Λ yield

Results: Efficiency and acceptance corrected yield (p_T/y projections)



True Λ yield



Reconstructed input signal without introducing any bias due to XGB model

Summary and outlook

- Λ baryon reconstruction in CBM@FAIR with Machine Learning techniques
  - Optimization of selection criteria performed via XGB
  - High signal purity and efficiency achieved
- Λ yield extraction and efficiency
  - Yield, extracted after XGB selection and (acceptance x efficiency) corrected is compatible with initial model spectra

Outlook

- Include more variables to improve XGB selection and signal to background ratio
- Study different Λ samples to minimize overfitting and investigate stability
  - multi-differential (p_T, y, centrality) XGB selection, test and training
- Evaluate systematic uncertainties
  - XGB selection variation
  - Yield extraction: variation of fit ranges, background and signal fit functions
- Apply developed procedure for multi-strange hadrons and hyper-nuclei

The CBM Collaboration

56 institutions, 12 countries, ~450 members

Germany
Darmstadt TU
FAIR
Frankfurt Univ. IKF
Frankfurt Univ. FIAS
Frankfurt Univ. ICS
GSI Darmstadt
Giessen Univ.
Heidelberg Univ. P.I.
Heidelberg Univ. ZITI
HZ Dresden-Rossendorf
KIT Karlsruhe
Münster Univ.
Tübingen Univ.
Wuppertal Univ.
ZIB Berlin

India
Aligarh Muslim Univ.
Bose Inst. Kolkata
Panjab Univ.
Univ. of Jammu
Univ. of Kashmir
Univ. of Calcutta
B.H. Univ. Varanasi
VECC Kolkata
IOP Bhubaneswar
IIT Kharagpur
IIT Indore
Gauhati Univ.

Korea
Pusan Nat. Univ.

Romania
NIPNE Bucharest
Univ. Bucharest

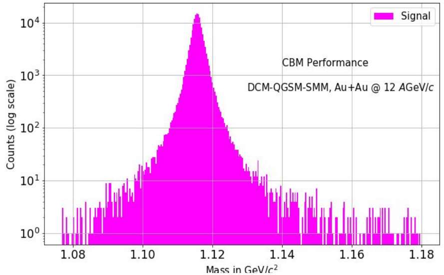
Poland
AGH Krakow
Jag. Univ. Krakow
Warsaw Univ.
Warsaw TU

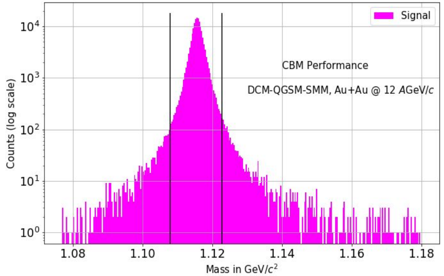
Russia
IHEP Protvino
INR Troitzk
ITEP Moscow
Kurchatov Inst., Moscow
MEPHI Moscow
PNPI Gatchina
SINP MSU, Moscow
JINR VBLHEP,

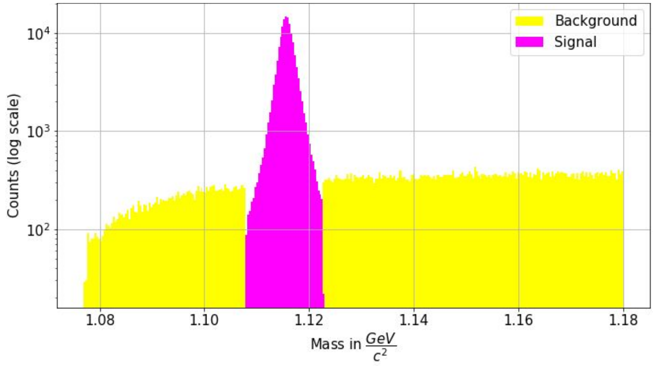
Ukraine
T. Shevchenko Univ. Kiev
Kiev Inst. Nucl. Research

Backup slides

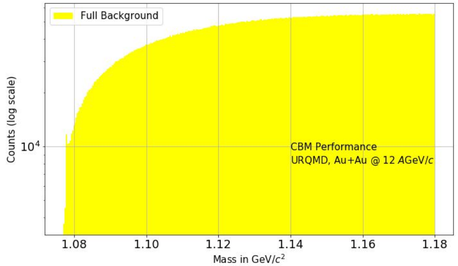
XGB implementation for Λ

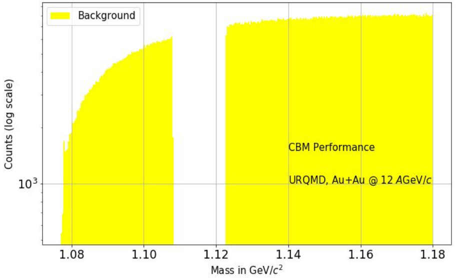


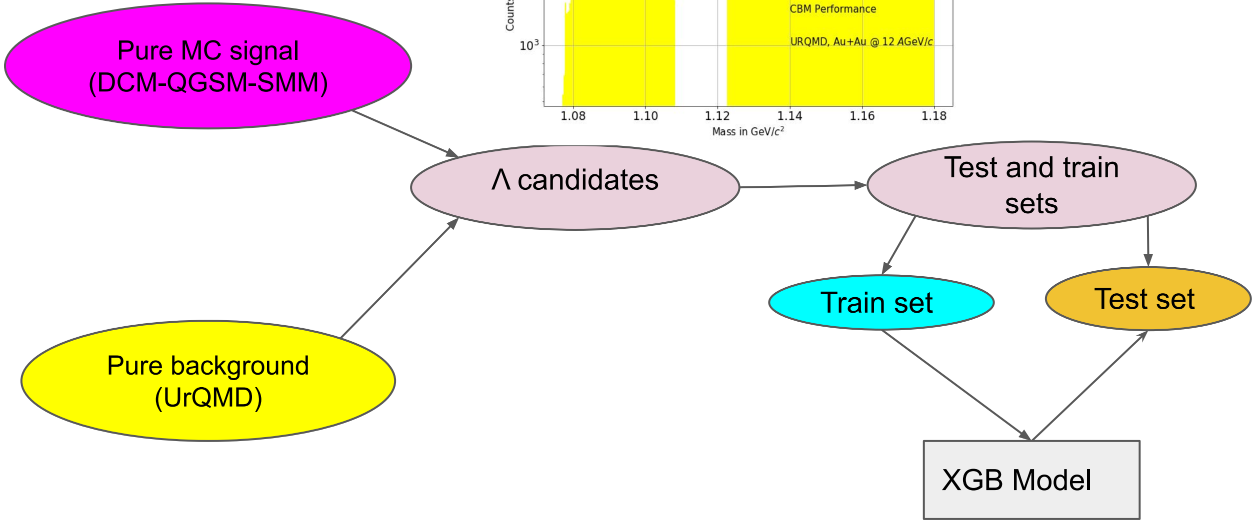




- UrQMD sample is taken as experimental data (pure background)
- DCM-QGSM-SMM sample as simulated data (pure signal)
- Λ candidates are cleaned by removing [nonphysical values]
- Λ candidates are divided into train and test s







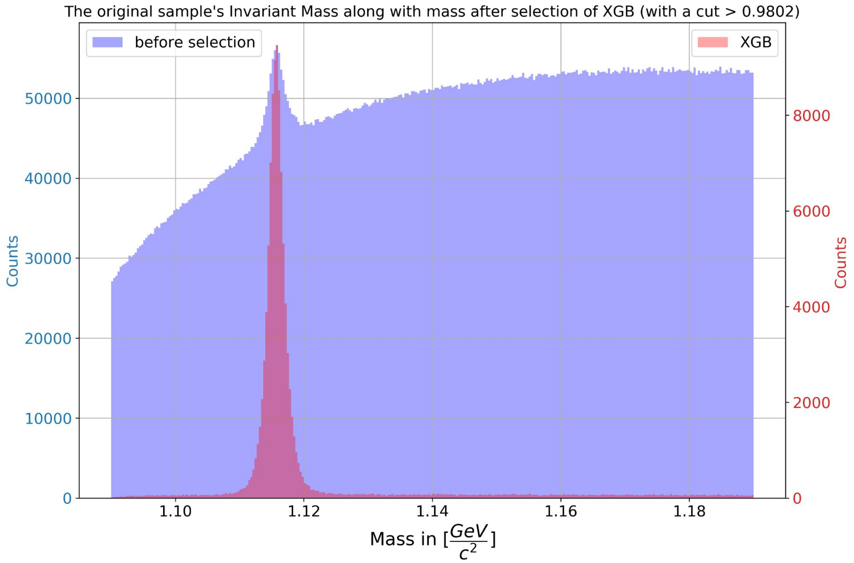
Background is selected ± 5σ away from the Λ peak mean

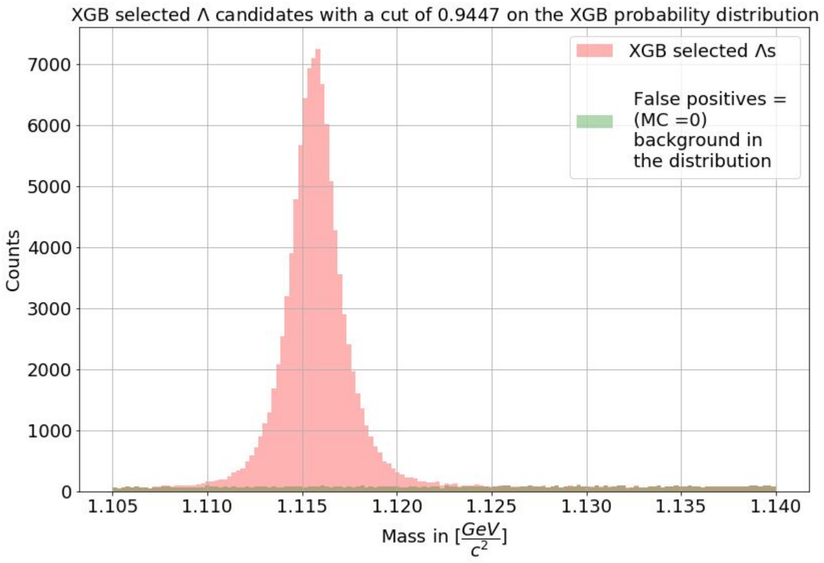
- Well go and check out the following
- Two easy to use jupyter notebooks are available on the following links
  - https://colab.research.google.com/drive/10fD3XNnf_0qt12DiAzlQunbW7IVEqqIE?usp=sharing

  - https://colab.research.google.com/drive/1yV3xboB67trorfOKy1-VLT1kLxYdN6dn?usp=sharing

- our code on github

Applying the model on the URQMD data set





The threshold on the ROC curve which maximizes AMS on the test data set is applied on the URQMD 100k events data set

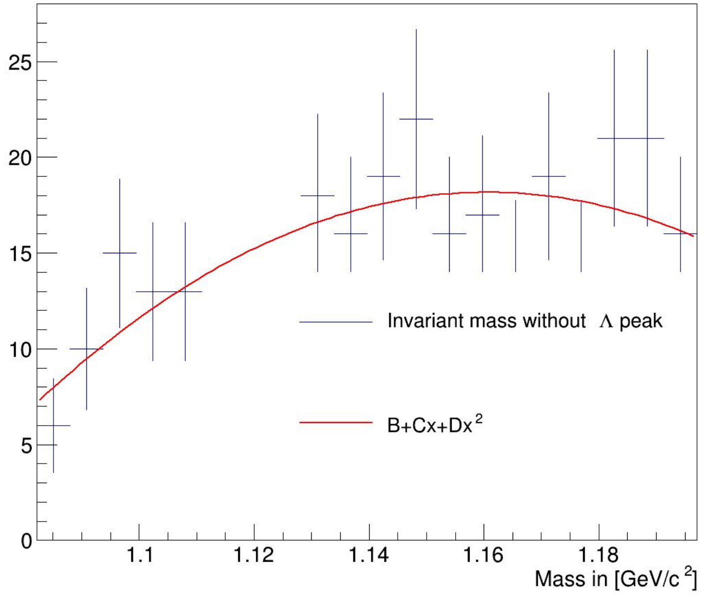
ML does not cut the background in an unexpected way, therefore, not introducing any bias

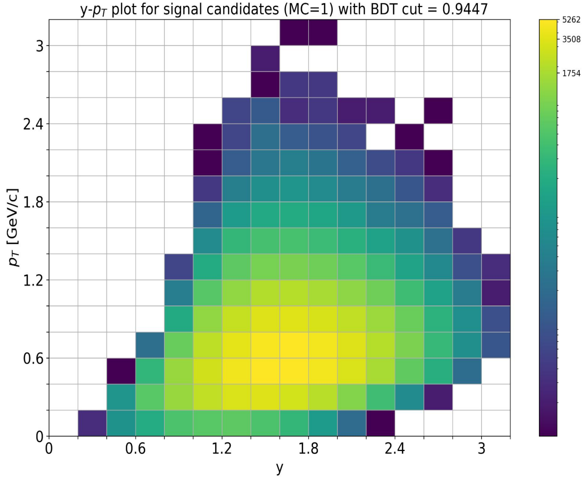
Yield Extraction: The Fitting Procedure

- Divide the data into p_T-y bins
- Applied fitting to all the bins individually
- Apply a mass cut of 1.13<m<1.108 for a 2nd order pol background fit

- Get the fit parameters and use them as initial fit parameters for the whole mass range, the fitting function is $A\frac{0.5\times0.0014}{(m-1.115683)^2+0.25\times0.0014^2}+B+Cm+Dm^2$

- Get the fit parameters and use them as initial parameters
- The final fit function $A\frac{0.5\Gamma}{(m-m_0)^2+0.25\Gamma^2}+B+Cx+Dx^2$

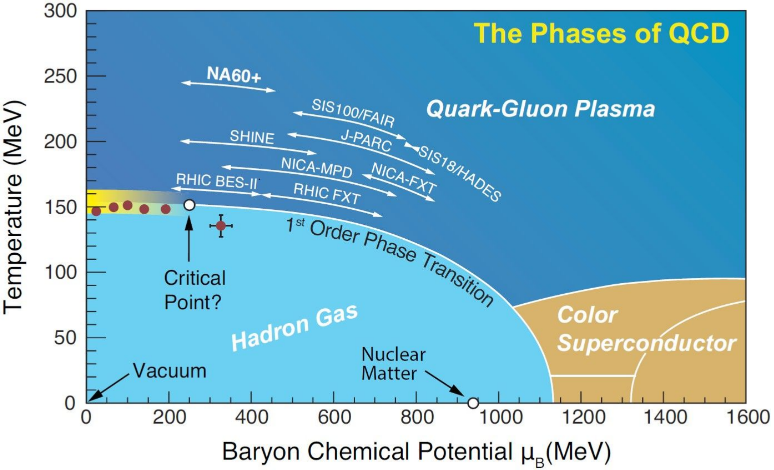




[pdf of pT rapidity bins divided data, with fitting]

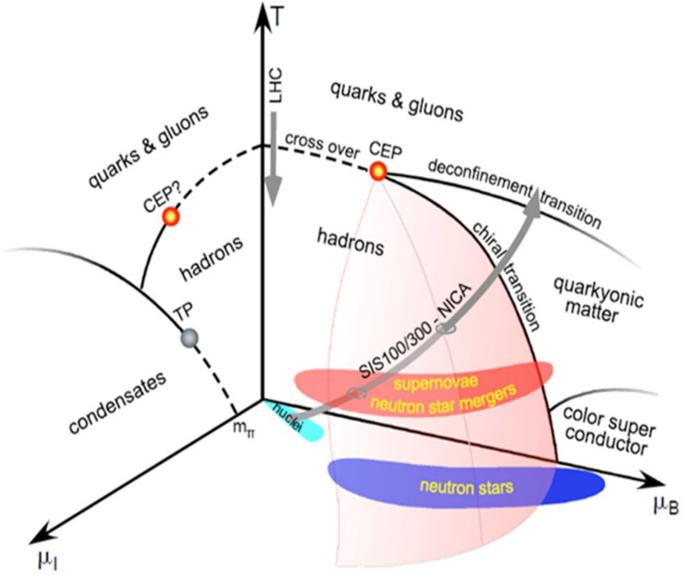
Phase diagram

Alessandro D. Falco, CPOD-2021

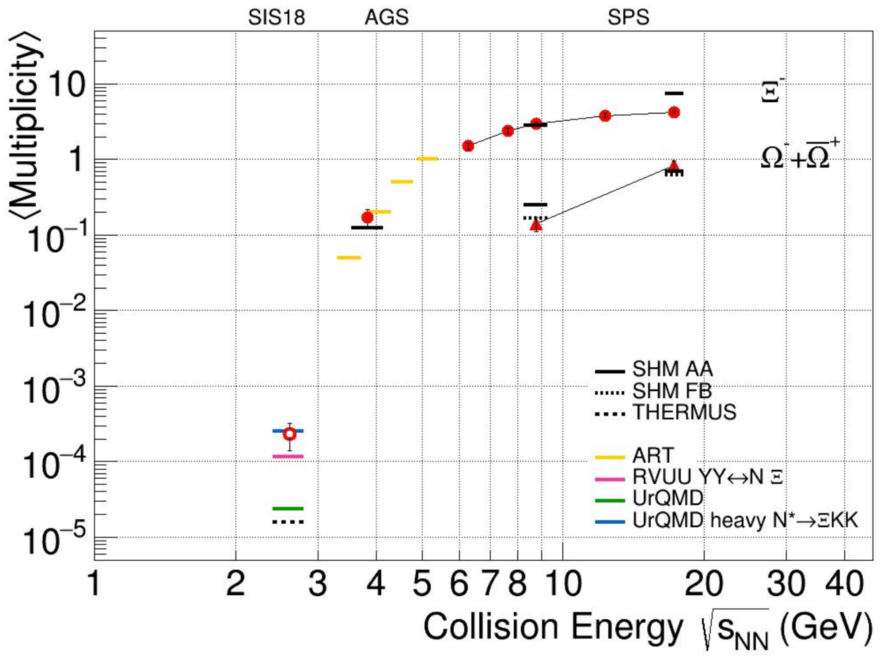


https://indico.cern.ch/event/985460/contributions/4264615/attachments/2211234/3742919/adf_cpod.pdf

NUPECC Long Range Plan 2017



Multi-strange yields

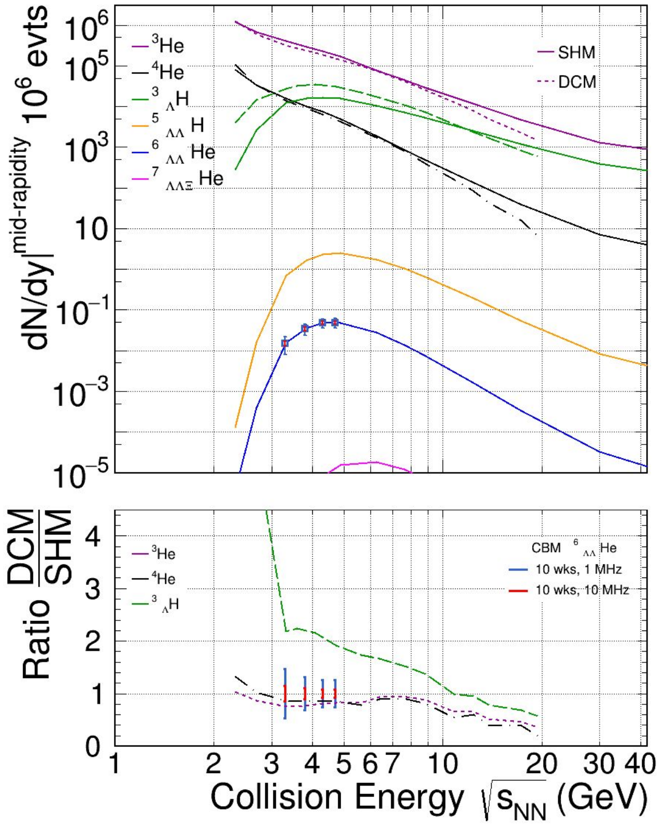


	$\sqrt{s_{NN}}$	Run time	$R_{int,}$ kHz	X^-	X^+	Ω^+
HADES (Ag)	2.6 GeV	4 wks	10	2.5×10^3		
MPD S1	11 GeV	10 wks	5	1.5×10^6	8×10^4	1.5×10^4
CBM	3.8 GeV	1 wk	1000	4×10^9	5×10^6	3.3×10^5

Compilation TG, QM2018

C. Blume, C. Markert, PPNP 66 (2011)
HADES Coll., PLB 778 (2018)
HADES Coll., PRL 103 (2009) 132301
RVUU: F. Li et al., PRC 85 (2012) 064902
UrQMD: J. Steinheimer et al., J.Phys. G43 (2016) 015104
ART: C.M. Ko et al., PLB595 (2004) 158-164
A. Andronic et al., NPA 772 (2006)
F. Becattini et al., PRC69 (2004) 024905
E. Seifert et al., PRC97 (2018)

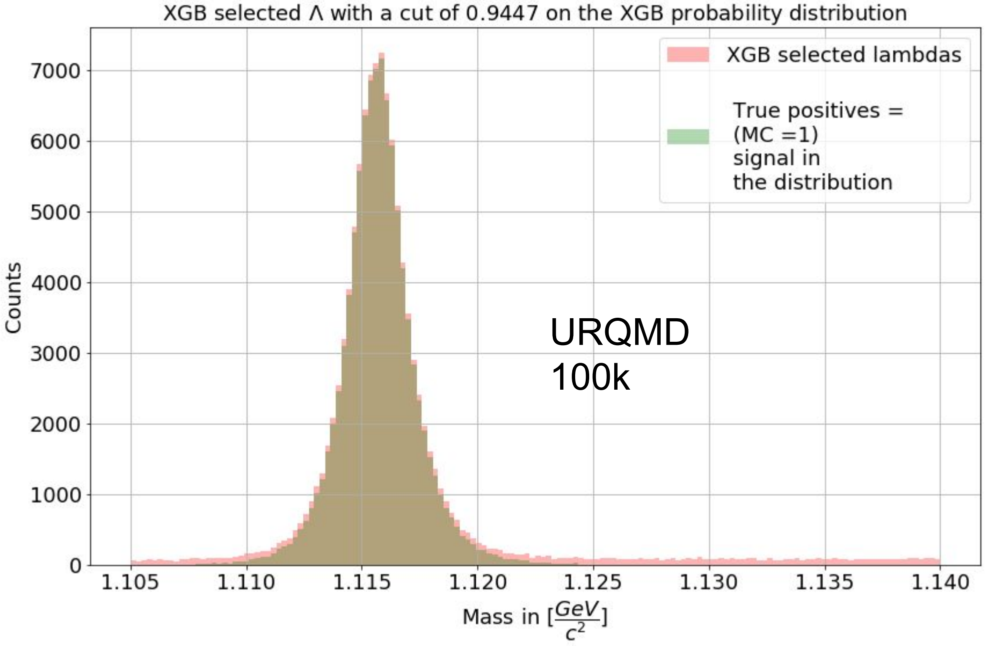
Hypernuclei yield: CBM projections

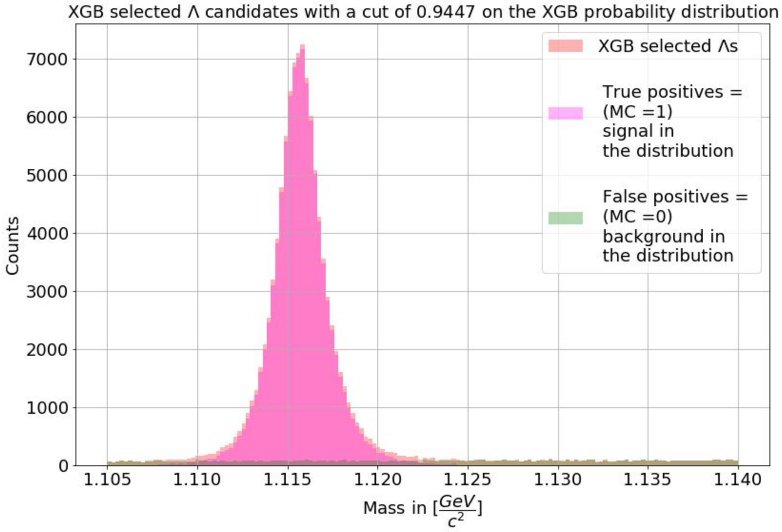


CBM	$\sqrt{s_{NN}}$	Run time	e %	R_{int}	*Duty F %*	Yield
$^{3}_{L}$H	4.7 GeV	1 wks	19	10 MHz	50	5.5×10^9
$^{4}_{L}$He	4.7 GeV	1 wks	15	10 MHz	50	2.7×10^8
$^{6}_{LL}$He	4.7 GeV	10 wks	1	10 MHz	50	146

Blue and red lines are the precision with which we can measure yields assuming various scenario

Compilation TG, QM2018


XGB selected Λ with a cut of 0.9447 on the XGB probability distribution
XGB selected lambdas
True positives = (MC =1) signal in the distribution
URQMD
100k
Counts
Mass in [$\frac{GeV}{c^2}$]
0
1000
2000
3000
4000
5000
6000
7000
1.105
1.110
1.115
1.120
1.125
1.130
1.135
1.140


XGB selected Λ candidates with a cut of 0.9447 on the XGB probability distribution
XGB selected Λs
True positives = (MC =1) signal in the distribution
False positives = (MC =0) background in the distribution
Counts
Mass in [$\frac{GeV}{c^2}$]
0
1000
2000
3000
4000
5000
6000
7000
1.105
1.110
1.115
1.120
1.125
1.130
1.135
1.140

Data cleaning/skimming

The data contains some entries which does not make sense, so we clean it by pre cuts

$p<20$	$pz > 0$	$0<X^2$primpos $< 1x10^6$	$0<l/dl<8000$
abs (x, y) < 50	$-1< z <80$	$0<X^2$geo $< 10^3$	cosineneg>0.1
$1<eta<6.5$	0<distance< 100	$0<X^2$primneg$<3x10^7$	Remove nan, infinite
$1.07<mass<1.3$	$l<80$	$0<X^2$topo$< 10^5$	$0<l/dl<8000$

Removes 3.2 % signal candidates from a set of 10k events (AU 12AGeV mbias URQMD)

But also removes 57 % background

mass can't be negative and we select mass greater or equal to the mass of proton and pion

Fixed target experiment, target position: (0,0,0)

X^2 can't be negative

Gradient boost: regressor, in simple words

- GB: Trees predicting residuals and a learning rate to prevent overfitting

2 samples

Using these

		X_1	X_2
variables	$Variable_1$	1	2
	$Variable_2$	3	4
	$Variable_3$	5	6
target	y	1	0
First prediction	Average of y = y’	0.5	0.5
Pseudo residuals	Residual = y - y’	0.5	-0.5
Predicts this	Tree=h_1	0.5	-0.5
2nd prediction	Predicted=y’’ =y’+tree	1	0
controls	Learning rate=0.1	0.1	0.1
3rd prediction	New prediction=y’’’=y’+ 0.1* (tree)	0.55	0.45

Over fits:
Low bias
High
variance

New residuals=y-y’’’	0.45	-0.45
Tree 2 = h_2	0.45	-0.45
Newest prediction= y’’’’ = y’’’+0.1 * (tree 2)	0.595	0.405
Goes on		

Step towards
the main target

Further reading
https://xgboost.readthedocs.io/en/latest/tutorials/model.html

Detailed Explanation GB

1. Input: Data $\{(x_i,y_i)\}^n_{i=1}$ and a differentiable **Loss Function** $L(y_i,F(x))$
   If we choose $L = ½ \{ y_i - F(x) \}^2$
   Then
   $d/dF(x) \{½ \{ y_i - F(x) \}^2\} =(-(y_i-F(x))) = F(x)-y_i$ =-(residuals)

   We minimize this $F(x)-y_i$ for all values

$$\sum_i^n F(x) - y_i = 0$$

   A predicted value which can minimize this sum is the average

$$F(x) = \frac{\sum_i^n y_i}{n}$$

   = average = $F_o(x)$

2. Fit m = 1 upto m=M number of trees
   a. Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]$ at $F(x) = F_{m-1}(x)$ for i =1,...,n

since first iteration so $F(x)=F_o(x)$

$-d/dF(x) \{½ \{ y_i - F(x) \}^2\} =-(-(y_i-F(x))) = y_i-F(x)$ Residuals

Detailed Explanation GB

- Fit m = 1 upto m=M trees
  a. Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]$ at F(x) = $F_{m-1}(x)$ for i =1,...,n
  b. Fit a regression tree to the r_{im} values and create terminal regions R_{jm}, for j=1,..,J_m (leaves but not with output values)
  c. Determine the output value for each leaf:
     for j=1,..,J_m compute
     again will turn out to be average if $L = \frac{1}{2}\{y_i - F(x)\}^2$

     $$\gamma_{jm} = argmin \sum_{x_i \epsilon R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$
  d. Update
     v is learning rate and the equation in the box is the tree we just made
     We started with F_0 so

     $$F_m(x) = F_{m-1}(x) + \nu \boxed{\sum_{j=1}^{J_m} \gamma_{jm} I(x \epsilon R_{jm})}$$
- Output $F_M(x)$ (The final classifier)

$$F_1(x) = F_0(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \epsilon R_{jm})$$