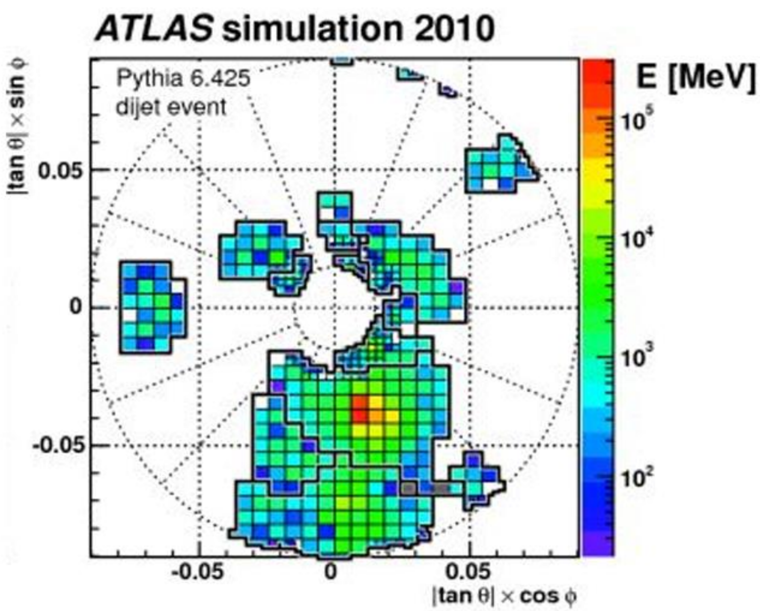
Energy Regression w/ ML

ELPH-ZDC Test Beam Meeting@20240604

Jen-Chieh Peng, Wen-Chen Chang, Chih-hsun Lin,
Chia Ming Kuo, Rong-Shyang Lu, Po-Ju Lin,
Kai-Yu Cheng, Chia-Yu Hsieh (presenter), **Yu-Siang Xiao, Shao-Yang Lu,**
Yuji Goto, Tatsuya Chujo, Motoi Inaba, Subaru Ito, Kentaro Kawade, YongsunKim,

Energy Regression Calibration with Machine Learning Method



- **Purpose of energy regression :** Energy deposited in the calorimeter may not always be directly proportional to the energy of the incident particle due **leakage**, **noise**, etc. By accurately estimating the particle energy, energy regression **improves the energy resolution.**

- **Machine learning techniques** can be used as a method to perform the energy regression.

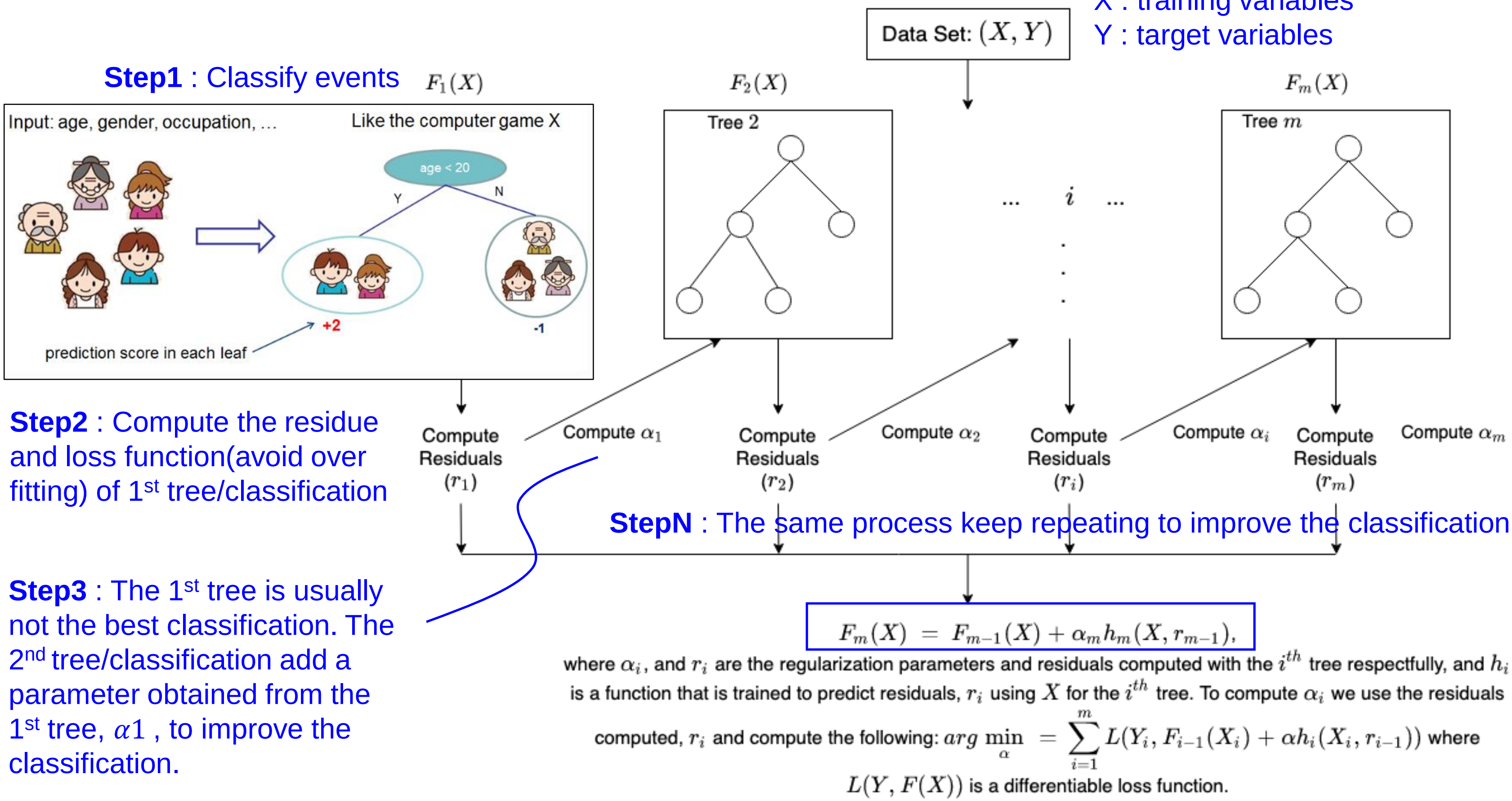
(1) Collect large MC sample and **select training parameters (Emax, E3x3, E5x5)** and **target parameters (ratio of Ebeam/E5x5).**
(2) Model training with large MC sample.
(3) Validate trained model with separated MC sample.
(4) Apply the trained MC to data.

- Attention : One have to make sure **MC and data are agreed at certain level. (We are still working on it!)**

XGBoost (Extreme Gradient Boosting)

Step0 : Data set
X : training variables
Y : target variables

Step1 : Classify events



Step2 : Compute the residue and loss function(avoid over fitting) of 1st tree/classification

StepN : The same process keep repeating to improve the classification.

Step3 : The 1st tree is usually not the best classification. The 2nd tree/classification add a parameter obtained from the 1st tree, $\alpha 1$, to improve the classification.

➔ Final output :
The predictions of all trees/classifications are combined to produce the final output.

Reference : https://xgboost.readthedocs.io/en/stable/tutorials/model.html
https://docs.aws.amazon.com/zh_tw/sagemaker/latest/dg/xgboost-HowItWorks.html

Training Conditions

- **XGBoost in Python**

- **Training MC sample**

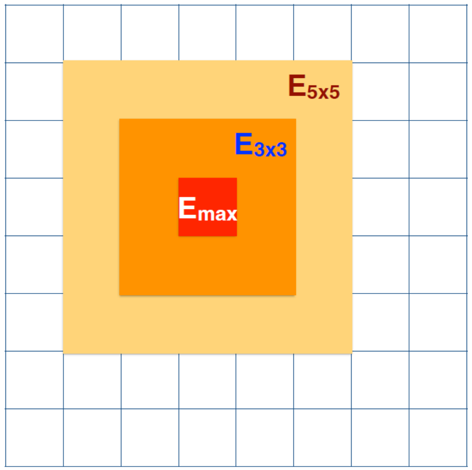
① 197MeV
② 30k events
(20% test, 80% training)

- **Training variables (X):**

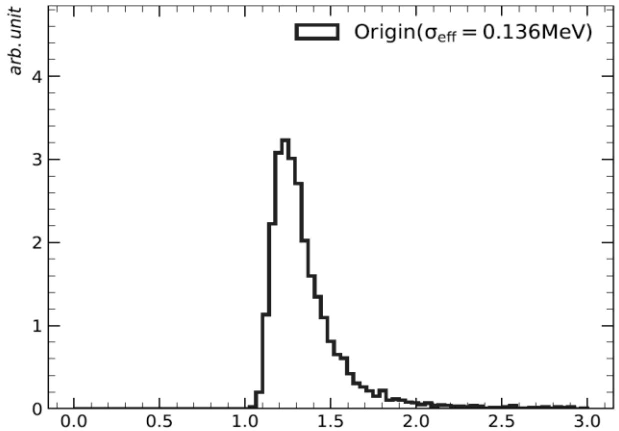
① E1x1
② E3x3
③ E5x5
④ E1x1/E5x5
⑤ E1x1/E3x3
⑥ E3x3/E5x5

- **Target variable (Y) :**

① Ebeam/E5x5



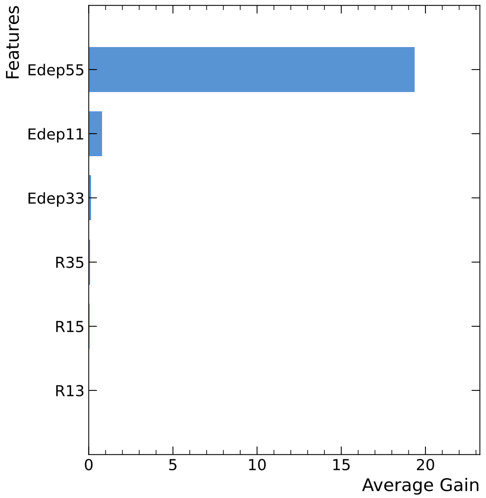
197MeV positron beam



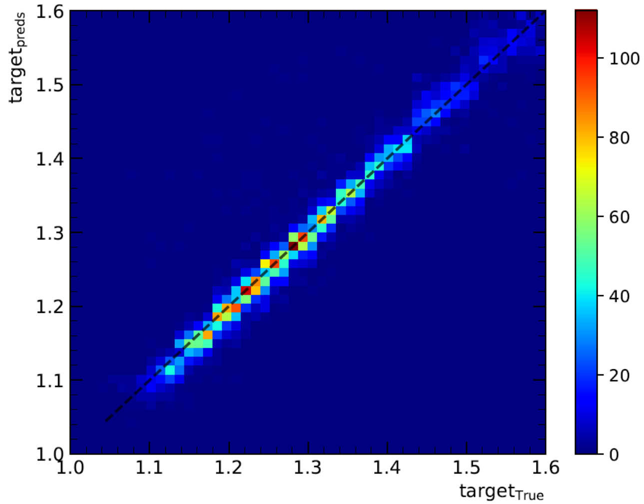
Target : Ebeam/E5x5

Validate ML Model

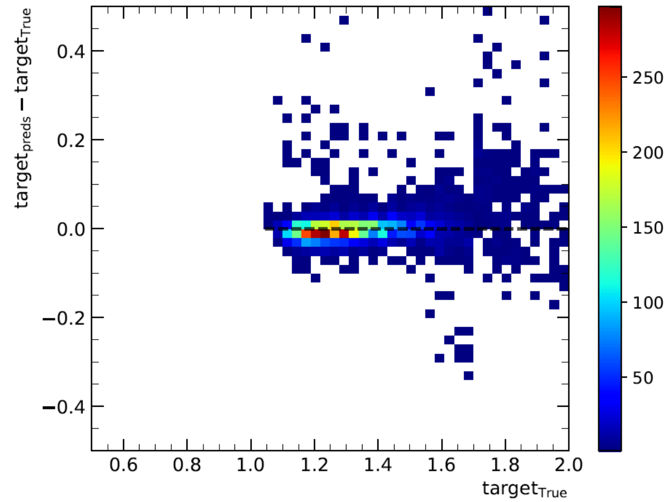
Importance of train variables, X



Target = Ebeam/E5x5
True VS predicted



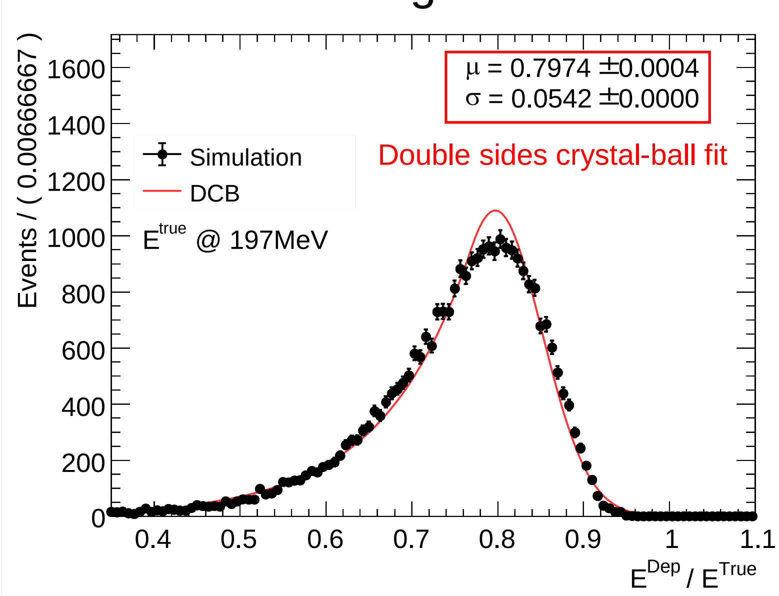
Target = Ebeam/E5x5
uncertainty



- Among all the training variables, E5x5 is the most important one.
- The training output shows reasonable prediction of target variable, Ebeam/E5x5, with less than 5% uncertainty.

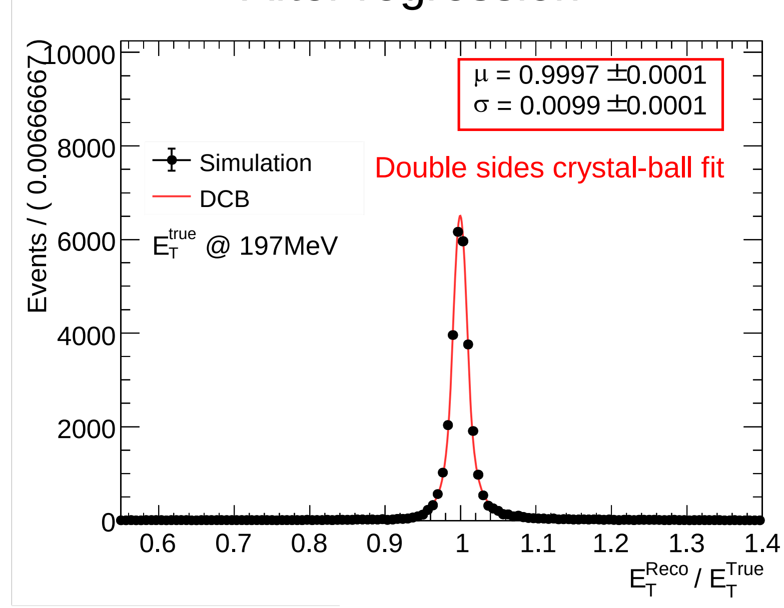
Impact of Energy Regression

Before regression



(E5x5/Ebeam)

After regression

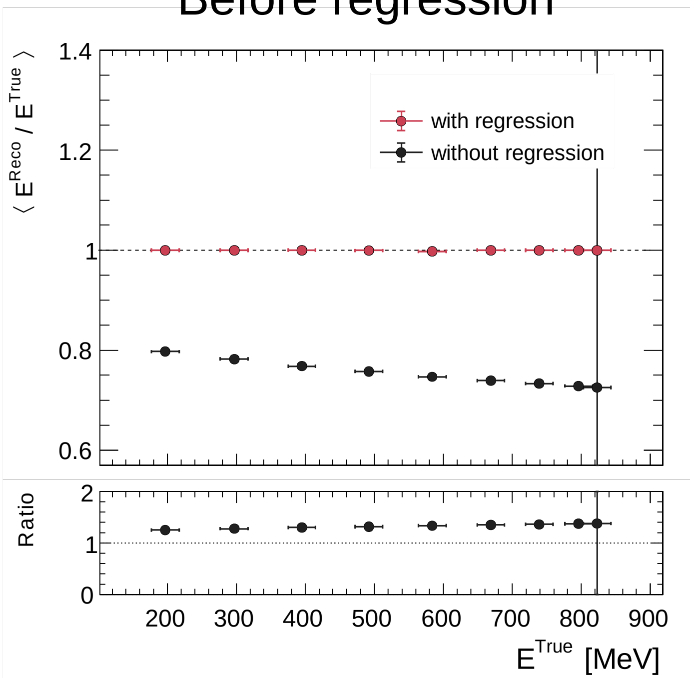


(Ebeam：Epredicted/Etrue)

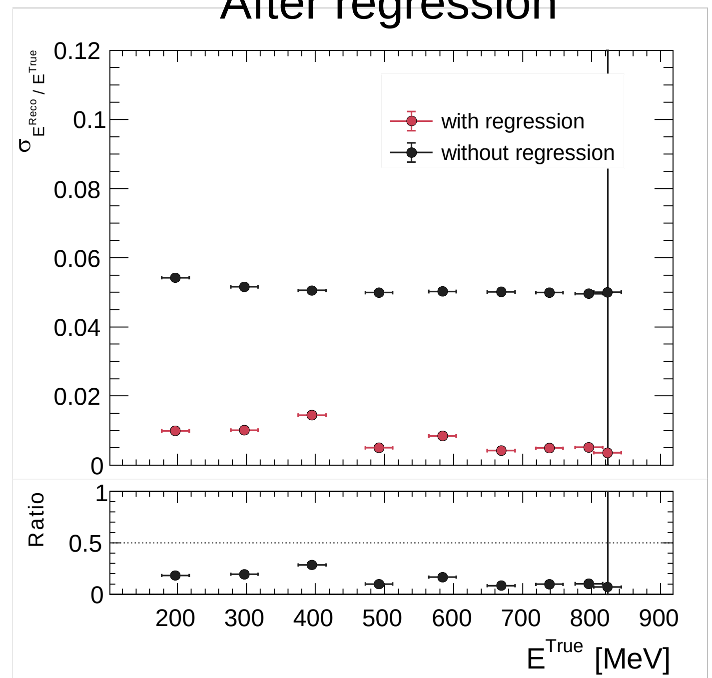
- A **new** MC sample generated w/ 197MeV positron beam w/ 30k events.
- After applying energy regression, the beam energy is will reconstructed by ML model and energy resolution improved from 5% to 1%.

Impact of Energy Regression

Before regression



After regression



- New MC samples with energy beam = 197MeV to 823 MeV are tested.
- Ebeam is well predicted and energy resolution is also improved after regression regardless the beam energy.

Summary and To Do

- A method of energy regression w/ machine learning technique is developed. We use XGBoost package provided by Python.

- MC sample is used as both training and test sample for now. The ML regression model shows a great performance to improve the prediction of beam energy (20%) and reduce the energy resolution (5%->1%).

- We will apply this technique once we have relatively more realistic MC simulation ready.