Energy reconstruction with machine learning techniques in JUNO: aggregated features approach

Arsenii Gavrikov[1,2], Yury Malyshkin[2], Fedor Ratnikov[1]

[1]HSE University, Moscow, Russia

[2]Joint Institute for Nuclear Research, Dubna, Russia

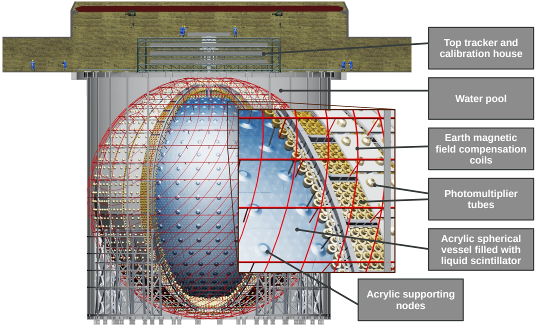
Moscow International School of Physics 2022, 24 July – 2 August 2022

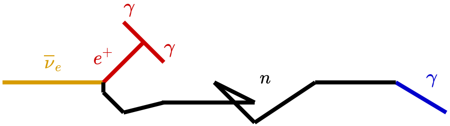
NATIONAL RESEARCH UNIVERSITY

JINR

Introduction to the JUNO experiment

1. Jiangmen Underground Neutrino Observatory:
   - multipurpose experiment
   - 53 km away from 8 reactor cores in China
   - data taking expected in ∼2023
   - JUNO Collaboration:
     - 77 institutions
     - 697 collaborators
2. The main goals of JUNO:
   - neutrino mass ordering (3σ in 6 years)
   - precise measure of oscillation parameters $\sin^2\theta_{12}, \Delta m^2_{21}, \Delta m^2_{31}$
3. The Central Detector:
   - detection channel: $\overline{\nu}_e + p \rightarrow e^+ + n$;
   - deposited energy converts to optical light
   - the largest liquid scintillator detector: 20 kt
   - 77.9% photo-coverage: 18k 20", 26k 3" photo-multiplier tubes (PMTs)

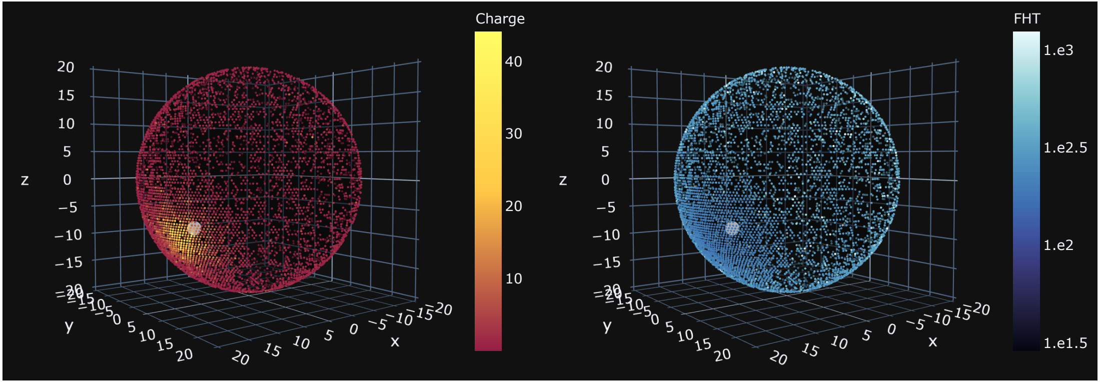




Machine Learning (ML) in HEP

- ML methods are used at all levels of data processing in many HEP experiments:
  - signal/background discrimination
  - event selection in the trigger
  - event simulation
  - anomaly detection
  - identification, etc.
- Why is ML useful for HEP?
  - **Faster**. More precisely, with proper training
  - **Adequate** for many purposes simultaneously: event simulation, analysis, reconstruction, identification, etc.
  - **GPU friendly** by construction, which is important for big data processing
- Machine-learning algorithms use statistics to find patterns in massive amounts of data
- Our task is a supervised learning problem (regression)

Problem statement



An example of a positron event with deposited energy ∼**6 MeV**. The grey sphere — the primary vertex.

Charge at PMT

First Hit Time (FHT) at PMT

We want to reconstruct:

Deposited energy E_{dep} with resolution 3% @ 1 MeV

Datasets

- Two datasets: for training and for testing
- generated by the Monte Carlo method
- full detector and electronics simulation
- using the official JUNO software

Data description:

1. positron events
2. uniformly spread in the volume of the central detector
3. $E_{\text{kin}} \in [0, 10]$ MeV. $E_{\text{dep}} = E_{\text{kin}} + 1.022$ MeV

- **Training dataset:**
  4. **5 million** events
  5. uniformly distributed in kinetic energy E_{kin}
- **Testing dataset:**
  4. subsets with discrete kinetic energies:
  5. 0, 0.1, 0.3, 0.6, 1, 2, ..., 10 [MeV]
  6. $\sum =$ **1.4 million** events: each subset contains 100k

Aggregated features

We use aggregated information from the whole array of PMTs as features for models:

1. `AccumCharge` — the accumulated charge on fired PMTs
2. `nPMTs` — the total number of fired PMTs
3. Coordinates of the center of charge:

$$(x_{\text{cc}},\ y_{\text{cc}},\ z_{\text{cc}}) = \vec{r}_{\text{cc}} = \frac{\sum_{i=1}^{N_{\text{PMTs}}} \vec{r}_{\text{PMT}_i} \cdot n_{\text{p.e.},i}}{\sum_{i=1}^{N_{\text{PMTs}}} n_{\text{p.e.},i}}$$

   and its radial component: $R_{\text{cc}} = |\vec{r}_{\text{cc}}|$

4. Coordinates of the center of FHT:

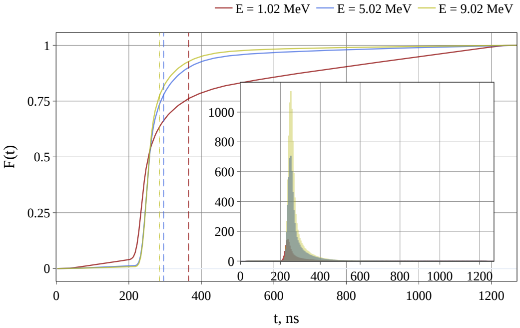
$$(x_{\text{cht}},\ y_{\text{cht}},\ z_{\text{cht}}) = \vec{r}_{\text{cht}} = \frac{1}{\sum_{i=1}^{N_{\text{PMTs}}} \frac{1}{t_{\text{ht},i}+c}} \sum_{i=1}^{N_{\text{PMTs}}} \frac{\vec{r}_{\text{PMT}_i}}{t_{\text{ht},i}+c},$$

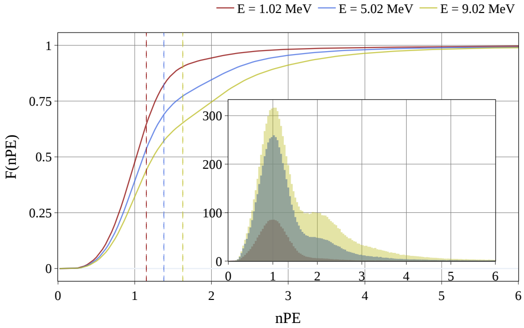
   and its radial component: $R_{\text{cht}} = |\vec{r}_{\text{cht}}|$

5. $\gamma_z^{\text{cc}} = \frac{z_{\text{cc}}}{\sqrt{x_{\text{cc}}^2+y_{\text{cc}}^2}}$
6. $\gamma_y^{\text{cc}} = \frac{y_{\text{cc}}}{\sqrt{x_{\text{cc}}^2+z_{\text{cc}}^2}}$
7. $\gamma_x^{\text{cc}} = \frac{x_{\text{cc}}}{\sqrt{z_{\text{cc}}^2+y_{\text{cc}}^2}}$
8. $\theta_{\text{cc}} = \arctan \frac{\sqrt{x_{\text{cc}}^2+y_{\text{cc}}^2}}{z_{\text{cc}}}$
9. $\phi_{\text{cc}} = \arctan \frac{y_{\text{cc}}}{x_{\text{cc}}}$
10. $J_{\text{cc}} = R_{\text{cc}}^2 \cdot \sin\theta_{\text{cc}}$
11. $\rho_{\text{cc}} = \sqrt{x_{\text{cc}}^2 + y_{\text{cc}}^2}$
12. with 7 similar features for the components of the center of FHT

Aggregated features

13. Percentiles of FHT and charge distributions:
    - $\{ht_{2\%}, ht_{5\%}, ht_{10\%}, ht_{15\%}, \ldots, ht_{90\%}, ht_{95\%}\}$
    - $\{pe_{2\%}, pe_{5\%}, pe_{10\%}, pe_{15\%}, \ldots, pe_{90\%}, pe_{95\%}\}$
14. Differences between percentiles for FHT:
    - $\{ht_{5\%-2\%}, ht_{10\%-5\%}, \ldots, ht_{95\%-90\%}\}$
15. Moments for FHT and charge distributions:
    - $\{ht_{\text{mean}}, ht_{\text{std}}, ht_{\text{skew}}, ht_{\text{kurtosis}}\}$
    - $\{pe_{\text{mean}}, pe_{\text{std}}, pe_{\text{skew}}, pe_{\text{kurtosis}}\}$





CDFs and PDFs for FHT (left) and charge (right) distributions. $R \simeq 0$ m, E_{kin} varied. Dashes lines show mean values.

Models description: BDT

A Decision Tree (DT) takes a set of input features and splits input data recursively based on those features.

Boosted Decision Trees (BDT):

- Ensemble model
- DT as base algorithm
- DTs in BDT are trained sequentially
- Each subsequent DT is trained to correct errors of previous DTs in the ensemble

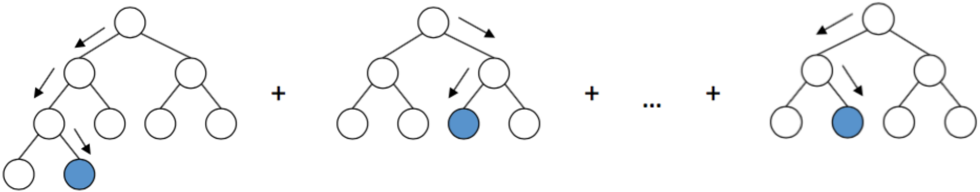


Figure: BDT demonstration. Source: https://arogozhnikov.github.io/

BDT: hyperparameters and benefits

Main tunable hyperparameters:

- **Max. depth**: The maximum depth of a tree (usually <12)
- **Learning rate**: This determines the impact of each tree on the final outcome (usually ≈ 0.1)
- **Number of trees**: How many trees in ensemble

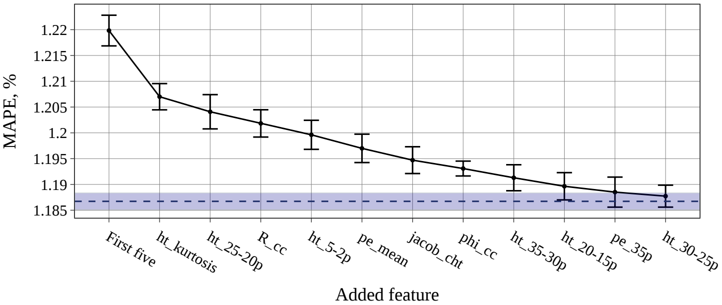
Benefits:

- Fast for training and prediction
- Easier to tune
- Minimalistic

BDT from XGBoost:

- Optimized **set of features** (sorted by *importance*):

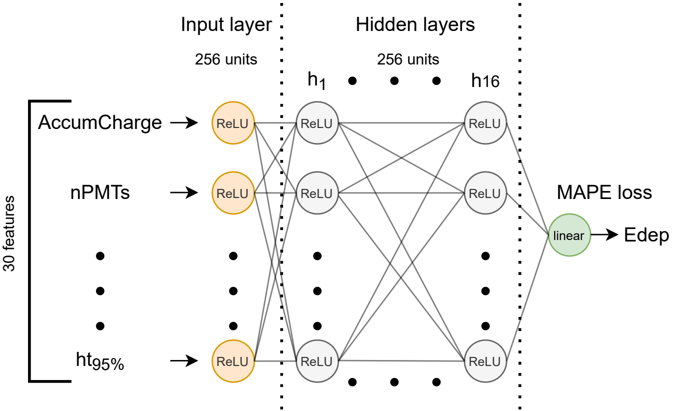
1. `AccumCharge`
2. R_{cht}
3. z_{cc}
4. pe_{std}
5. `nPMTs`
6. $\text{ht}_{\text{kurtosis}}$
7. $\text{ht}_{25\%-20\%}$
8. R_{cc}
9. $\text{ht}_{5\%-2\%}$
10. pe_{mean}
11. J_{cht}
12. ϕ_{cc}
13. $\text{ht}_{35\%-30\%}$
14. $\text{ht}_{20\%-15\%}$
15. $\text{pe}_{35\%}$
16. $\text{ht}_{30\%-25\%}$



- Optimized **hyperparameters** (using Grid Search):
  1. The maximum depth of the tree: 10
  2. Number of trees in the ensemble: ≃500
  3. Learning rate: 0.08

Models description: FCDNN

Fully-connected deep neural network (**FCDNN**):



- The search for hyperparameters was performed using *BayesianOptimizer*
- Training with *early stopping*
- Validation dataset: *400k events*
- *Selected features* provided the same performance as full set:

1. AccumCharge
2. nPMTs
3. R_{cc}
4. R_{cht}
5. ρ_{cc}
6. ρ_{cht}
7. pe_{mean}
8. pe_{std}
9. pe_{skew}
10. $pe_{kurtosis}$
11. Percentiles of FHT distribution: $\{ht_{2\%}, ht_{5\%}, ht_{10\%}, ht_{15\%}, ..., ht_{90\%}, ht_{95\%}\}$

Results

Metrics:

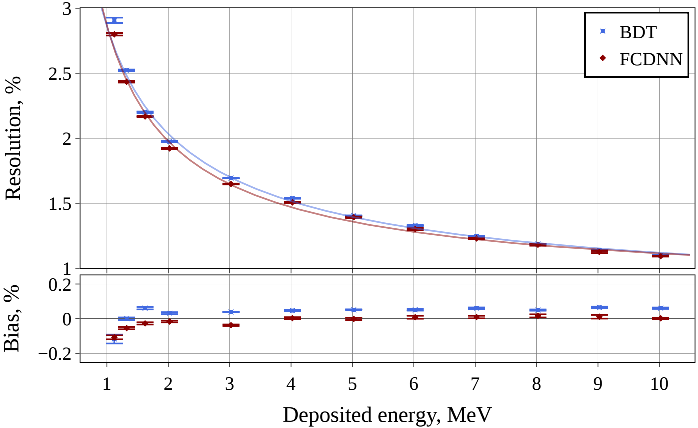
- Defined by a Gaussian fit of the $E_{\text{predicted}} - E_{\text{dep}}$ distributions
- *Resolution*: σ / E_{dep}, where σ — standard deviation of the fit
- *Bias* μ / E_{dep}, where μ — mean of the fit

Parameterization:

$$\frac{\sigma}{E_{\text{dep}}} = \sqrt{\left(\frac{a}{\sqrt{E_{\text{dep}}}}\right)^2 + b^2 + \left(\frac{c}{E_{\text{dep}}}\right)^2}$$

Models' pred. time and memory usage:

	BDT	FCDNN
Pred. time, sec/100k	3.5	17
Size, MB	50	12



Summary

- **Energy reconstruction** using the information collected by PMTs
- *Aggregated* features approach
- The following ML models are used: **BDT, FCDNN**
- As a result *achieved*:
  1. High **quality** 3% @ 1 MeV, requared for physics goals of JUNO
  2. Great **computation speed**, thanks to a small set of aggregated features (in $10^4 - 10^5$ times faster than traditional methods)

References and more details

Publications:

- **A. Gavrikov**, et al. arXiv: 2206.09040 (2022)
- **A. Gavrikov**, et al. EPJ Web Conf. 251 (2021), 03014
- Z. Qian, et al. NIM-A 1010 (2021), 165527