Machine Learning methods at the JUNO Neutrino physics experiment

Benoit Viaud

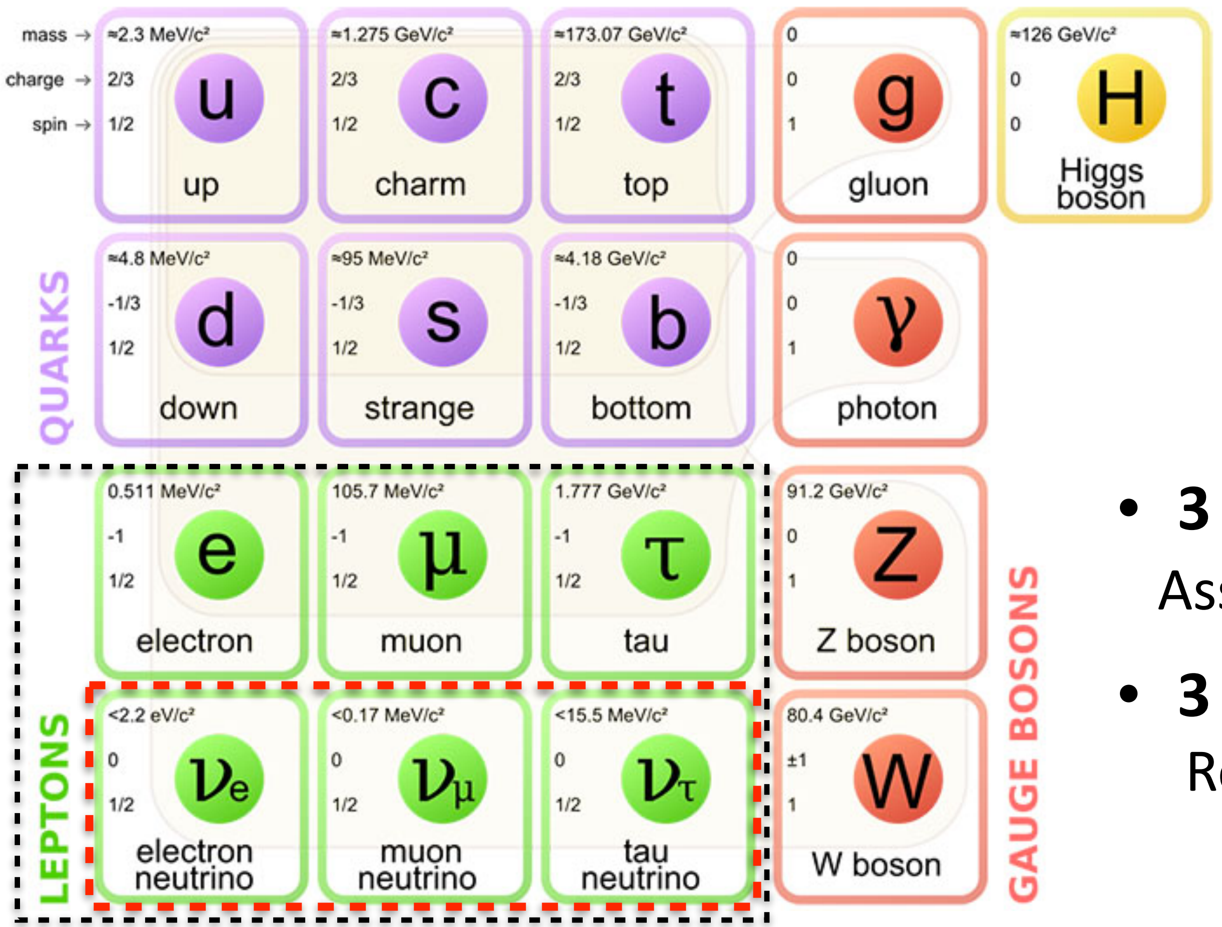
Subatech Laboratory (Nantes U., IMT Atlantique, CNRS)

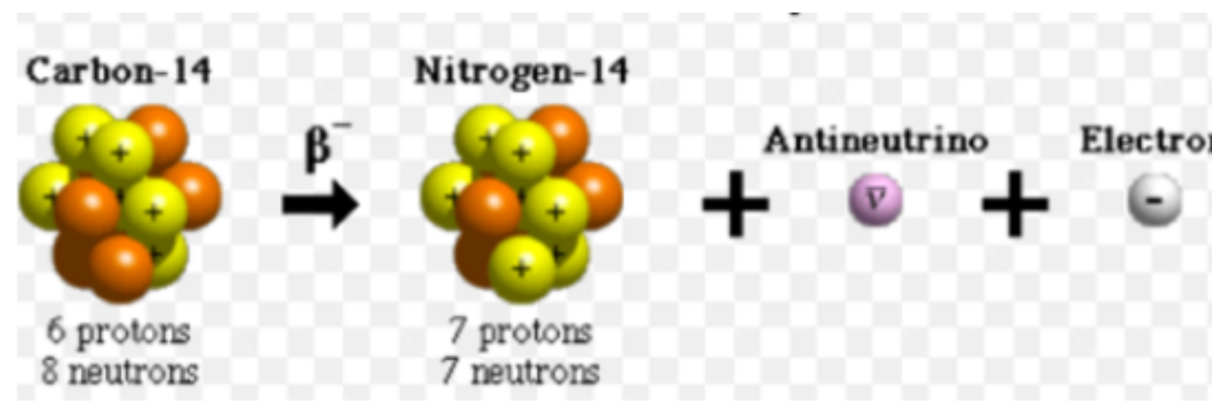
Speaking on behalf of
L. Imbert (Neutrino group @ Subatech)
F. Yermia (Neutrino group @ Subatech)
Gilles Grasseau (Calculus group @ Subatech)
and of
*The **JUNO** Collaboration*

Congrès général de la Société Fr. de physique, atelier MC10 "Physique et intelligence artificielle », 03-07-2023

Neutrino (ν) physics in less than a nutshell.

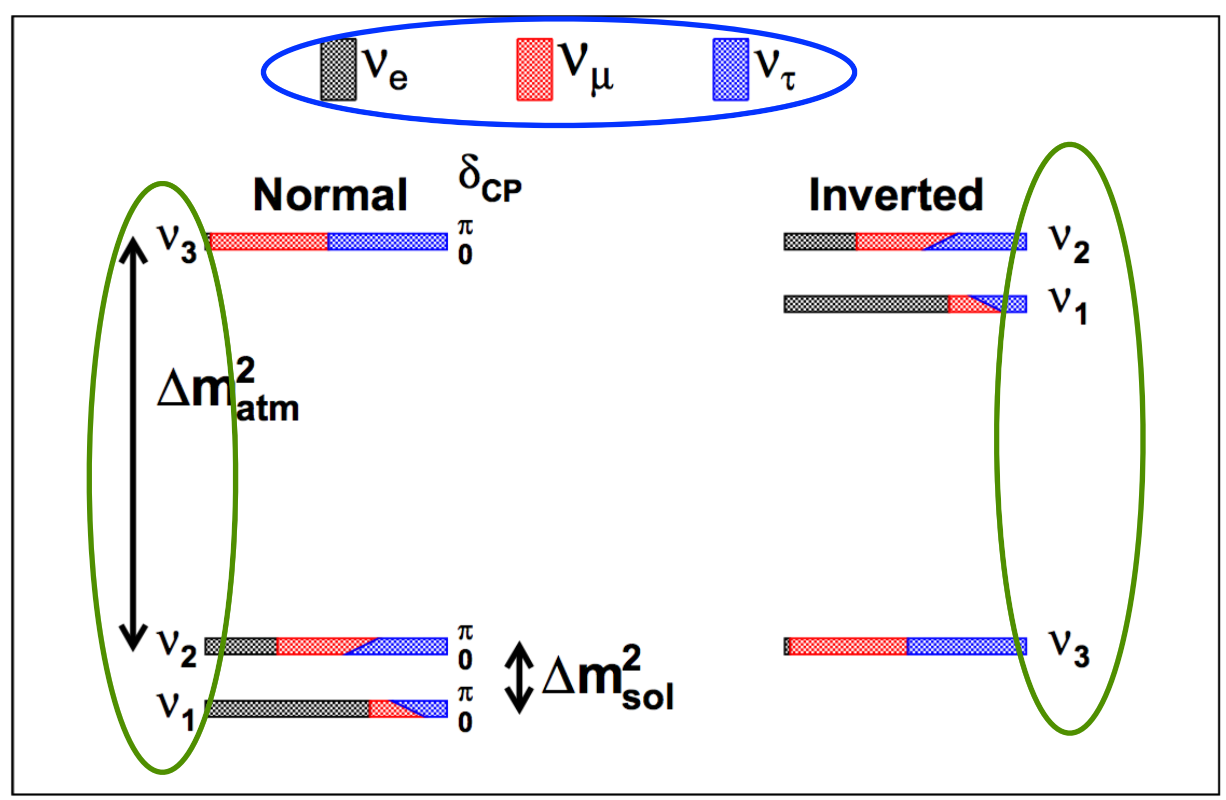
Known elementary particles (Standard Model)





- **3 known flavors of neutrinos**
  Associated to the lepton also produced when a ν is.
- **3 quantum mass states of neutrinos**
  Relationship with flavor states is not fully known.

Is ν_3 the heaviest or the lightest ?



- **Determining the Neutrino Mass Ordering (Normal or Inverted) is one of the hottest questions in particle physics.**
- **This is JUNO's main objective.**

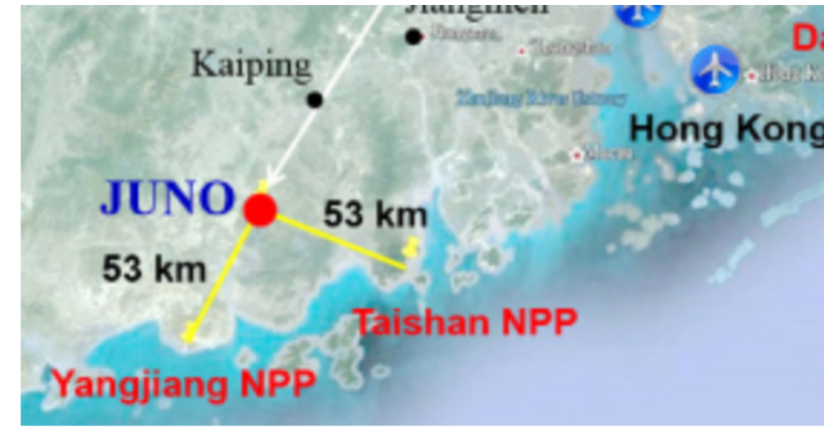
Neutrino Mass Ordering with the JUNO experiment.

- Exploit **Neutrino Oscillation** *Spontaneous change of flavor between creation & detection.*

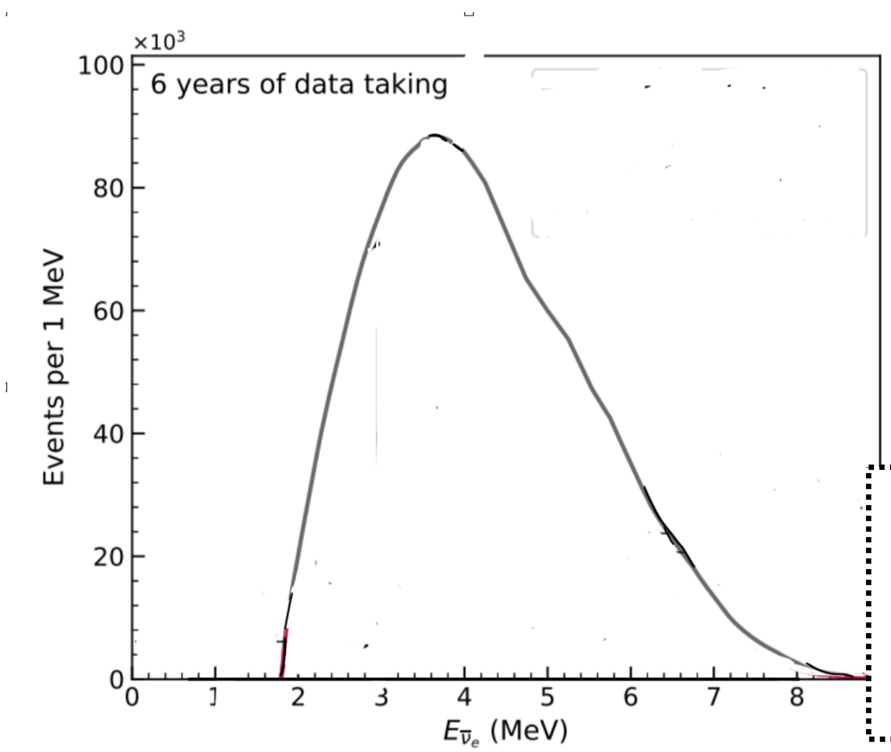
$$\nu_e \rightarrow \nu_\mu \; ; \; \nu_e \rightarrow \nu_\tau \; ; \; \nu_\mu \rightarrow \nu_e$$

2015

- Measure the **inprint of oscillation** on the **Energy spectrum** of antineutrinos produced by nuclear reactors.



Production of $\overline{\nu_e}$ in reactors



52.5 km

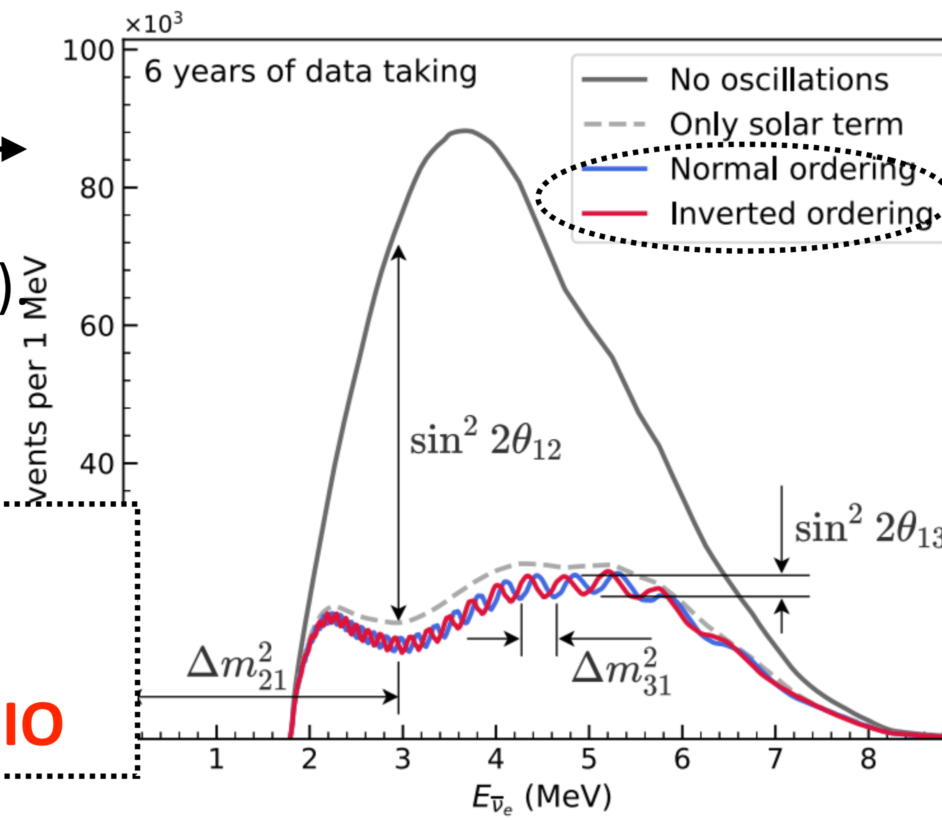
$\overline{\nu_e}$ disappear

(oscillate to other flavors)

$$\overline{\nu_e} \rightarrow \overline{\nu}$$

NMO determination:
detecting the very small
dephasing between **NO** and **IO**

Detection in JUNO



=> Necessits to reconstruct the Energy of the $\overline{\nu_e}$ with an **extreme** precision.

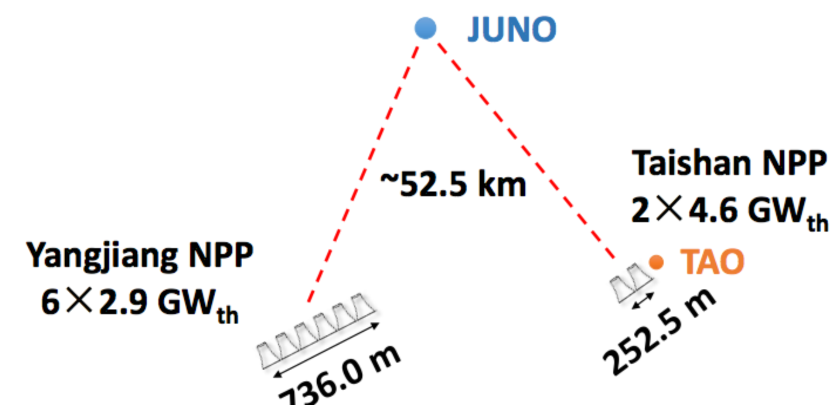
Neutrino Mass Ordering with the JUNO experiment.

- Exploit **Neutrino Oscillation** *Spontaneous change of flavor between creation & detection.*

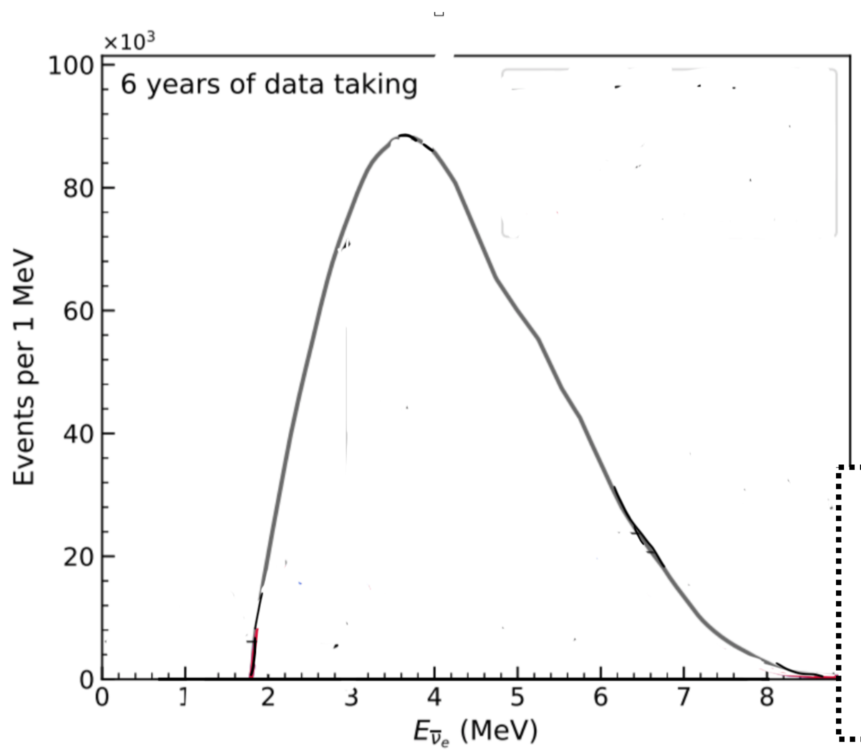
$$\nu_e \rightarrow \nu_\mu \; ; \; \nu_e \rightarrow \nu_\tau \; ; \; \nu_\mu \rightarrow \nu_e$$

2015

- Measure the **inprint of oscillation** on the **Energy spectrum** of antineutrinos produced by nuclear reactors.



Production of $\overline{\nu}_e$ in reactors



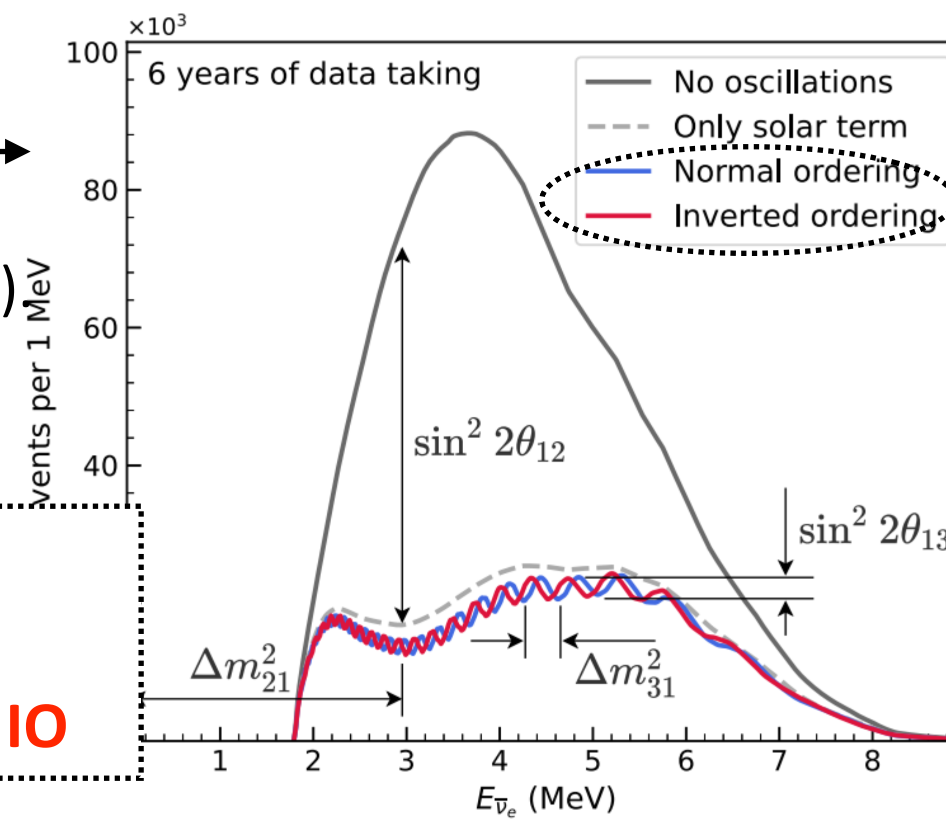
52.5 km

$\overline{\nu}_e$ disappear

(oscillate to other flavors)

$$\overline{\nu}_e \rightarrow \overline{\nu}$$

Detection in JUNO

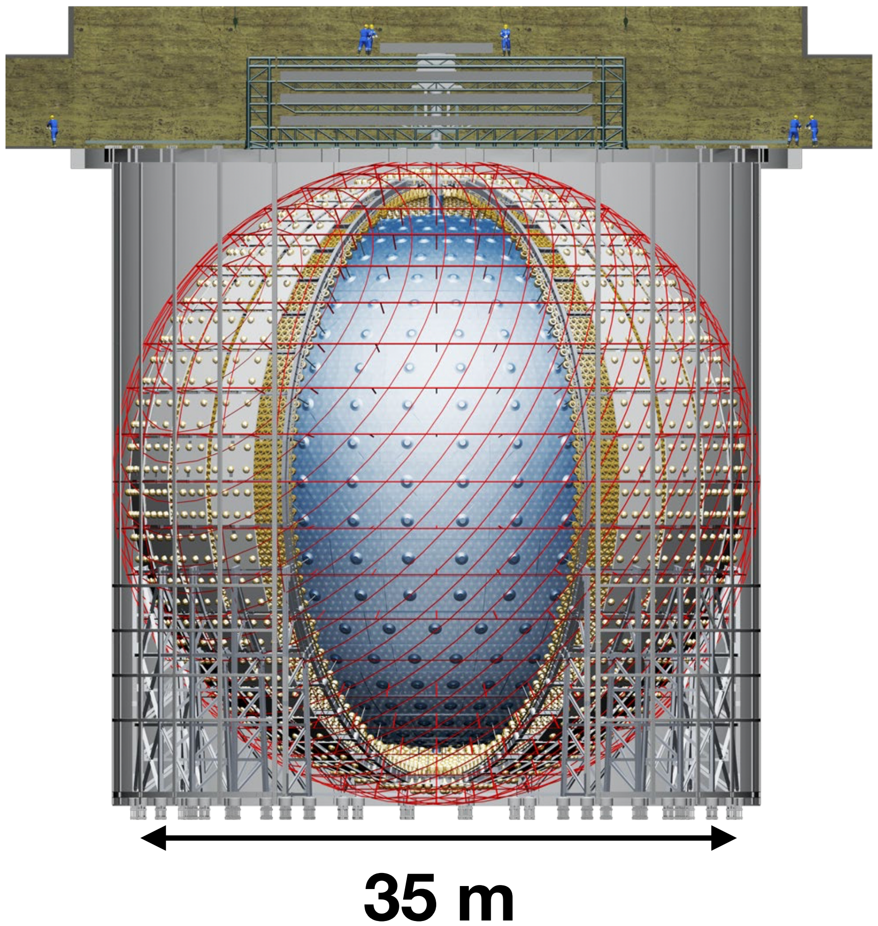


NMO determination:
detecting the very small dephasing between **NO** and **IO**

=> Necessits to reconstruct the Energy of the $\overline{\nu}_e$ with an **extreme** precision.

The Jiangmen Underground Neutrino Observatory (JUNO)

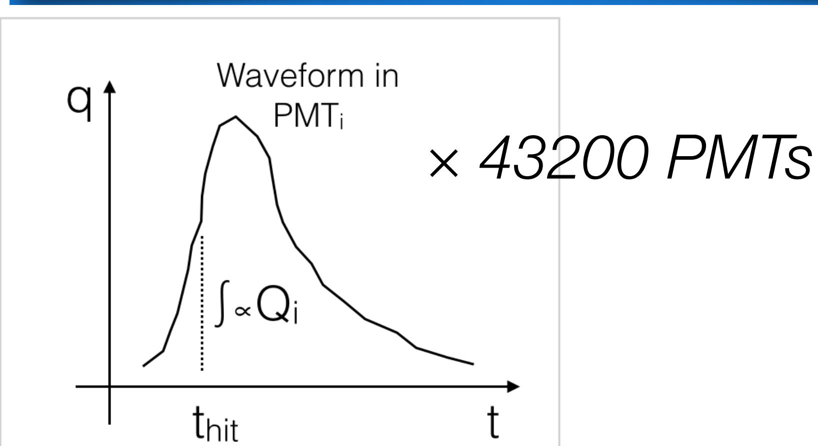
- One of the 4 major neutrino physics next generation experiments.

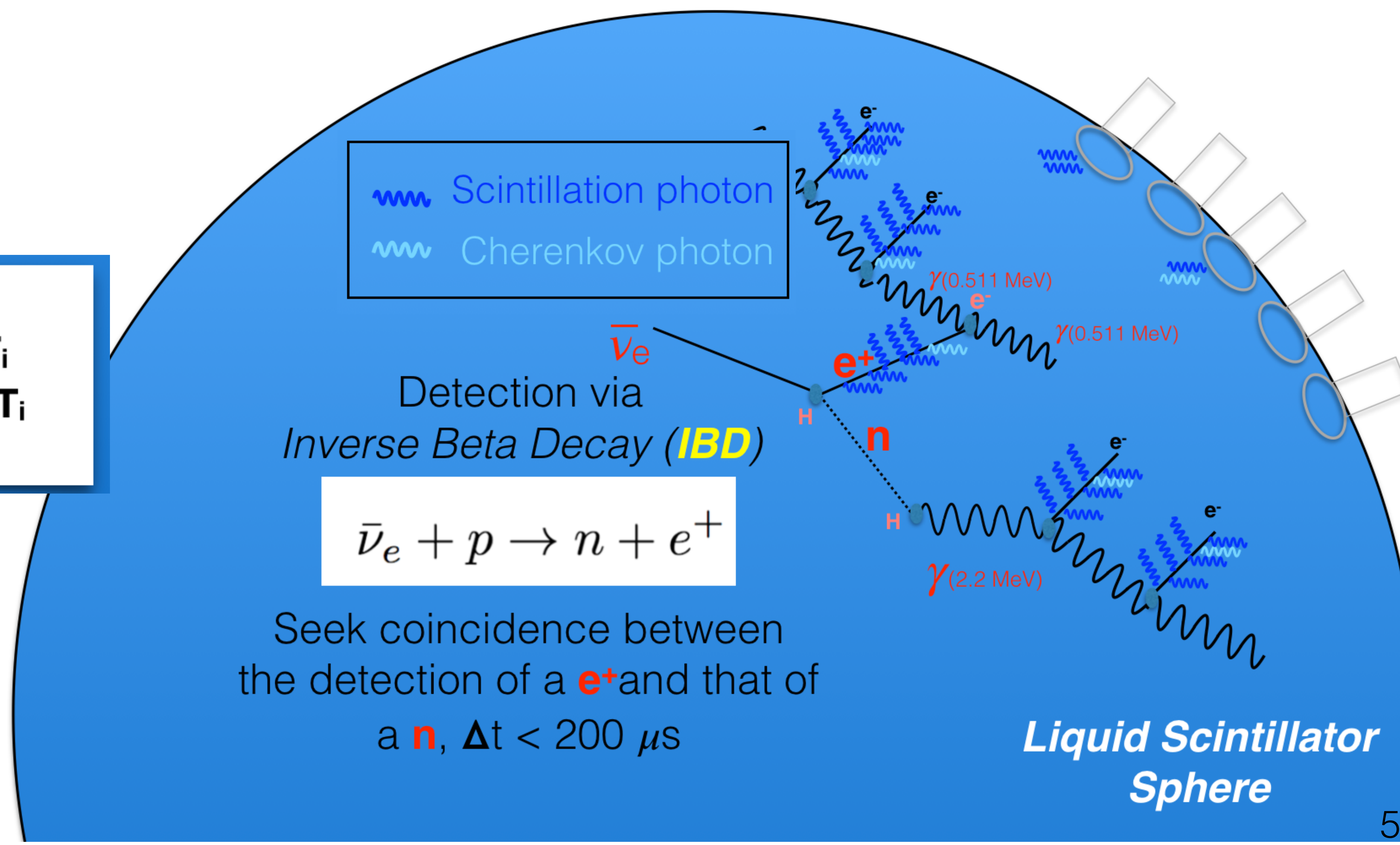


- A **35 m diameter** sphere filled with Liquid Scintillator (20 kt).
- Readout by a **double calorimetric system** :
  17600 20-inch and 25600 3-inch PMTs.
- Under construction in China, **700 m underground**.
  Data taking expected to start late 2024.

- **Reactor $\bar{\nu}_e$ measurement principle (goal: E_{IBD}, $\vec{X}_{IBD}$)**

Inputs to reco algorithms
- **Charge Q_i ∝ #Photons that hit PMT_i**
- **t_i ∝ Estimated time of 1st hit in PMT_i**
- **Other Waveform parameters....**





Classical Energy reconstruction at JUNO

▶ **Important feature in subatomic physics:** measurements most often rely on the comparison of data with models stemming from very detailed and realistic simulations.

- Tuned and/or complemented with real data control samples (e.g. from calibration sources).

  (I'll refer to ''Modelled'' data in this presentation)

▶ In JUNO, a rather ''simple", homogenous detector, this allows to predict the distribution of **charge** & **hit times** all over the PMTs, given the true E & Position of the IBD.

=> *We can regress the Energy by maximising a Likelihood based on those probabilities.*

$$\mathcal{L}(q_1, q_2, ..., q_N; t_{1,r}, t_{2,r}, ..., t_{N,r} | \mathbf{r}, t_0, E_{\text{vis}}) =$$

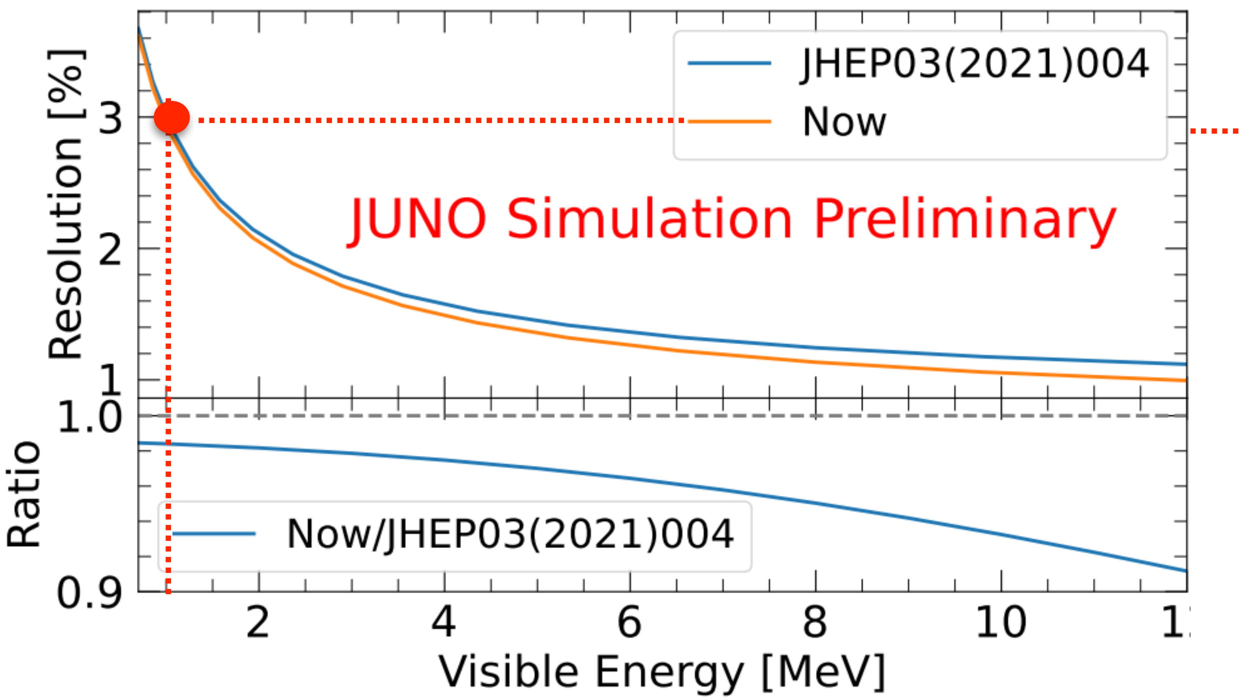
$$\prod_{unfired} e^{-\mu_j} \prod_{fired} \left(\sum_{k=1}^{+\infty} \underbrace{P_Q(q_i|k) \times P(k, \mu_i)}_{\textit{PDF of Charge } Q_i \textit{ distribution}} \right) \prod_{T-\text{valid } hit} \left(\underbrace{\frac{\sum_{k=1}^{K} P_T(t_{i,r}|r, d_i, \mu_i^l, \mu_i^d, k) \times P(k, \mu_i^l + \mu_i^d)}{\sum_{k=1}^{K} P(k, \mu_i^l + \mu_i^d)}}_{\textit{PDF of } t_i \textit{ distribution.}} \right)$$

Classical Energy reconstruction at JUNO

- Classical methods use low level data (Q_i, t_i) from all PMTs , make minimal assumptions.

  => *The loss of detector information and generality is small*

  => *Classical reconstruction performs very well.*



The necessary performance for a 3σ sensitivity to NMO in 6 years of data taking.

$$\frac{\sigma_E}{E} = 3\% \quad @ \quad E = 1\ MeV$$

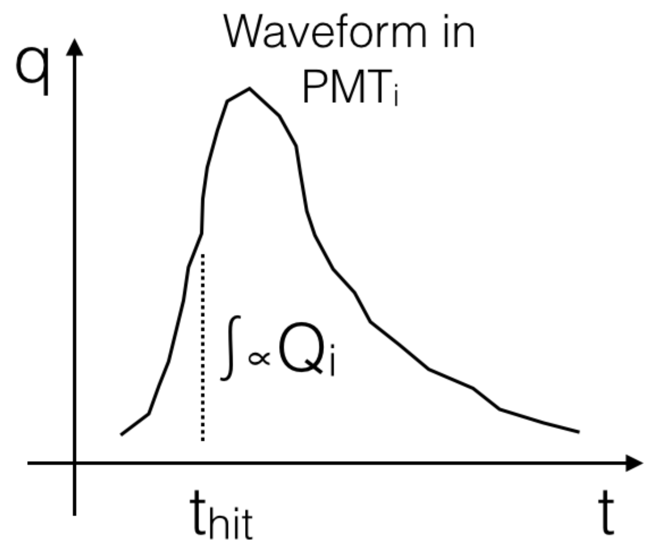
=> *A challenge for machine learning methods to do better.*

Motivation for ML methods

- Resolution and bias in the E_{IBD} reconstruction must be understood very precisely.

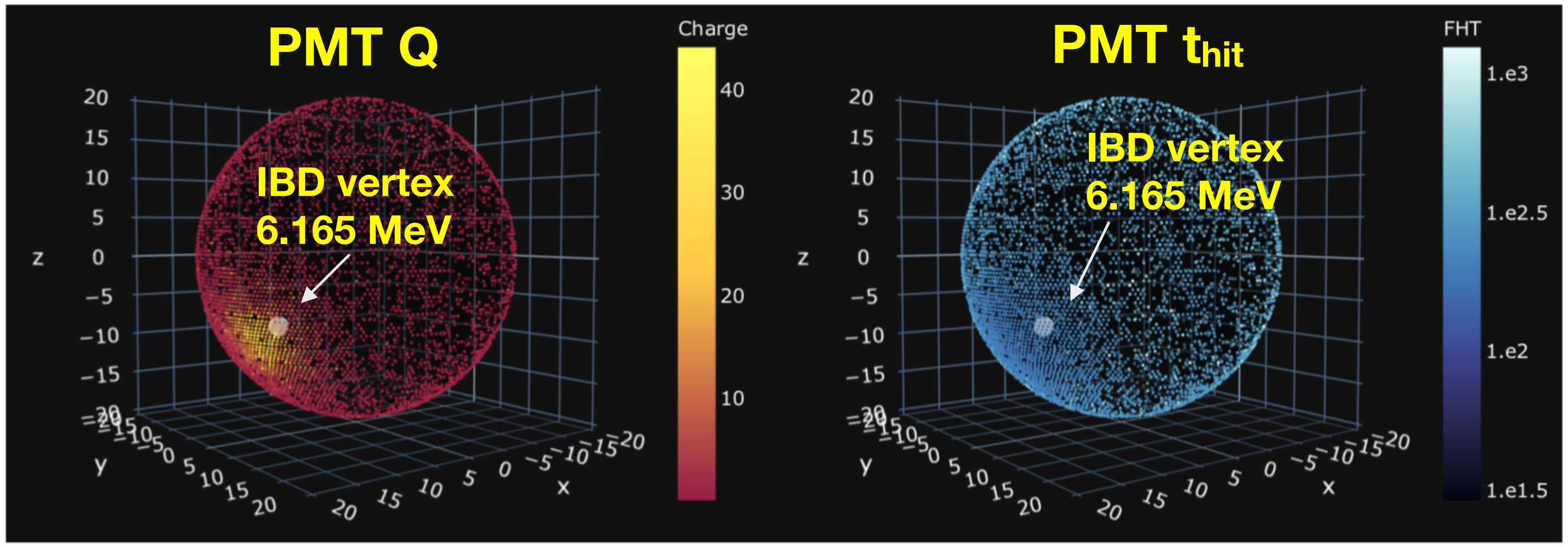
  => *Alternative methods brings robustness: not all depend the same way on mismodelling effects.*

- DL methods: Might use information classical methods don't;
  Might rely less on assumptions.
  - Most often use low level signal : (Q,t) from every PMTs
  - Or even the lowest : full Waveform information.
  - Besides a potential gain on resolution and bias : execution speed !



- **Tempting** to use DL since **JUNO events look like (spherical) images**.

Can we benefit of the advances that occured over the last decade in image recoginition ?

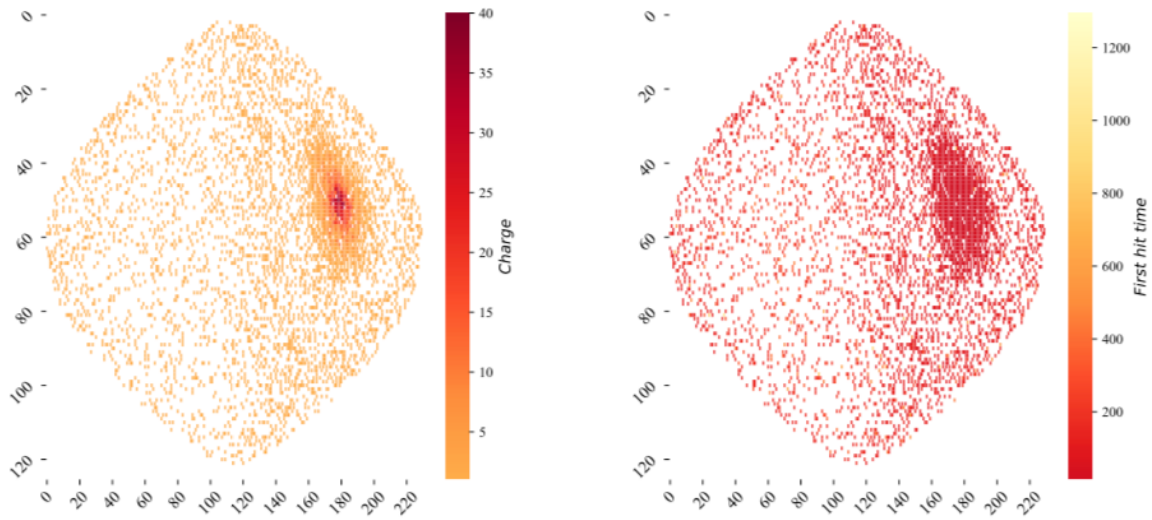


- Presented today: some methods applied to reactor antineutrino reconstruction.

Convolutional Neural Networks

(arXiv:2101.04839v2 ; arXiv:2205.04039v1)

- Input to the CNN : N-channel image; each pixel is a PMT. Ex:



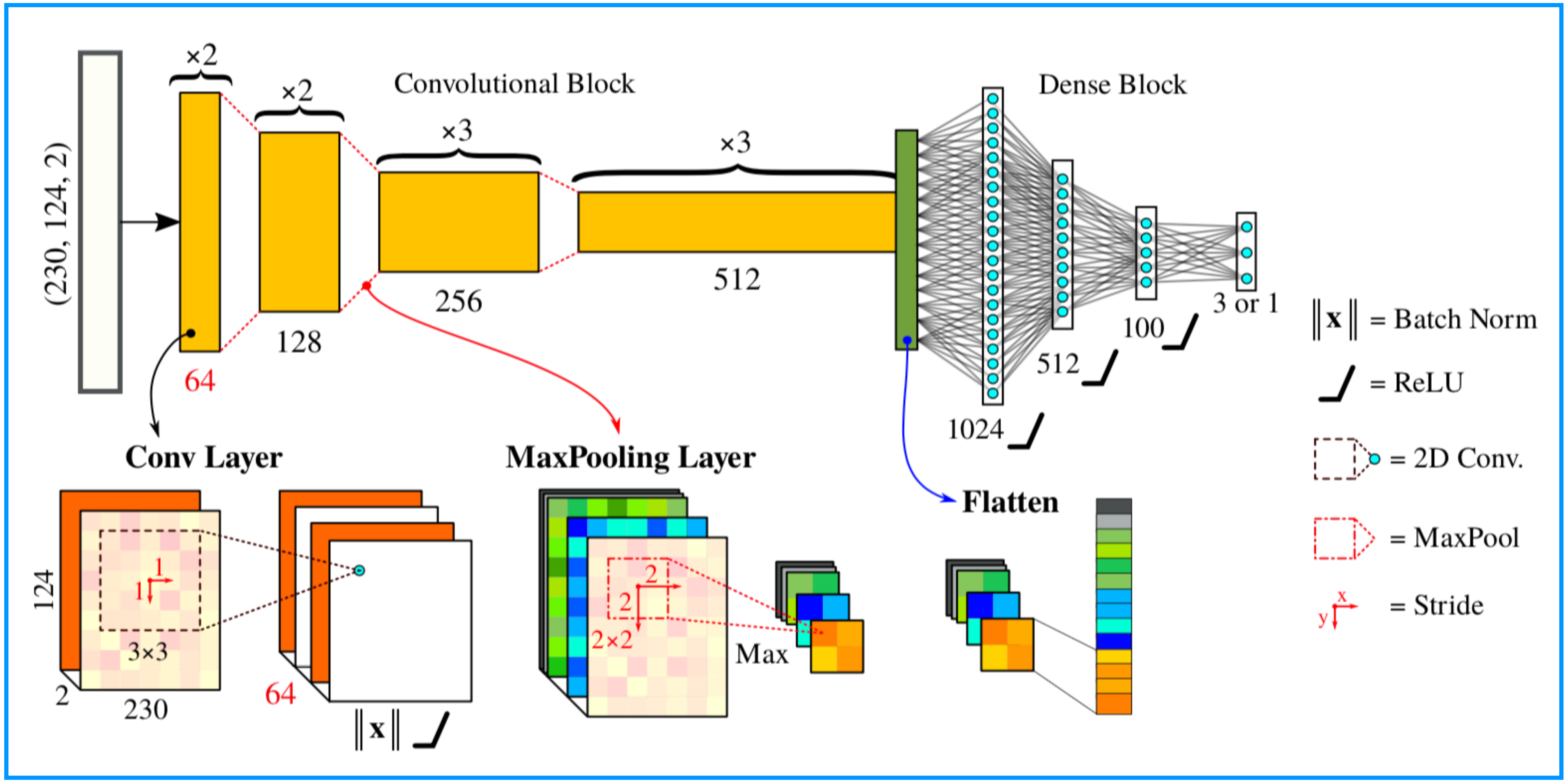
(b) Charge channel. **(c)** First hit time channel.

CNNs work on d-dimensional domains.

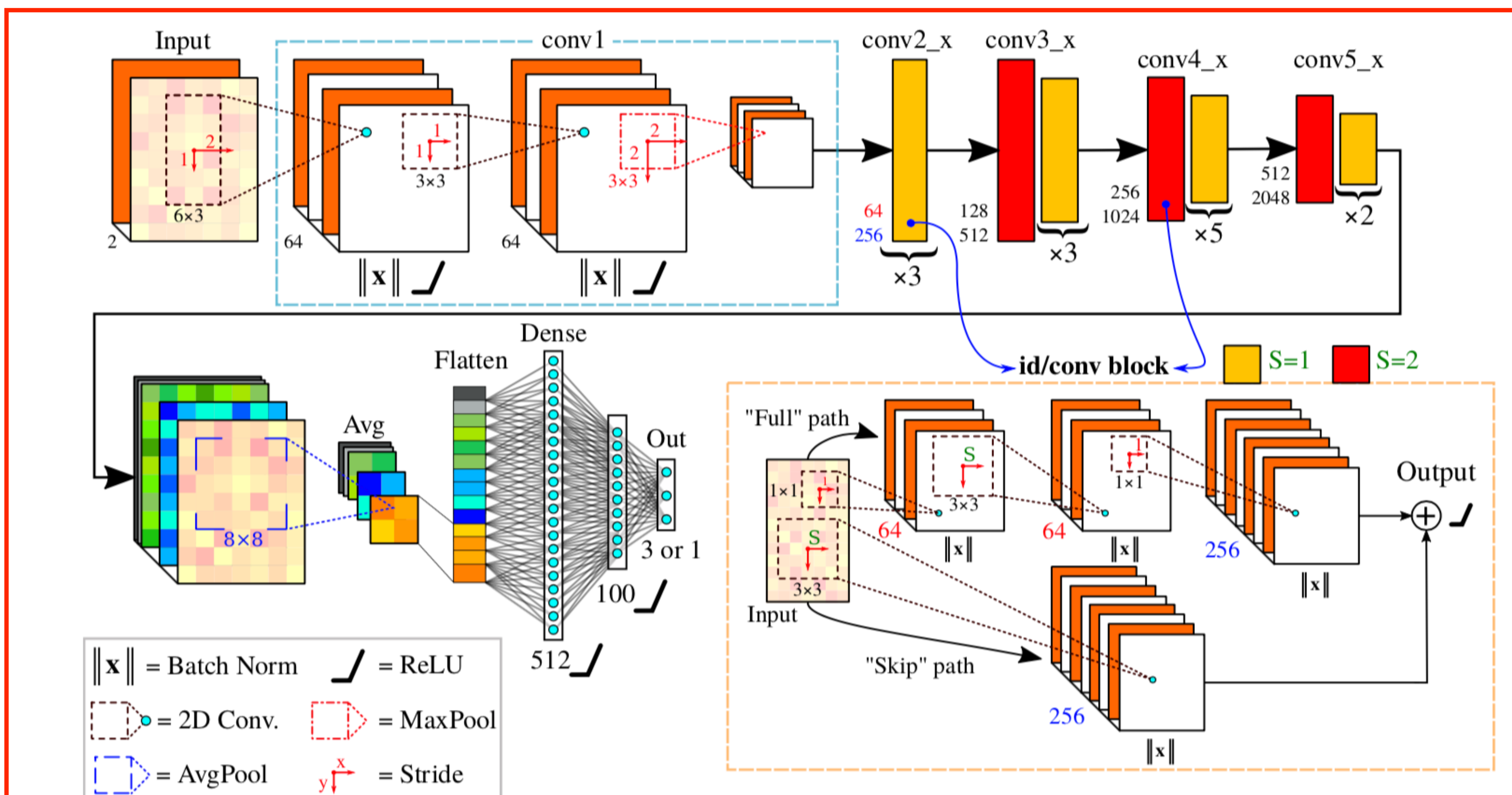
Spherical image -> planar image.
***Projection** conserving distance between PMTs.*

- **Strategy**: slightly adapt well established algorithms to JUNO : VGG-**J** and ResNet-**J**
  - One key question : necessary level of complexity ? (layers, parameters
  - Comparing these two algorithms answers it to a large extent.

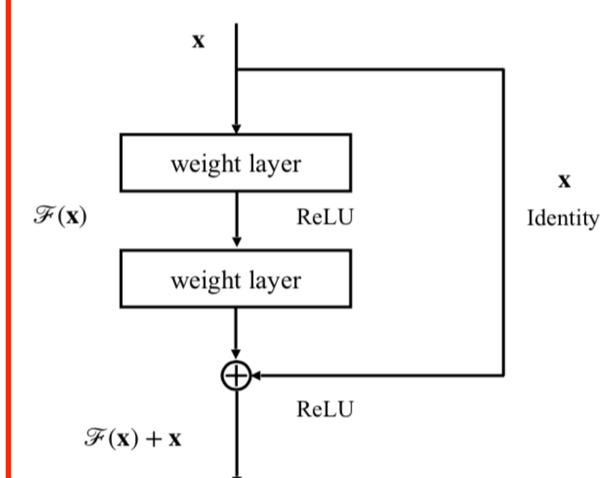
	VGG-J	ResNet-J
Layers	17	53
Parameters	26 310 035	38 352 403



VGG-J



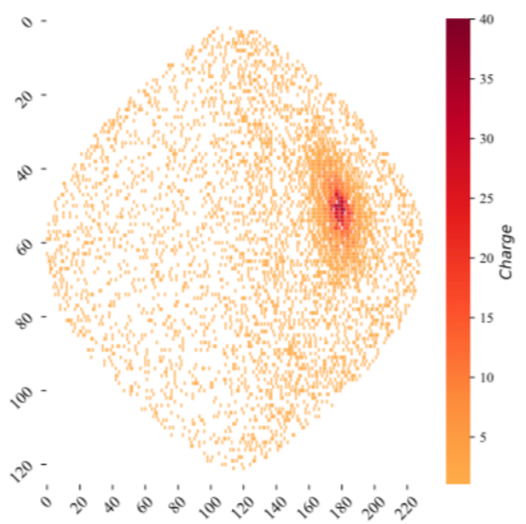
ResNet-J



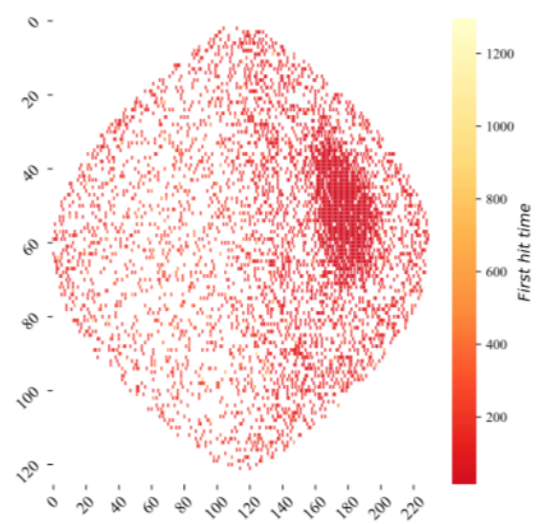
- Residual block structure in ResNet network.

Convolutional Neural Networks (arXiv:2101.04839v2 ; arXiv:2205.04039v1)

- Input to the CNN : N-channel image; each pixel is a PMT. Ex:



(b) Charge channel.



(c) First hit time channel.

CNNs work on d-dimensional domains.

Spherical image -> planar image.
***Projection** conserving distance between PMTs.*

- **Strategy**: slightly adapt well established algorithms to JUNO : VGG-**J** and ResNet-**J**
  - Optimisation 1 : Architecture complexity (e.g. # of layers & parameter
    These 2 algorithms vary a lot in this respect.
  - Optimisation 2 : Inputs !
    First version : 2-channel input (Q_1, t_{first})
    But : 2 types of Large PMTs in JUNO : separate them.
    Also use the time of the second it in each PMT.

6-channel input:
(Q_1, $t_{first,1}$, $t_{sec,1}$, Q_2, $t_{first,2}$, $t_{sec,2}$)

	VGG-J	ResNet-J
Layers	17	53
Parameters	26 310 035	38 352 403

Two types of 20-inch PMTs :
5000 Hamamastu Dynode PMT
12612 NNVT Micro-channel Plate (MCP)

	Dynode	**MCP**
Detection efficiency [%]	28.4	30.1
Dark noise rate [kHz]	15.3	29.6
Charge resolution [%]	27.9	32.9
Transit time spread [ns]	2.8	12.0

Convolutional Neural Networks (arXiv:2101.04839v2 ; arXiv:2205.04039v1)

- **Training details**: sample of 5 M e^+ interactions (full simulation of all phenomena in JUNO)
  - — Flat distribution in Energy - [0, 10] MeV - and position in Juno.
  - — 10 %used for validation
  - — 13, 10k events, testing samples (E = 0.3, 0.6, 1, 2, ..., 10 MeV)
  - — Tested several configurations of the readout electronics.

  Essentially valid for all algorithms presented today.

  - — Each version (e.g. hyperparameter configuration) completed in 4 days on a single V100 GPU.

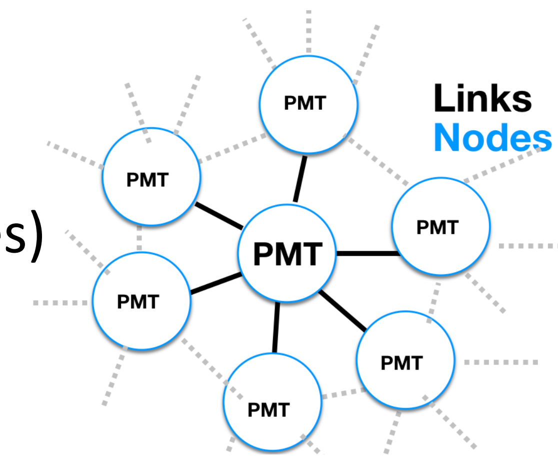
- **Performance**: similar to classical methods. (see synthesis slide later)

- **One caveat**: planar projection. Can we do better if we keep spherical ?

- **Graph structure**: more flexible way to combine Nodes information than CNN filters with Pixels.

  JUNO : the way PMTs' information is treated can be invariant under 'translations' over the sphere.

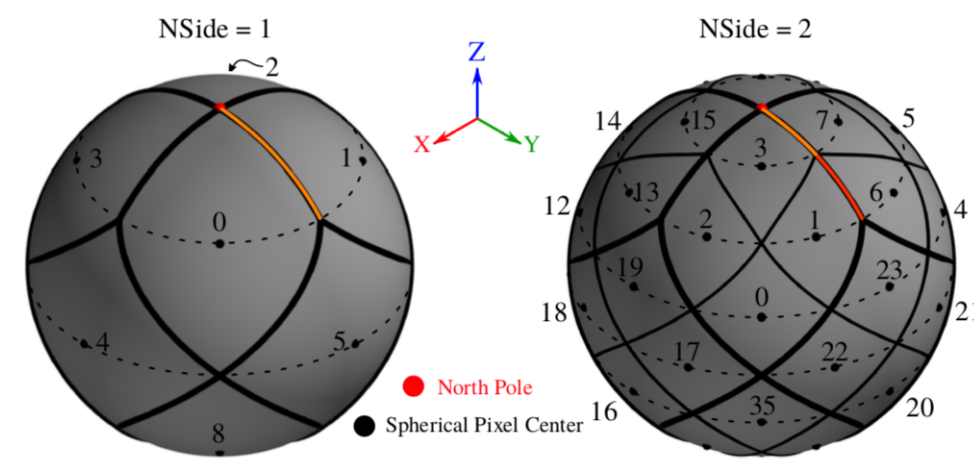
- **Computation limitation**: cannot link all 17600 LPMTs together or even have one PMT per node.



  • First layer : use HEALPix algorithm to define 3072 identical regions (pixels = nodes)

  5-6 LPMT per pixel

  HEALPix also convenient for pooling is subsequent layers.



- **Main characteristics**

  • Convolutional Graph NN based on DeepSphere and VGG-16

  Convolution : Chebyseh convolutional layers.

  • Nodes input feature :

  Total charge in each pixel (∑PMT)

  Earliest thit in the pixel.

  • One training : 22h on a single V100 GPU.

- **Performance**: similar to classical methods. (see synthesis slide later)

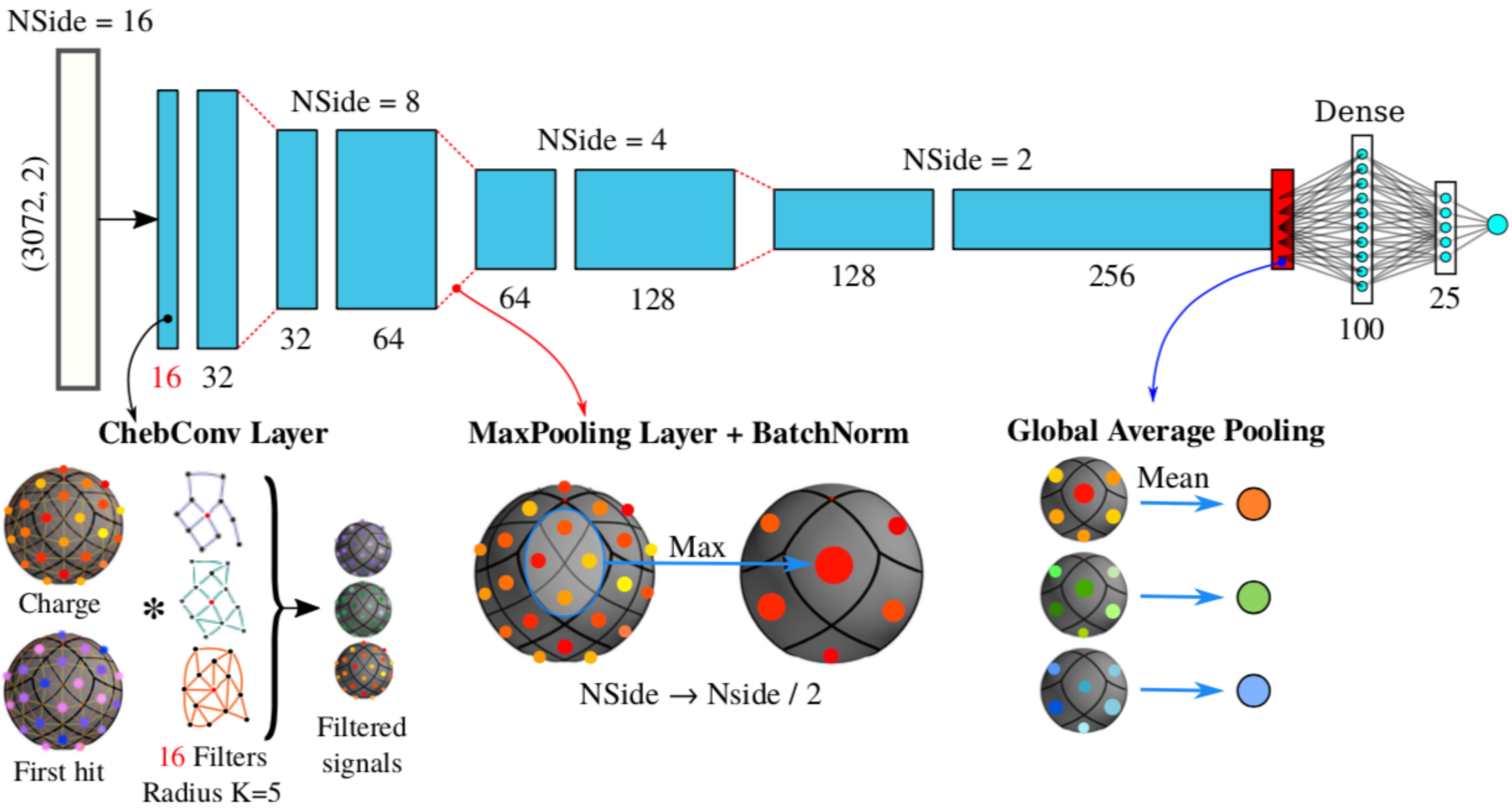
- **One limitation**: each node linked only to its direct neighbours ; weight based on distance between connected nodes.

$$W_{ij} = \exp\left(-\frac{\|\boldsymbol{v_i} - \boldsymbol{v_j}\|_2^2}{2\overline{d^2}}\right), \qquad \overline{d^2} = \frac{1}{|\mathcal{E}|}\sum_{(\boldsymbol{v_i},\boldsymbol{v_j})\in\mathcal{E}} \|\boldsymbol{v_i} - \boldsymbol{v_j}\|_2^2,$$

@ Subatech : development of a GNN inter-connecting nodes from all over the sphere.

Graph neural networks

(arXiv:2101.04839v2)



Methods with engineered inputs (iso Q & t from all PMTs) (arXiv:2206.09040v2)

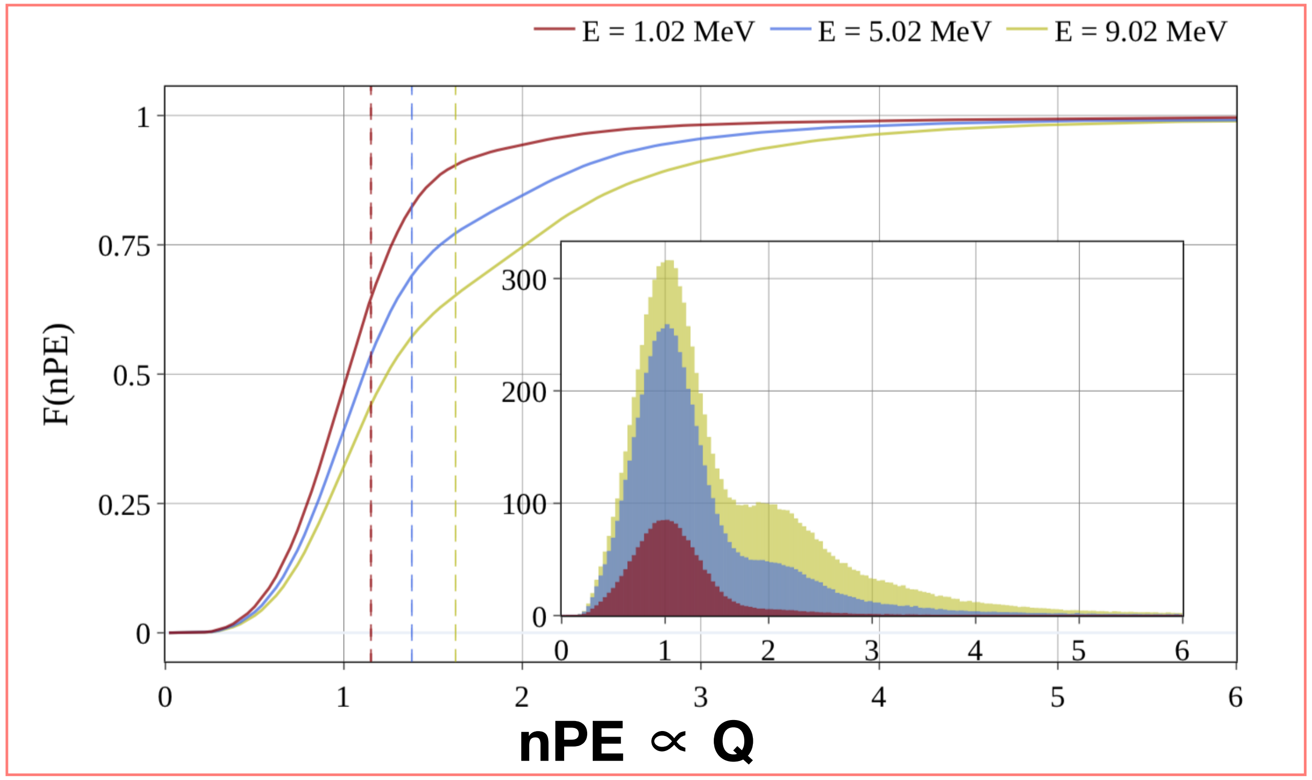
▶ Can the necessary info actually be contained in a small set of engineered variables ?

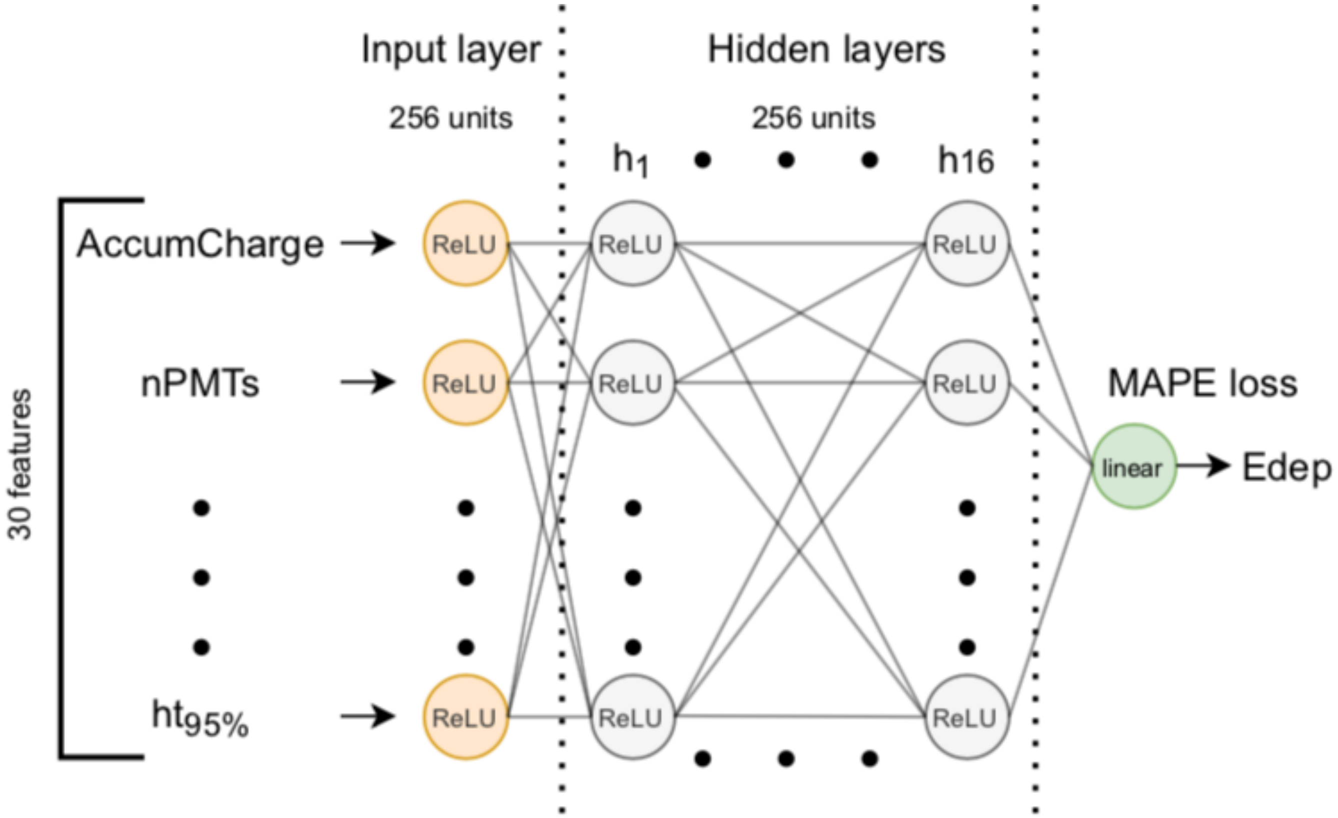
=> Designed 91 "aggregated" variables, correlated to the E_{IBD} & $\vec{X}_{IBD}$, based on the knowledge of what happens in JUNO when an IBD occurs (completed by simulation studies). Exemples :

- Total charge in the event + number of hit PMTs
  Quasi proportional to interaction E.
- Variables linked to X_{IBD} (Q & t barycenters of PMT positions)
  Helps to exploit the dependence on position.
- Percentiles of the distribution of the Q per PMT, and of the distribution of t_{hit}.
  Shaped by E and $\vec{X}$ of the interaction.

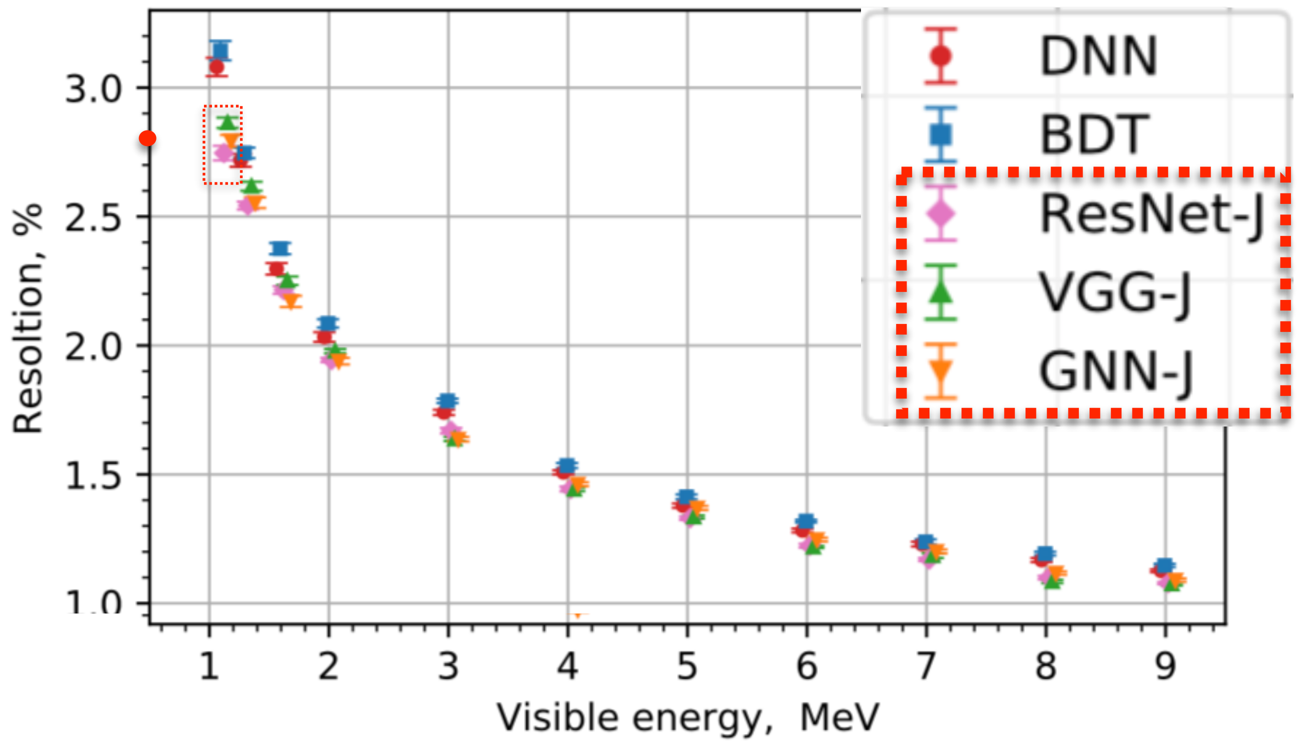
▶ One BDT & one <u>FCDNN</u> developed to exploit these variables.

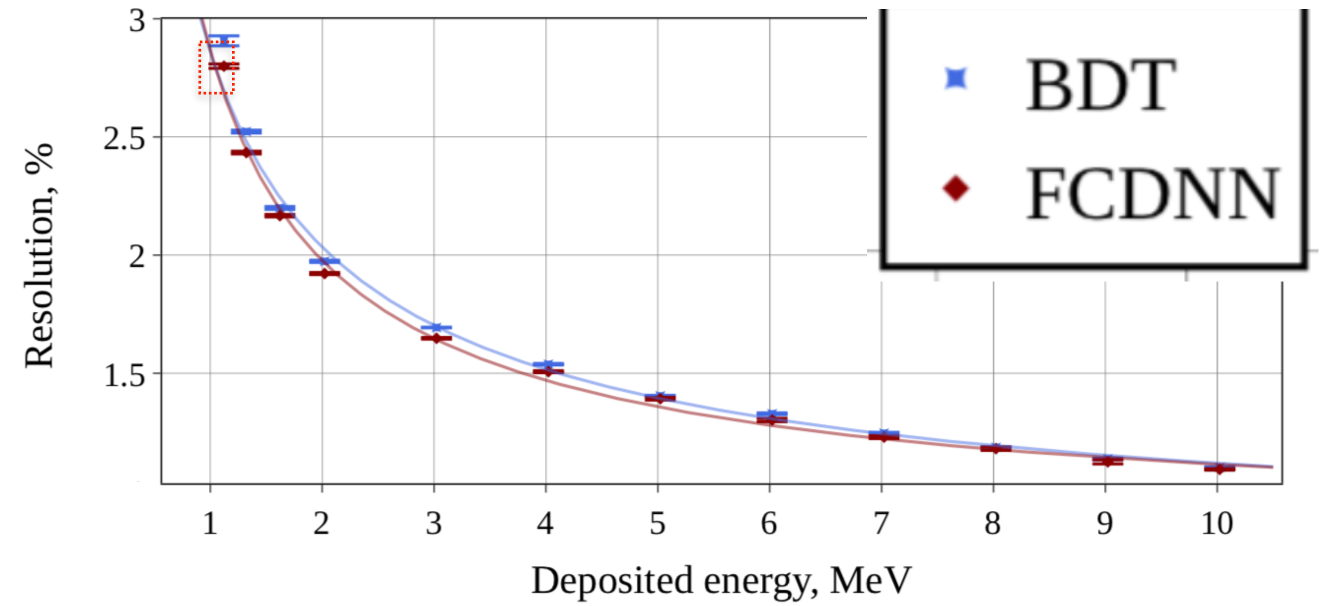
- Compared subsets of the 91 variables:
  Select best (30 variables)





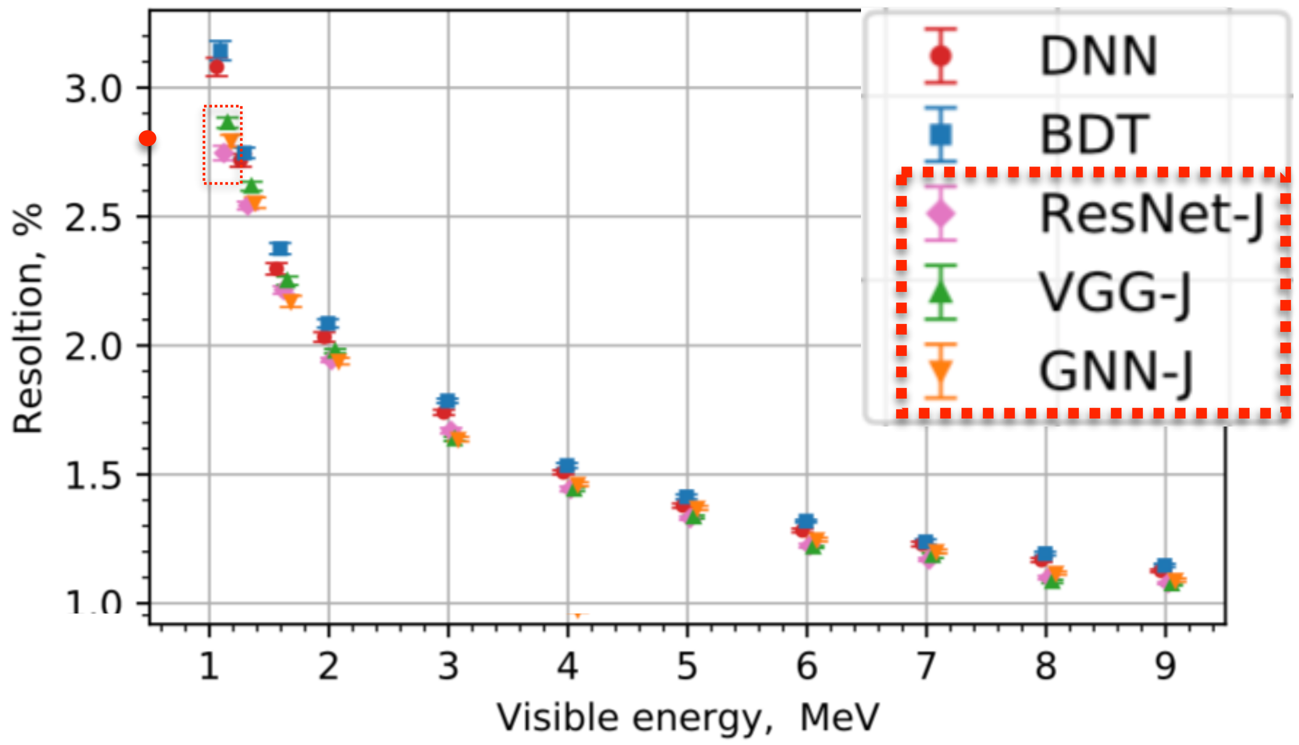
Performance and discussion

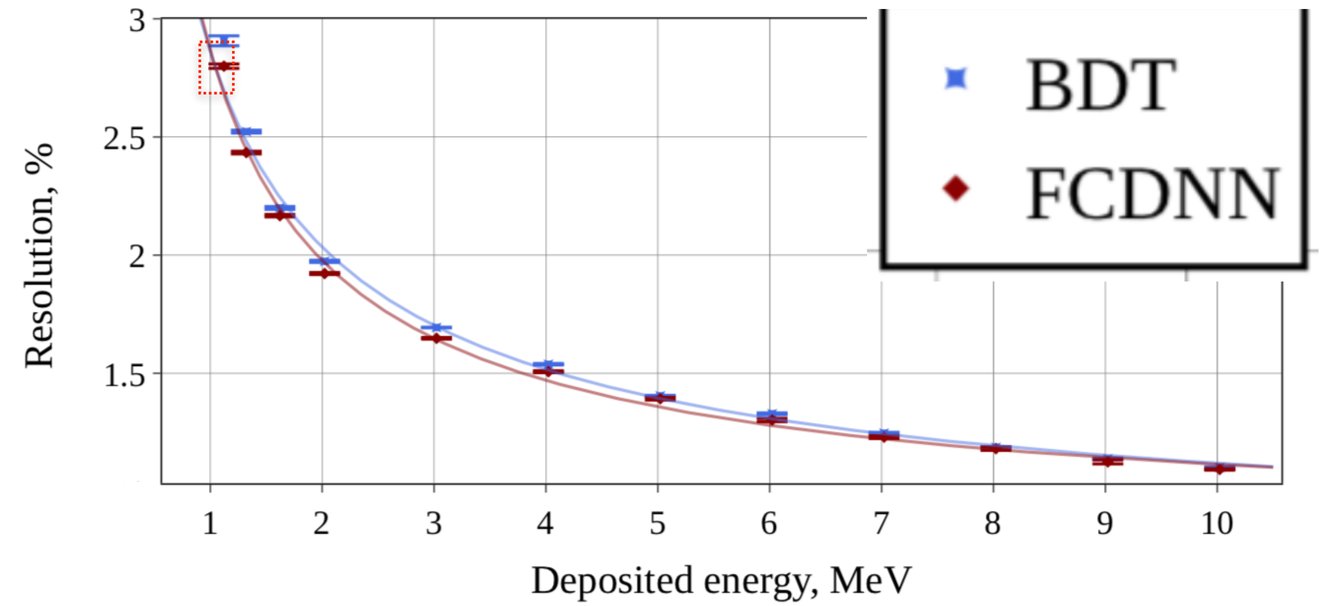




- ML reconstruction methods appear to reach the needed resolution $\frac{\sigma_E}{E} = 3\% \quad @ \quad E = 1\ MeV$
- Performance in the same ballpark as classical methods. Some hope to eventually do better.

Performance and discussion





- Now that the potential of ML is established : crucial to start working on **reliability**
  - So far, training samples from simulation. **Will be improved using real IBD-less data.**
    Calibration sources, beta decays from environmental radioactivity, …
  - There might be in such modelled samples information absent from IBD events in real data.
    If it is used by ML algorithms: potential biases in physics results !
  - Even more critical if we try to improve performance using the full Waveform information.
  - **Remember:** we need to understand the E spectrum **very** precisely.

    => Even subtle discrepancies between modelled and real data must be anticipitated.

ML reliability : first steps at JUNO.

1. Start with ways reconstruction biases are usually dealt with in HEP.

- **Probe the scale of the problem**: develop many methods, hope not all biased the same way.
  - Scale ~ Variation in results of the oscillation analysis, performed on the same IBD sample.

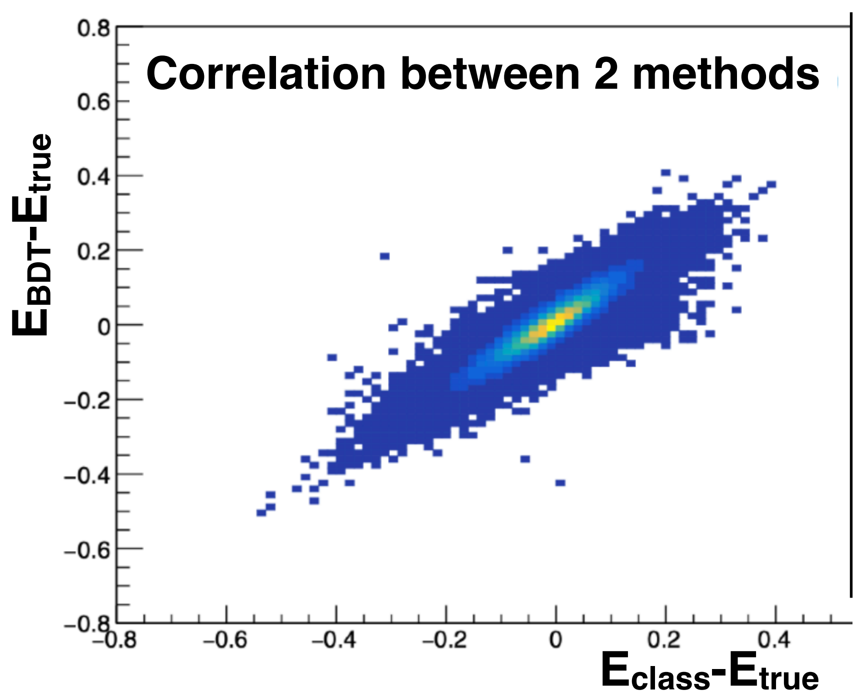
- **Test stability of ML methods vs. parameters of the simulation**
  - Varied within uncertainties evaluated after adjustements based on real data.
  - Re-training until independent from MC tuning.

- **Include real data control samples in the validation and/or training**
  - Ex: train on best modelled data, verify on calibration sources (E and X are known), retrain on them.

- **Seek where differences come from. Which information used by which method ?**
  - Overlap and differences between methods.
    Requires to include all methods in JUNO's software.
    (work on-going at Subatech)

  Use event per event comparison, to evaluate e.g. Mutual information between:

  - Energy estimators from various methods
  - Estimators and various engineered variables.



2. Develop ML methods to identify systematically scenarios a physicist might not think of ?

An Adversorial NN @ JUNO to explore ML reliability (under dev @ Subatech)

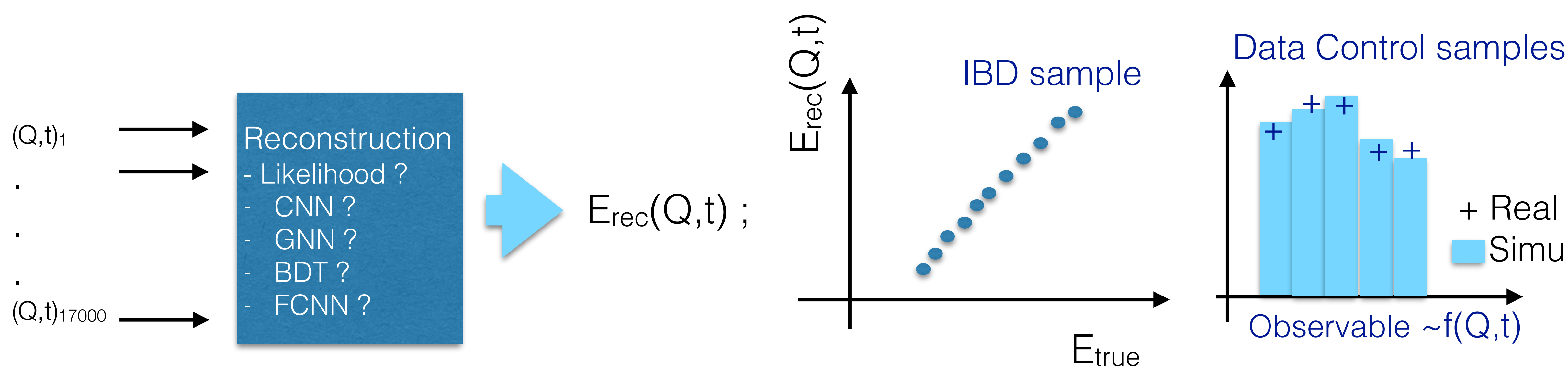
▶ **Aim**: incorporates the extensive real data samples used by JUNO to calibrate, tune/complement simulations, understand the detector…

- Calibration sources (radioactive decays with well known Energy and position) ;
- Background from natural radioactivity ;

… into an algorithm that automatically generates discrepancies that could still bias JUNO's results.

If these distortion patterns look physically sound => derive systematic uncertainties from this.
If none are found => a proof of robustness for the attacked reconstruction method.

▶ Adaptation of ''AI Safety for High Energy physics'', B. Nachman (LBL, Berkeley), C. Shimmin(Yale U.), arXiv:1910.08606



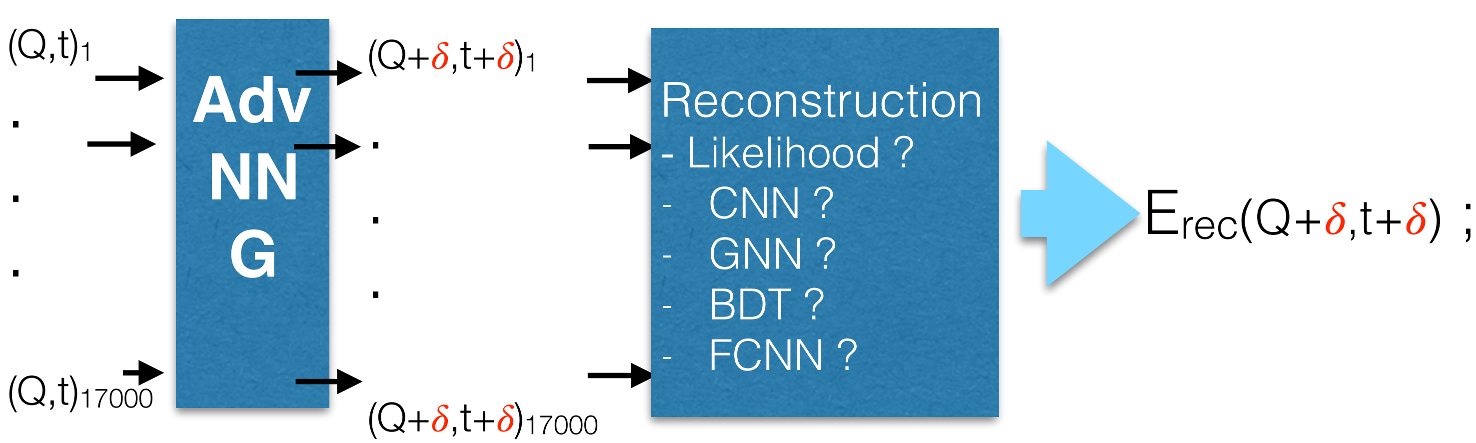
An Adversorial NN @ JUNO to explore ML reliability (under dev @ Subatech)

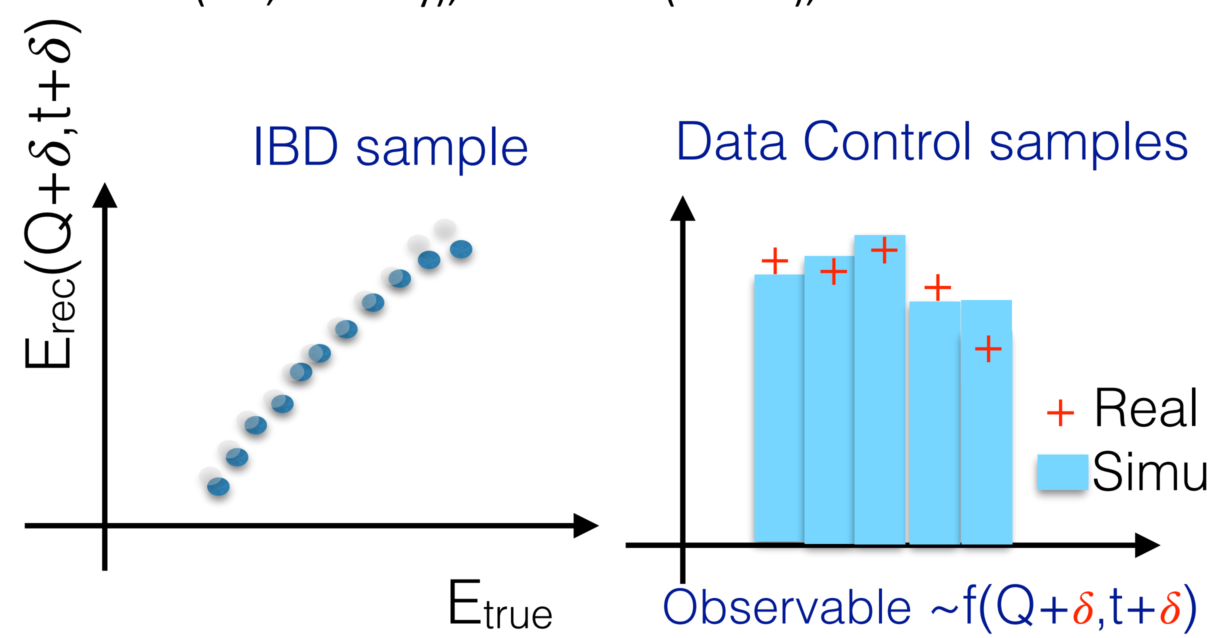
- **Aim**: incorporates the extensive real data samples used by JUNO to calibrate, tune/complement simulations, understand the detector…
  - Calibration sources (radioactive decays with well known Energy and position) ;
  - Background from natural radioactivity ;

… into an algorithm that automatically generates discrepancies that could still bias JUNO's results.

If these distortion patterns look physically sound => derive systematic uncertainties from this.
If none are found => a proof of robustness for the attacked reconstruction method.

- Adaptation of ''AI Safety for High Energy physics'', B. Nachman (LBL, Berkeley), C. Shimmin(Yale U.), arXiv:1910.08606

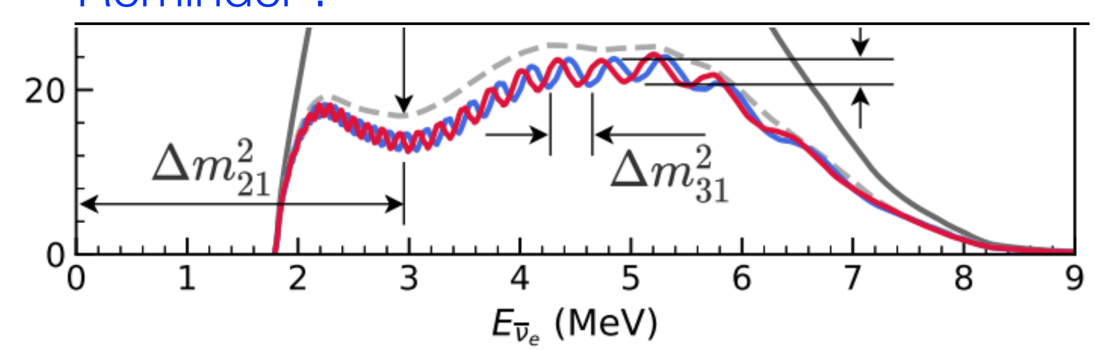




Loss : $$\underbrace{\mathcal{L} = \mathcal{L}_{Adv}}\, + \mathcal{L}_{Reg}$$

Breaks the reconstruction ;
Regularises with Control samples.

Reminder :



An Adversorial NN @ JUNO to explore ML reliability (under dev @ Subatech)

Presently under development, i.e. treating these issues :

- Number of inputs to modify > 35000. Ex: Q_i, t_i -> $Q_i + \delta Q_i$, $t_i + \delta t_i$ for each PMT i.
  - Find a systematic way to modify Q and t, learn only the parameters of a function.

$$\begin{bmatrix} \delta Q \\ \delta t \end{bmatrix}_i = F(Q_i, t_i, X_i, Y_i, Z_i, E^{true}_{dep}, V^{true}_{dep}, \ldots)$$

Also a way to guide distortions toward "physical" ones.

- What Loss Function to yield relevent distortions ? (e.g. random variations may not bias physics results.)
- Generality of the Adv NN (if not need one NN per reconstruction algorithm).

- What control samples ?
  - Copious calibration data, but must be representative enough of the physics data.

- What control variables ?
  - Interaction Energy and Position, engineered variables, raw PMT signal (Q, t, waveform), … ??

Key takeaways

- JUNO's main goal : Neutrino Mass Ordering (data taking start : 2024 ; main results : 2030)
- Takes a very precise and well understood reconstruction of reactor antineutrinos Energy
- Performant classical reconstruction methods have been developed, as well as several ML methods that perform in the same ballpark, with hopes to improve.
- ML reliability: an issue JUNO starts to work on (involvment of Subatech's ν & calculus groups).

Questions session…

- More Neutrino physics ?
- More on JUNO ?
- More details on ML methods at JUNO ?
- More on ML reliability at JUNO ?
- ….

Back up slides

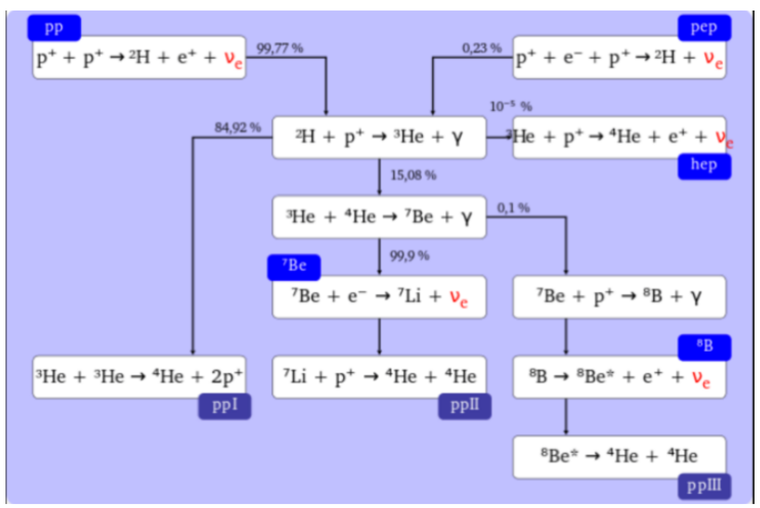
More on JUNO

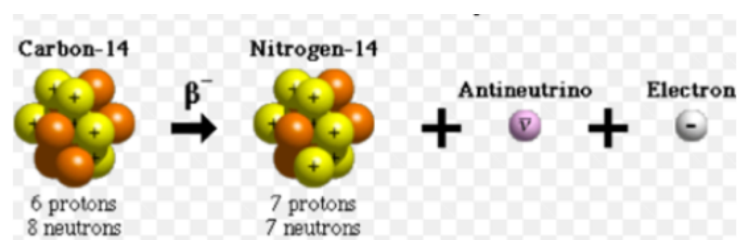
Neutrino (ν) physics in less than a nutshell.

- The Standard Model of particle physics lacks fine answers to some fundamental questions

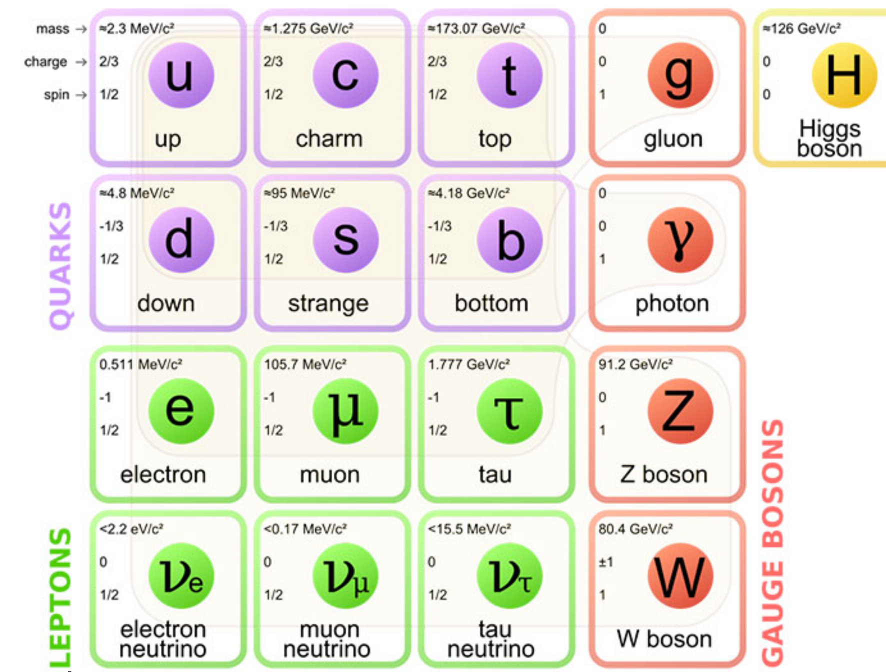
  <u>Ex</u>: Precise origin of Mass ? Why has antimatter disappeared from the early Universe ?

- Studying neutrino physics can help answering them.
  - — Neutrinos are elementary particles.
  - — Produced naturally in stars, radioactive decays, …

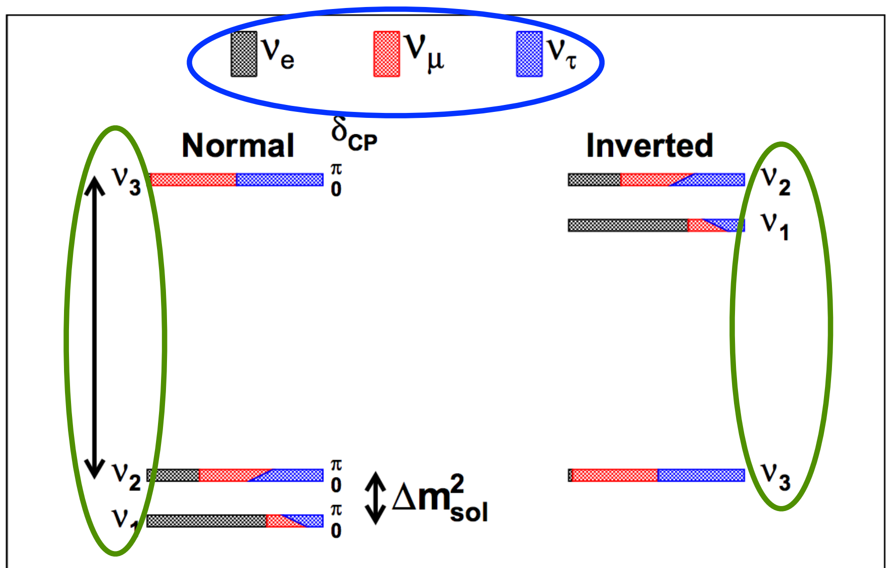




Each second:
10000 billions solar ν's thru your head.



- — 3 known ν **flavors**, associated to the lepton that's also produced when a ν is produced.
- — 3 possible quantum **mass** states : the relationship with flavor states is not well known



Flavor states

Mass states

— Determining the Neutrino Mass Ordering (Normal or Inverted) is one of the hottest questions in particle physics.

— This is JUNO's main objective.

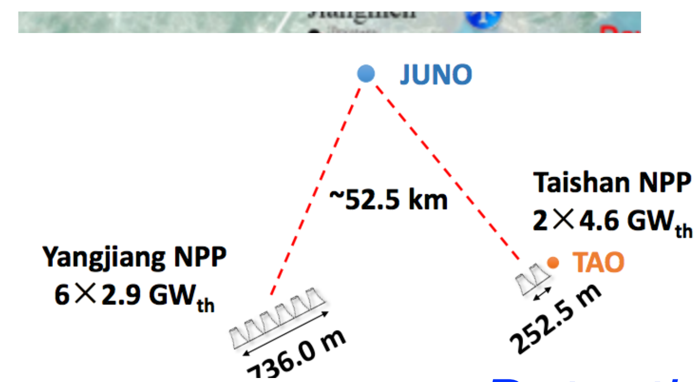
Neutrino Mass Ordering with the JUNO experiment.

- All over the World, many experiments (will soon) try to determine NMO.
- Most use **neutrino oscillation**, a phenomenon providing a lot of info. on neutrino physics

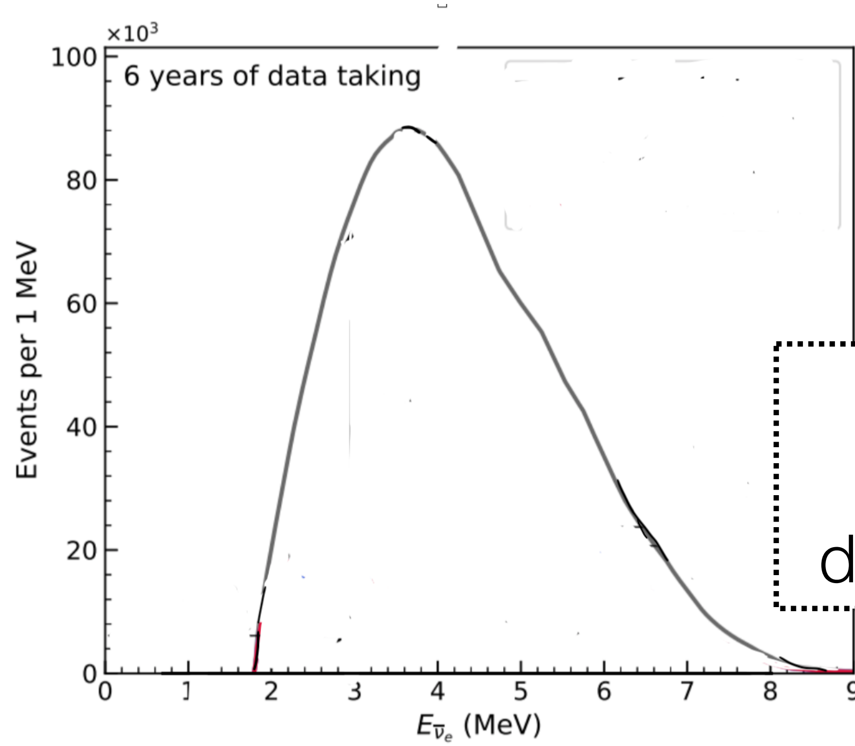
Spontaneous change of flavor between creation & detection.

$$\nu_e \rightarrow \nu_\mu \;;\; \nu_e \rightarrow \nu_\tau \;;\; \nu_\mu \rightarrow \nu_e$$

- **JUNO**: try to determine NMO via the **inprint of oscillation** on the **Energy spectrum** of antineutrinos produced by nuclear reactors.



Production in reactor



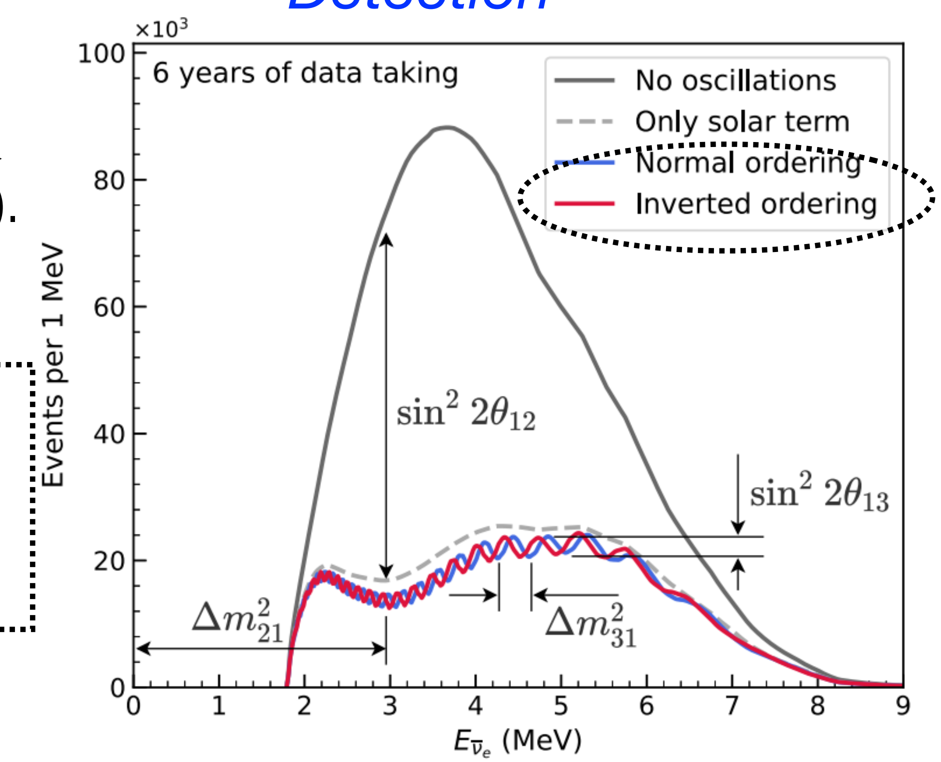
52.5 km

lisappear (oscillate into other flavors).

$$\nu_e \rightarrow \nu$$

NMO determination:
detecting the very small
dephasing between NO and IO

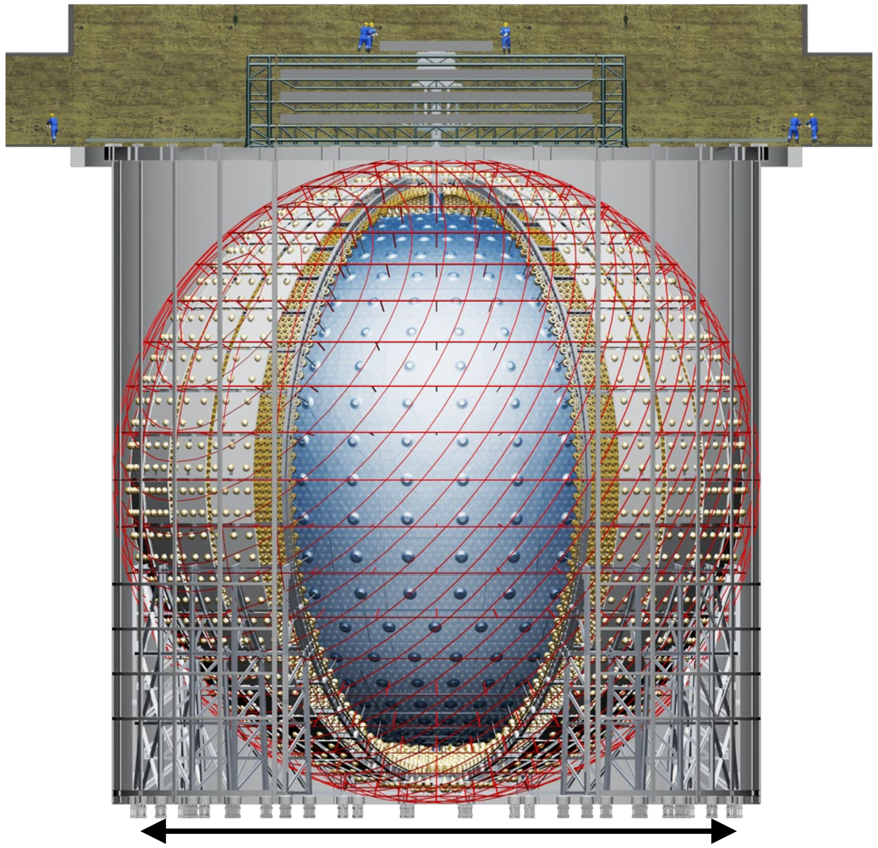
Detection



=> Necessits to reconstruct the Energy of the ν_e with an **extreme** precision.

The Jiangmen Underground Neutrino Observatory (JUNO)

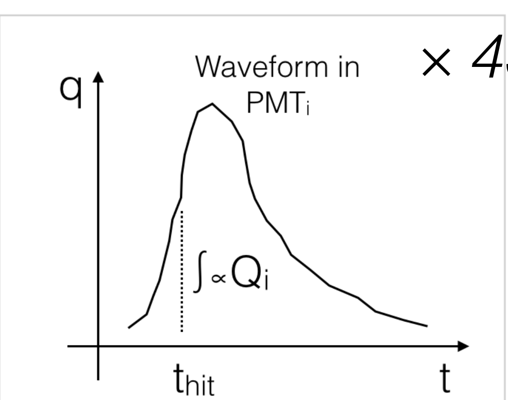
- One of the 4 major neutrino physics next generation experiments.



- — A **35 m diameter** sphere filled with Liquid Scintillator (20 kt).
- — Readout by a **double calorimetric system** :
  17600 20-inch and 25600 3-inch PMTs.
- — Under construction in China, **700 m underground**.
  Data taking expected to start late 2024.
- — International collaboration:
  18 countries, 75 institutes, 650 scientists.
- — A very **rich, multipurpose physics program**.
  Goes far beyond MO determination we focus on here.

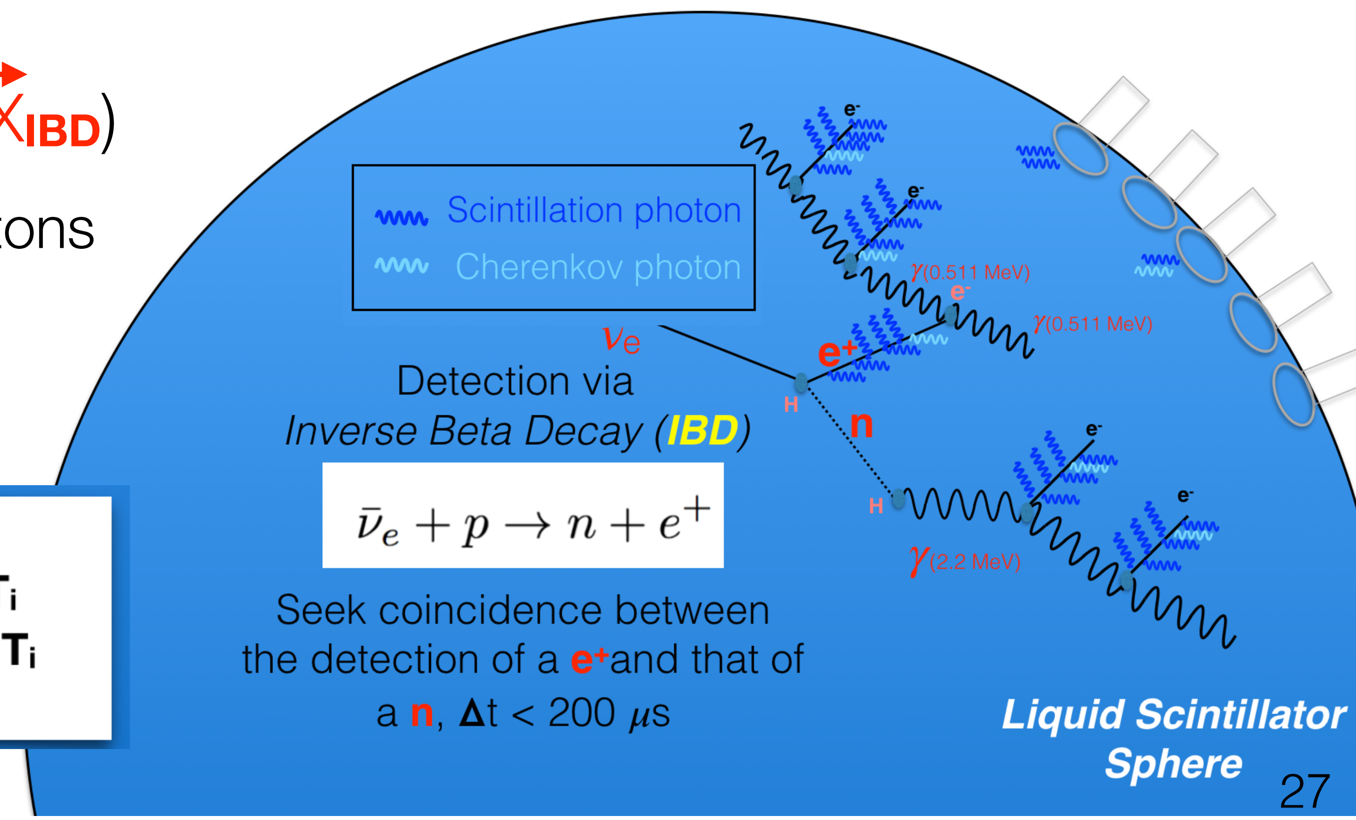
- Reactor $\bar{\nu}_e$ measurement principle (E_{IBD}, $\vec{X}_{IBD}$)

— Collect signals seen in all PMTs hit by Scint. Photons
(more rarely: Cherenkov photons)



× *43200 PMTs*

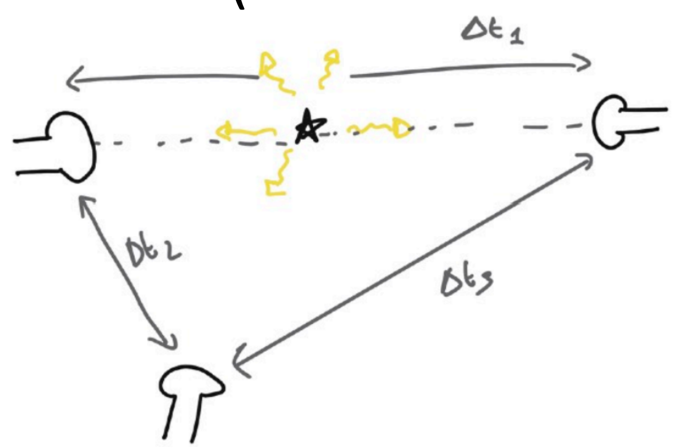
Inputs to reco algorithms
- **Charge Q_i ∝ #Photons that hit PMT_i**
- **t_i ∝ Estimated time of 1st hit in PMT_i**
- **Other Waveform parameters....**



Graph neural networks

Preliminary !
Under development @ Subatech.

- Energy better reconstructed if interaction position is known (+useful for many tasks in JUNO)



PMT hit times: the crucial information here.

Comparing hit times of opposite regions (and over the full detector) is key.

- Previous algorithms: global detector information gathered via successive poolings.

=> Development at Subatech of an alternative GNN

From the start (1st layer), link nodes from all over the sphere, while trying to keep local info.

Fired nodes :
All PMTs with at least 1 Photon hit.
Linked to their corresponding Mesh Node.
Input features : low level signals (Q, t_{hit})

Mesh nodes :
768 regional pixels, all connected to each other.

Global node :
Connected to all Mesh nodes.
Inputs : Ex : powers of spherical harmonics decomposition (spherical image).
Output : Energy & Position.

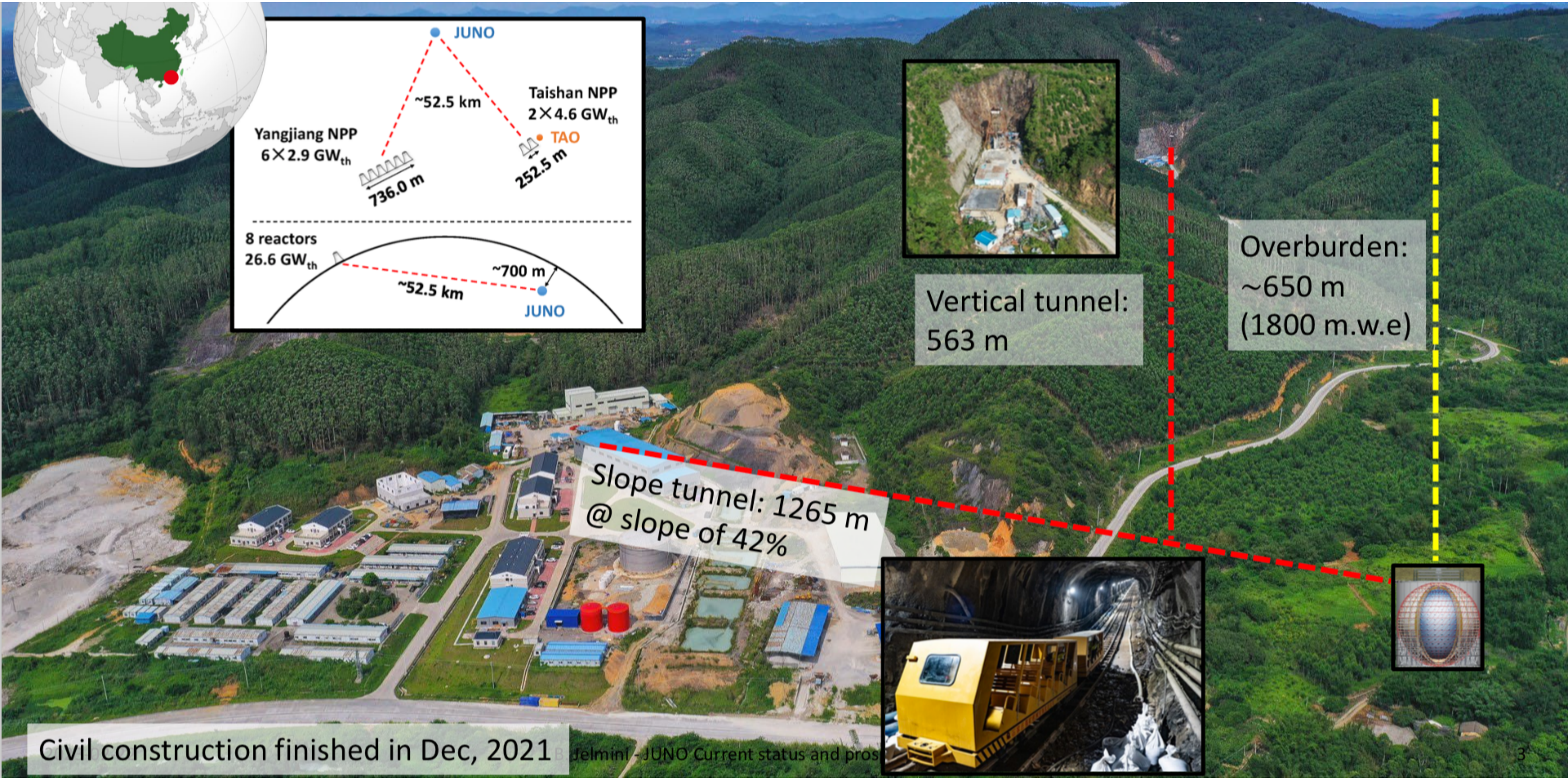
Mesh-Mesh links : Engineered features to help inductive bias.
Ex: ratios of relative timings and total charges (same if interaction lies on the link)

JUNO : an extensive physics programme

Besides the reactor neutrino program, JUNO will study several fields.

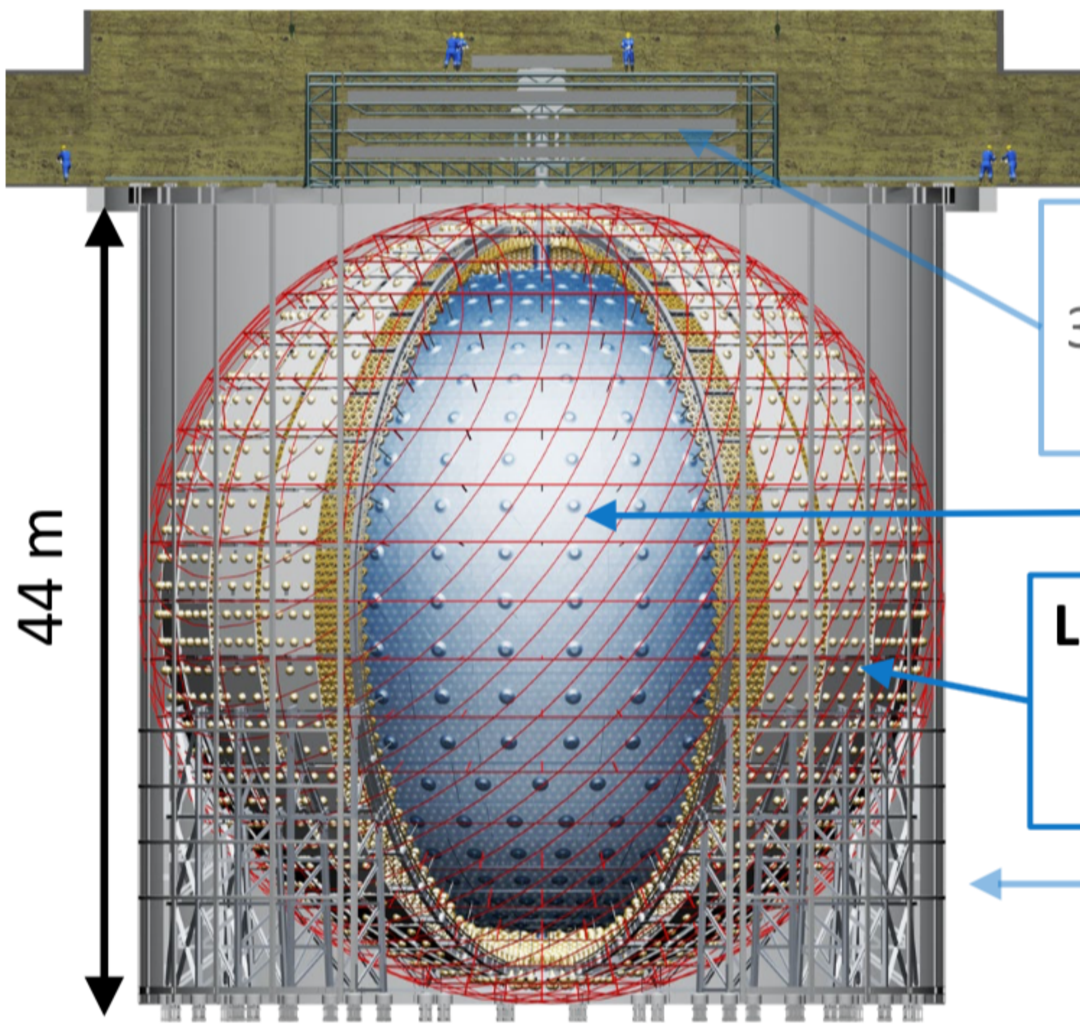
- Neutrino Physics beyond NMO :
  - Precision study of the oscillation
  - Physics beyond the standard model via evidence of additional neutrino states
  - Other new physics studies.
- Atmospheric neutrinos — Neutrino physics, like NMO.
- Geoneutrinos — Geosiences.
- Solar neutrinos — Neutrino physics, astrophysics.
- Core Collapse Supernovea.
- Diffuse Supernovae Neutrino Background.
- Sterile Neutrinos Searches using TAO near detector
- Nuclear reactor physics using TAO near detector

More on JUNO



From "JUNO Current status and prospects",B. Jelmini @ LLWI 2023

More on JUNO



KamLAND	1 kton	34%	6% @ 1 MeV	250
JUNO*	20 kton	78%	3% @ 1 MeV	>1300

* Prog. Part. Nucl. Phys. 123 (2022) 103927

Top Tracker (TT):
3 plastic scintillator layers
Precision muon tracking

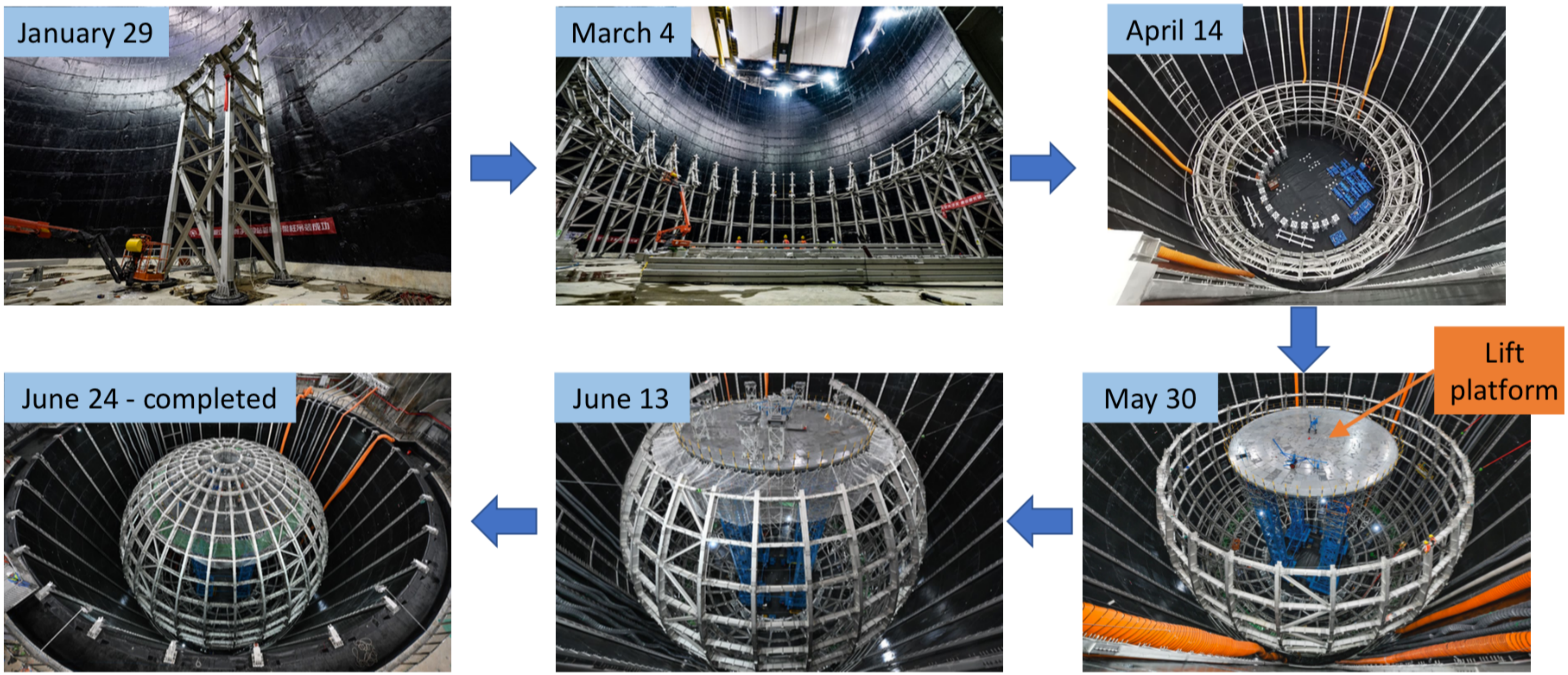
Central Detector (CD):
20 kton Liquid Scintillator (LS)
Acrylic vessel (Ø 35.4 m)
Steel structure (Ø 40.1 m)

Light detection system:
17612 20-inch PMTs
25600 3-inch PMTs

Water Cherenkov Detector (WCD):
35 kton ultra-pure water
2400 20-inch PMTs

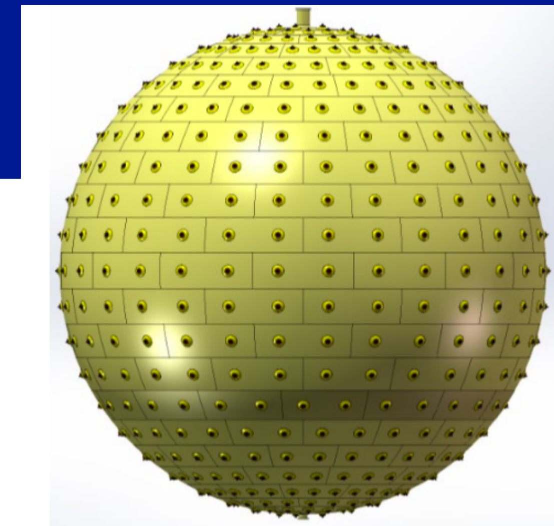
From "JUNO Current status and prospects'',B. Jelmini @ LLWI 2023
"

More on JUNO

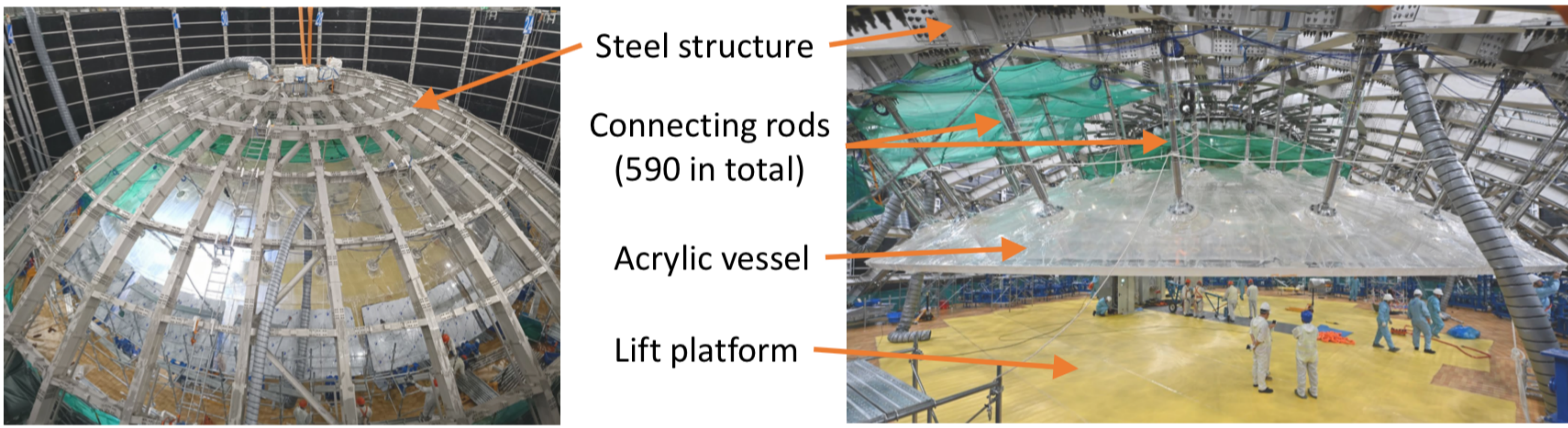


From "JUNO Current status and prospects",B. Jelmini @ LLWI 2023
"

More on JUNO



Acrylic Vessel



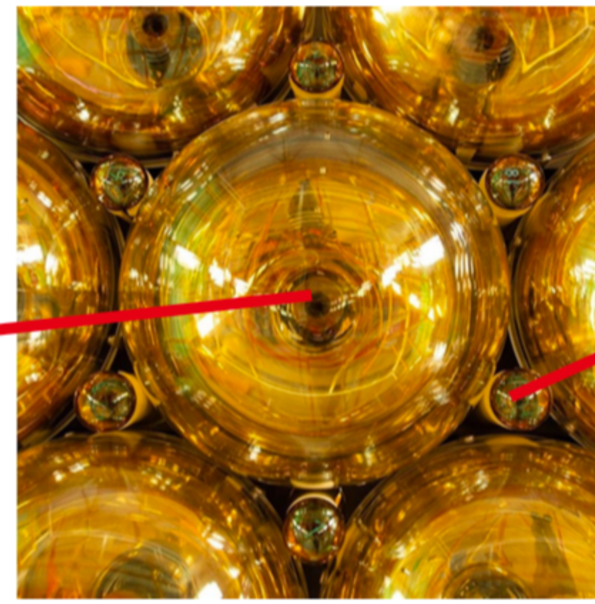
From "JUNO Current status and prospects'',B. Jelmini @ LLWI 2023
"

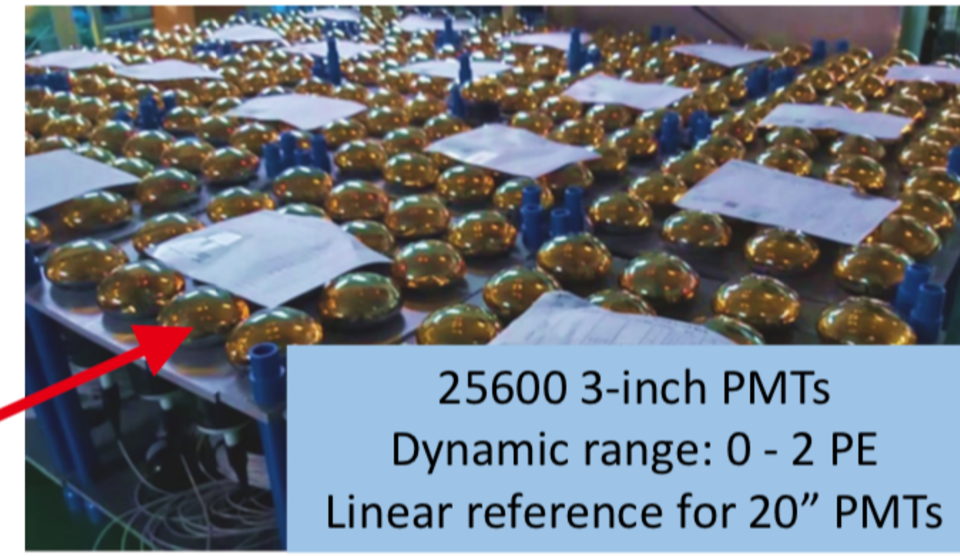
More on JUNO

Photomultiplier Tubes

Total photocathode coverage: 77.9%

17612 (CD) + 2400 (veto) 20-inch PMTs
Dynamic range: 0 - 100 PE





25600 3-inch PMTs
Dynamic range: 0 - 2 PE
Linear reference for 20” PMTs

20-inch PMTs:

- All potted and tested
- Protection cover under production

3-inch PMTs:

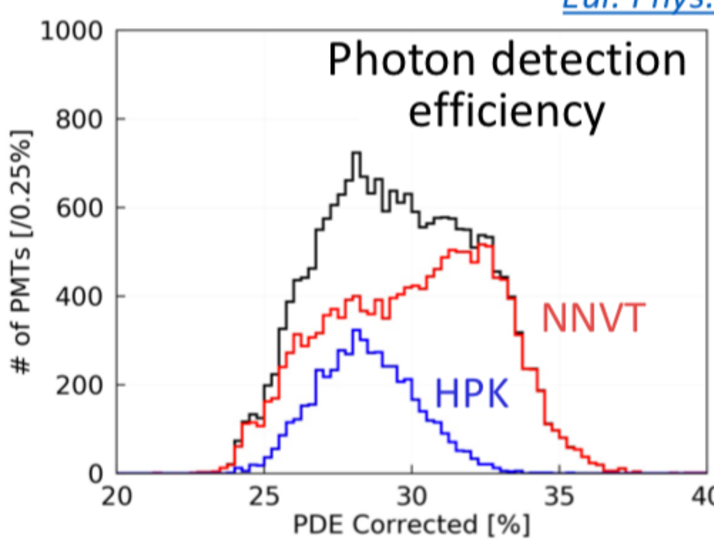
- All potted and tested

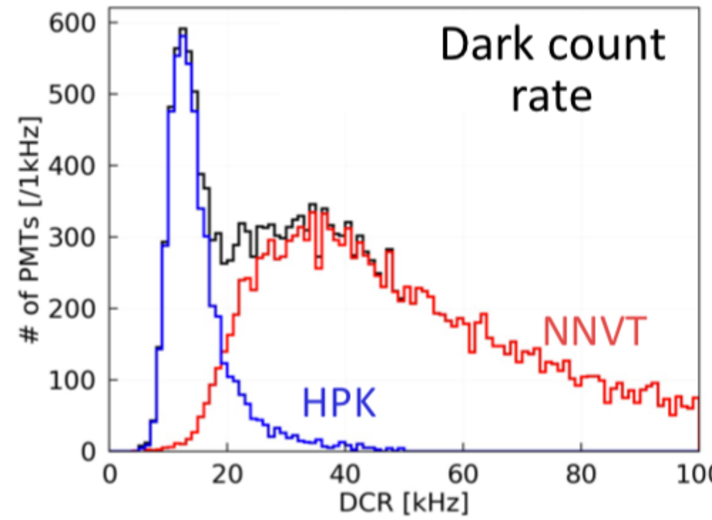
First PMTs installed!

15012 Micro-channel Plate PTMs from Northern Night Vision Technology (NNVT)

5000 dynode PMTs from Hamamatsu Photonics K. K. (HPK)

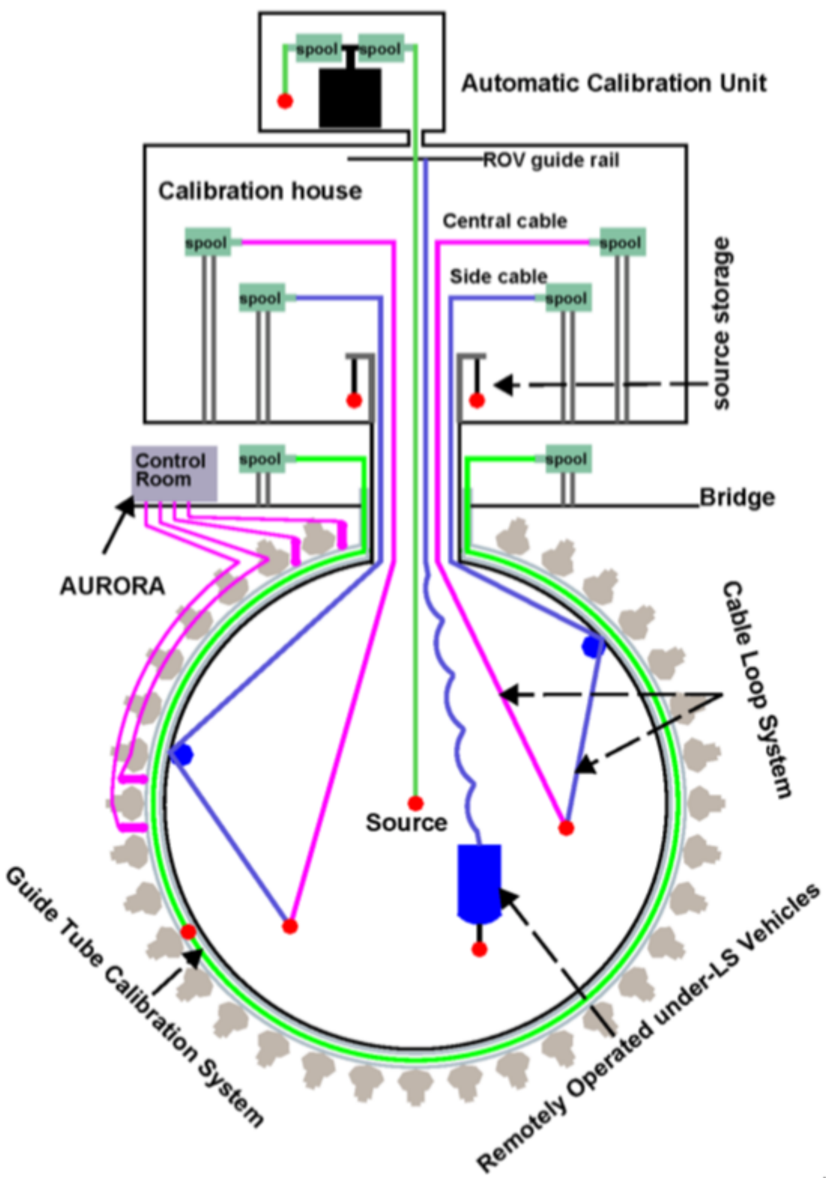
Eur. Phys. J. C **82**, 1168 (2022)





From "JUNO Current status and prospects”,B. Jelmini @ LLWI 2023
"

Calibration system



4 sub-systems JHEP 03(2021)004

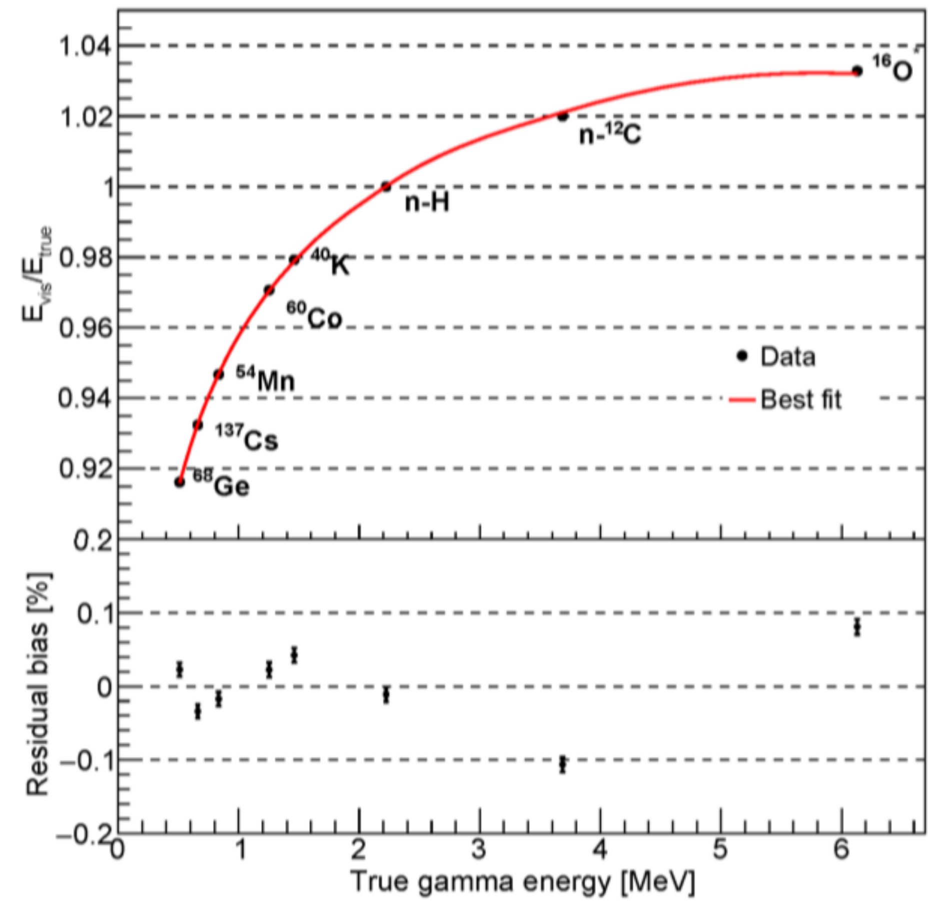
Liquid scintillator non-linearity:
- 5 gamma sources
- 2 neutron sources
- Continuous ^{12}B spectrum

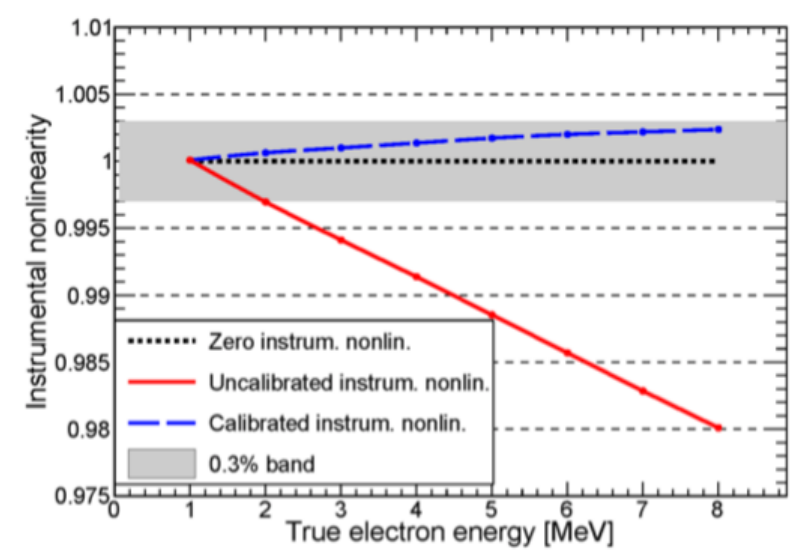
Instrumental non-linearity:
- Tunable UV laser
- Gamma source

Dual Calorimetry Calibration*
- Use 3-inch PMTs as linear reference
- Correct 20-inch PMT channel-wise non-linearity

→ Residual NL < 0.3%

*Yang Han, https://hal.archives-ouvertes.fr/tel-03295420v1



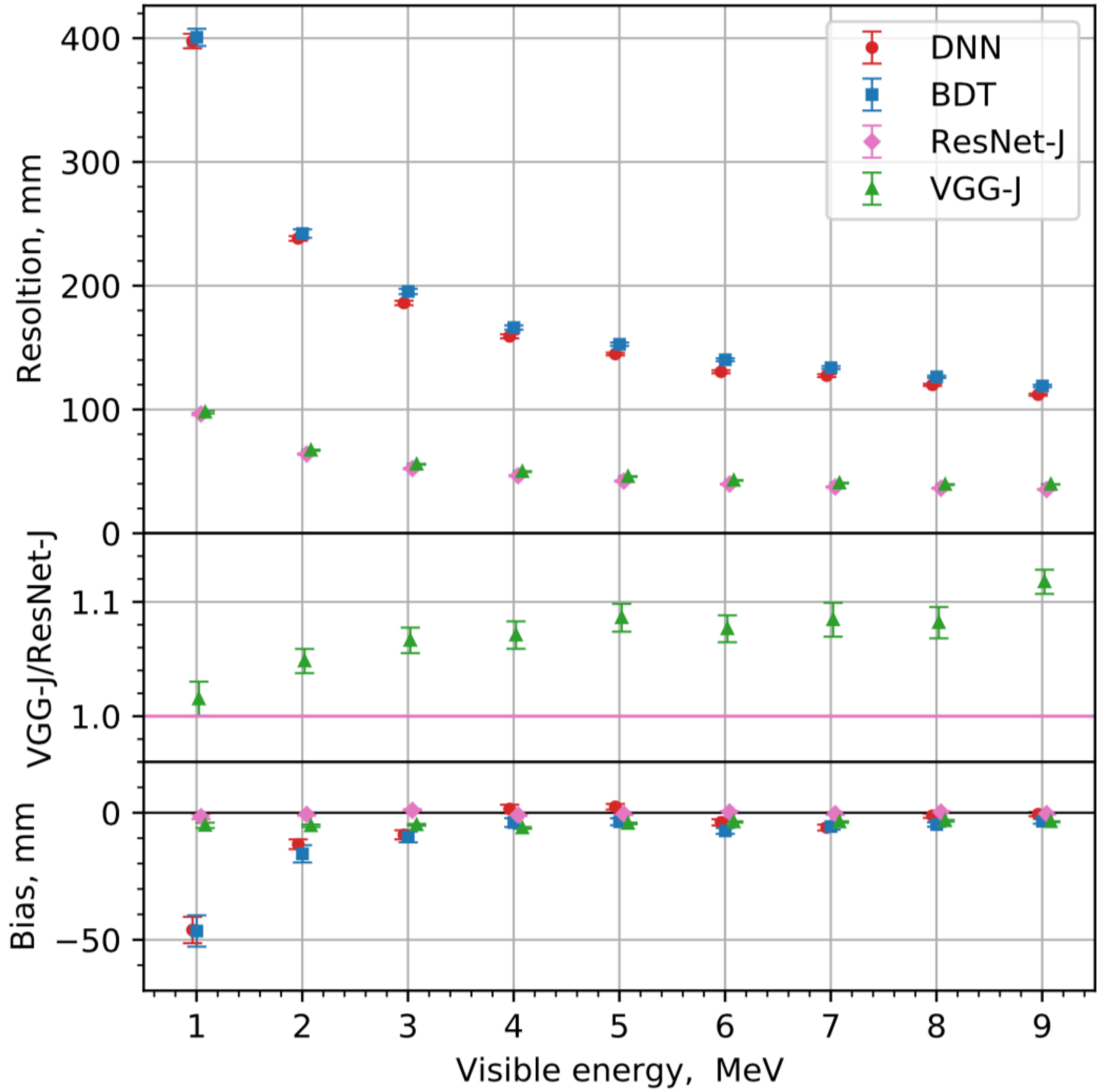


From "JUNO Current status and prospects",B. Jelmini @ LLWI 2023

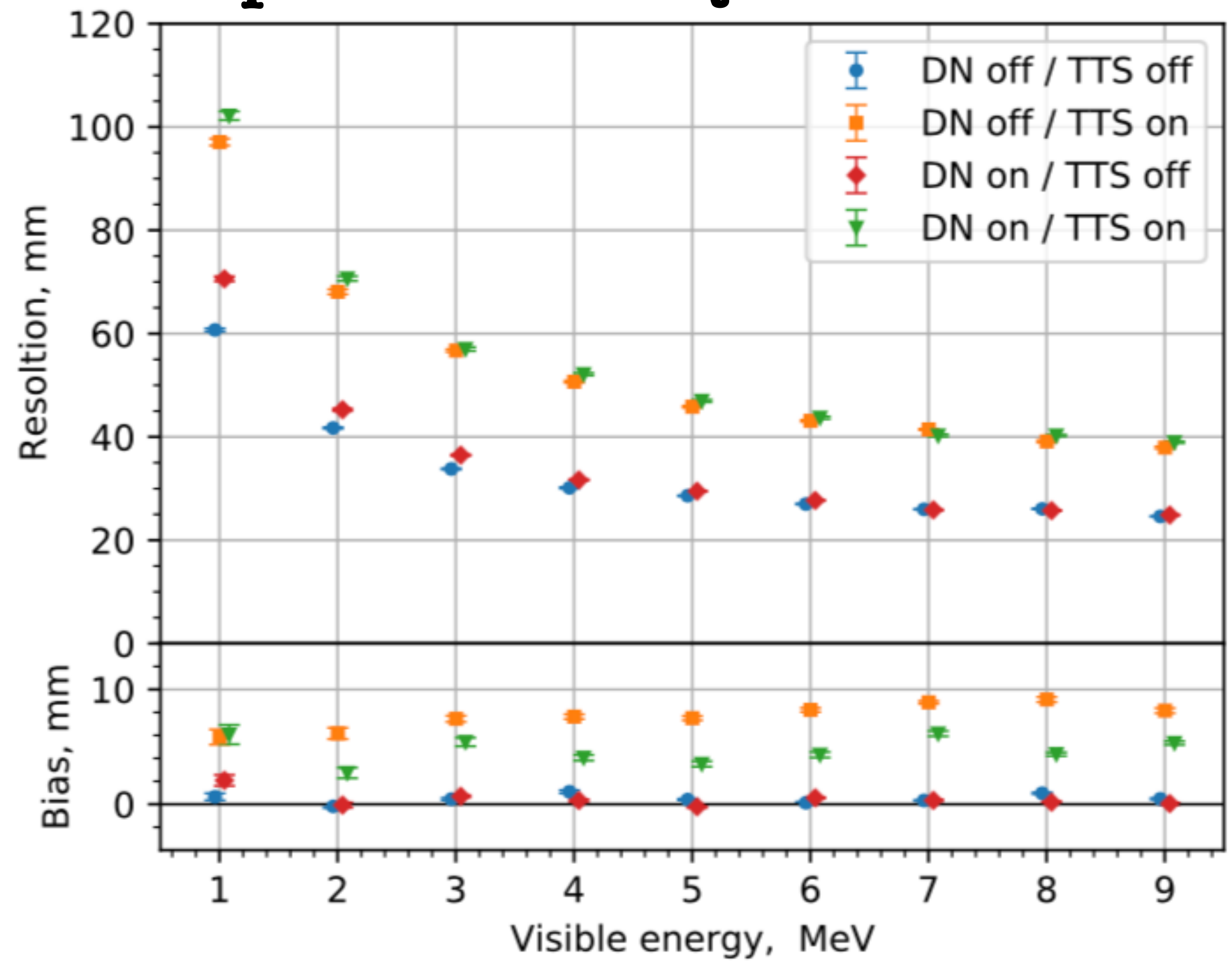
More on performance

Vertex resolutions and biases

Global (2 -channel case, still valid in the 6-channel case).



Importance of DAQ effects. VGG



Essentially valid for other methods.

Generically :

- biases of a few mm.
- up to ~20 mm for classical.
- ML in general a bit better.
- Actually: bias sometimes in different regions for ML and classic
  => possible compensation, opportunity to understand origin

Vertex bias, classical methods.

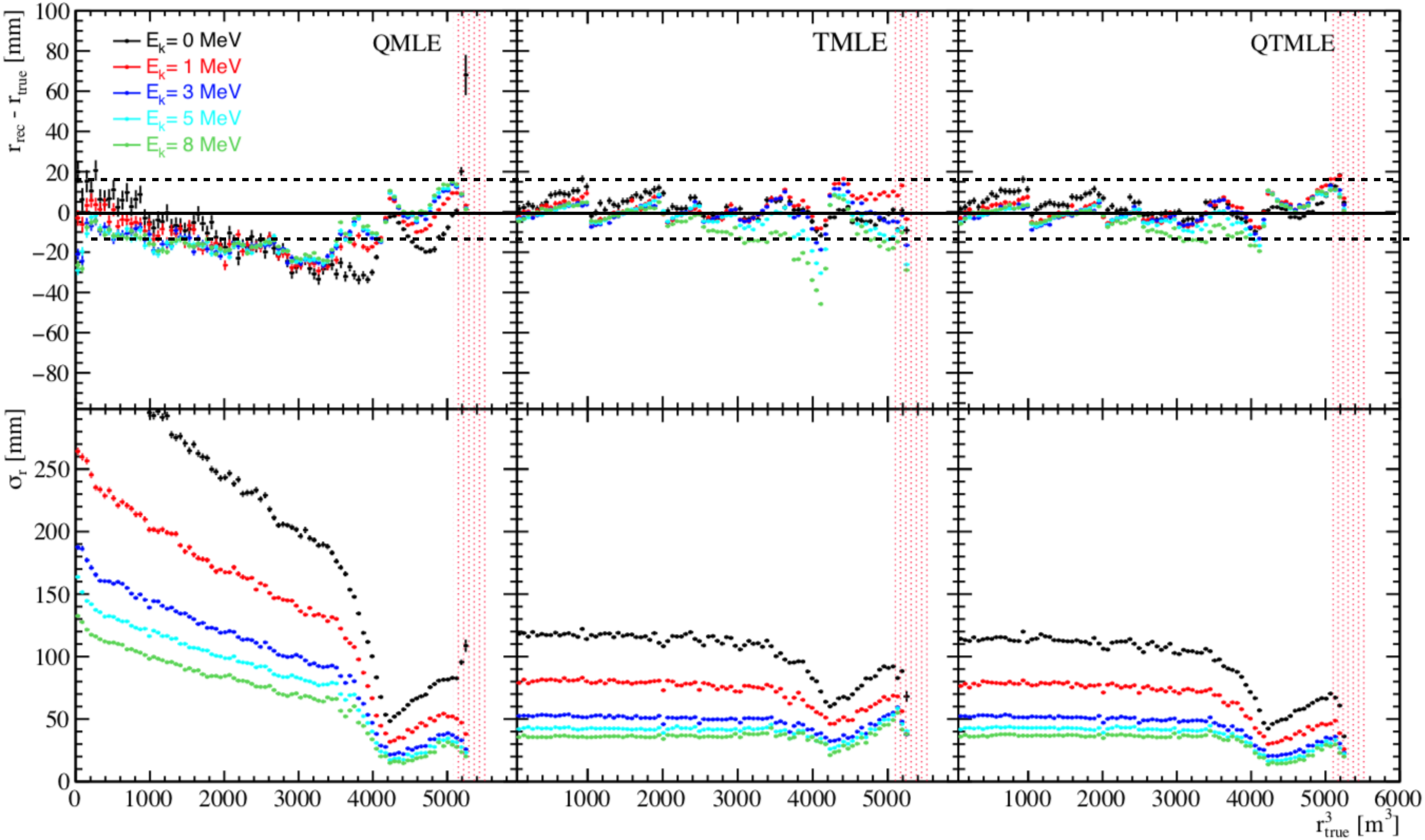
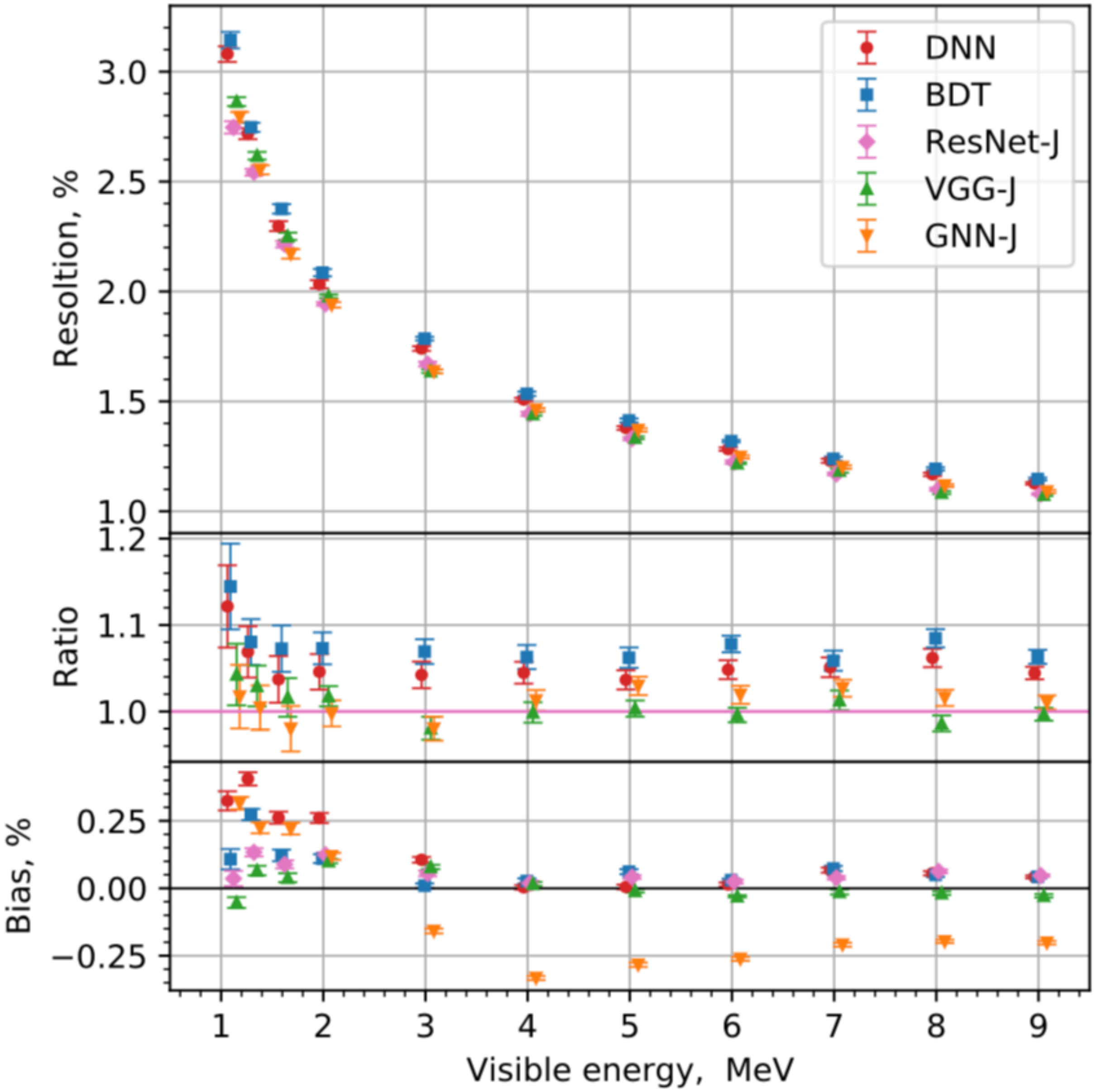


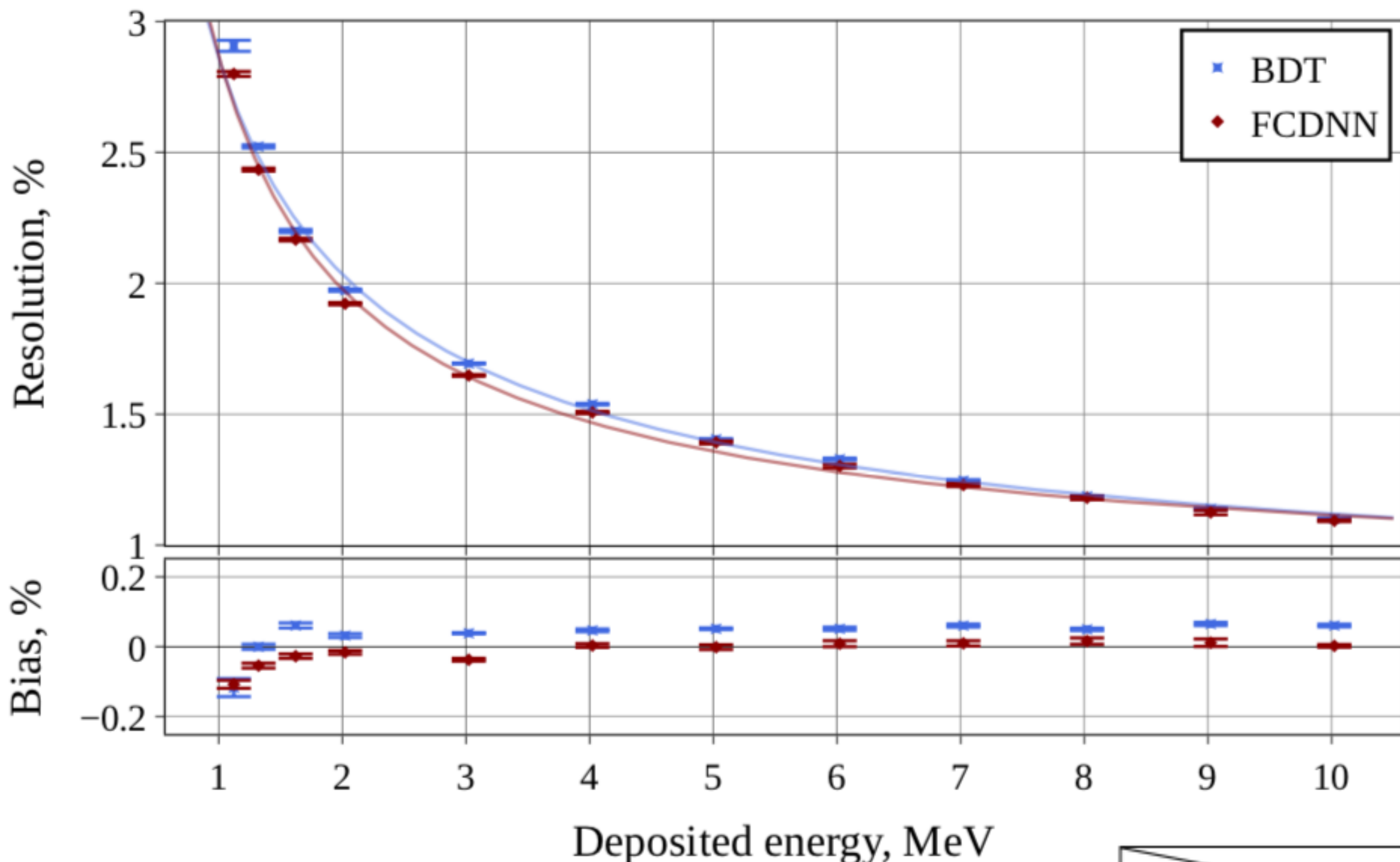
Fig. 7. Vertex reconstruction performances. The left, middle and right columns correspond to the QMLE, TMLE and QTMLE methods, respectively. The top row shows the vertex bias and the bottom row shows the vertex resolution.

Energy resolution and bias.

Global (2 -channel case, still valid in the 6-channel case).



Energy reconstruction and bias with aggregated variables.



Generically :

- Res: Similar to PMT-wise methods
- Bias : slightly worse at very low E (in this case)
- Bias of **classical** methods:
  - Same remarks as for vertex reconstruction
  - Below 0.3%

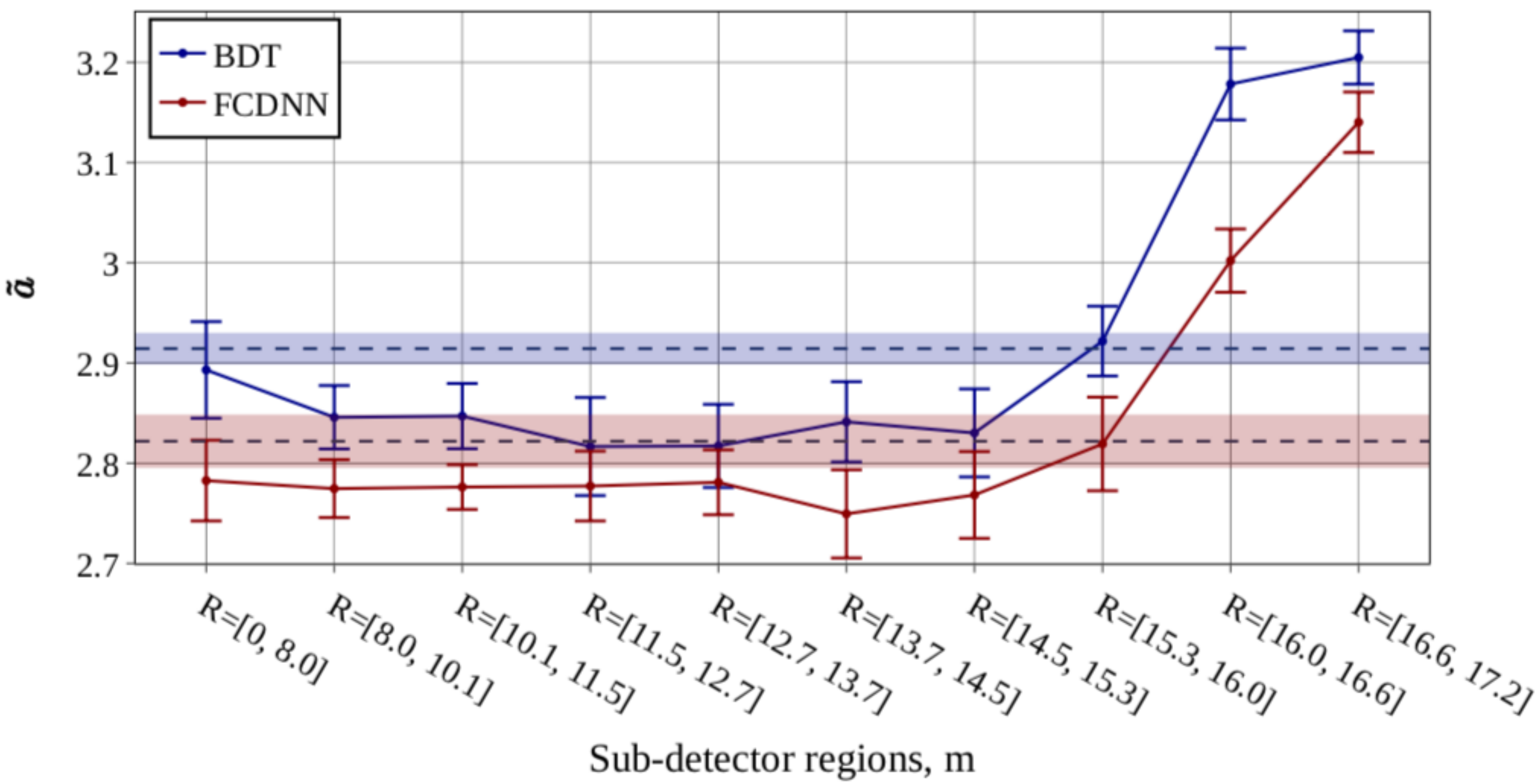
Model / *Parameter*	BDT	FCDNN
$a \pm \Delta a$	2.573 ± 0.097	2.316 ± 0.139
$b \pm \Delta b$	0.763 ± 0.045	0.827 ± 0.054
$c \pm \Delta c$	0.990 ± 0.394	1.474 ± 0.285
$\tilde{a} \pm \Delta \tilde{a}$	2.914 ± 0.016	2.822 ± 0.027

E_{res} 2.864712

✗ Classical methods

Case	a	b	c	E_{res}	Relative improvement
Default	2.614	0.640	1.205	2.948	-

Energy reconstruction with aggregated variables.



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

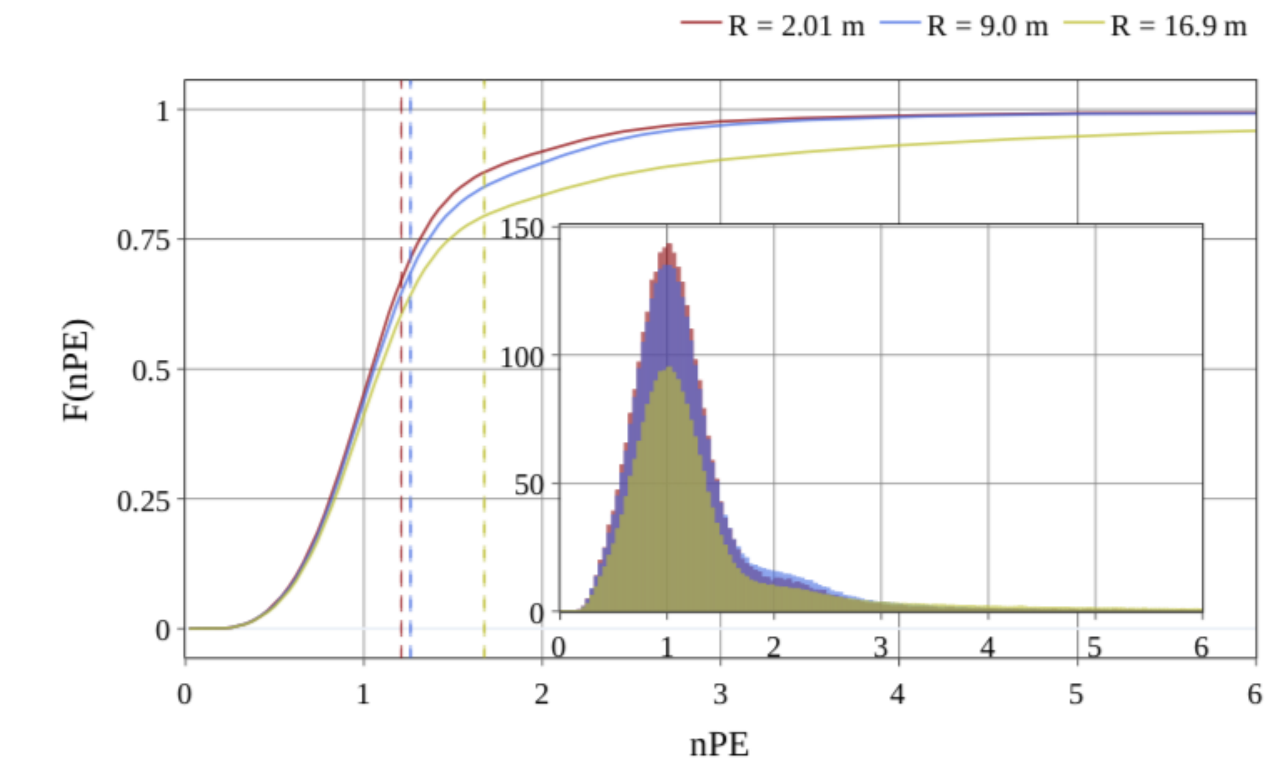
$$\tilde{a} \equiv \sqrt{(a)^2 + (1.6 \times b)^2 + \left(\frac{c}{1.6}\right)^2}$$

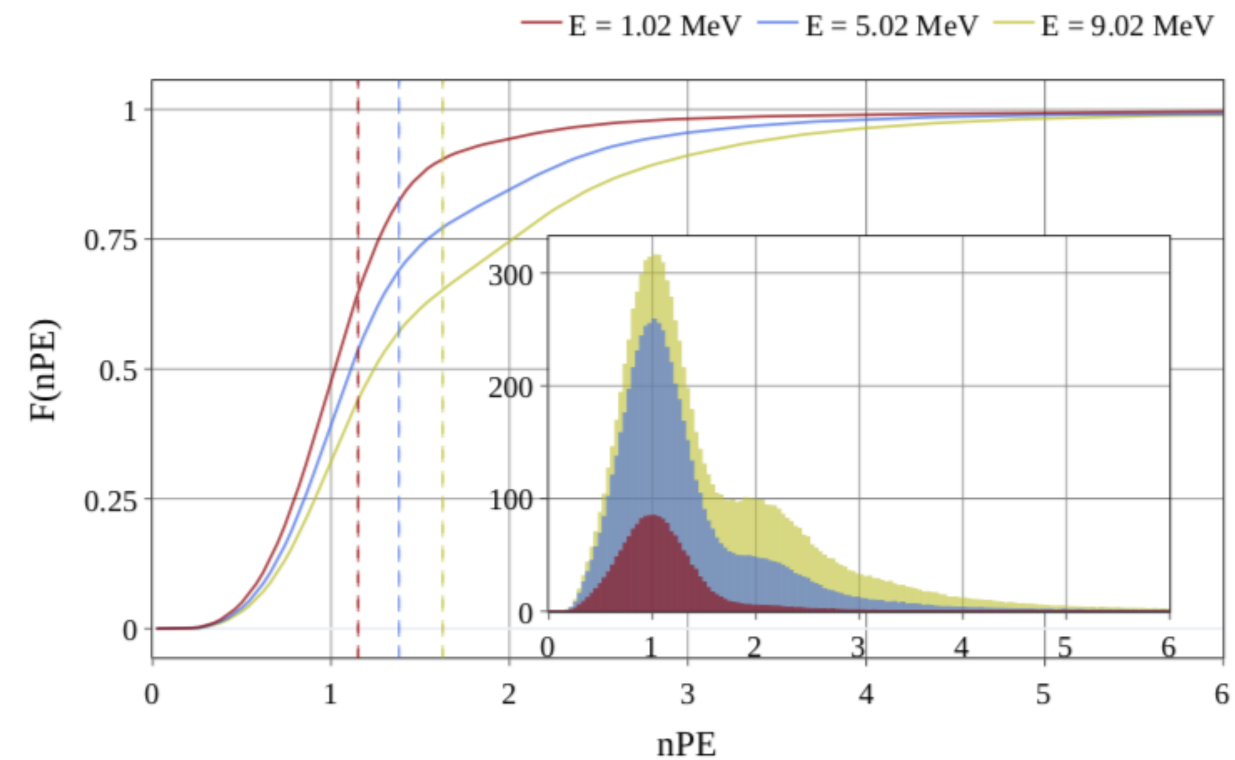
Computing performance (ResNet-J, VGG-J, GNN-J)

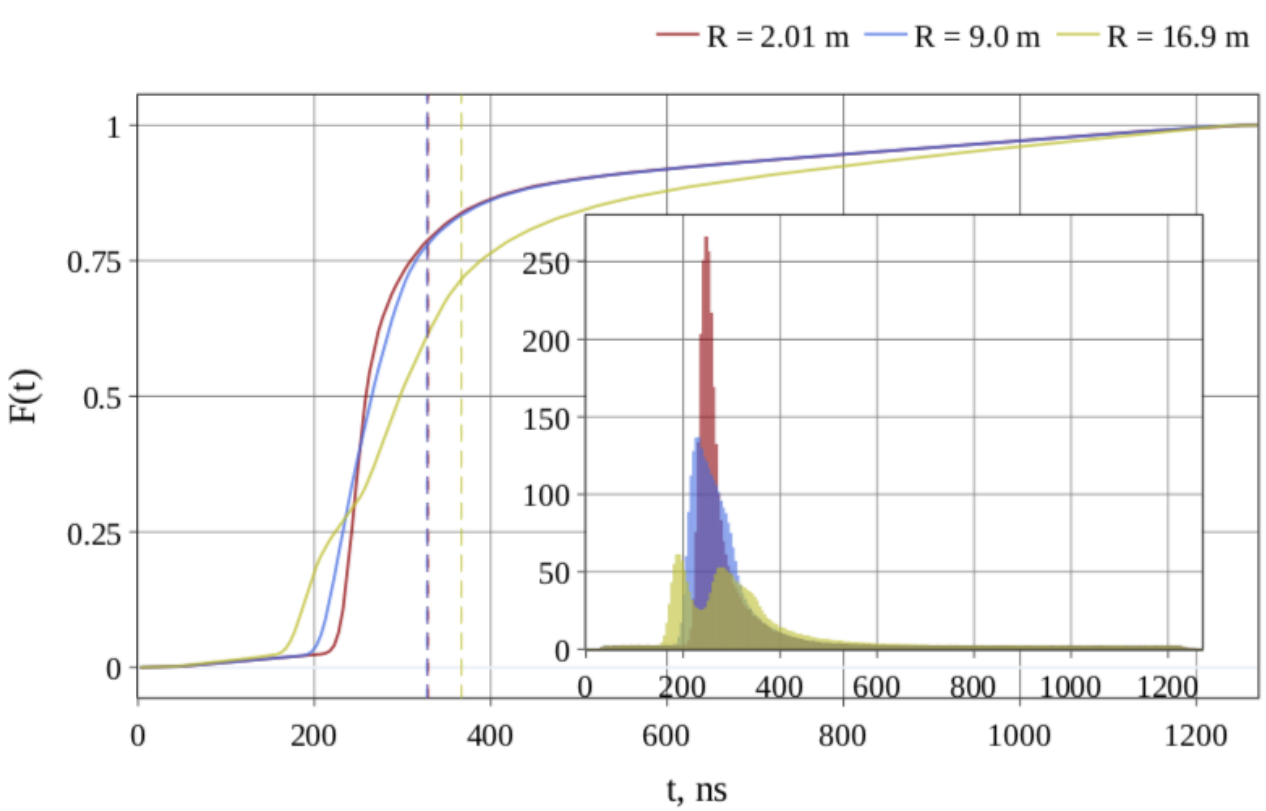
			Planar CNN		Spherical
Architecture	BDT	DNN	ResNet-J	VGG-J	GNN-J
Prediction time, sec/100k events	<1	<1	235	155	110
Prediction batch size	100 000	100 000	100	100	10 000
Number of weights		6625	38 352 403	26 310 035	353 979
Memory occupied by weights, MB	17	0.073	146	100	4.2
Training time, min/1M events	5	1000	1543	840	265
Training batch size		700	64	64	64

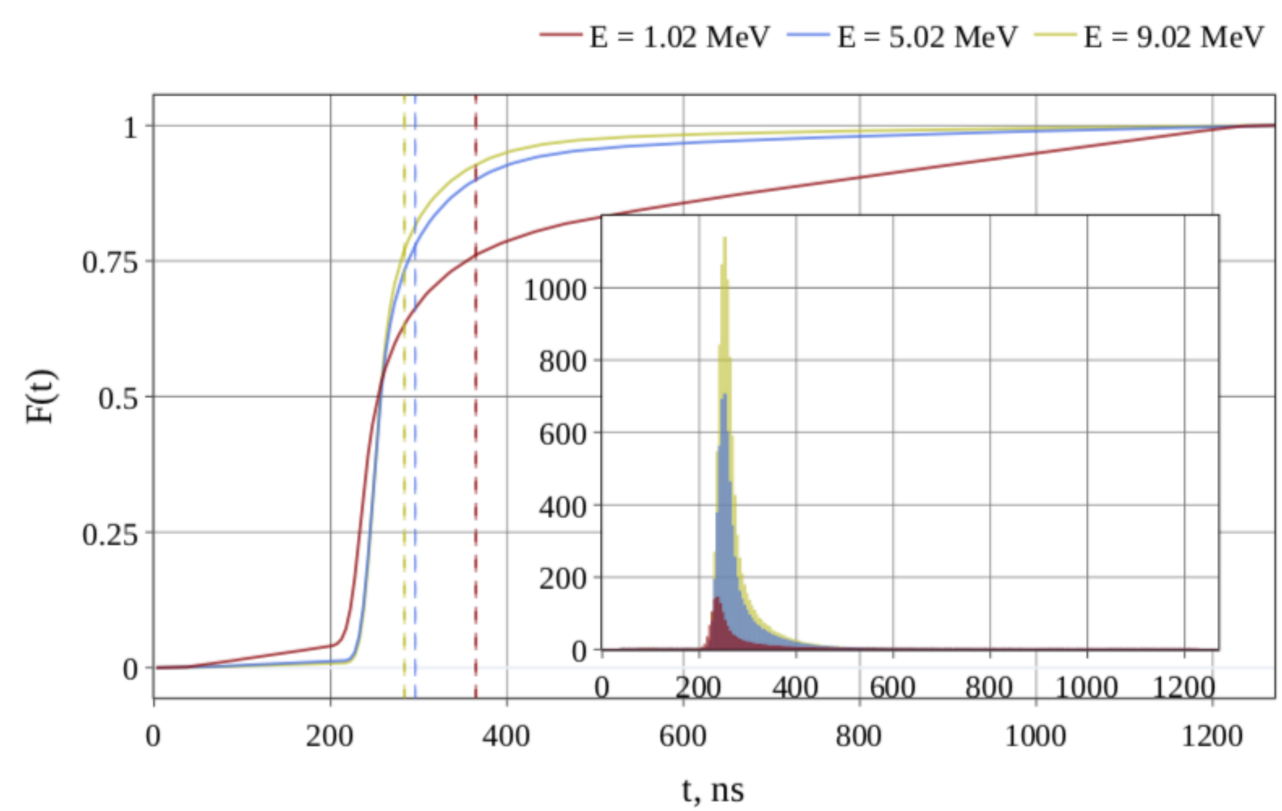
GNN Subatech : 0.5 M (far less param, since no dense layer)

Methods with engineered inputs (iso Q & t from all PMTs) (arXiv:2206.09040v2)









More on Methods :

Archi, hyperparameters and more.

On VGG-J and ResNet-J

- **Algorithms**: Also tried : AlexNet and GoogleNet
- **Projection** : Have also tried Mercator
- **VGG-J**: 17 layers, and 4 in the dense layers. There, compared with original VGG, 2 layers of 4096 nodes have been removed. This reduces by 65 percents the number of parameters.
- **ResNet-J**: ResNet chosen in order to avoid overfitting although far more layers. The residual mapping is easier to optimize (not the full amplitude of the weights).
- Hyper param: probably Grid search.

Parameter	**Value**
Loss	Mean Squared Error
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate	Linearly increasing from 0 to 10^{-3} during the first epoch, then exponential decay to 10^{-8}.
Batch size	64
N. Epochs	15

Table 5 – Hyperparameters for VGG-J and ResNet-J.

On GNN 1

- ✗ They also tried one PMT per pixel, but no better performance.
- ✗ This is an undirected graph.
- ✗ À quoi servent les liens et la matrice d'adjacence, sachant les fitres K=5 ? Vraiment de liens avec seulement les premiers voisins ?
- ✗ Inspired by VGG-16. Minor modifications in the number of layers and filters (brought a 5% improvement).
- ✗ Hyperparameter : manual search. Not enough CPU to do more…
- ✗ Pooling layers divide Nside by 2 -> Ncell = 12Nside^2 divided by 4.
- ✗ For this one : loss = MAPE

Parameter	**Value**
Loss	Mean Absolute Percentage Error
Optimizer	Adam ($\beta_1 = 0.8$, $\beta_2 = 0.9$)
Learning rate	Fixed at 0.001 for $N_{\text{epoch}} < 3$, then exponential decay at rate -0.1.
Batch size	64
N. Epochs	10

Table 6 – Hyperparameters for GNN-J.

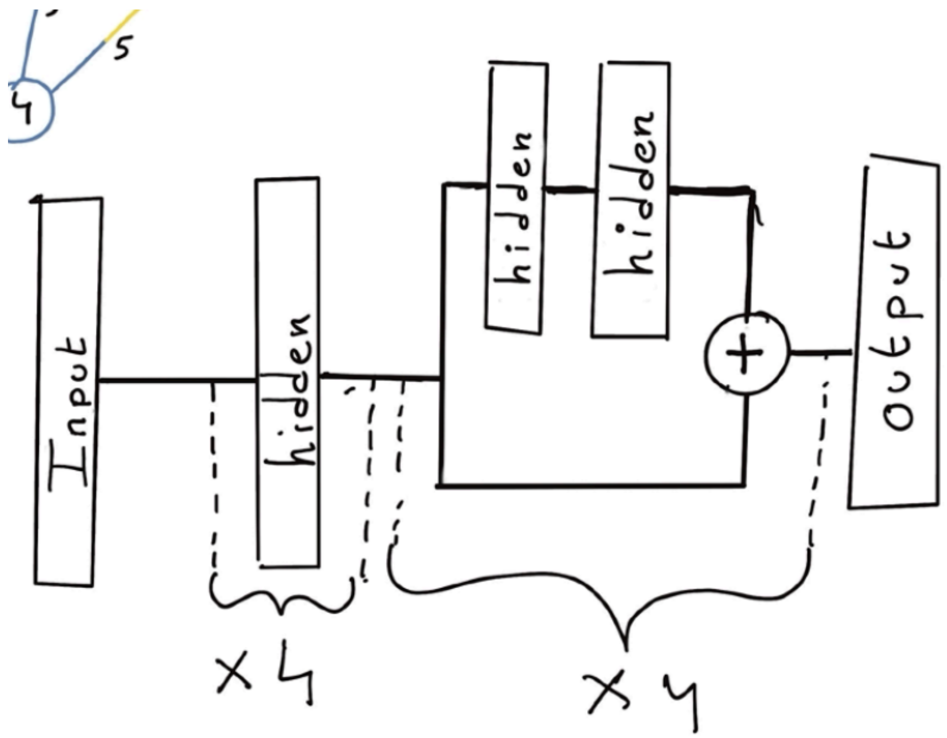
FCDNN, Aggregated variables. Hyperparameters.

Optimization of hyperparameters for FCDNN is performed using the BayesianOptimization tuner from the KerasTuner library for Python [36]. To train the model, we use TensorFlow [37]. The MAPE loss for reconstructed energy and true energy is used as a loss function. All input features were normalized with a standard score normalization. The training process is performed with an early stopping condition on the validation dataset with a patience of 25 and with the batch size 1024. Table 2 shows the search space and the selected hyperparameters.

Hyperparameter	*Range*	*Selected*
Units in input layer	[1, 512]	**256**
Units in hidden layers	[1, 512]	**256**
Number of hidden layers	[1, 32]	**16**
Activation [38–40]	ReLU, ELU, SELU	**ReLU**
Optimizer [41, 42]	Adam, SGD, RMSprop	**Adam**
Learning rate	[0.0001, 0.01]	**0.0016**
Scheduler type [43]	Exponential, None	**Exponential**
Input layer weights initialization Hidden layers weights initialization	normal, lecun-normal, uniform	**normal**

- 4 serial layers and 4 ResNet blocks.

— **Number of trainable parameters** : 100k to 1.5 M
— **Main hyperparameters** : number and nature var on each vtx, each link, 12 layers, loss (MSE, aussi testé relatives), Vtx or E and Vtx, Batch size (32-64 memory !), n epoch : no early stop so 500, learning rate (= 1e-8 + decay = *0.99 at each epoq => Very small, but exploded) and variation,
— **Why we decided to learn slow** : numerical instability… Due to aggregation function (since 1000 links)
— **ADAM** (SGD tended to get stuck in local mins)
— **Batch size** 8 (memory), 800 per epoch.
— At end of epoch look at loss on validation, keep current model if loss better. At the end, we kept the best of all selected this way, plus the last model (useful for stability studies).
— **Size** : 35G in training phase. Cause : very big adjacency matrix (essentially empty, but need memory allocation)
— **Inference time**: 100 ms for inference.
— **Training time** : 15-92h A100, 40G GPU



- Also : bi-directionnal links (mirror variables)

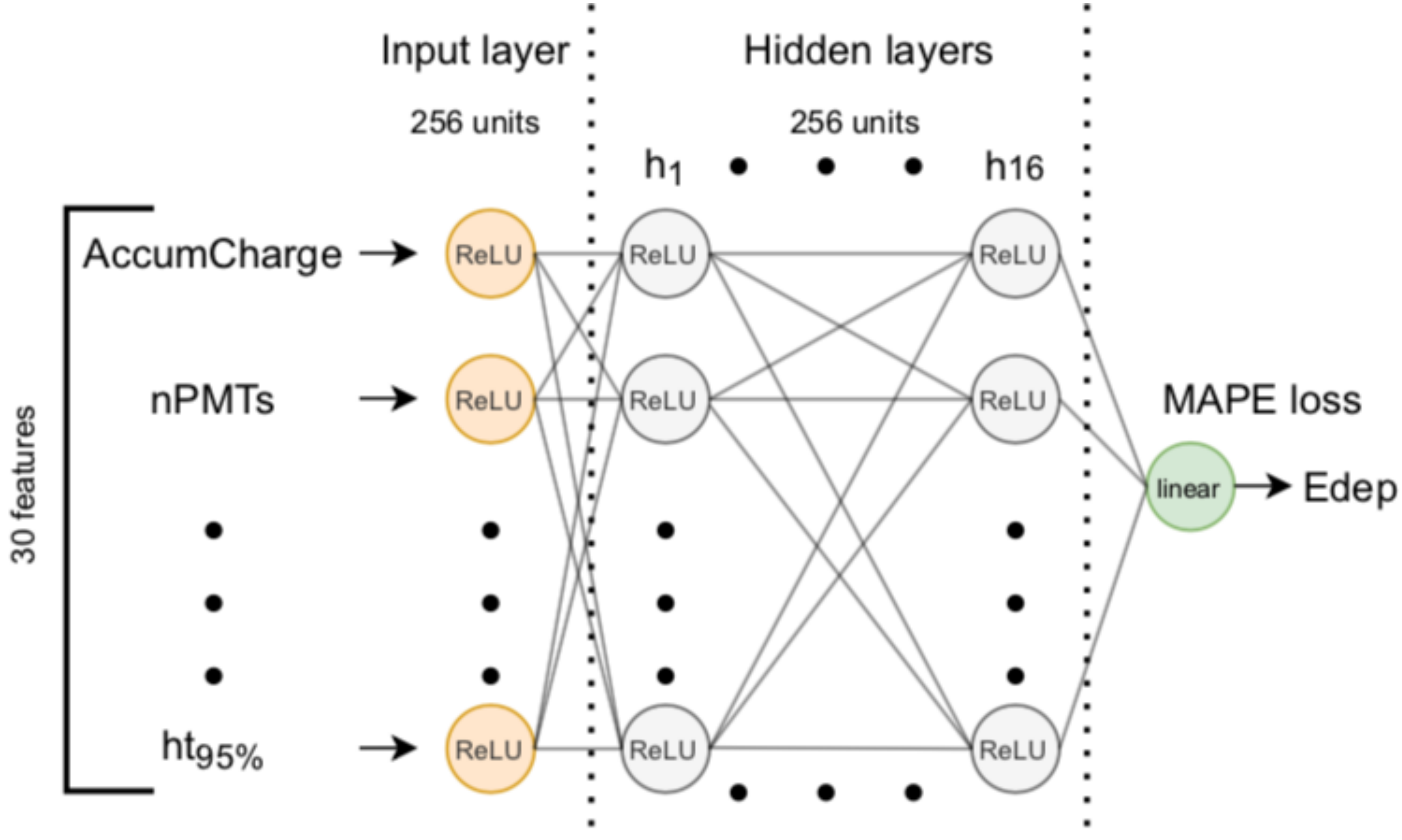
Methods with engineered inputs (iso Q & t from all PMTs) (arXiv:2206.09040v2)

- Loss function for the FCDNN

Mean Absolute Percentage Error

$$\mathrm{MAPE} = \frac{100\%}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \left| \frac{\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}}{\mathrm{y_i}} \right|$$

y = true E $\hat{y}$ = reconstructed E



Reliability : ML methods

The original method by Nachman et al.

- This adversorial method is thought as an adaptation of

AI Safety for High Energy Physics

Benjamin Nachman[*]
Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA[†]

Chase Shimmin[*]
Department of Physics, Yale University, New Haven, CT 06511, USA[‡]
(Dated: October 22, 2019)

arXiv:1910.08606

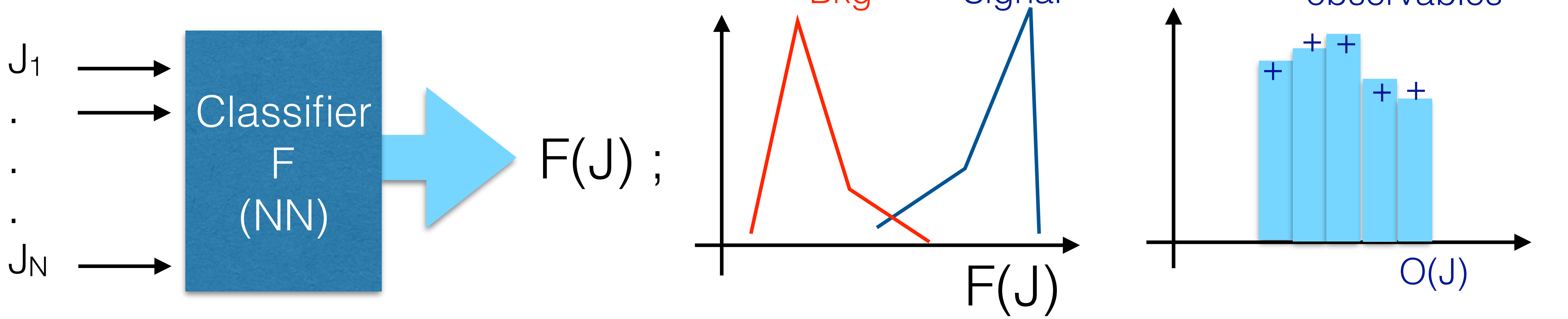
- In a nutshell

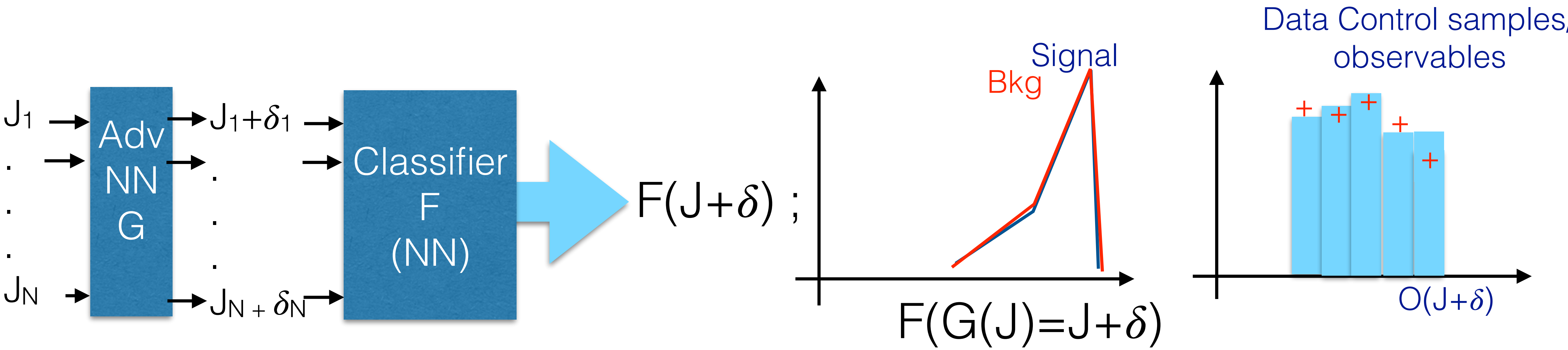
— Adversorial attack on a classifier F(J) identifying S vs. B jets based on ~200 input variables.

— Finds how to modify each J_i into a J_i' so that the distribution of the score F(J') looks like the distribution of background events even when events are signal events.

— Control: the same modifications are applied to data control samples : the distribution of some observables of interest must be stable enough to not change data/MC quality.

The original method by Nachman et al.





Adversorial NN trained by minimising two Loss functions :

$$\mathcal{L}_{\rm sig} = \log(1 - f(g(J))),$$

$$\mathcal{L}_{\rm bg} = \lambda_{\rm cls}(f(J) - f(g(J)))^2 + \sum_i \lambda_{\rm obs}^{(i)}(\mathcal{O}^{(i)}(J) - \mathcal{O}^{(i)}(g(J))^2$$

- Interesting features of this method
  - — Allows to determine midmodelling effects δ_i's that can't be detected with control data.
  - — Automated perturbations to a large number of inputs (no need to « think » to all of them).
  - — All inputs shifted simultaneously : can reveal effect of subtle correlations.
  - — If it indeed manages to cause $F(J) \neq F(J+\delta)$, then a systematic uncertainty can be derived.

On ML reliability : have we considered other methods ?

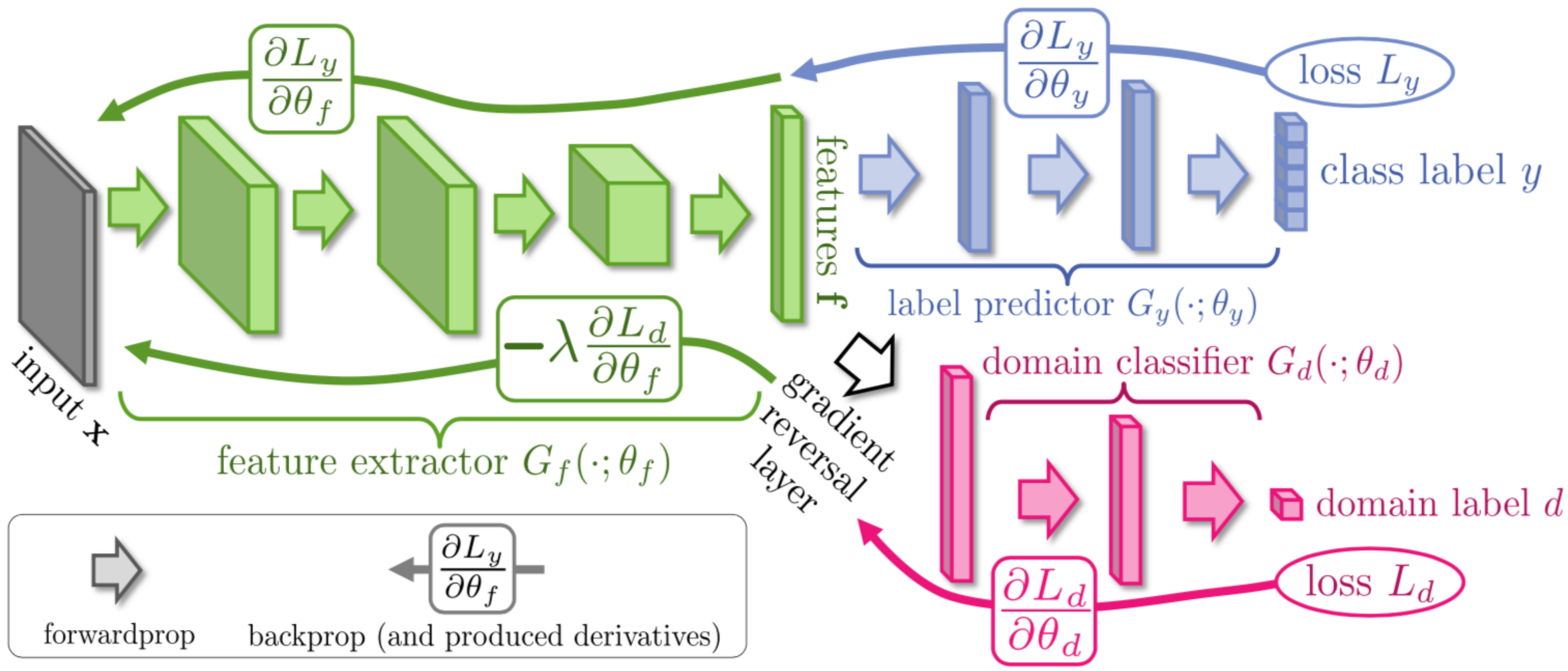


Figure 1. The **proposed architecture** includes a deep *feature extractor* (green) and a deep *label predictor* (blue), which together form a standard feed-forward architecture. Unsupervised domain adaptation is achieved by adding a *domain classifier* (red) connected to the feature extractor via a *gradient reversal layer* that multiplies the gradient by a certain negative constant during the backpropagation-based training. Otherwise, the training proceeds in a standard way and minimizes the label prediction loss (for source examples) and the domain classification loss (for all samples). Gradient reversal ensures that the feature distributions over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus resulting in the domain-invariant features.

On ML reliability : have we considered other methods ?

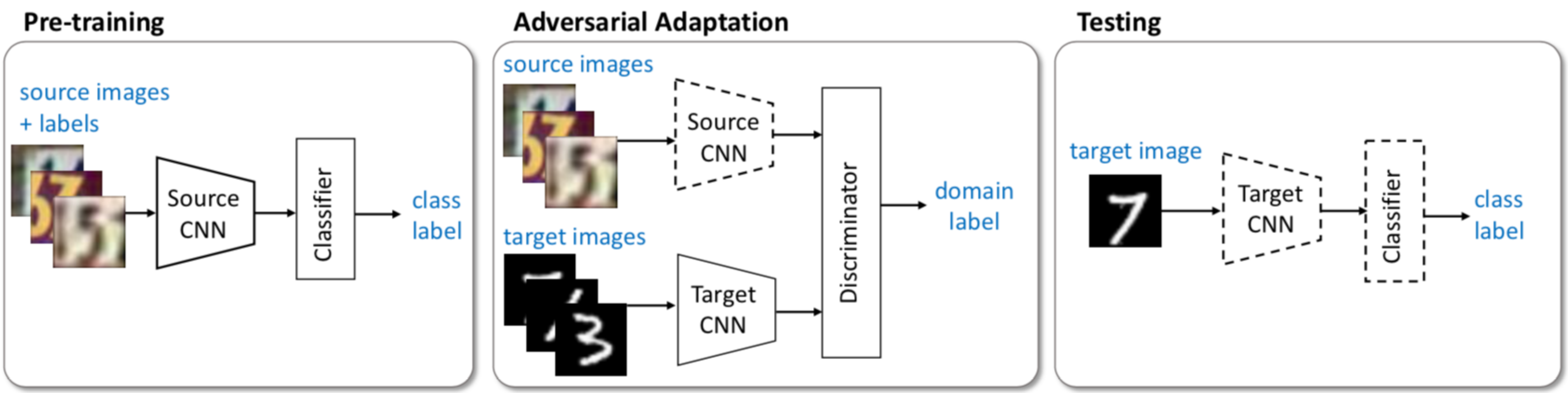


Figure 3: An overview of our proposed Adversarial Discriminative Domain Adaptation (ADDA) approach. We first pre-train a source encoder CNN using labeled source image examples. Next, we perform adversarial adaptation by learning a target encoder CNN such that a discriminator that sees encoded source and target examples cannot reliably predict their domain label. During testing, target images are mapped with the target encoder to the shared feature space and classified by the source classifier. Dashed lines indicate fixed network parameters.

Loss functions : $\mathcal{L}_{Adv}$

- Idea: try to decorrelate $E_{rec(Q+\delta, t+\delta)}$ from E_{true}.

$$\mathcal{L}_{Adv} = \lambda_{IBD} \times \frac{1}{\sigma_{E_{rec}} \sigma_{E_{true}}} \frac{1}{N_{IBD}} \sum_{i=1}^{N_{IBD}} \left(E_{rec,i} - \bar{E}_{rec}\right) \times \left(E_{true,i} - \bar{E}_{true}\right)$$

— A first idea…

— Probably necessary at some point to find something smarter.

— Actually, we'll probably need to find ways to guide the (δQ,δt) perturbation pattern
- Cannot simply be random variations between PMTs (or will only affect the E resolution).
- Ex : A loss that would favor a wrong Mass ordering ?
- Ex : something causing patterns involving in some way the response linearity ?
  (like the Q linearity in the electronics) ?
- Must remain rather simple (batch learning).

- One technical challenge : too difficult to regress on 34000 input parameters.

—> Find a systematic way to modify Q and t.
We plan to design a function with a limited number of learnable parameters.

For each PMT i $\begin{bmatrix} \delta Q \\ \delta t \end{bmatrix}_i = F(Q_i, t_i, X_i, Y_i, Z_i, E_{dep}^{true}, V_{dep}^{true}, \ldots)$

Loss functions : $\mathcal{L}_{Reg}$

$$\mathcal{L}_{Reg} = \mathcal{L}_{Reg,1} + \mathcal{L}_{Reg,2} + \ldots \qquad \mathcal{L}_{Reg,k} = \sum_{i=1}^{N_{Reg,k}} \sum_{j=1}^{N_{obs}} \lambda_{reg}^{k,j} \left(O_{kij} - O'_{kij}\right)^2$$

- Here, we need to determine what control data samples to use, and what observables.
- Observables
  - Energy ?
  - Vertex ?
  - Aggregated features like this used in this recent [1] JUNO publication ?
  - Lower level ?

- Data control samples ?
  - A lot of calibration data will be available : but need to be sure they are representative enough of the physics data we need.
  - Physics runs data : enriched background samples (Ex : cosmogenics after-muon events, …)

[1] Ex: "Energy reconstruction for large liquid scintillator detectors with ML techniques: aggregated features approach", Gavrikov et al, Eur.Phys.J.C 82 (2022) 1021.

A few examples we looked at and/or that have been mention within JUNO

- Domain adaptation [1, 2] : Tools trained to produce discriminative features that are the same in source and target domains.
- GANs
- Methods providing the regressed quantity + an uncertainty.

  Not convinced such methods would indicate directly the scenarios at detector level, that still can change the oscillation analysis, even when all control data is used to train them.

- Pivot method [3] : an Adv NN to eliminate the dependence of an algorithm on nuisance parameters.

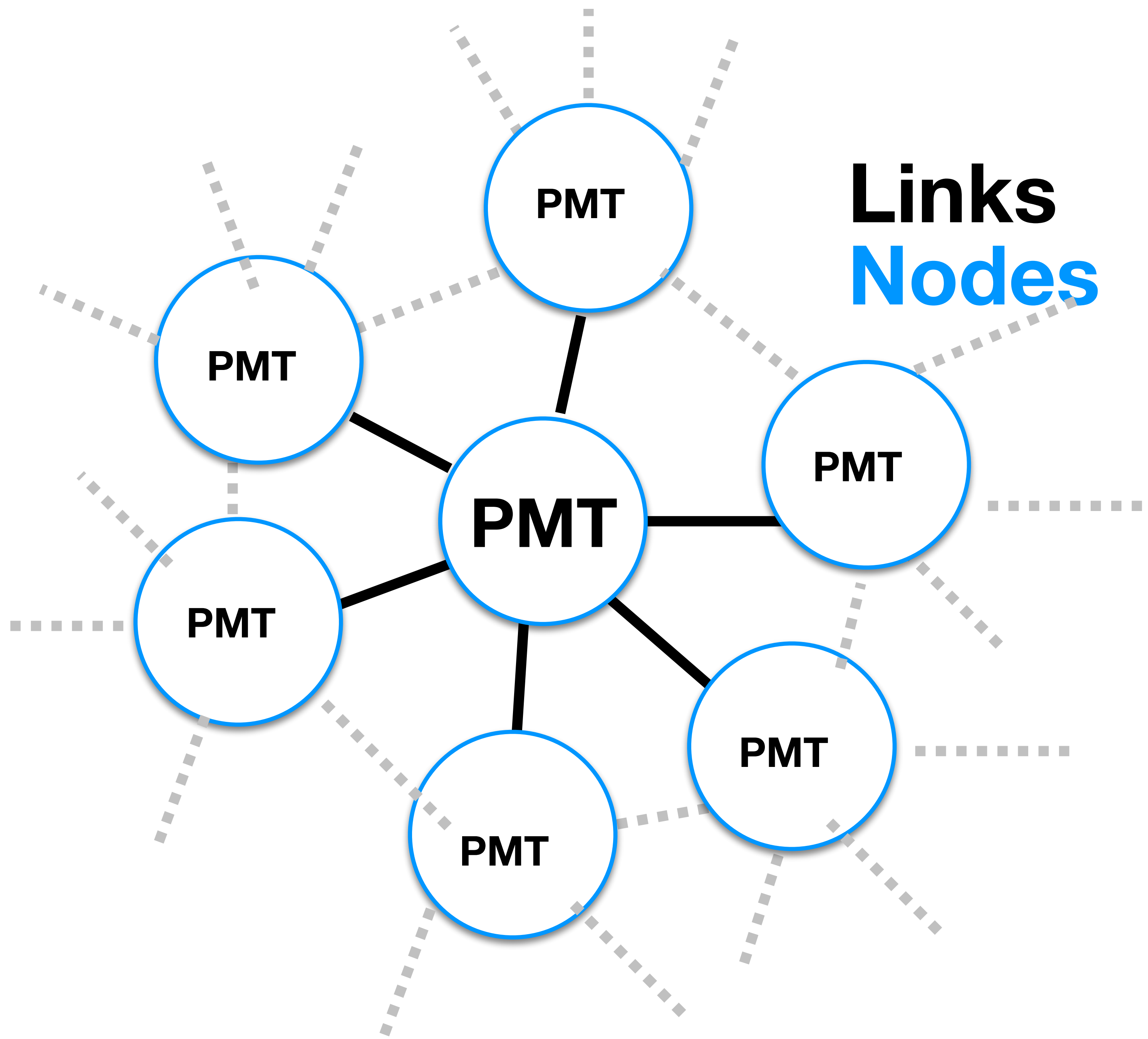
  The adaptation to > 35k nuisance parameters is not obvious.

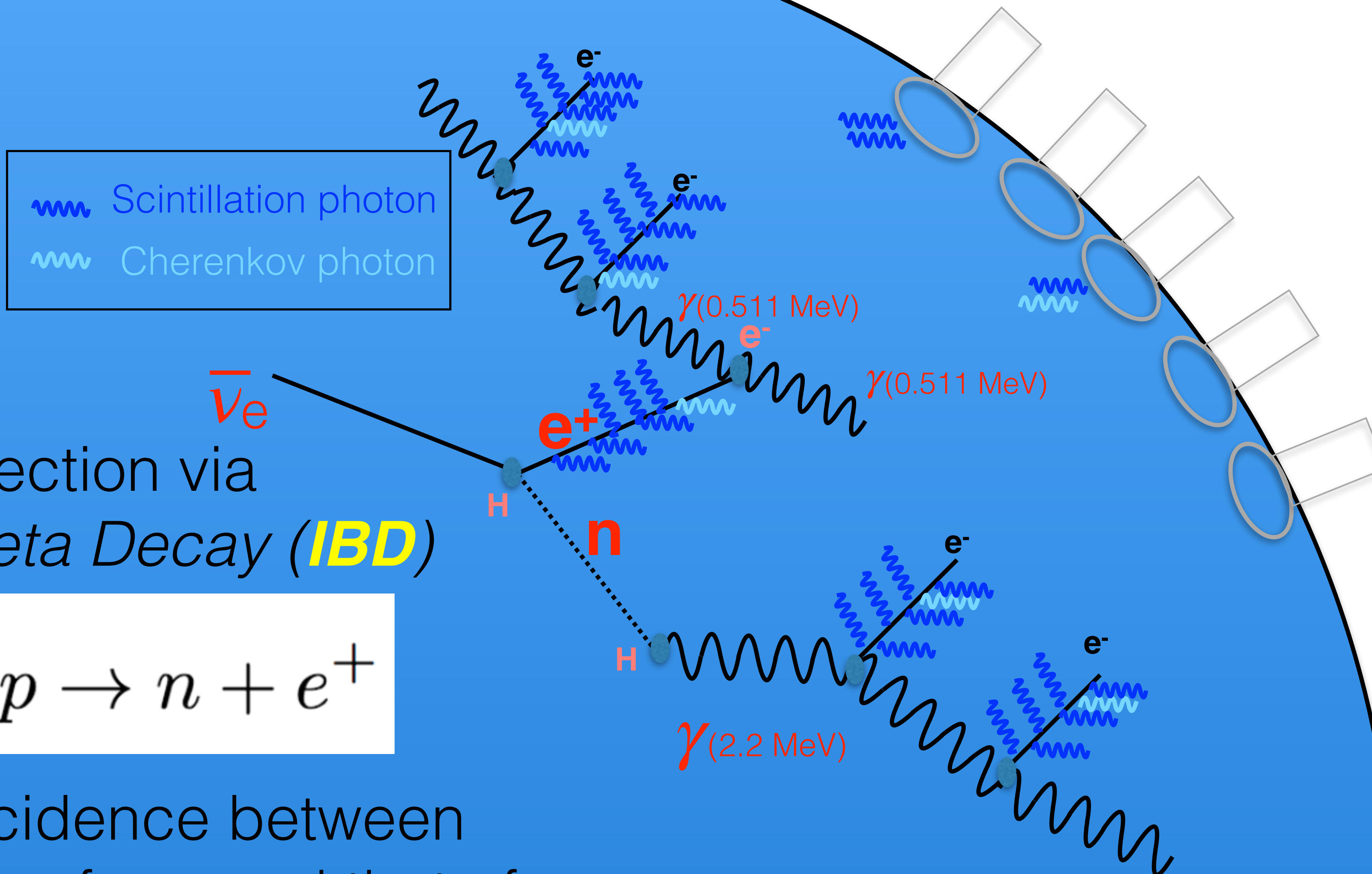
- Also see https://iml-wg.github.io/HEPML-LivingReview
  - Esp. sections on : Decorrelation methods, Generative models and density estimation, uncertainty quantification, etc.

[1] « Learning to Pivot with Adversarial Networks », Gilles Louppe(NY U.), Michael Kagan, Kyle Cranmer(NY U.), arXiv:1611.01046

[2] « Unsupervised Domain Adaptation by Backpropagation», Y. Ganin and V. Lempitsky, Proceedings of the 32nd International Conference on Machine Learning, https://proceedings.mlr.press/v37/ganin15.pdf

[3] « Adversarial Discriminative Domain Adaptation », E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, arXiv:1702.05464


PMT
PMT
PMT
PMT
PMT
PMT
PMT
Links
Nodes



Detection via

*Inverse Beta Decay (**IBD**)*

$$\bar{\nu}_e + p \rightarrow n + e^+$$

Seek coincidence between the detection of a **e+** and that of a **n**, Δt < 200 μs

Liquid Scintillator Sphere

Inputs to reco algorithms

- **Charge $Q_i \propto$ #Photons that hit PMT_i**
- **$t_i \propto$ Estimated time of 1st hit in PMT_i**
- **Other Waveform parameters....**