



AI and Machine Learning Applications at the Near Detector of the T2K Experiment

Anaëlle Chalumeau

for the T2K collaboration & ND280 AI/ML working group

EPS-HEP Marseille, 07/07/2025

↳ T16 - AI for HEP

Overview

- The T2K experiment and its Near Detector
- The AI/ML working group
- Overview of the group activities:
 1. SFGD Momentum reconstruction and PID with a BDT (TMVA)
 2. Global ND280 PID with BDT (XGBoost)
 3. Identify EM shower with PointNet
 4. 2D+3D CNN for e/γ classification
 5. Other projects using ND280 data (Omnifold, Normalizing Flows)
 6. PID in the SFGD with a Transformer

The T2K experiment & its Near Detector



The T2K experiment & its Near Detector



particle accelerator to
create neutrino beam

The T2K experiment & its Near Detector

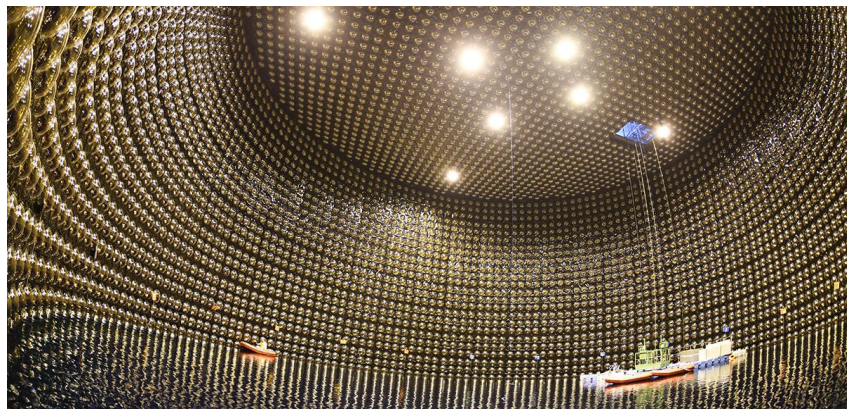


near detector: ND280



particle accelerator to
create neutrino beam

The T2K experiment & its Near Detector



far detector: Super-Kamiokande



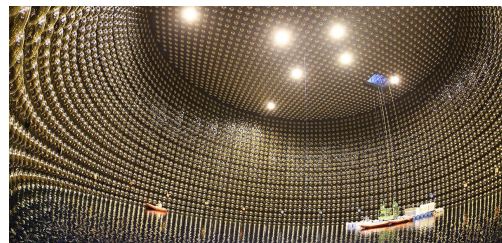
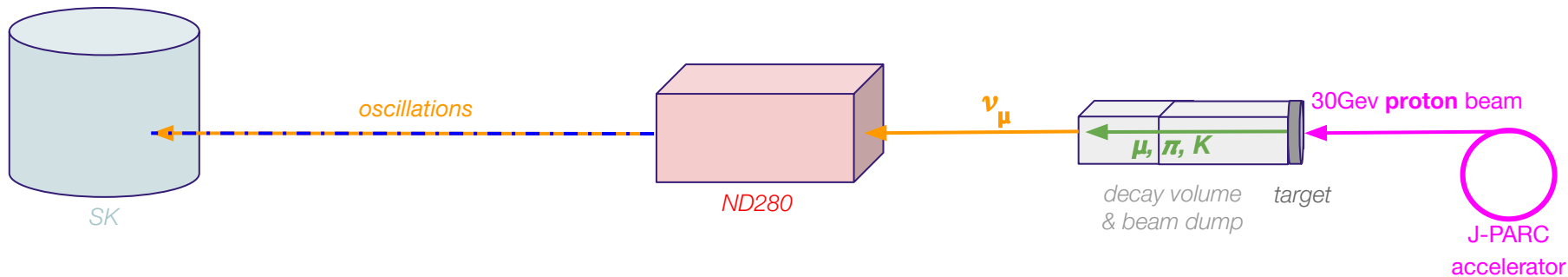
near detector: ND280



particle accelerator to
create neutrino beam

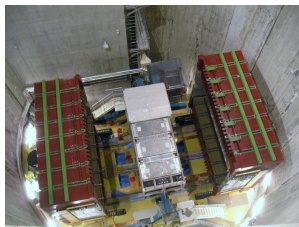
The T2K experiment & its Near Detector

goal: measure neutrino oscillation parameters



Super-Kamiokande

295km



ND280

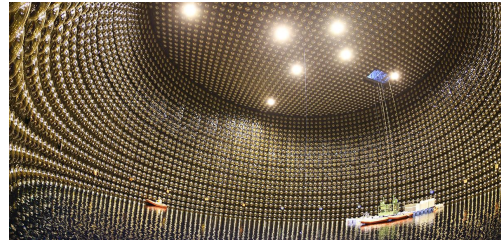
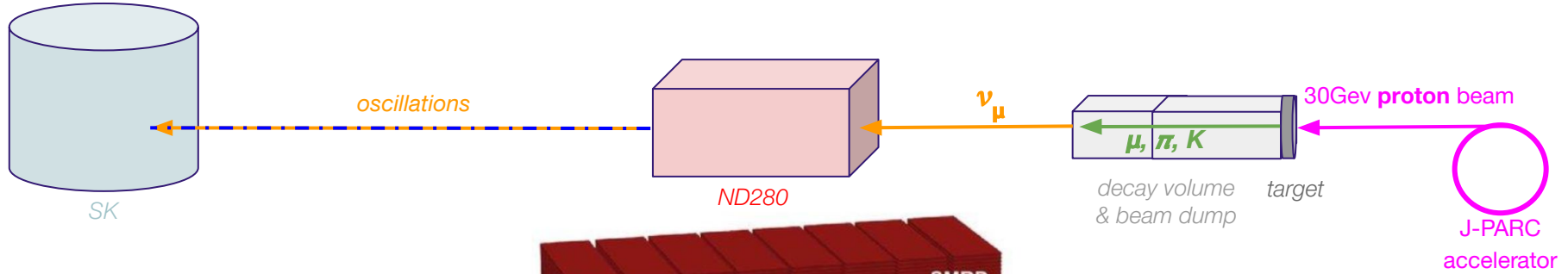
280m



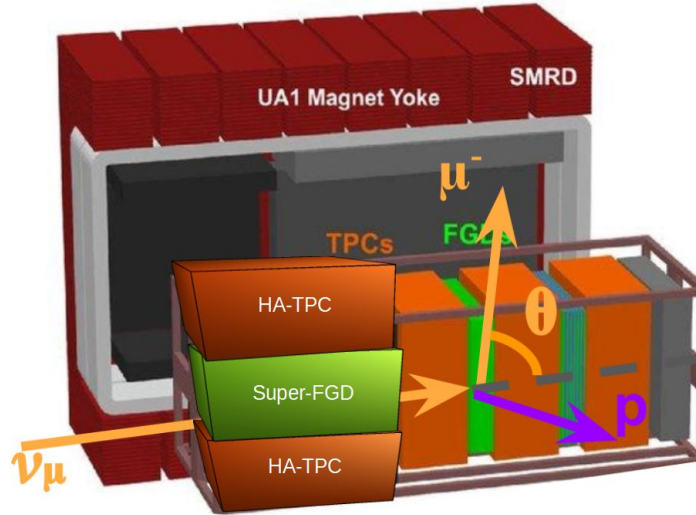
neutrino beam

The T2K experiment & its Near Detector

goal: measure neutrino oscillation parameters



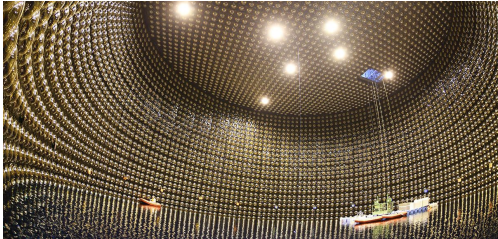
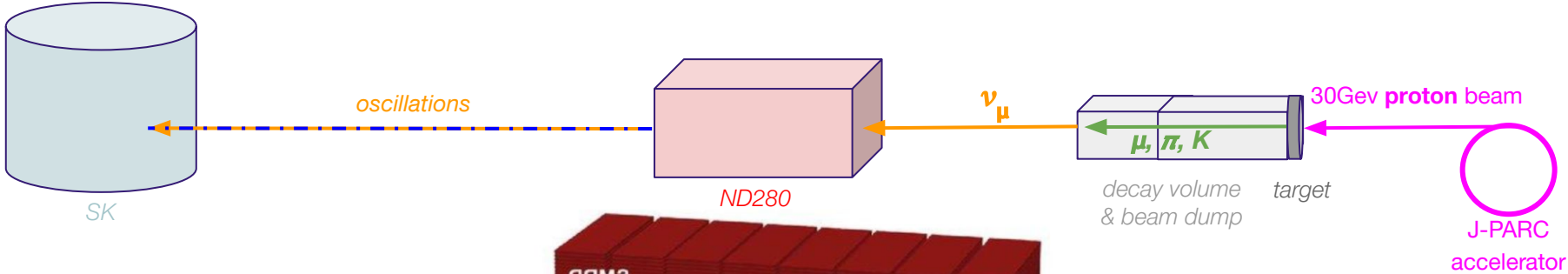
Super-Kamiokande



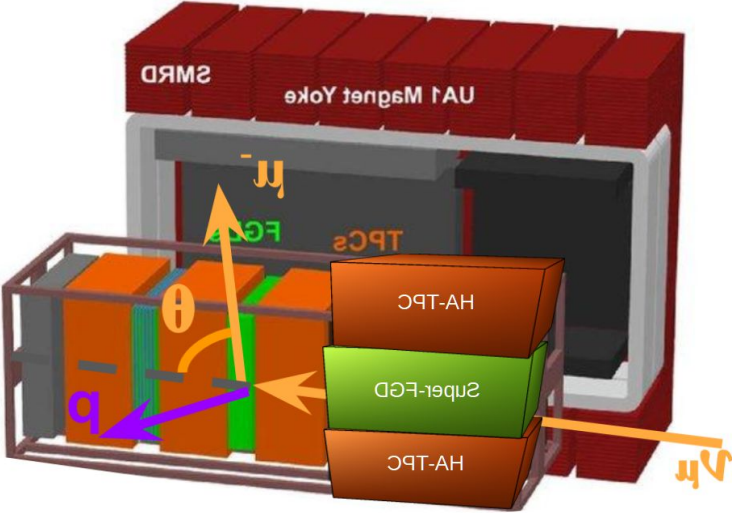
neutrino beam

The T2K experiment & its Near Detector

goal: measure neutrino oscillation parameters



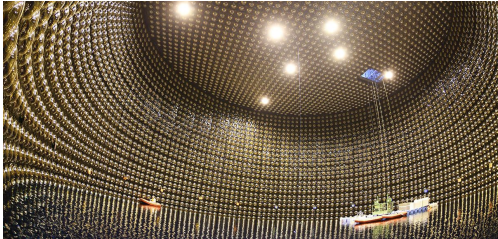
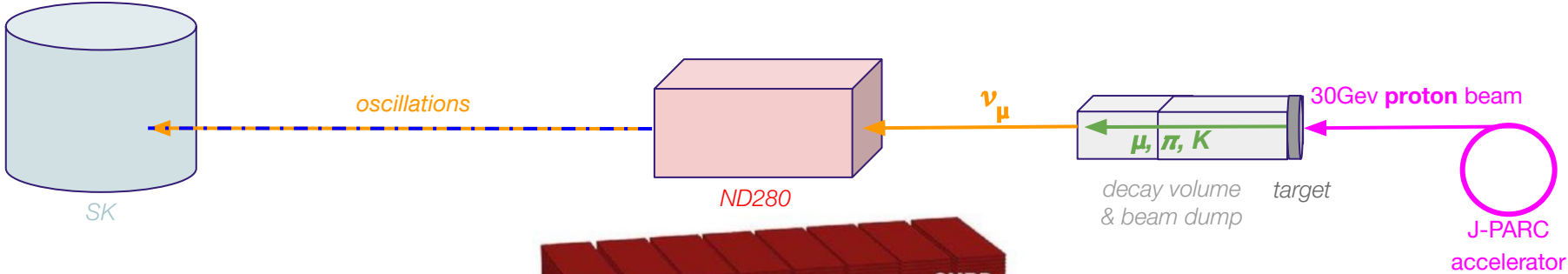
Super-Kamiokande



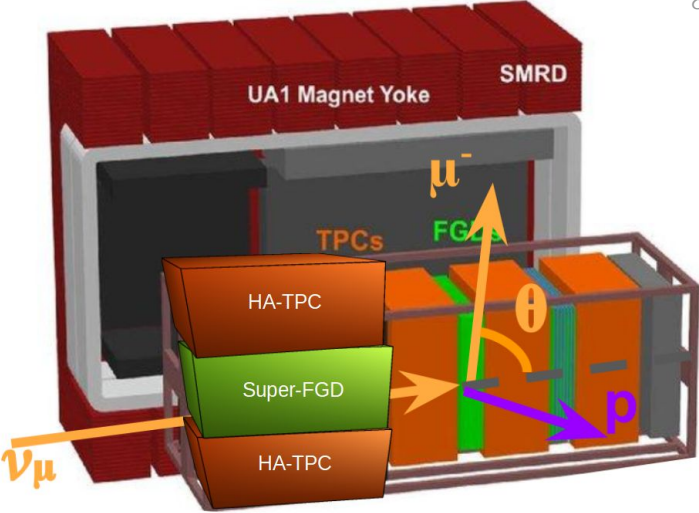
neutrino beam

The T2K experiment & its Near Detector

goal: measure neutrino oscillation parameters



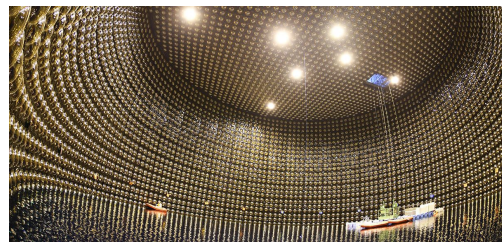
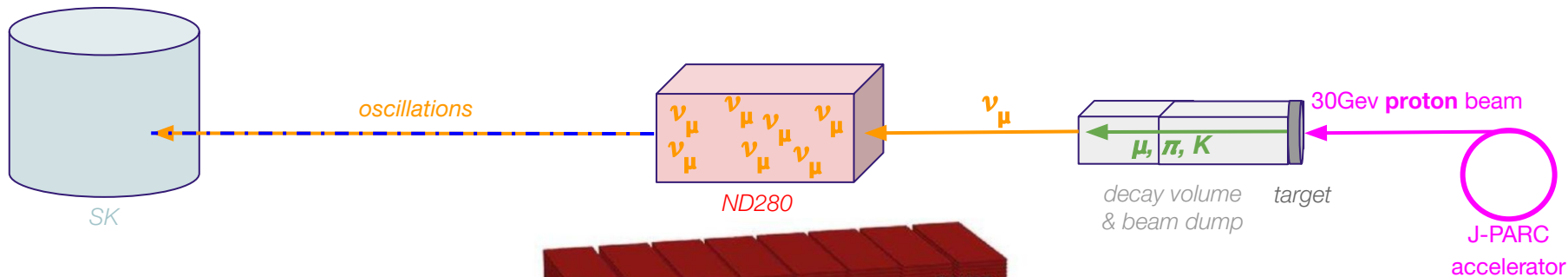
Super-Kamiokande



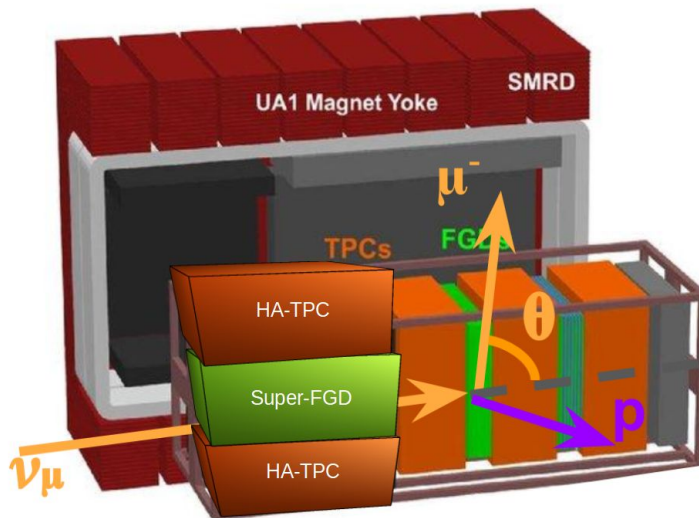
neutrino beam

The T2K experiment & its Near Detector

goal: measure neutrino oscillation parameters



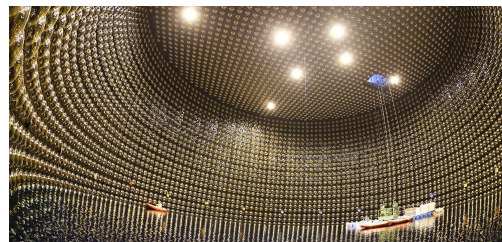
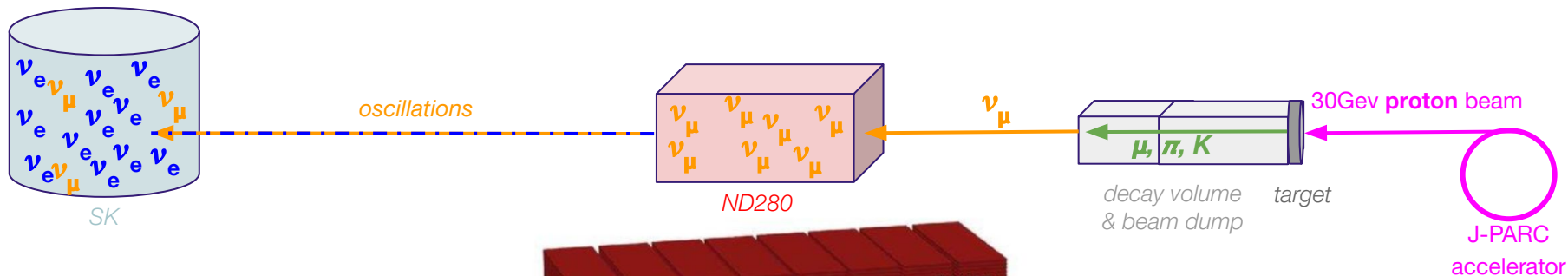
Super-Kamiokande



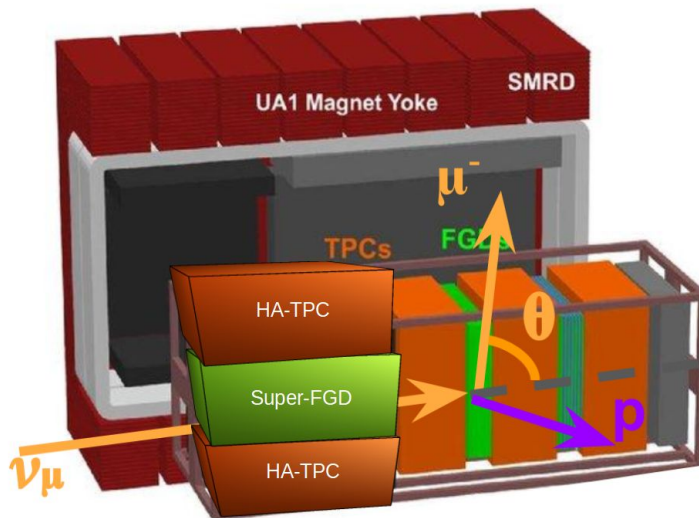
neutrino beam

The T2K experiment & its Near Detector

goal: measure neutrino oscillation parameters

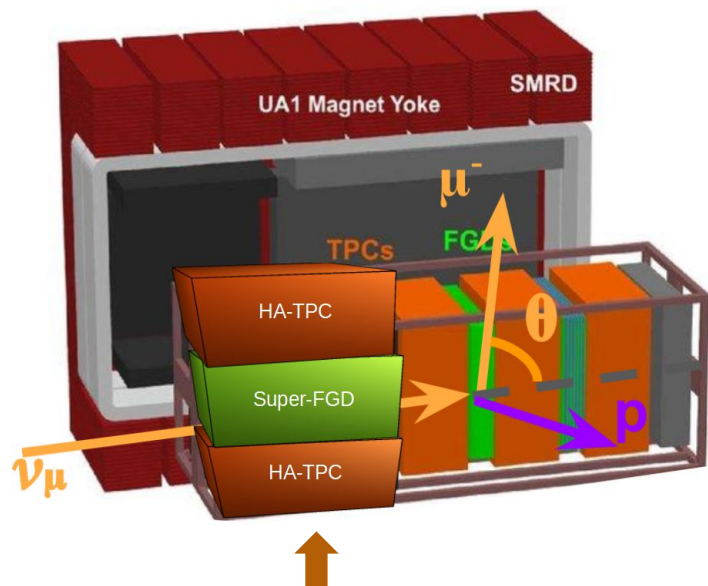


Super-Kamiokande



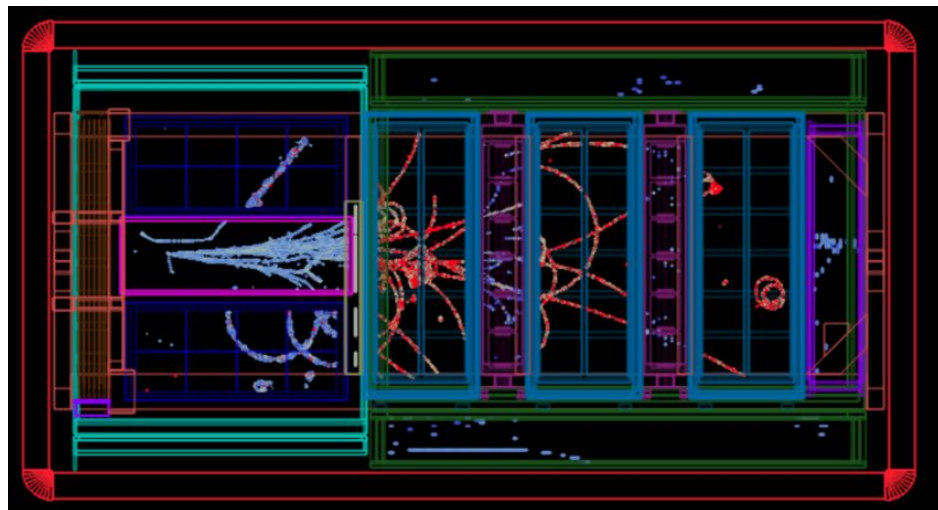
neutrino beam

The T2K experiment & its Near Detector

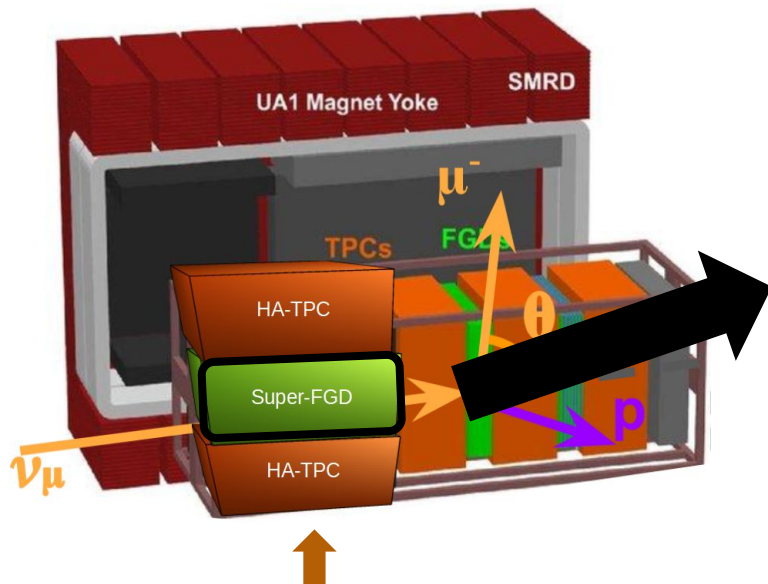


ND280 Upgrade installed last year!

& data taking since end of 2024



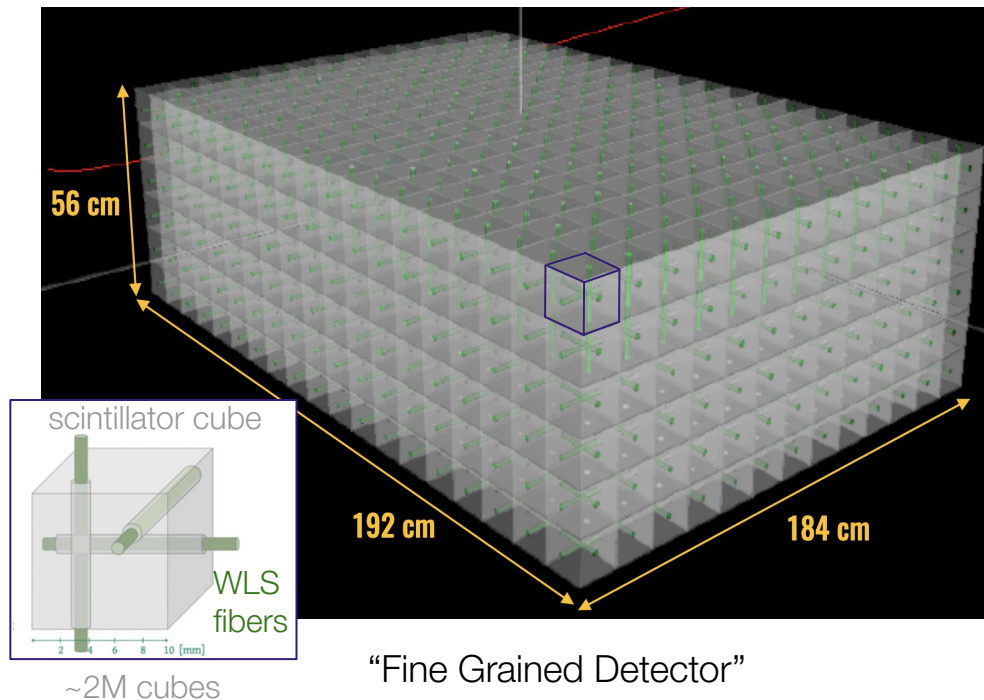
The T2K experiment & its Near Detector



ND280 Upgrade installed last year!

& data taking since end of 2024

The Super-FGD: a Scintillator Detector

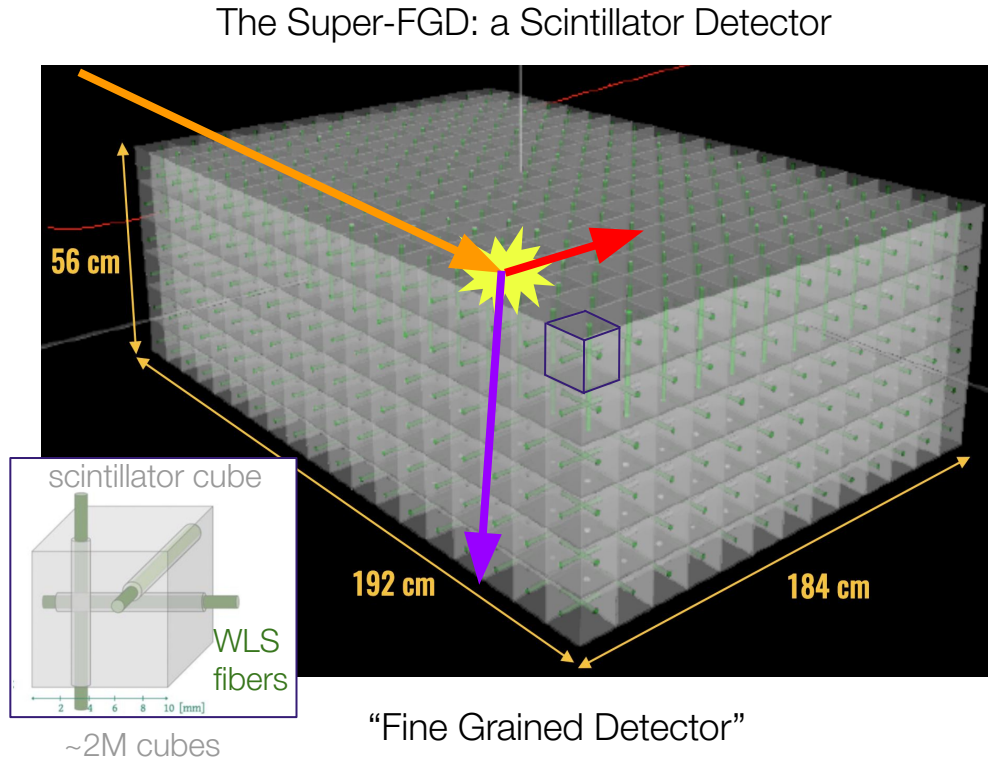


“Fine Grained Detector”

The T2K experiment & its Near Detector

New detector technology

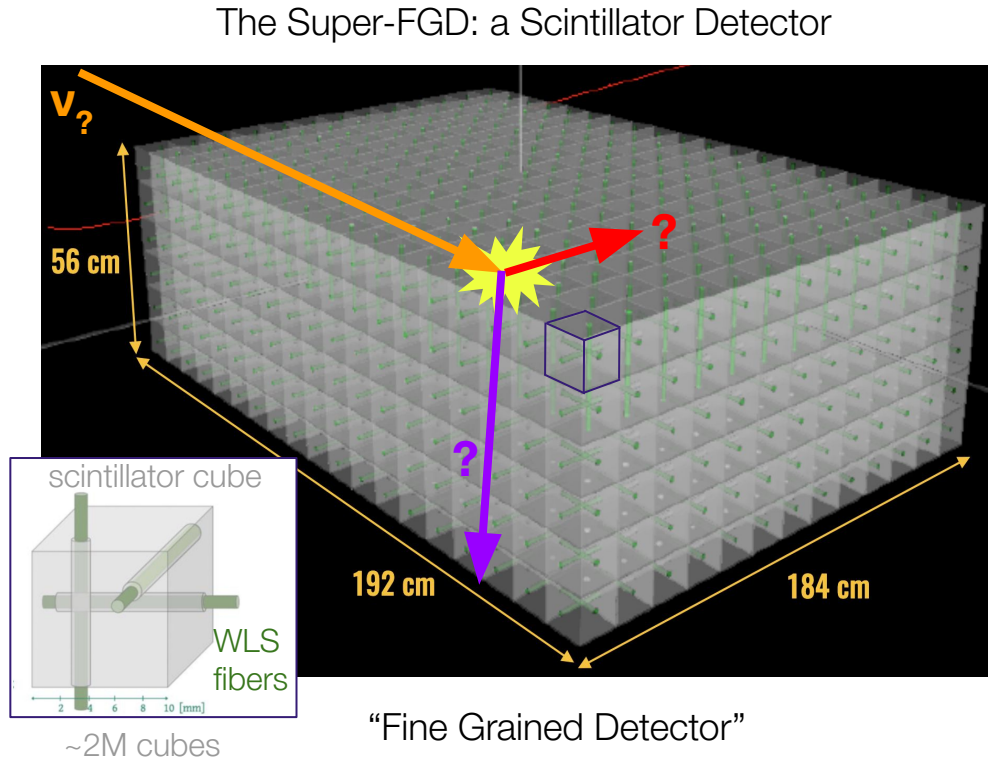
⇒ need new tools to identify the particle types (PID) from neutrino interaction using charge deposition in the detector



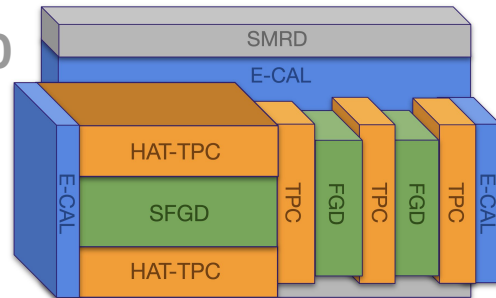
The T2K experiment & its Near Detector

New detector technology

⇒ need new tools to identify the particle types (PID) from neutrino interaction using charge deposition in the detector

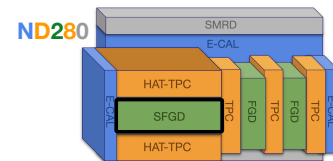


The ND280 AI/ML working group



- Recent working group dedicated to AI/ML methods (since Nov. 2024)
- Convener is Saúl Alonso Monsalve (ETH Zurich)
- ~ 10 active analysers
- In different part of the experiment, various task focused around the Near Detector:
 - ★ **reconstruction**: vertex activity, track fitting, momentum reco → 3 projects
 - ★ **analysis**: PIDs, unfolding, modelling posterior systematics → 7 projects
 - ★ **simulation**: cross-section sampling → 1 project

1. SFGD Momentum reconstruction & PID with BDT (TMVA)

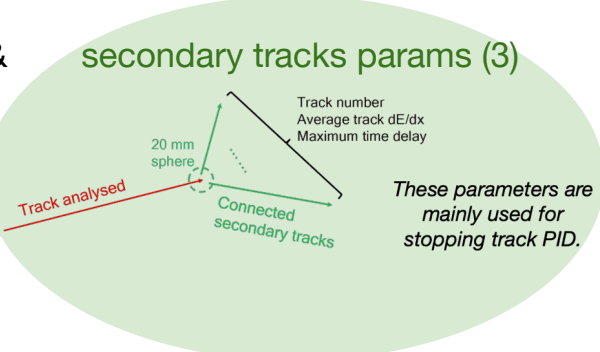
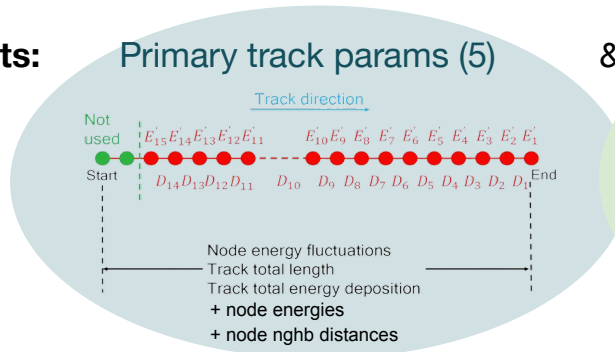


Inputs:

Primary track params (5)

&

secondary tracks params (3)



BDT HP: 5 to tune (done by hand):

Hyperparameter	Value
Decision tree number M	2000
Division point number K at each node	20
Decision tree maximum depth D_{max}	3
Shrinkage ν	0.05
Stochastic boosting fraction f	0.5

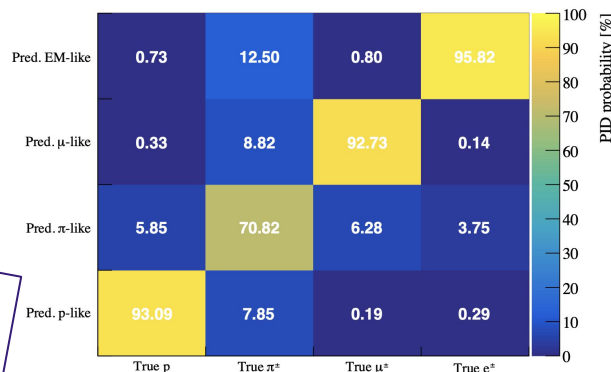
same for regression & classification

Training:

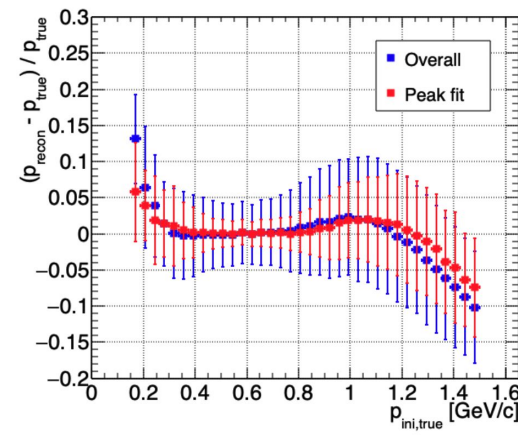
- on particle-gun MC data (i.e. 1 particle /event): p , $\pi^\pm$, $\mu^\pm$, $e^\pm$
- classifiers & momentum regressors

Test:

classification results



momentum resolution:

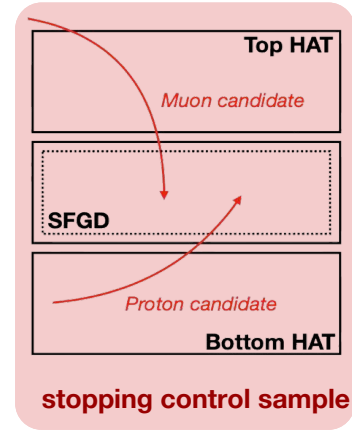


Currently in use in official T2K analysis software

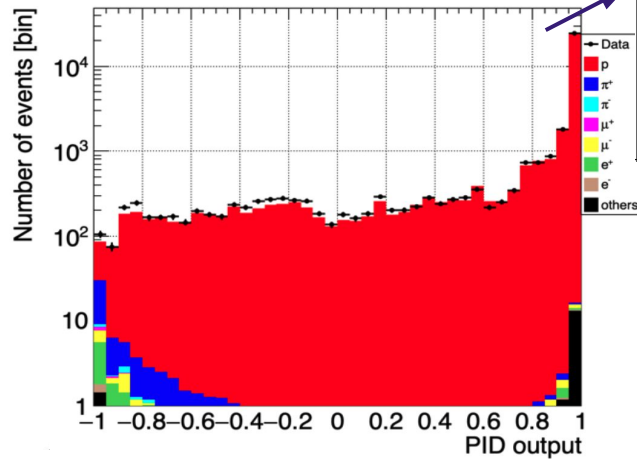
1. SFGD Momentum reconstruction & PID with BDT (TMVA)

BDT Systematics:

- for real application, apply BDT to MC & data $\Rightarrow$ check if have similar performances!
- If not $\rightarrow$ need to evaluate the difference
- Propage this difference as systematic source in the analysis
- Evaluation of systematics: **use stopping control sample**



PID efficiency:

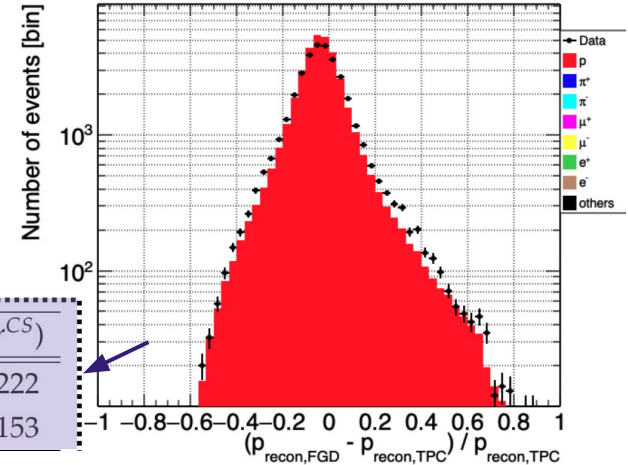


Control Sample	Data / MC
Proton	0.99 ± 0.00
Pion (π^+)	2.17 ± 0.06
Muon (μ^-)	1.32 ± 0.05
Electron / Positron	1.41 ± 0.15

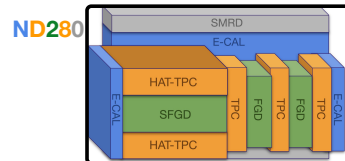
those values are propagated
as systematics

Control Sample	Data / MC (r^{CS})
Proton	1.2034 ± 0.0222
Muon (μ^-)	0.8840 ± 0.0153

Momentum resolution:

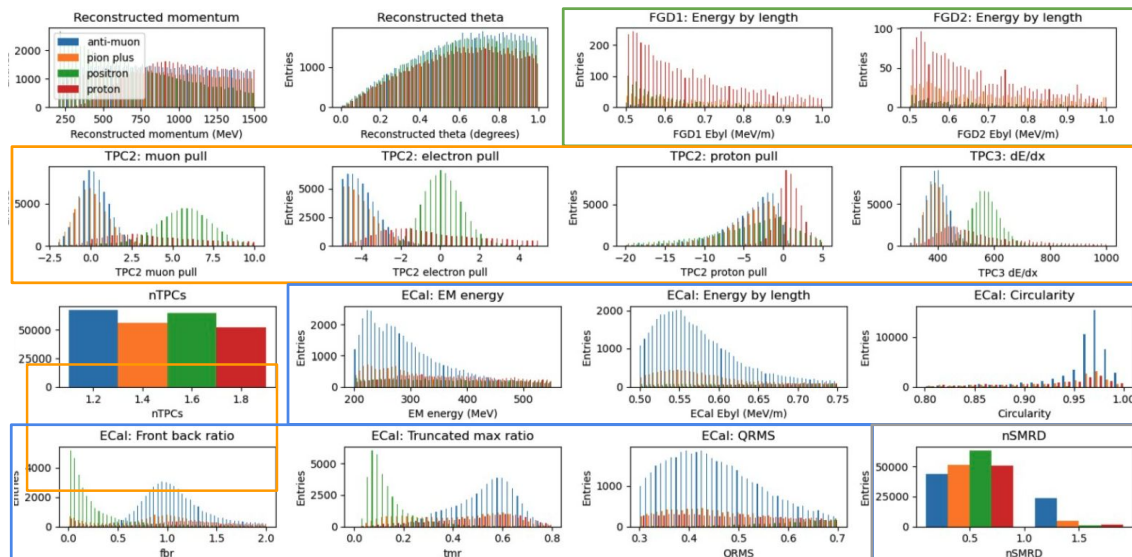


2. Global ND280 PID with BDT (XGBoost)

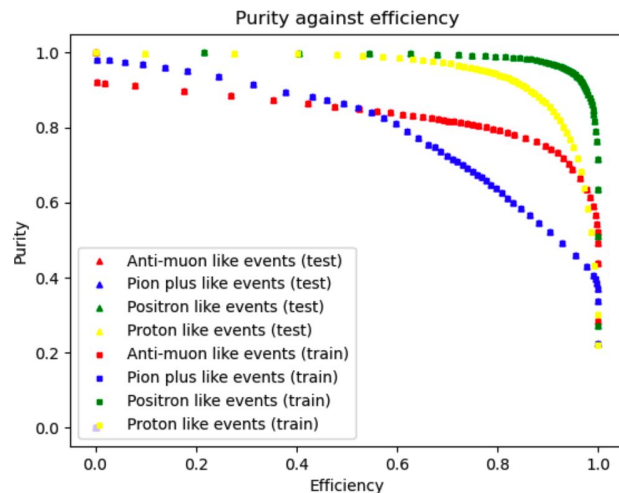


Goal: use inputs from many ND280 sub-detectors to get a global PID tool

Inputs: 16 variables from 4 of ND280 sub-detectors



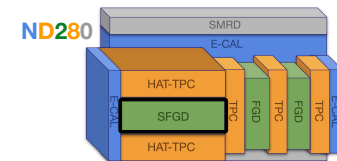
Results:



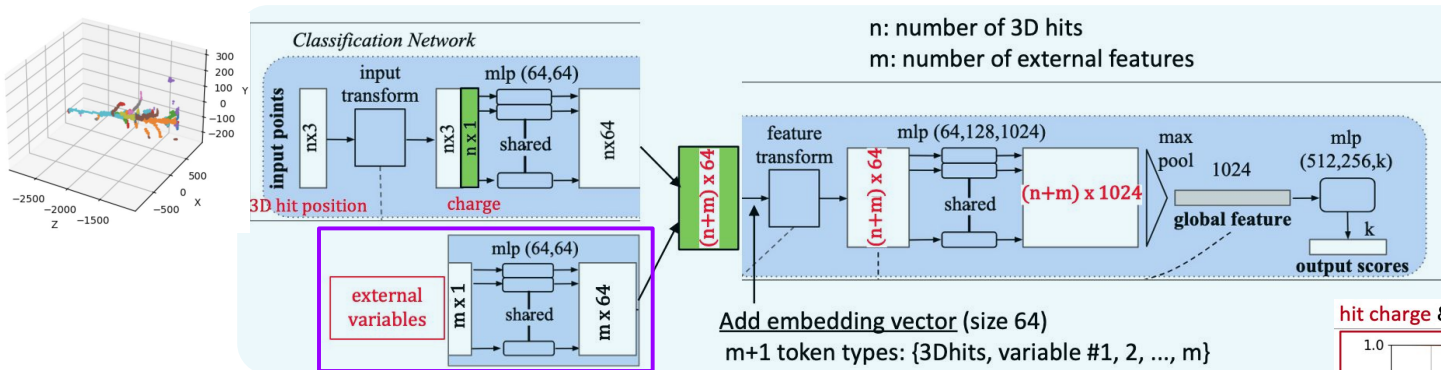
Training:

- on particle-gun MC data: p , π^+ , μ^+ , e^+ with 250k events
- use HYPEROPT for the HPO

3. Identify EM shower in SFGD: PointNet

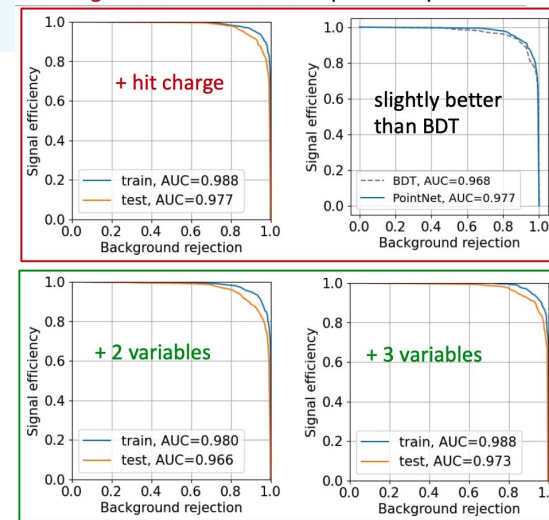


Architecture: PointNet (DNN for 3D point cloud data) with modifications



Results:

hit charge & external variables improve the performance



Goal: distinguish EM shower-like (e,y) particle from non EM ones (μ, π, p)

Training data: pgun MC data of e^- , μ^- with 8000 patterns +data augmentation

External features: adding global feature of the event increase performances!
will be tested: add all BDT variables as external features

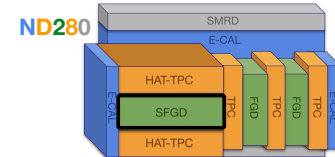
- total charge in the event
- total number of hits
- size of the shower

3. Identify EM shower in SFGD: PointNet

Question:

- segmentation network not used right? (from archi image) why don't we need it here
- how manageable is it to rm the 512 pt limit and make it changeable?
- plot updates with all eff,pur on same one?
- perf update with all BDT var?

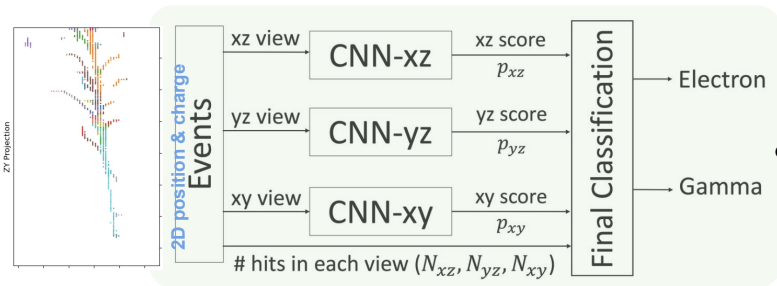
4. 2D+3D CNN for e/ γ classification



Architecture: experimental combination of 2D CNNs + sparse 3D CNN

ResNet50

SSCN



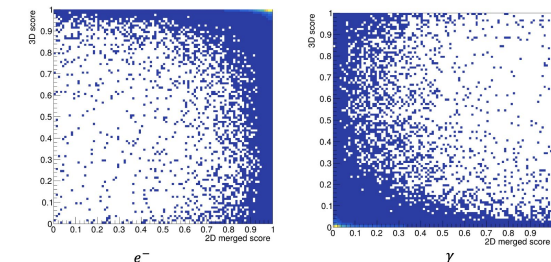
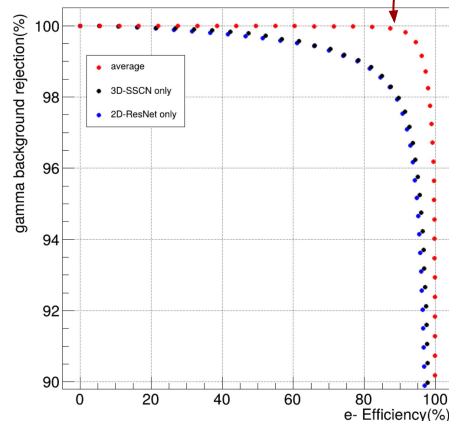
Sparse Submanifold
convolutional networks

$$(final\ score) = \frac{(2D\ score) + (3D\ score)}{2}$$

Training: pgun MC data of e^- and γ with 400k training event

Results:

model	e^- efficiency	γ rejection
3D-SSCN	95.5	95.3
3D-ResNet	94.9	95.8
2D-ResNet	95.3	94.4
3DSS+3DRes	96.1	96.4
3DSS+2DRes	98.3	97.9



2D vs 3D scores: not much correlations

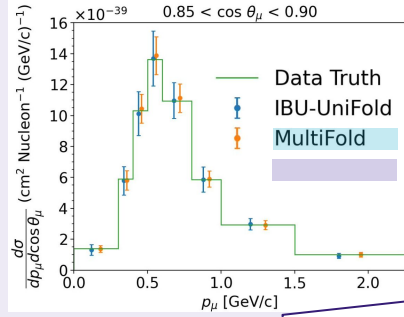
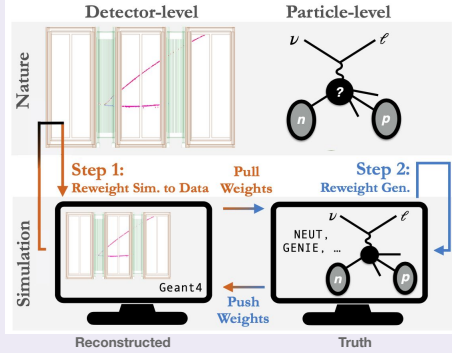
**e/γ events looks similar because of $\gamma \rightarrow e^+e^-$ creating similar shower of particles
 $\Rightarrow$ challenge is to distinguish them**

5. Other projects using ND280 data (Omnifold, Normalizing Flows)

Omnifold:

work of Roger Huang

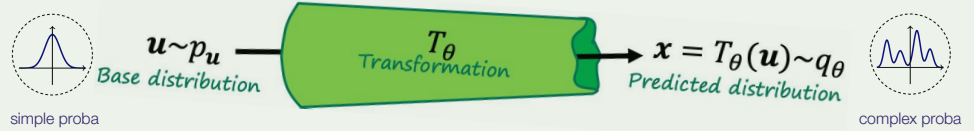
unbinned method to unfold ND280 data faster



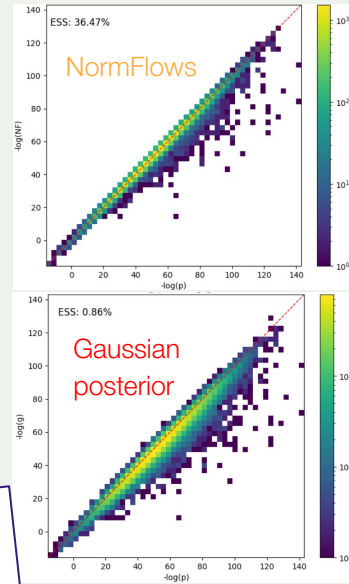
Discussions to incorporate these in T2K softwares

Normalizing flows:

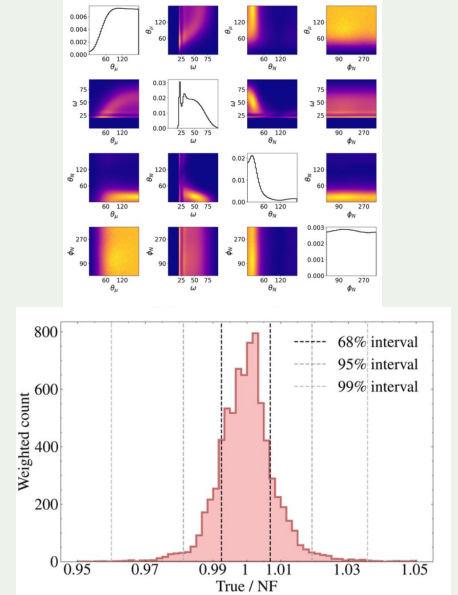
work of Mathias El Baz



Modeling posterior systematics



Efficient CCQE cross-section sampling



ND280

SMRD

E-CAL

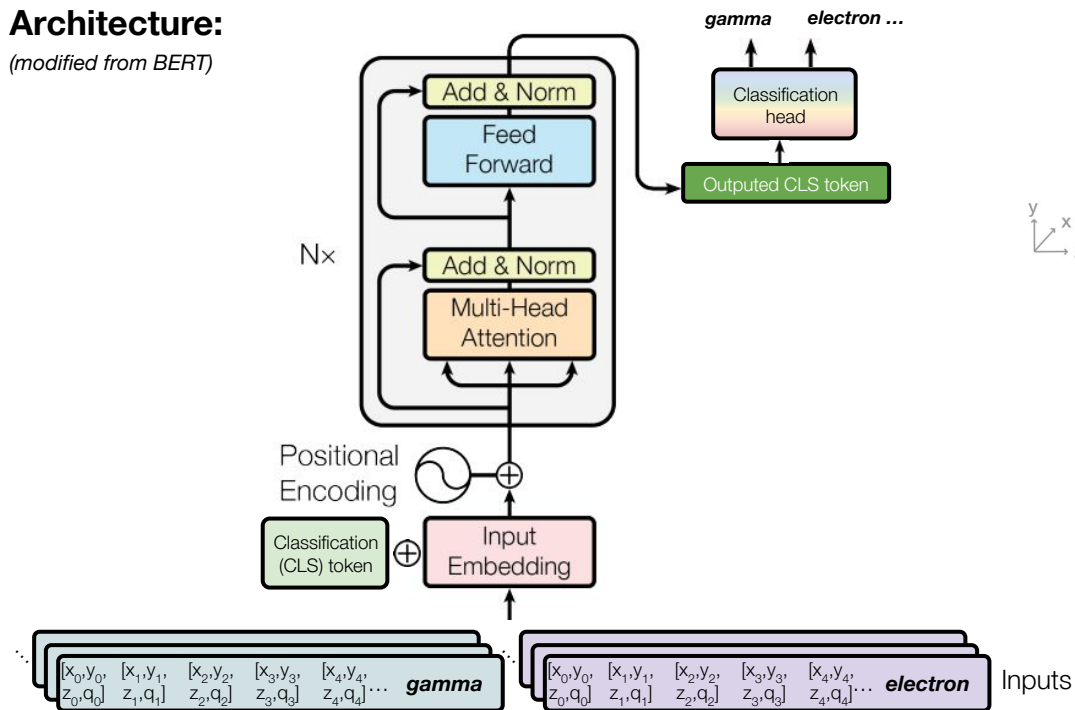
HAT-TPC

SFGD

FGD

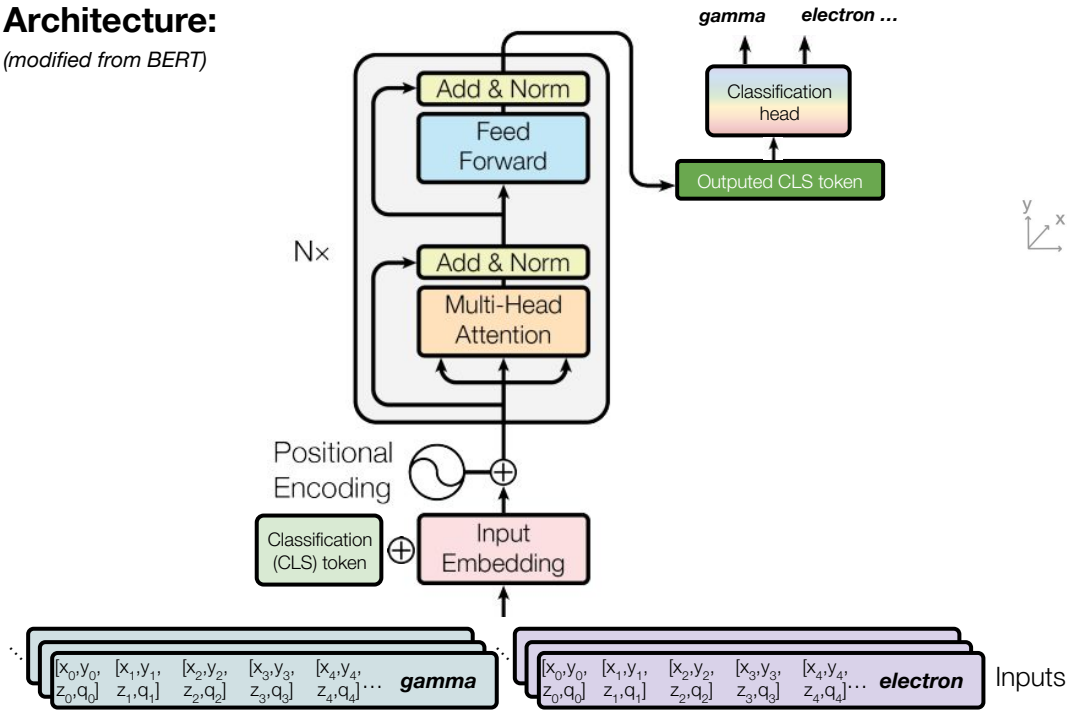
TPC

(modified from BERT)

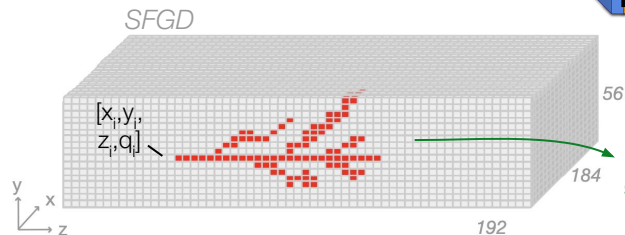


6. PID in the SFGD with a Transformer

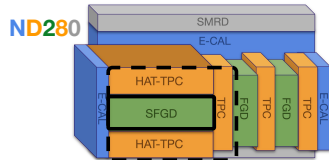
Architecture: (modified from BERT)



Inputs:

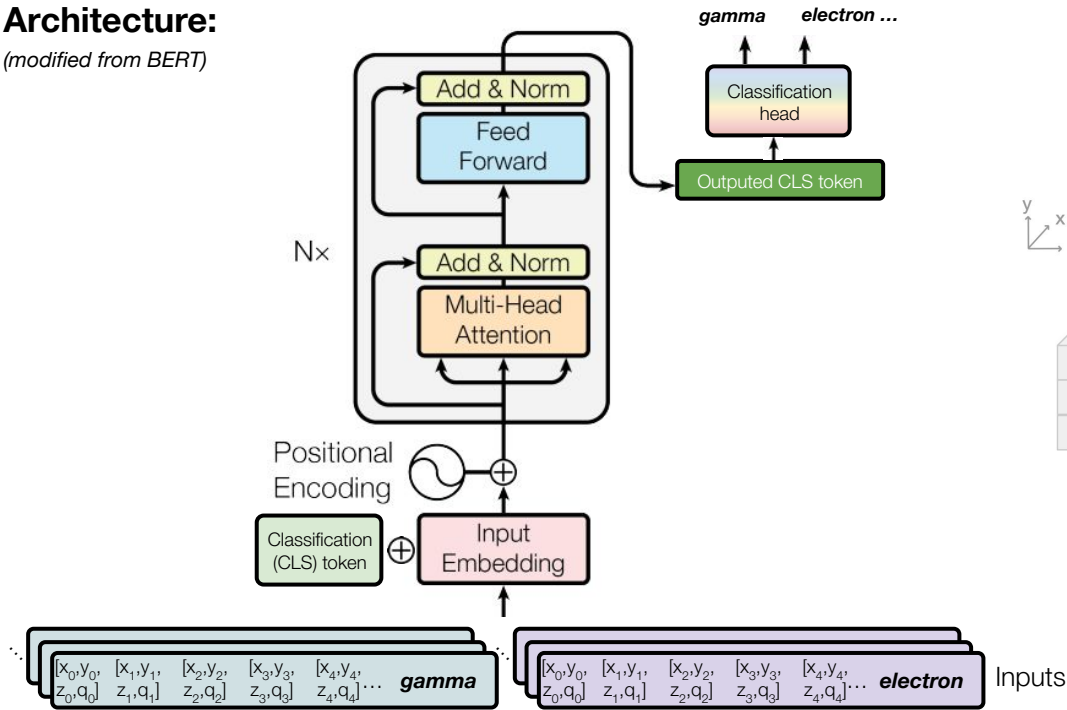


Too long
sequences of hits:
use Vision
Transformer
principle

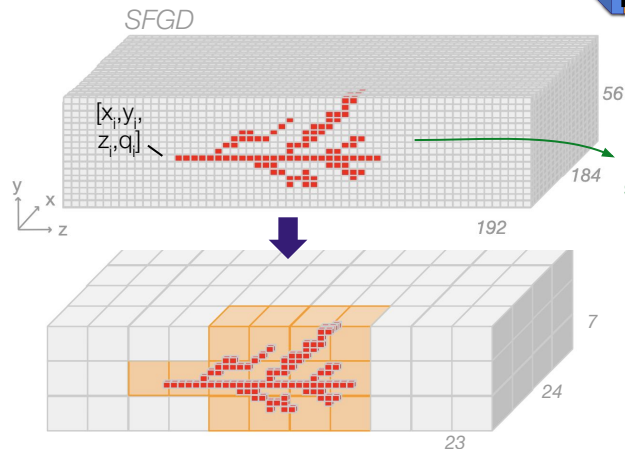


6. PID in the SFGD with a Transformer

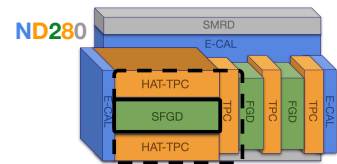
Architecture: (modified from BERT)



Inputs:



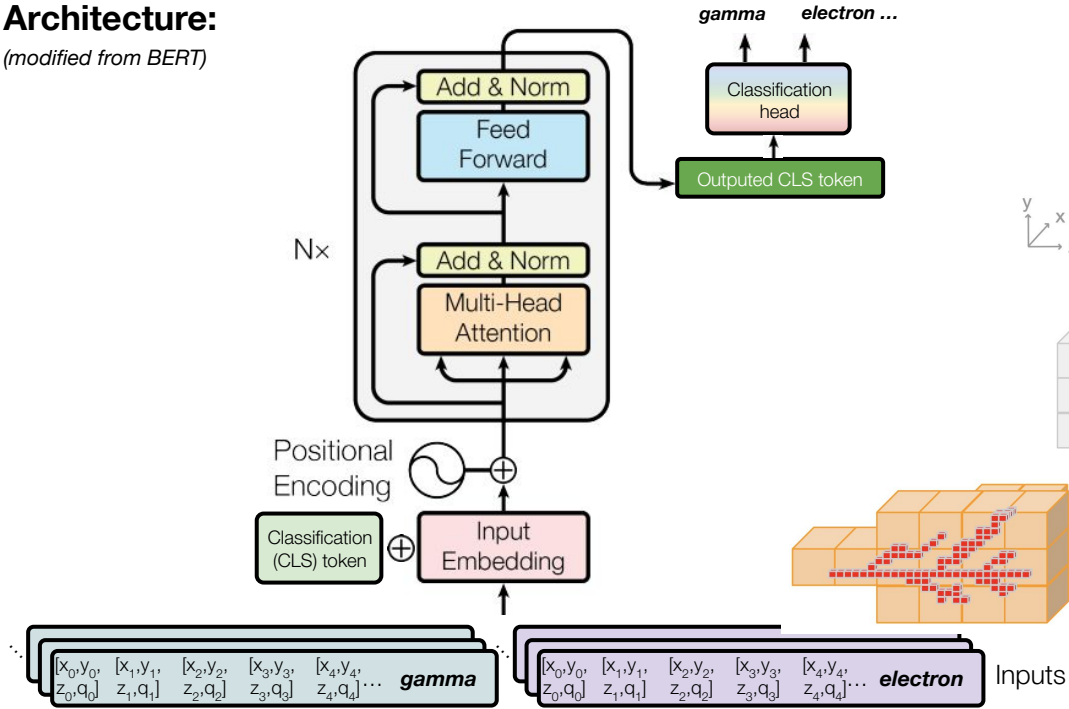
Too long
sequences of hits:
use Vision
Transformer
principle



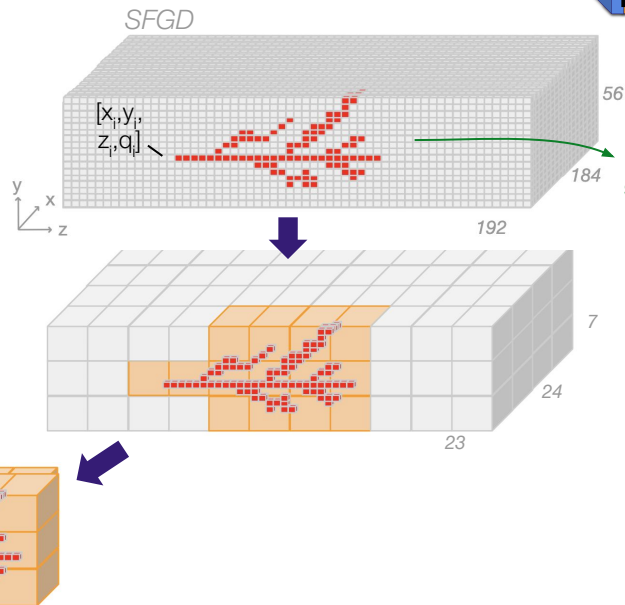
6. PID in the SFGD with a Transformer

Architecture:

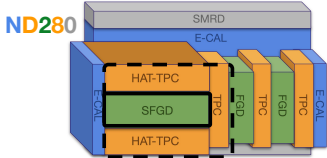
(modified from BERT)



Inputs:



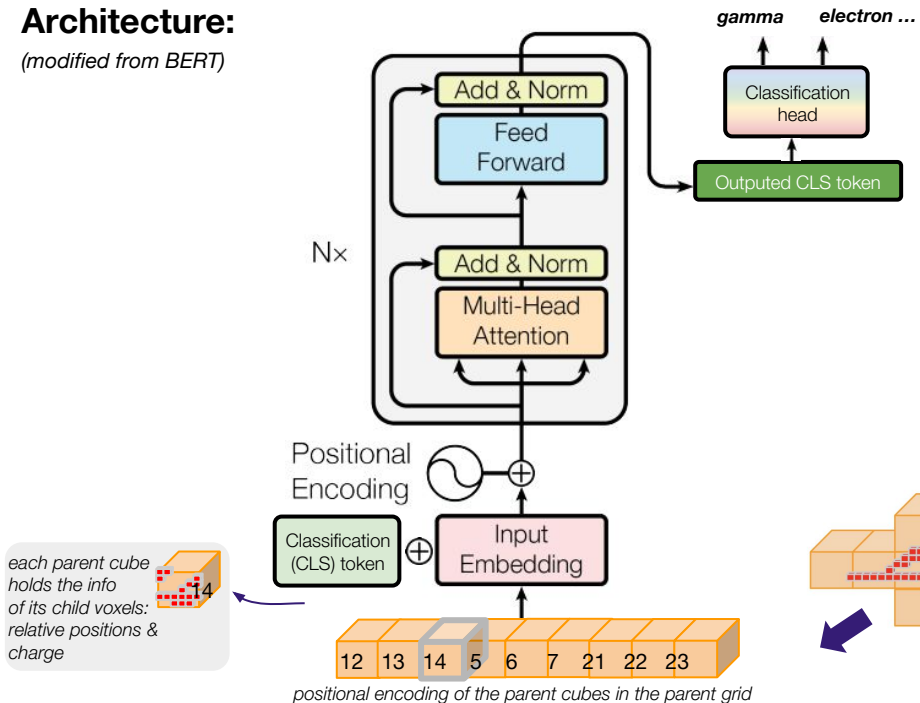
Too long sequences of hits: use Vision Transformer principle



6. PID in the SFGD with a Transformer

Architecture:

(modified from BERT)



Inputs:

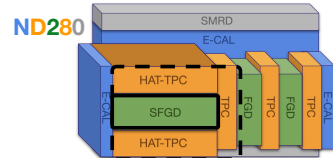
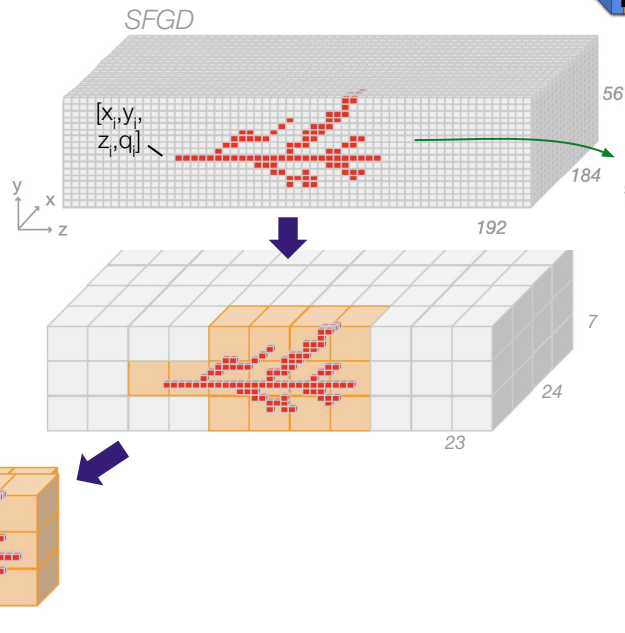
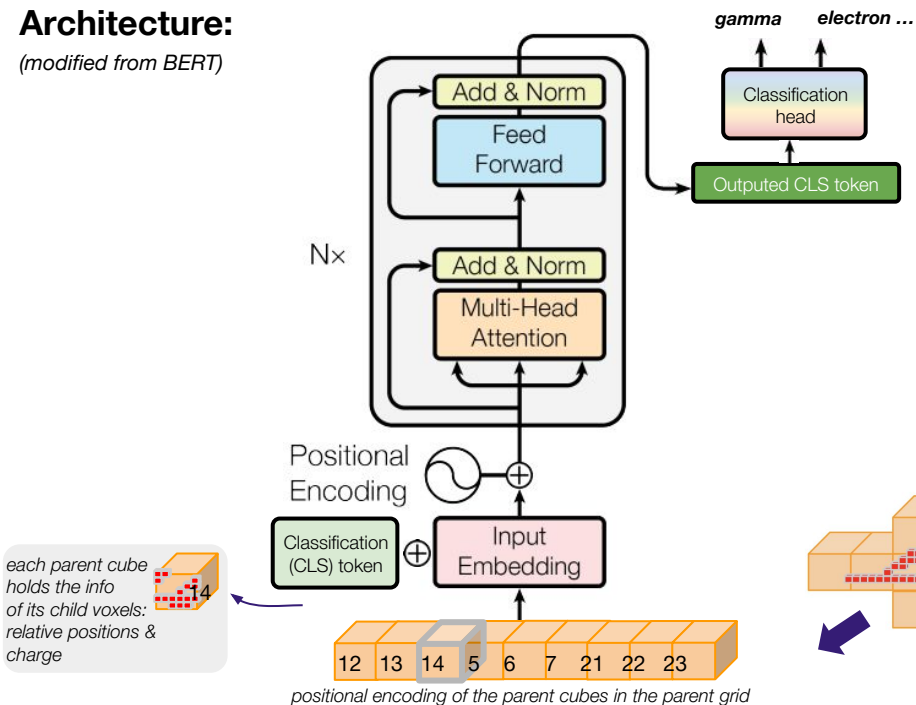


Diagram illustrating the ND280 detector structure, showing layers including SMRD, E-CAL, HAT-TPC, SFGD, and TPC.

(modified from BERT)



Too long
sequences of hits:
use Vision
Transformer
principle

Training:

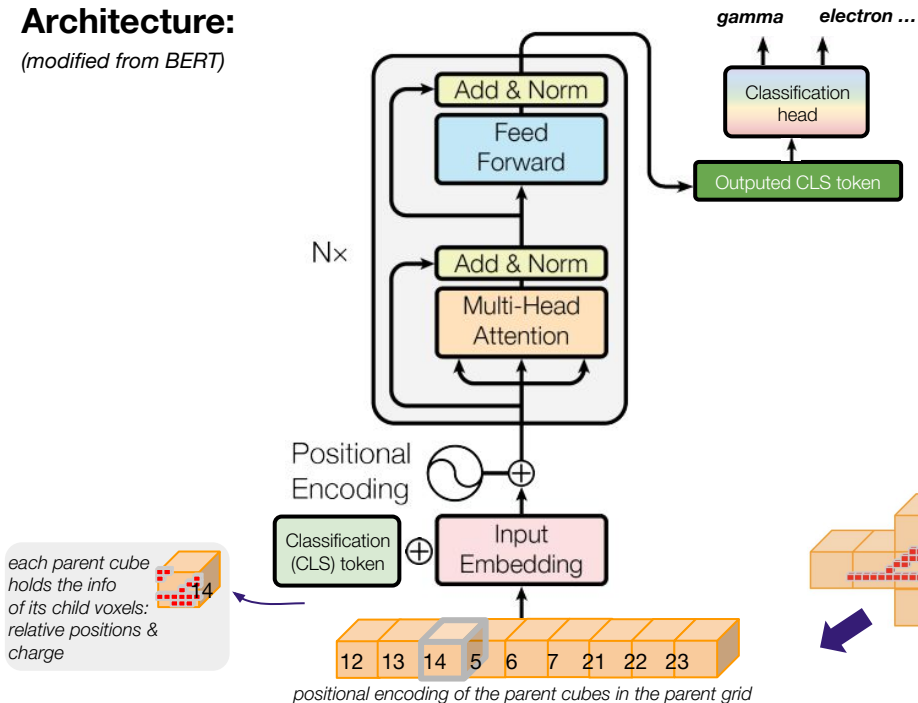
- particle-gun MC data: e^- , γ , π^- , μ^-
- either along 1 direction or isotropic

'z-pgun' *'iso-pgun'*

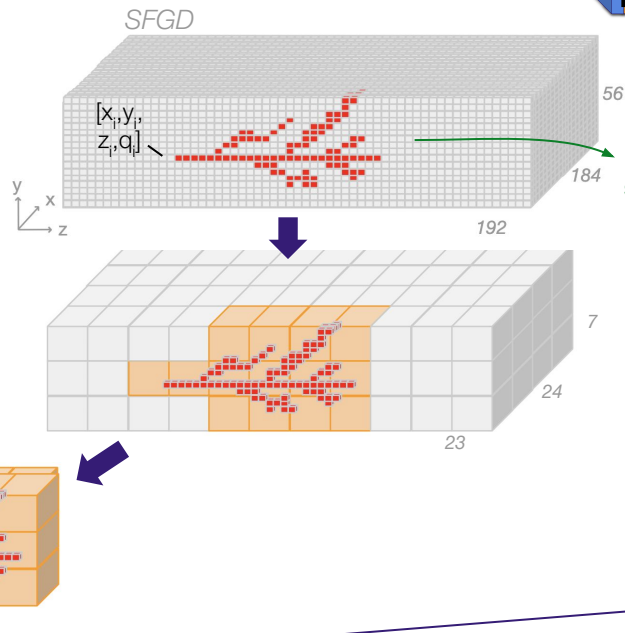
6. PID in the SFGD with a Transformer

Architecture:

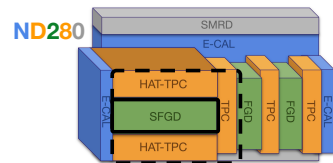
(modified from BERT)



Inputs:



Too long sequences of hits: use Vision Transformer principle

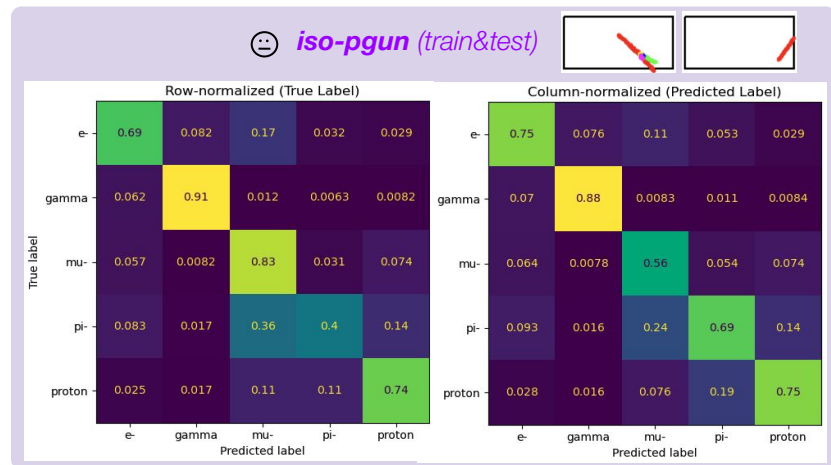
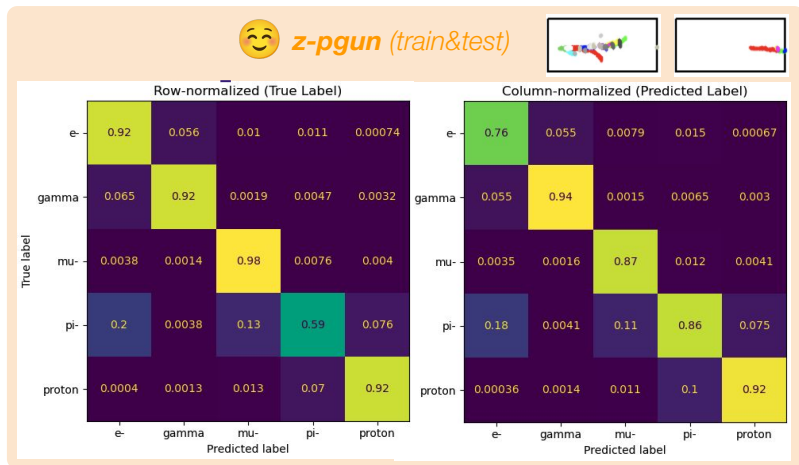


Training: - particle-gun MC data: e^- , γ , π^- , μ^-
 - either **along 1 direction** or **isotropic**
 'z-pgun' 'iso-pgun'

- architecture converted in C++ (Libtorch)
 - saved weights from training
 → 1st complex ML model implemented in analysis
 soft!

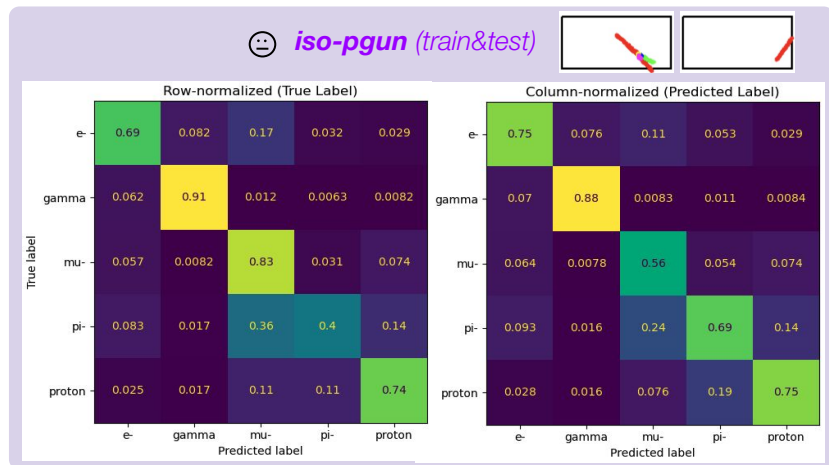
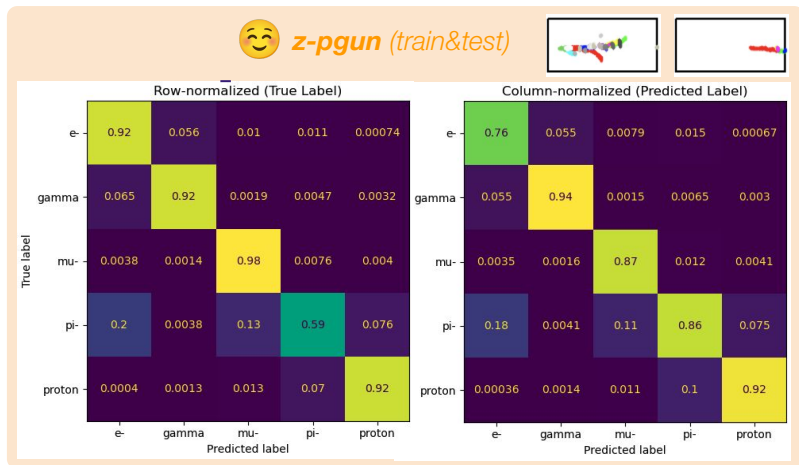
6. PID in the SFGD with a Transformer

Test:

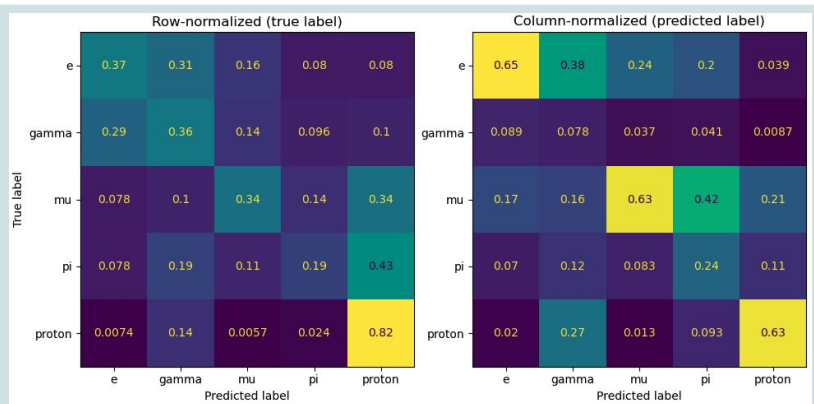


6. PID in the SFGD with a Transformer

Test:

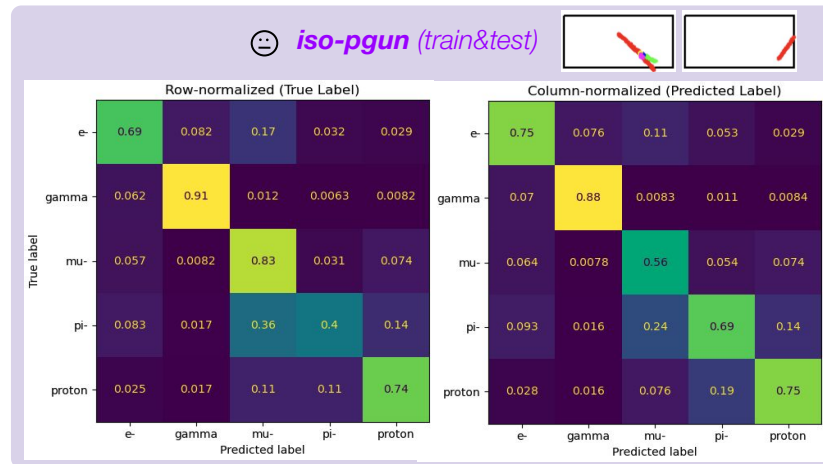
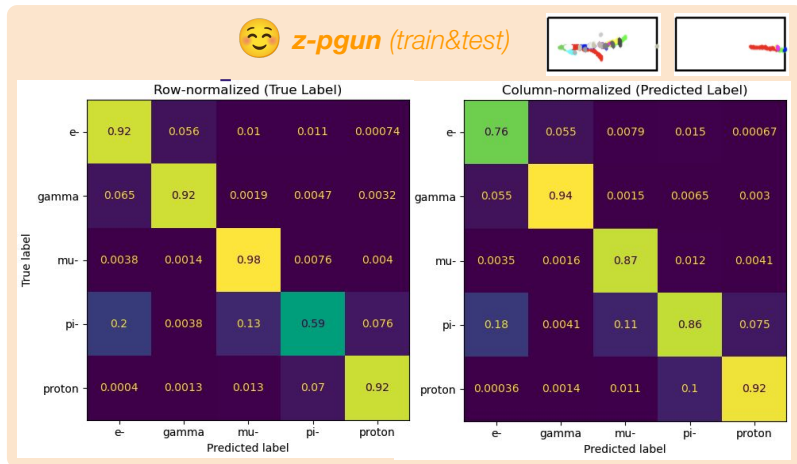


Test on
neutrino
interactions:
(iso-pgun training)

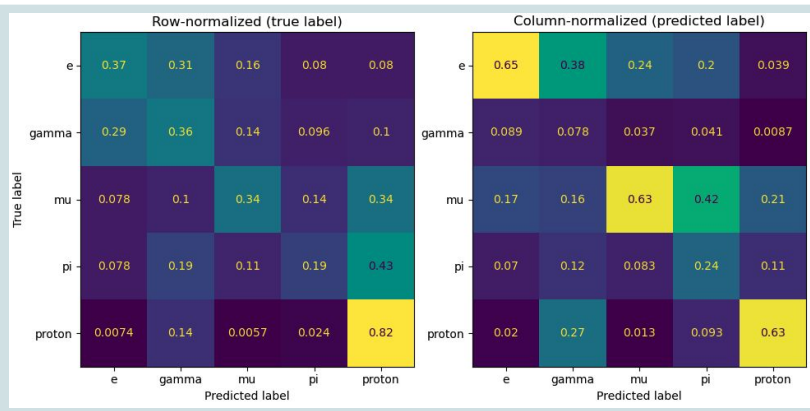


6. PID in the SFGD with a Transformer

Test:

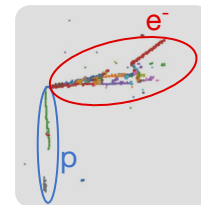


Test on
neutrino
interactions:
(iso-pgun training)



On-going improvements:

- refine pattern recognition before PID (if bad, then model cannot succeed)
- use time of each hits
- add e^+ , π^+ , μ^+ to training
- add extra token to the input sequence:
 - total nhits
 - total charge
 - pulls from ngbh detectors (HAT/TPCs)



Summary

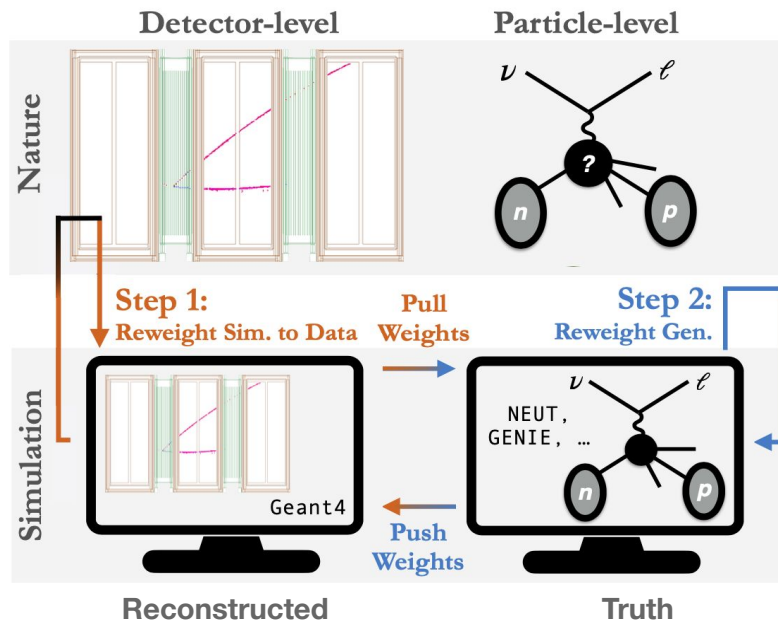
- Increasing number of ML projects at T2K ND
- In all stages of the experiment: more project in the analysis part, then reconstruction then simulation
- Methods are starting to be used in the official software (BDT, Transformer), paving the way for the other ones
- Challenges to come:
 - more friendly integration possibilities for these methods
 - strategy for systematic propagation

Thank you!

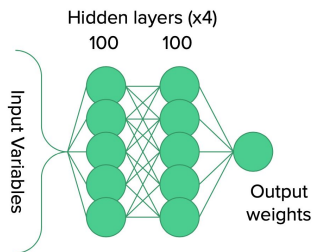
Back-up

Unfolding of ND280 data: Omnifold

Working principle:



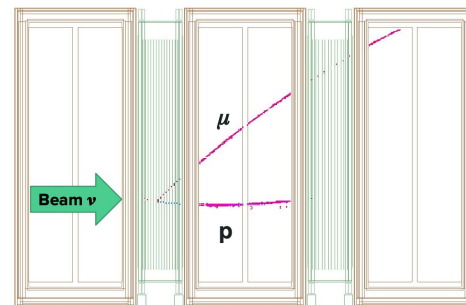
Archi:



1NVIDIA A100 on a NERSC
Perlmutter node: takes < 30 min
to run 15 OmniFold iterations on
one set of data/MC

Data:

- 1.2M simulated ND280 evts $\approx$ 20k measured evts
- with π^+ and leading p kinematics



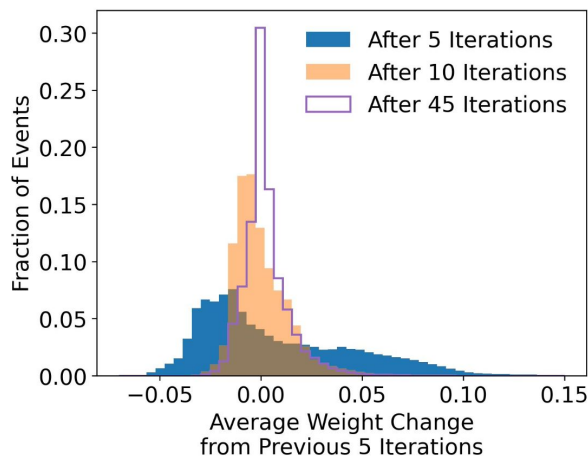
- for test: create fake dataset with a BeRPA-based modification to the true interaction rates

Inputs:

- kinematic observables (p_μ , $\cos \theta_\mu$, p_p , δp_T , $\delta \alpha_T$, $\delta \phi_T$)
- detector sample ID
- interaction topology (CC0 π 0 p , CC0 π 1 p , CC0 π N p , CC1 π , CCothers)

Unfolding of ND280 data: Omnifold

Test:



Comparison with conventional-like unfoldings:

- not straightforward since Omnifold is unbinned
- use Omnifold in a way that it is mathematically equivalent to IBU (Iterative Bayesian Unfolding): inputs limited to bin indices

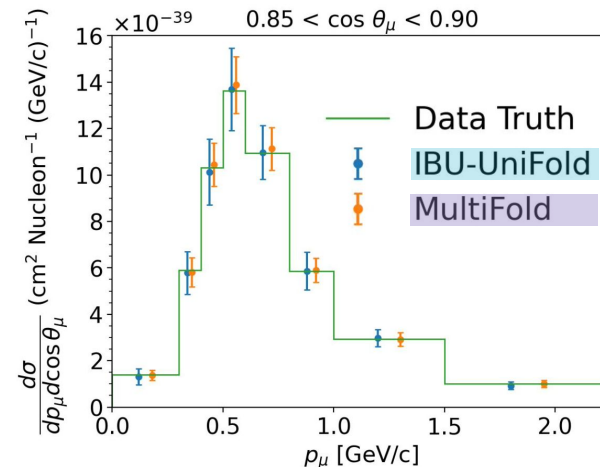
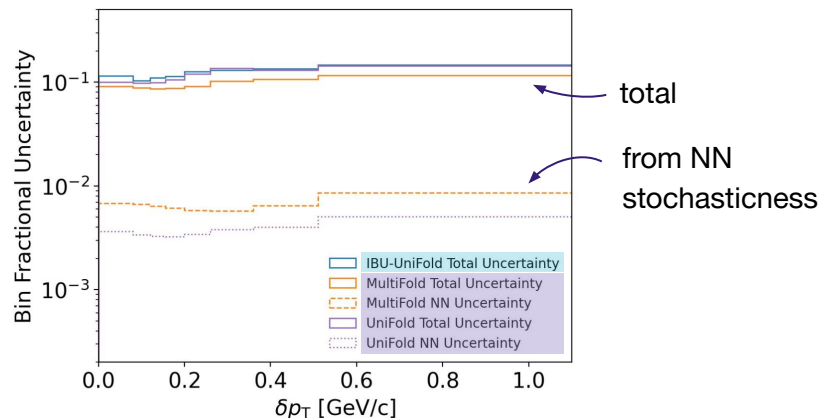
conventional-like method

Method	χ^2			
	$(p_\mu, \cos \theta_\mu)$ DoF=58	δp_T DoF=8	$\delta \alpha_T$ DoF=8	$\delta \phi_T$ DoF=8
Prior	298.2	2.3	5.9	4.9
IBU-UniFold	2.1	0.2	0.4	0.1
Binned UniFold	21.4	1.4	0.9	0.5
UniFold	27.1	1.1	0.6	1.1
MultiFold	3.1	0.3	0.2	0.3
Omnifold	10.0	0.8	1.1	0.4

Omnifold variations (inputs choice)

Method	Triangular Discriminator			
	$(p_\mu, \cos \theta_\mu)$	δp_T	$\delta \alpha_T$	$\delta \phi_T$
Prior	545.6	27.5	31.2	26.7
IBU-UniFold	17.1	1.9	3.4	0.8
Binned UniFold	29.9	2.8	6.0	1.9
UniFold	17.3	5.7	5.8	1.7
MultiFold	2.7	0.7	0.6	0.6
Omnifold	9.4	1.7	3.0	2.1

Uncertainties:



Modeling posterior systematic: Normalizing Flows

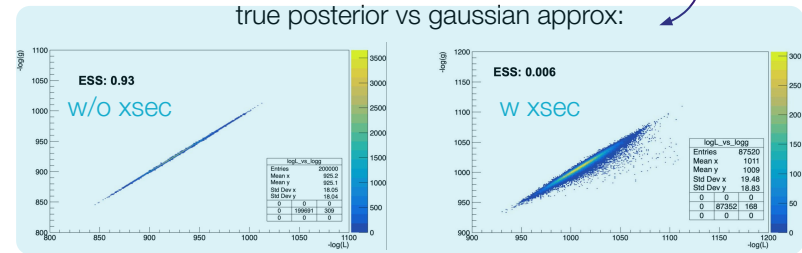
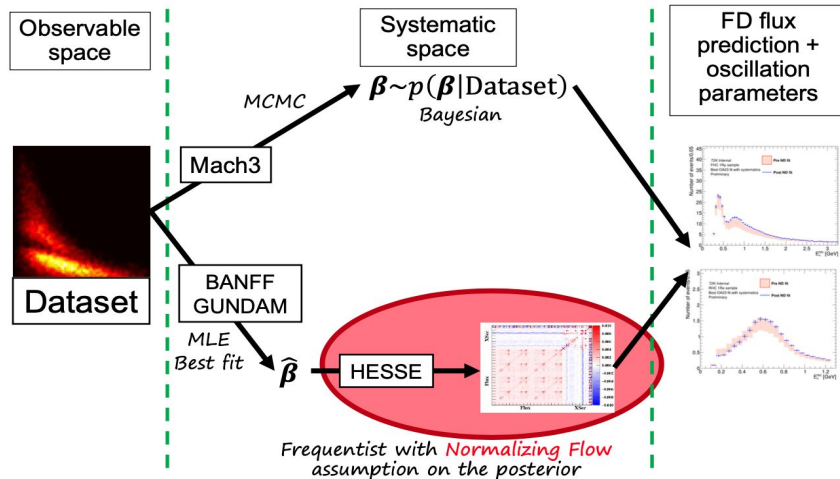
ND fit is a constrain on systematic uncertainties using ND280 observations

Context: ND280 likelihood of systematics depends on >700 variables (from **flux**, **detector** and **xsec** uncertainties)

Goal: Learn the posterior probability distribution of neutrino flux binned in neutrino energy

Conventional methods:

- Semi-frequentist (*GUNDAM*): gaussian assumption on the posterior, get best-fit params from MLE + analytical - miss xsec non-gaussianities
- Bayesian (*Mach3*): sample from the posterior using MCMC + capture non-gaussianities - pt cloud estimation (not analytical)

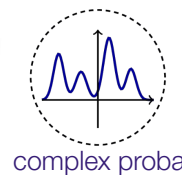
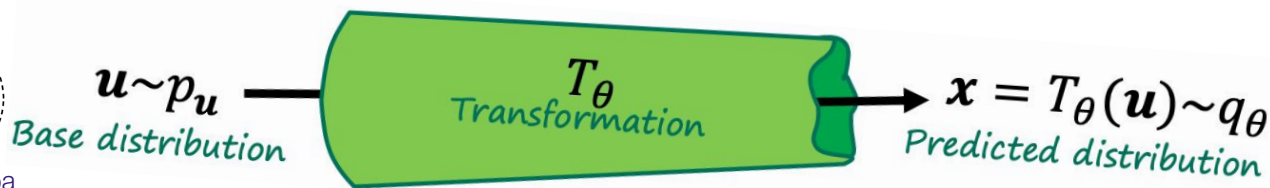
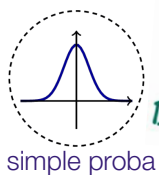


Where ML comes in:

replace *GUNDAM* gaussian approximation by something more complex to capture **non-gaussianities** in **xsec** params

Modeling posterior systematic: Normalizing Flows

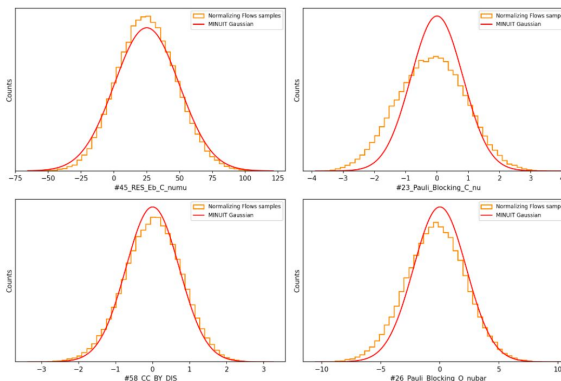
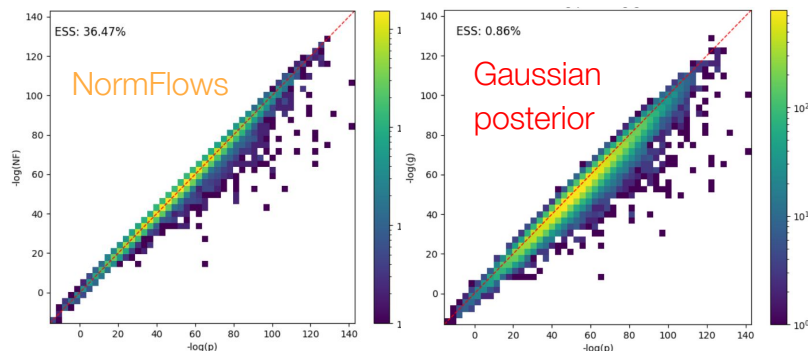
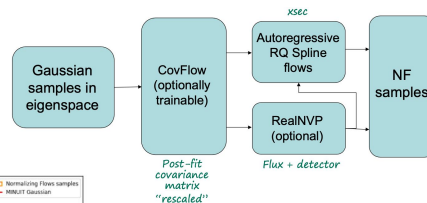
NormFlows:



↳ straightforward to: sample $x = T_\theta(u) \sim q_\theta$ with $u \sim p_u$ & evaluate the proba $q_\theta(x) = p_u(u) |\det(J_{T_\theta}(u))|^{-1}$

Test: on full sets of systematics (OA 2022 config) to learn the 59 **xsec** probas conditioned on the 652 **flux** + **detector** systematics

Archi: 5 RQ-NSF splines, MANN-parametrized flows



Fast method:
10M sample / day
vs ?

Modeling posterior systematic: Normalizing Flows

Questions:

- running time comparison with non-ML method?
- is loss func the KDL or derived from it?

CCQE cross-section sampling: Normalizing Flows

Goal: efficient MC sampling for CCQE exclusive cross-section of neutrino-nucleus (^{12}C)

Conventional method:

use sophisticated **nuclear models**

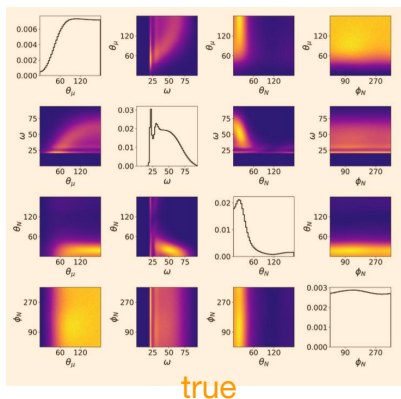
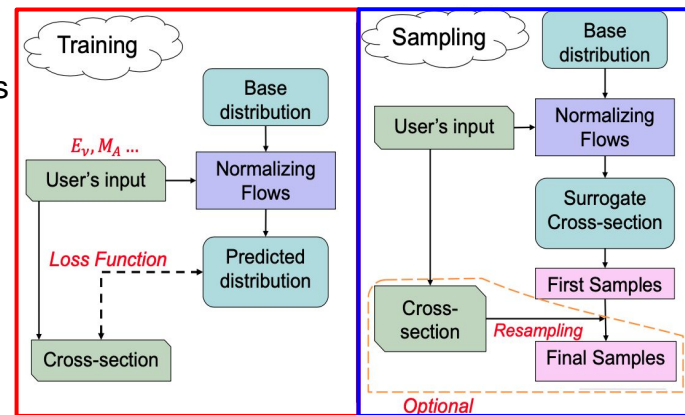
$$\frac{d^6\sigma}{dE_\nu d\omega d\Omega_\mu d\Omega_N} \propto L^{\mu\nu} W_{\mu\nu} \rightarrow \text{long computations}$$

ML method: same Normalizing Flow architecture as project 5

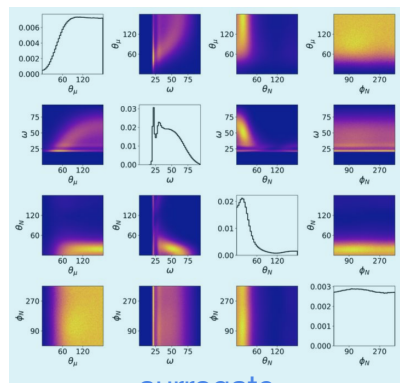
(slightly different loss function derived from KLD)

Train: to model 1p1h i.e. $p(\theta_\mu, \omega, \theta_N, \phi_N | E_\nu, \alpha)$ on 2 shells for many E_ν

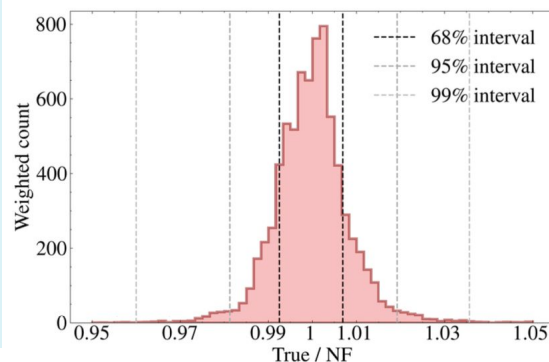
Test: sample e.g for (600 MeV, 1s shell)



true



surrogate



Fast method:
1M sample /25min/GPU
vs 1 day/CPU

Next:

2p2h will be more complicated
because higher dimension