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
- Presentation of 6 selected projects:

    1. Momentum reconstruction and track PID with a BDT
    2. EM shower PID with PointNet
    3. PID in the SFGD with a Transformer
    4. Unfolding of ND280 data with Omnifold
    5. Modeling posterior systematic with Normalizing Flows
    6. CCQE cross-section sampling with Normalizing Flows

–in order of complexity (pov)

The T2K experiment & its Near Detector

The T2K experiment & its Near Detector

particle accelerator to create neutrino beam

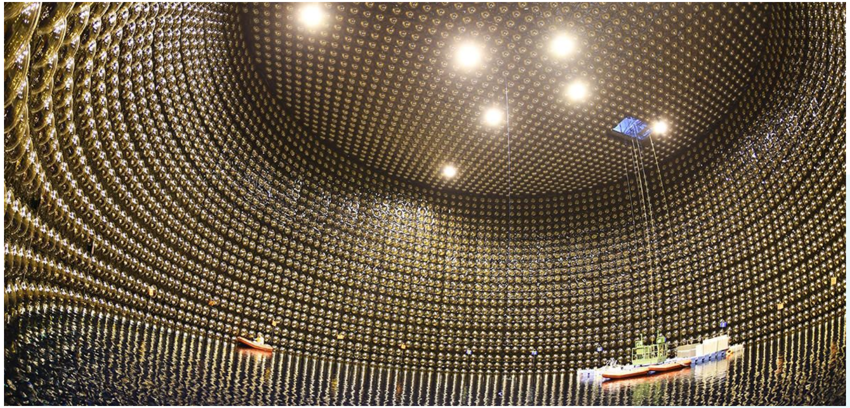
The T2K experiment & its Near Detector

near detector: ND280

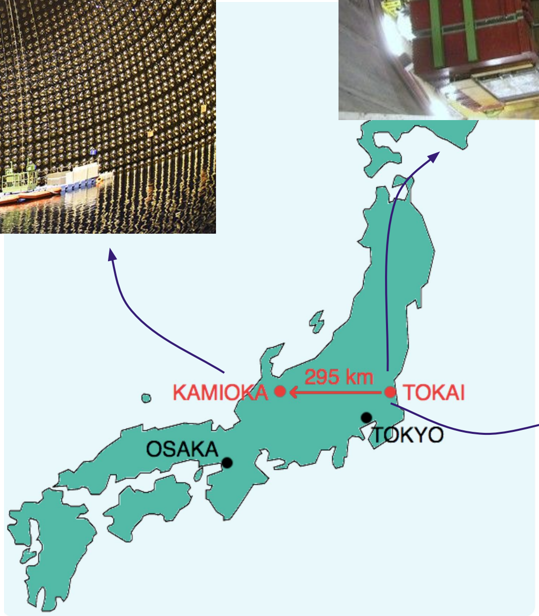
particle accelerator to create neutrino beam

The T2K experiment & its Near Detector

near detector: ND280

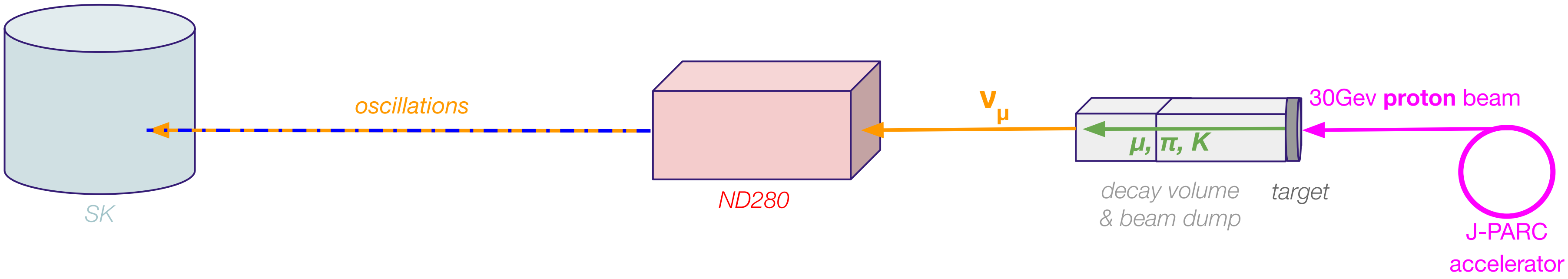


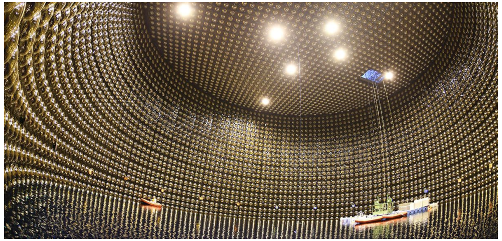
far detector: Super-Kamiokande



particle accelerator to create neutrino beam

The T2K experiment & its Near Detector





Super-Kamiokande

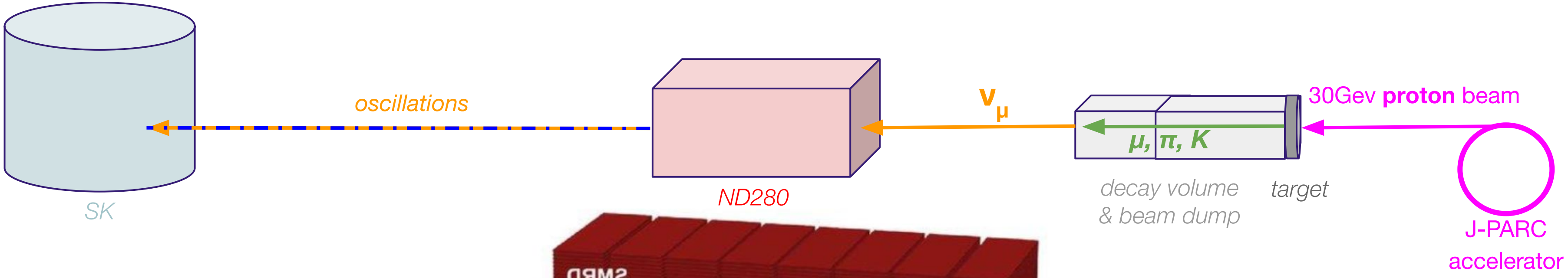
295km

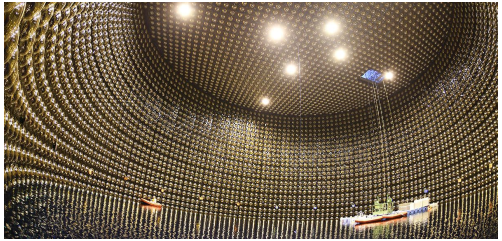
ND280

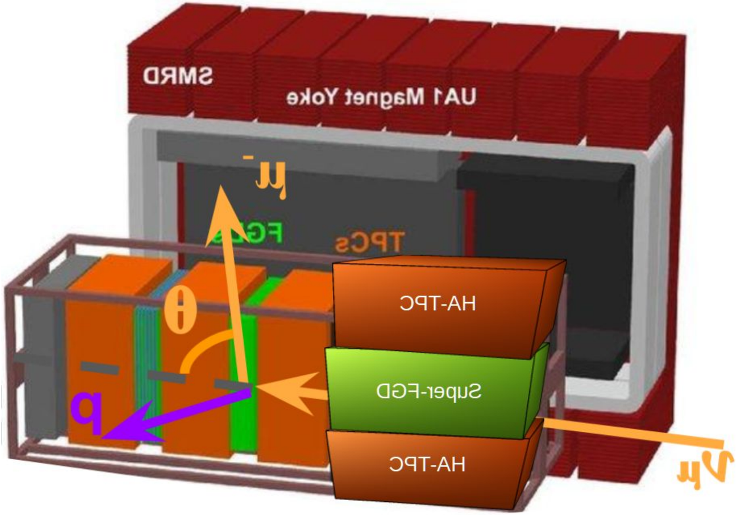
280m

neutrino beam

The T2K experiment & its Near Detector

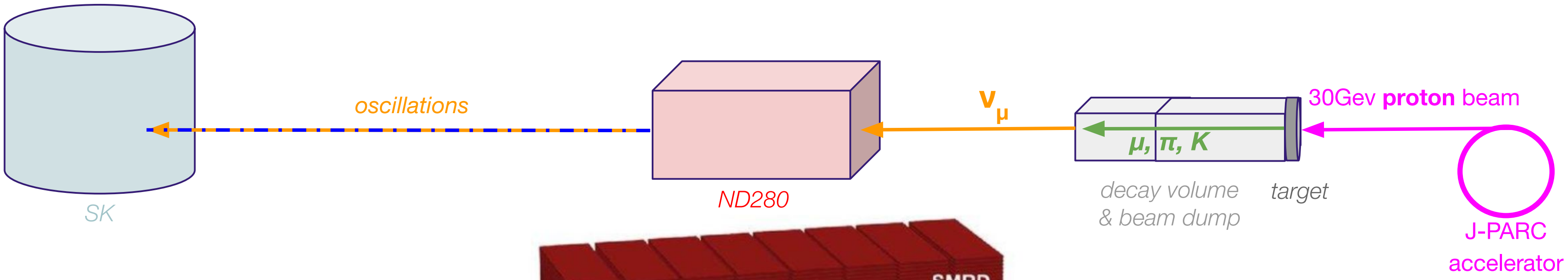


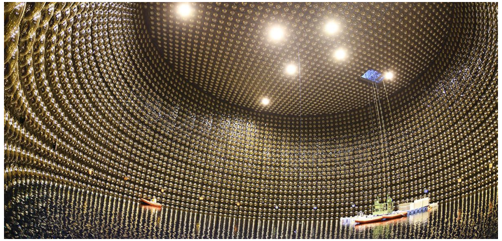

Super-Kamiokande

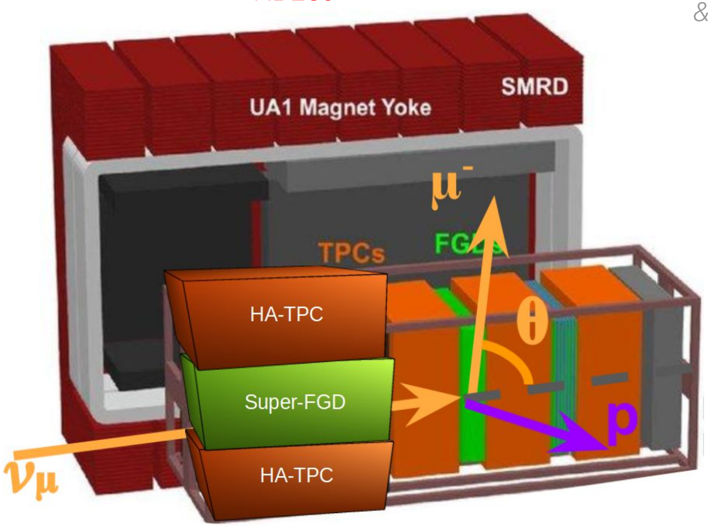



neutrino beam

The T2K experiment & its Near Detector

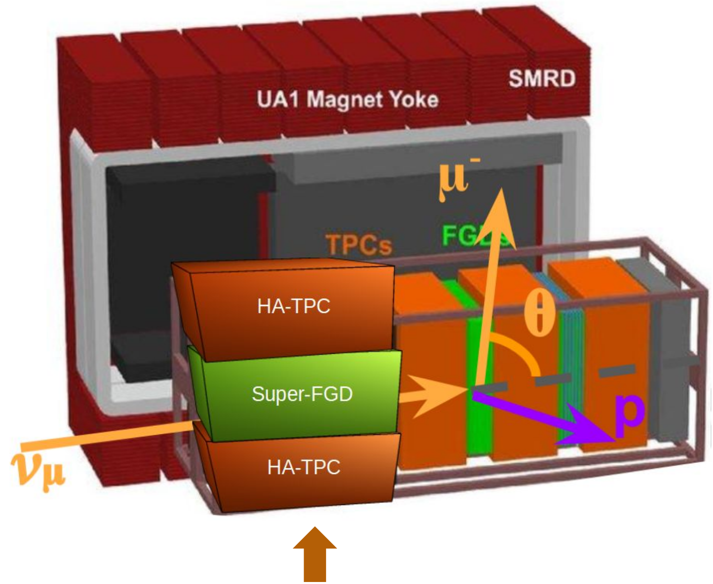


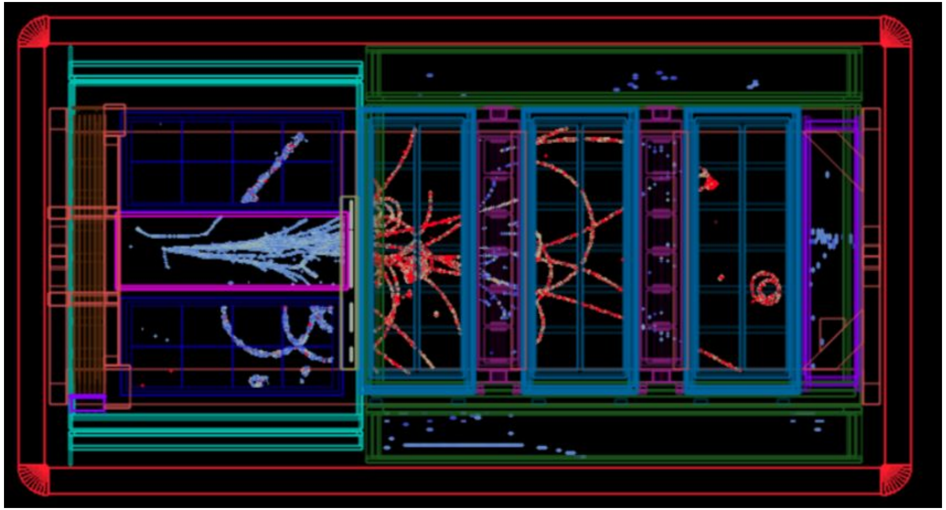

Super-Kamiokande




neutrino beam

The T2K experiment & its Near Detector

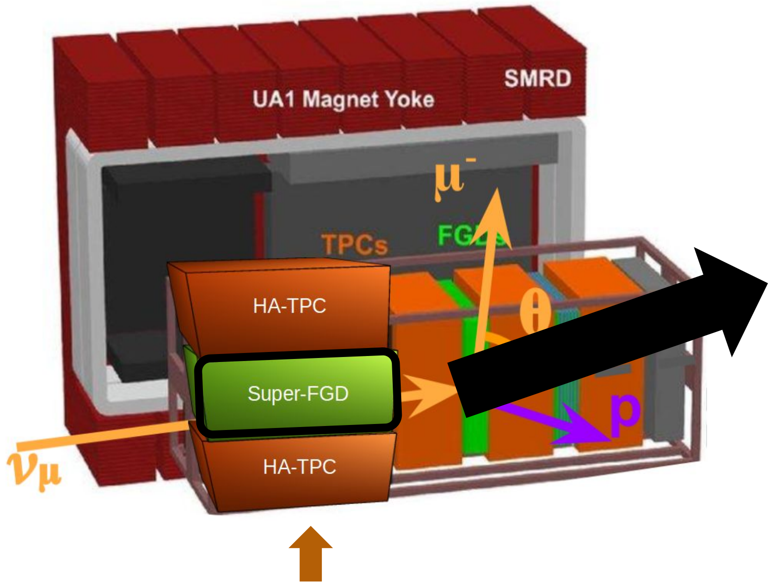




ND280 Upgrade installed last year!

& data taking since end of 2024

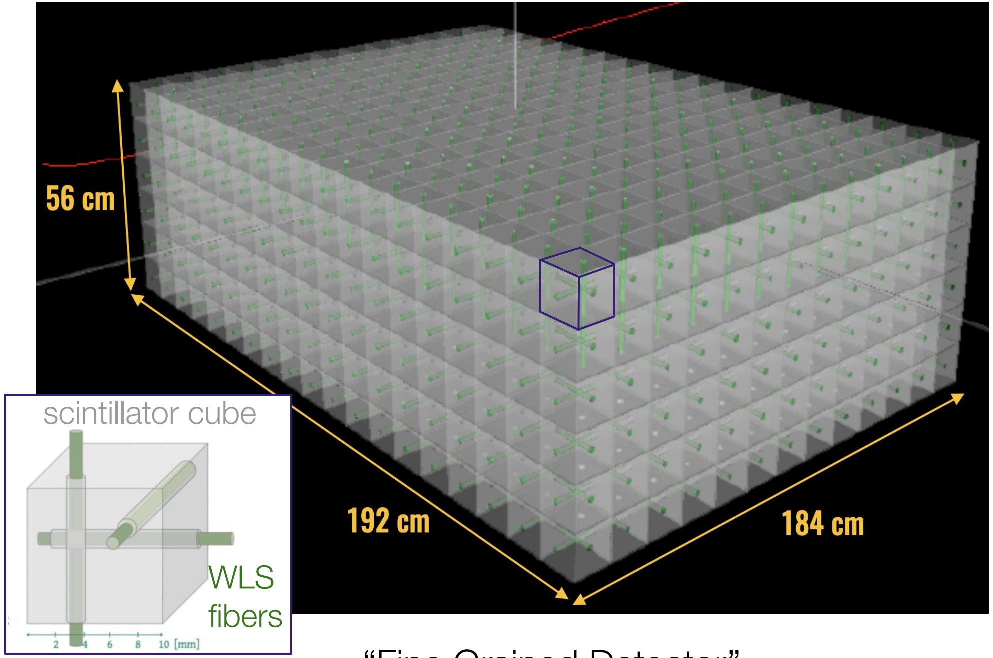
The T2K experiment & its Near Detector



ND280 Upgrade installed last year!

& data taking since end of 2024

The Super-FGD: a Scintillator Detector



~2M cubes

"Fine Grained Detector"

The ND280 AI/ML working group

- Recent working group dedicated to AI/ML methods (since Nov. 2024)
- Convener is Saùl Alonso Monsalve (ETH Zurich)
- ~ 10 active analysers
- In different part of the experiment, various task focused around the Near Detector:

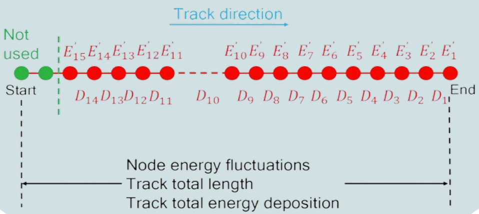
★ ***reconstruction***: vertex activity, track fitting, momentum reco ⟶ 3 projects

★ ***analysis***: PIDs, unfolding, modelling posterior systematics ⟶ 7 projects

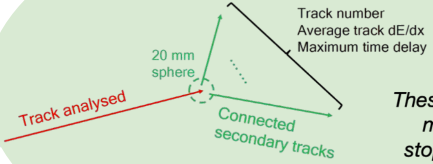
★ ***simulation***: cross-section sampling ⟶ 1 project

1. Momentum reconstruction & track PID with BDT

Inputs: Primary track params (5) & secondary tracks params (3)



+ node energies
+ node nghb distances



These parameters are mainly used for stopping track PID.

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

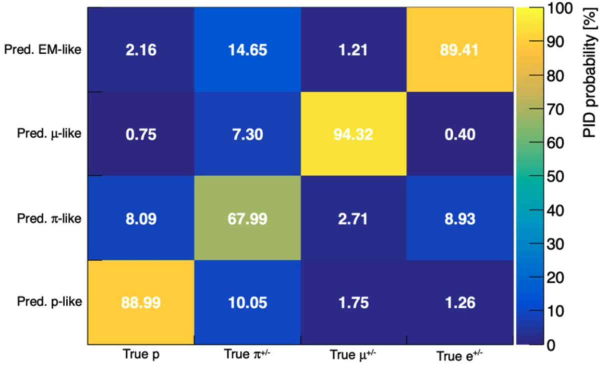
Hyperparameter	Value
Decision tree number M	2000
Division point number K at each node	20
Decision tree maximum depth D_{max}	3
Shrinkage ν	0.05
Stochastic boosting fraction f	0.5

same for regression & classification

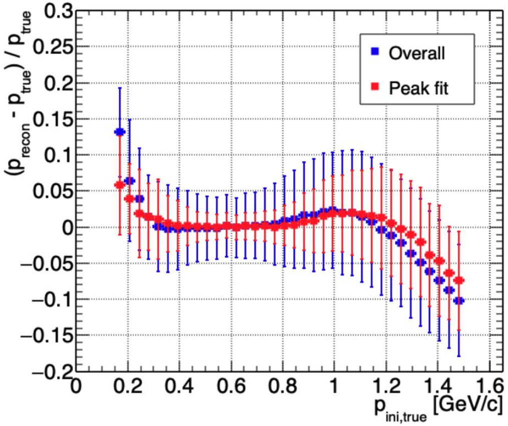
Training:
- on particle-gun MC data (i.e. 1 particle /event): p, $\pi^{\pm}$, $\mu^{\pm}$, $e^{\pm}$
- 4 independent PID classifier & 3 independent momentum regressors

Test:

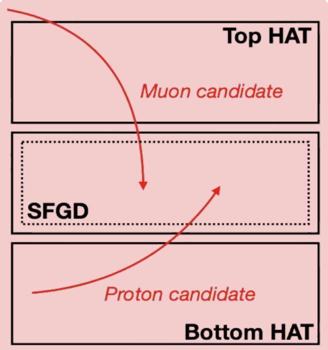
classification results



momentum resolution:



1. Momentum reconstruction & track PID with BDT

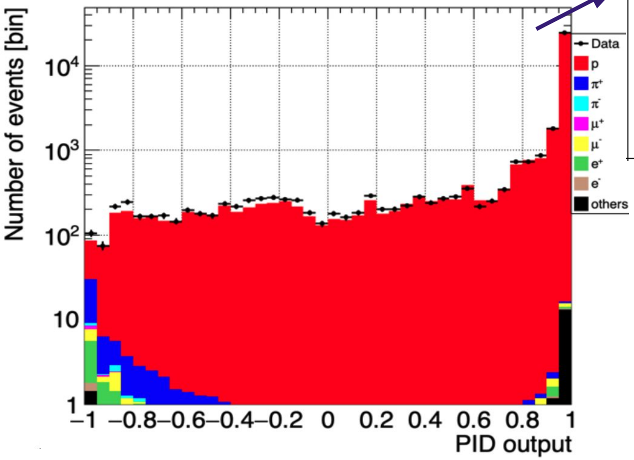


stopping control sample

BDT Systematics:

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

PID efficiency:

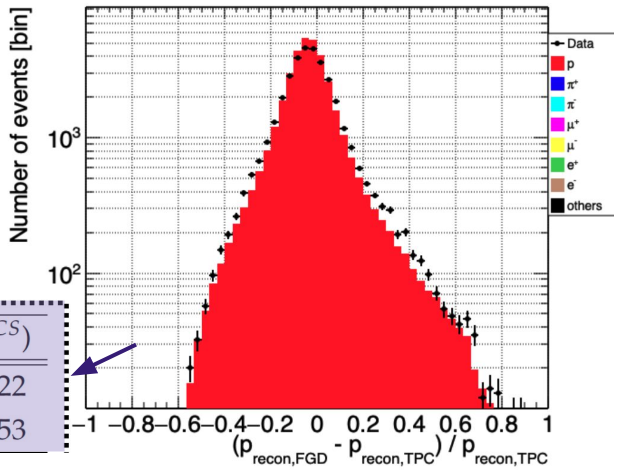


Control Sample	Data / MC
Proton	0.99 ± 0.00
Pion (π^+)	$\mathbf{2.17 \pm 0.06}$
Muon (μ^-)	1.32 ± 0.05
Electron / Positron	1.41 ± 0.15

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:



1. Momentum reconstruction & track PID with BDT

Questions

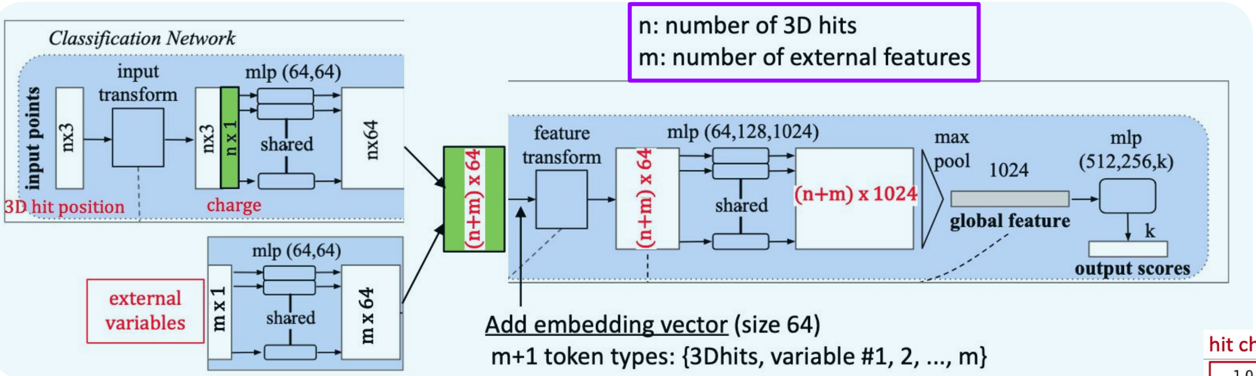
- is the momentum regression used in SFG Recon?
- are the systematics currently propagated/where/what is the plan otherwise?
- in previous table, no pi-, no mu+ why
- controle sample: for each particle type? ID only with hats/tpcs? from outside events?
- 4 independent PID classifier & 3 independent momentum regressors

  but CM was classifier with 4 particles: just for the result plots?

  why 3 momentum regressors?

2. EM shower PID in SFGD: PointNet

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



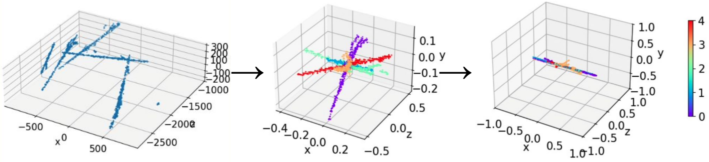
Inputs:

- pgun of e⁻, μ⁻ for now
- use the pattern reco algo on the pgun (used for neutrino interaction)
- select shower-like, contained patterns

↳ 8000 patterns

Preprocessing:

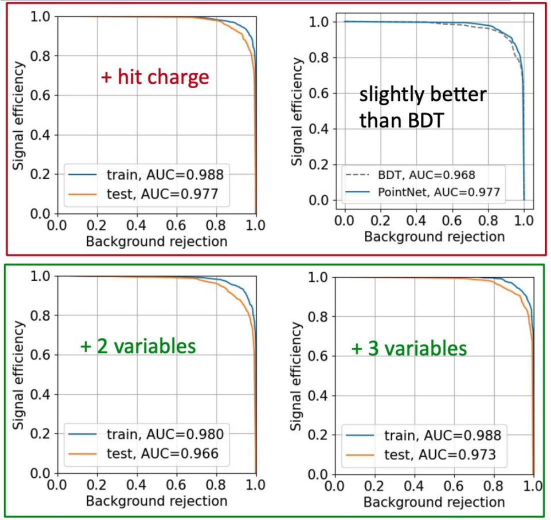
- center and align showed better perf



- PointNet take 512 pts:
  - If nHits > 512: Randomly sample
  - If nHits < 512: Repeatedly sample

Test:

hit charge & external variables improve the performance



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

2. EM shower PID in SFGD: PointNet

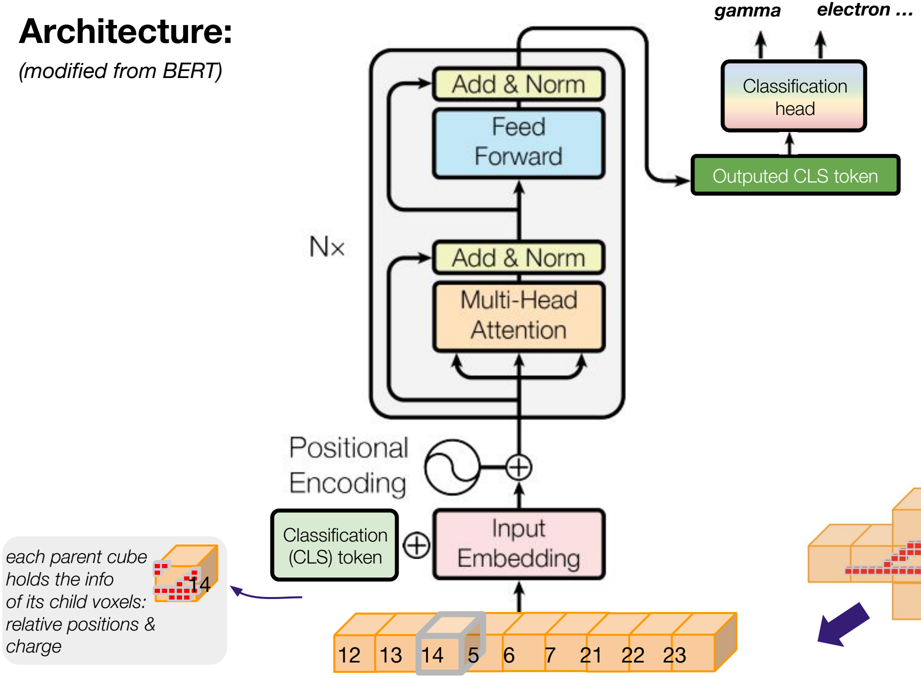
Question:

- 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?

3. PID in the SFGD: Transformer

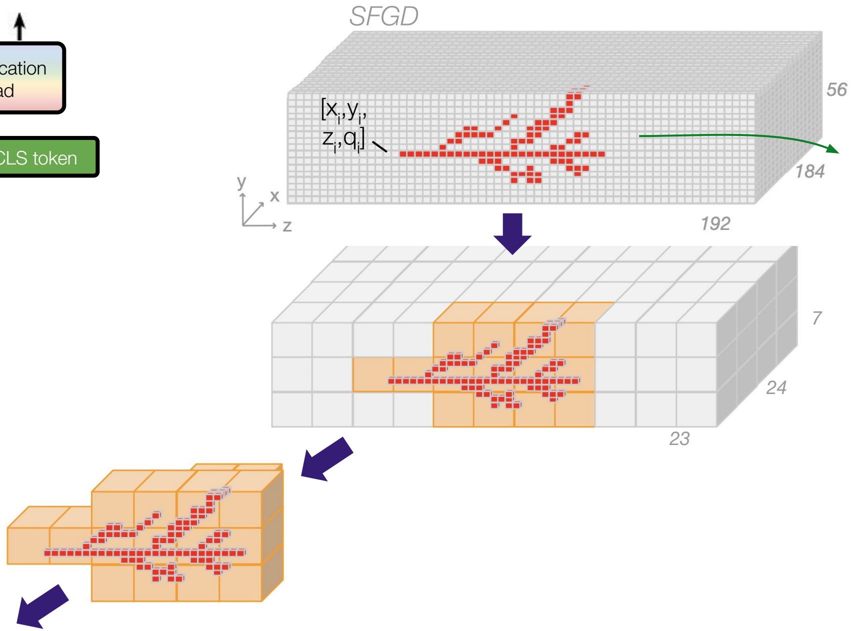
Architecture:

(modified from BERT)



positional encoding of the parent cubes in the parent grid

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'*

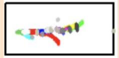
- architecture converted in C++ (Libtorch)
- saved weights from training

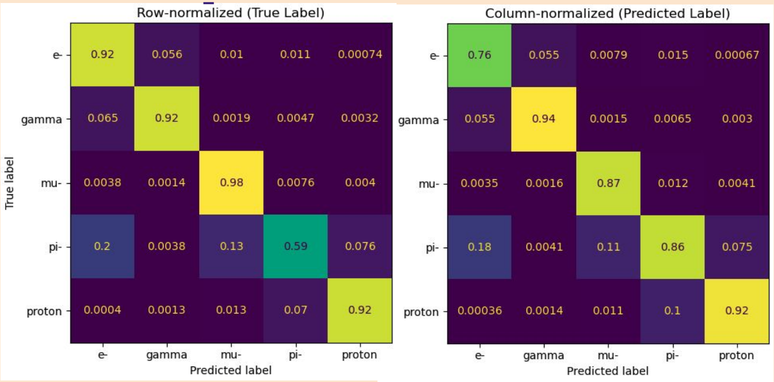
➡ 1rst complex ML model implemented in analysis soft!

3. PID in the SFGD: Transformer

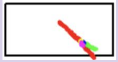
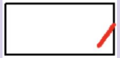
Test:

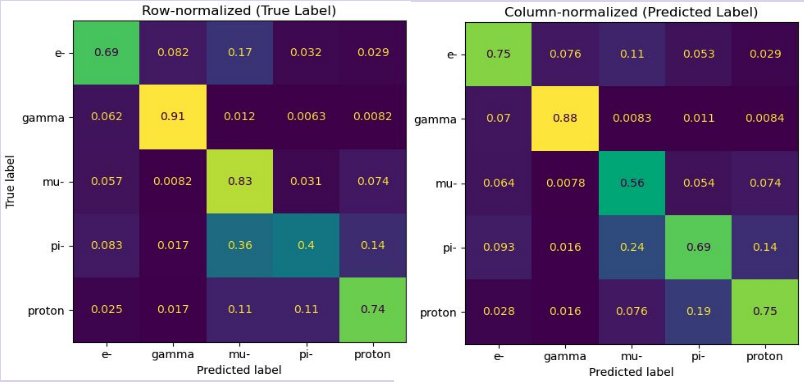
☺️ ***z-pgun*** *(train&test)*



😐 ***iso-pgun*** *(train&test)*

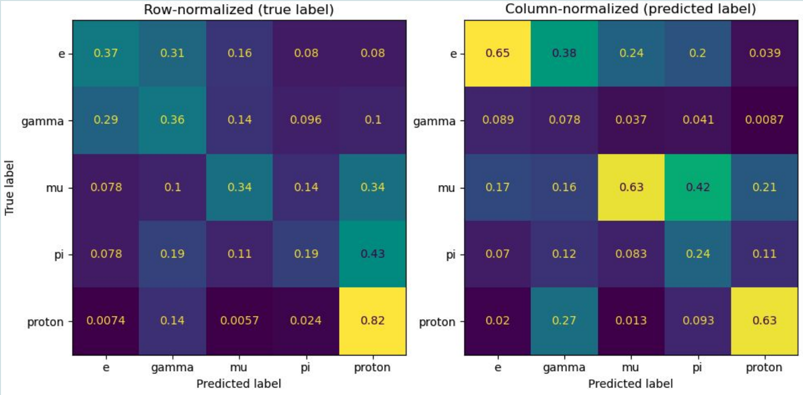





Test on neutrino interactions:

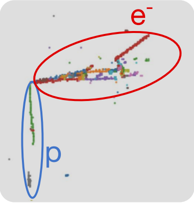
(iso-pgun training)

☹️



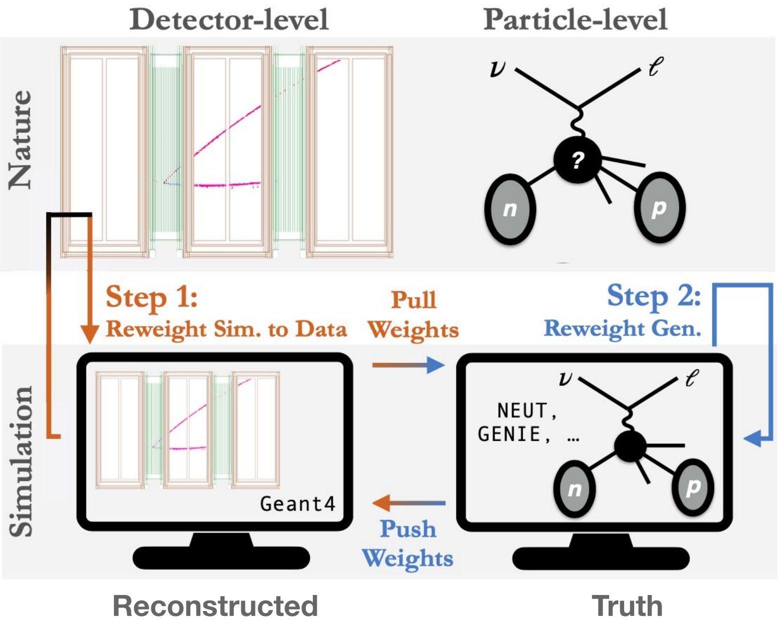
On-going improvements:

- 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 nghb detectors (TPCs)



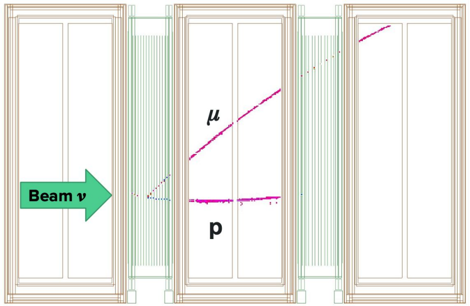
4. Unfolding of ND280 data: Omnifold

Working principle:



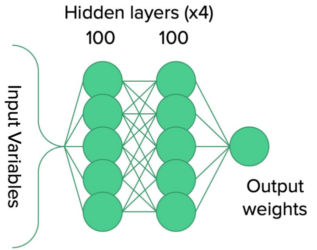
Data:

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



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

Archi:



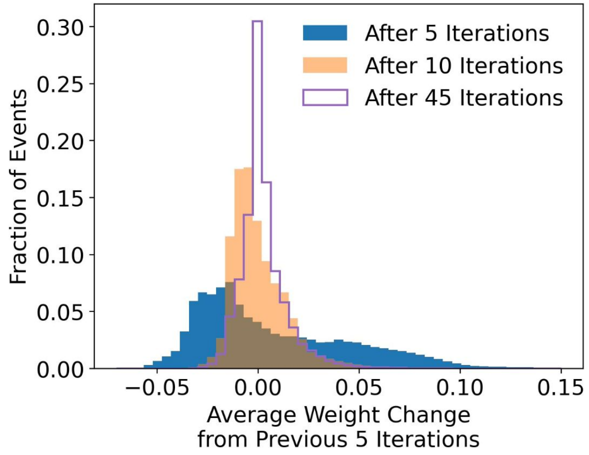
1NVIDIA A100 on a NERSC Perlmutter node: takes < 30 min to run 15 OmniFold iterations on one set of data/MC

Inputs:

- kinematic observables (p_μ, $\cos\theta_\mu$, p_p, δp_T, $\delta\alpha_T$, $\delta\phi_T$)
- detector sample ID
- interaction topology (CC0π0p, CC0π1p, CC0πNp, CC1π, CCother)

4. Unfolding of ND280 data: Omnifold

Test:



Comparision 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

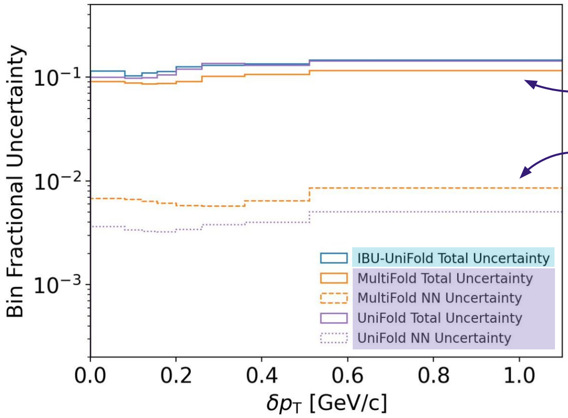
conventional-like method: IBU-UniFold, Binned UniFold

Omnifold variations (inputs choice): UniFold, MultiFold, OmniFold

Method	χ^2 $(p_\mu, \cos\theta_\mu)$ DoF=58	χ^2 δp_T DoF=8	χ^2 $\delta\alpha_T$ DoF=8	χ^2 $\delta\phi_T$ 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

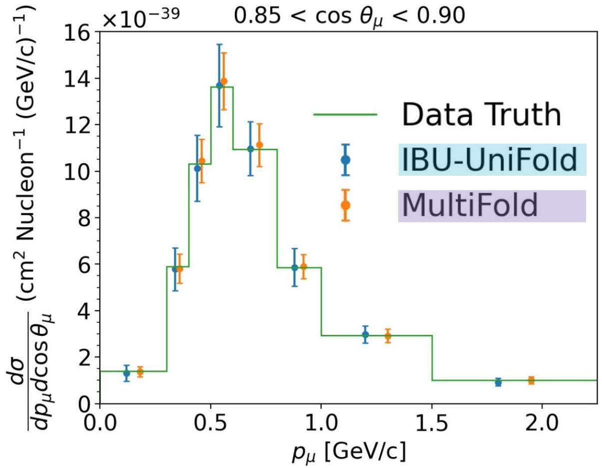
Method	Triangular Discriminator $(p_\mu, \cos\theta_\mu)$	δp_T	$\delta\alpha_T$	$\delta\phi_T$
Prior	545.6	27.5	31.2	26.7
IBU-UniFold	17.1	1.9	3.4	0.8
Binned UniFold	29.9	2.8	6.0	1.9
UniFold	17.3	5.7	5.8	1.7
MultiFold	2.7	0.7	0.6	0.6
OmniFold	9.4	1.7	3.0	2.1

Uncertainties:



total

from NN stochasticness



4. Unfolding of ND280 data: Omnifold

Questions

5. Modeling posterior systematic: Normalizing Flows

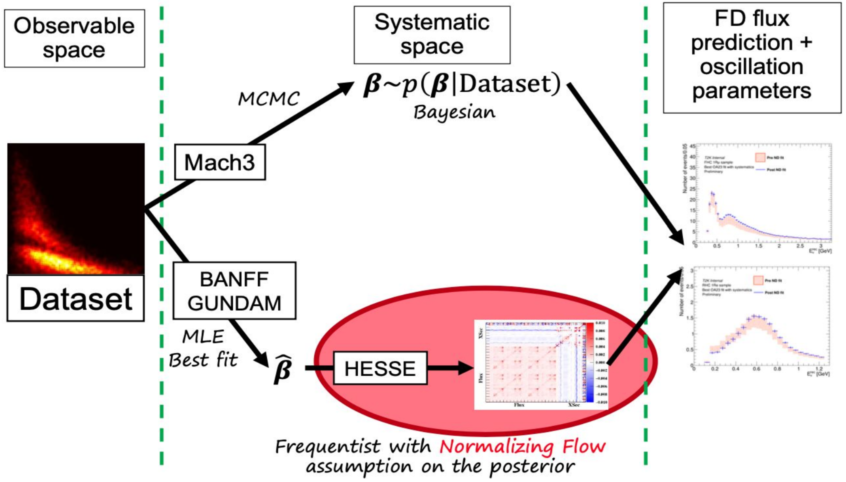
ND fit is a constrain on systematic uncertainties using ND280 observations

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

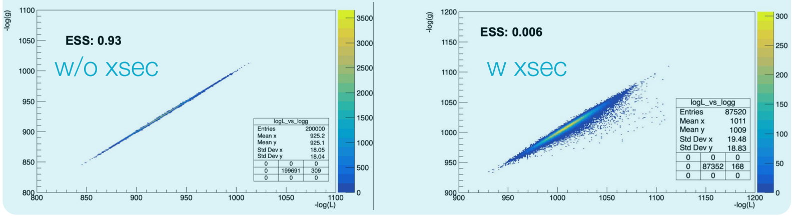
Goal: Learn the posterior probability distribution of neutrino flux binned in neutrino energy

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-gaussianties
  - pt cloud estimation (not analytical)



true posterior vs gaussian approx:



Where ML comes in:

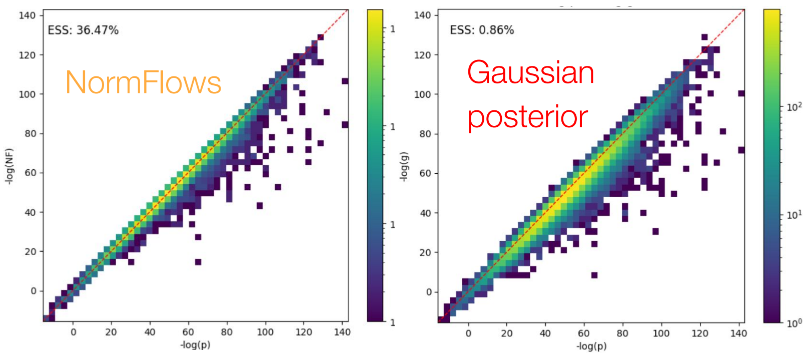
replace *GUNDAM* gaussian approximation
by something more complex
to capture non-gaussianities in xsec params

5. Modeling posterior systematic: Normalizing Flows

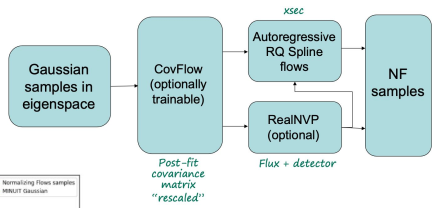
NormFlows:

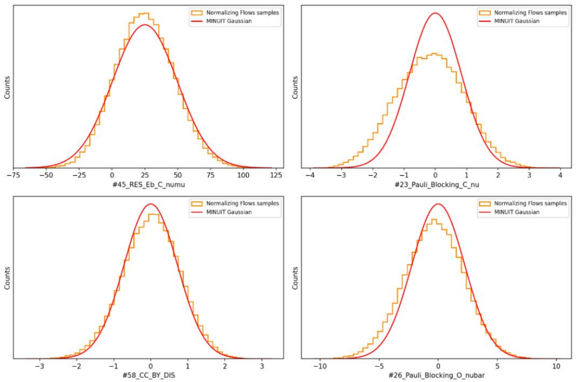
↳ straighforward to: sample $\boldsymbol{x} = T_\theta(\boldsymbol{u}) \sim q_\theta$ with $\boldsymbol{u} \sim p_{\boldsymbol{u}}$ & evaluate the proba $q_\theta(\boldsymbol{x}) = p_{\boldsymbol{u}}(\boldsymbol{u}) \,|\det\left(\mathrm{J}_{T_\theta}(\boldsymbol{u})\right)|^{-1}$

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 ?

5. Modeling posterior systematic: Normalizing Flows

Questions:

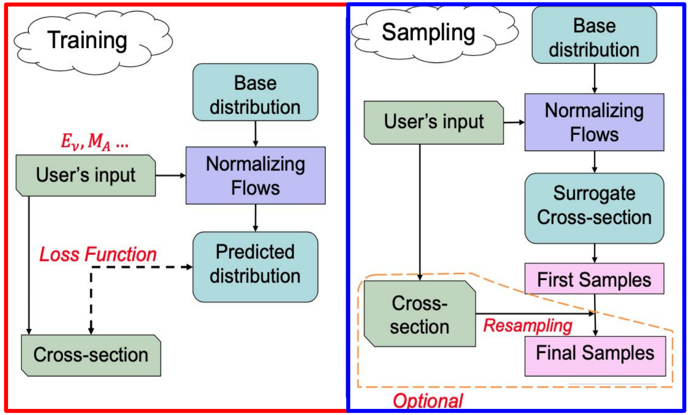
- running time comparison with non-ML method?
- is loss func the KDL or derived from it?

ref. https://arxiv.org/pdf/2502.14452

6. CCQE cross-section sampling: Normalizing Flows

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

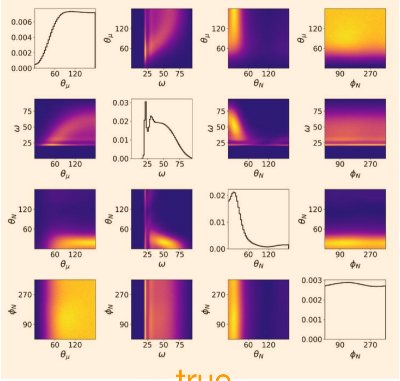
Conventional method:
use sophisticated nuclear models $\frac{d^6\sigma}{dE_\nu d\omega d\Omega_\mu d\Omega_N} \propto L^{\mu\nu} W_{\mu\nu} \rightarrow$ long computations



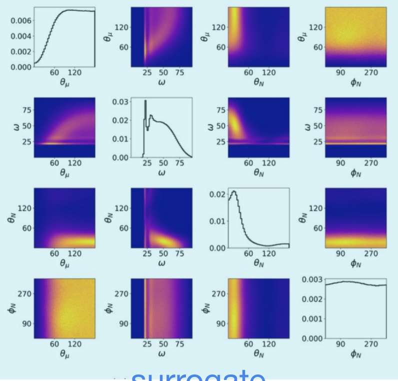
ML method: same Normalizing Flow architecture as project 5
(slightly different loss function derived from KLD)

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

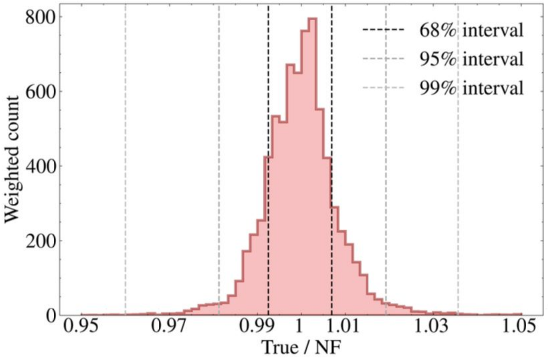
Test: sample e.g for (600 MeV, 1s shell)



true



surrogate



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

Next:
2p2h will be more complicated because higher dimension

6. CCQE cross-section sampling: Normalizing Flows

Questions:

-

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:
  - integration within reconstruction softwares is more tricky?
  - more friendly integration possibilities for these methods?
  - strategy for systematic propagation