ML tools for a 3D highly segmented plastic scintillator

Work conducted under the supervision of
Saúl Alonso-Monsalve & Davide Sgalaberna

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

Summary

I. Introduction
Overview of such experiment

II. Hit tagging
Task of tagging the hits in SFGD

III. Track fitting
Task of fitting the trajectory points

IV. Vertex activity
Task of decomposing the vertex activity

V. Conclusion
Key takeaways of the projects

I.
Introduction

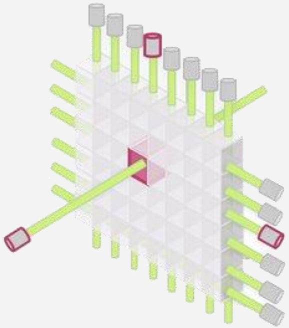
Overview of such experiment

I.A Case study: SFGD

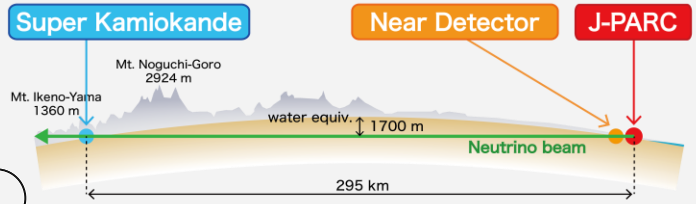
Composition

Plastic scintillator cubes with 3 axis optical fibres

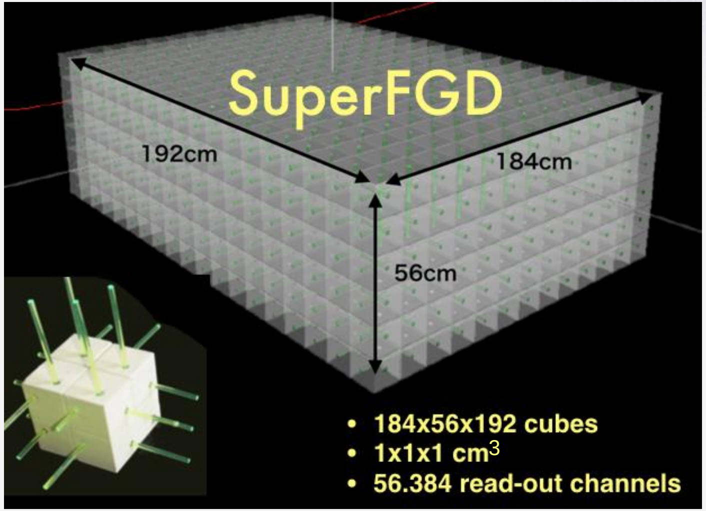
T2K Near Detector upgrade : reduce systematic uncertainties in search for CP violation



SFGD readout principle



T2K devices location

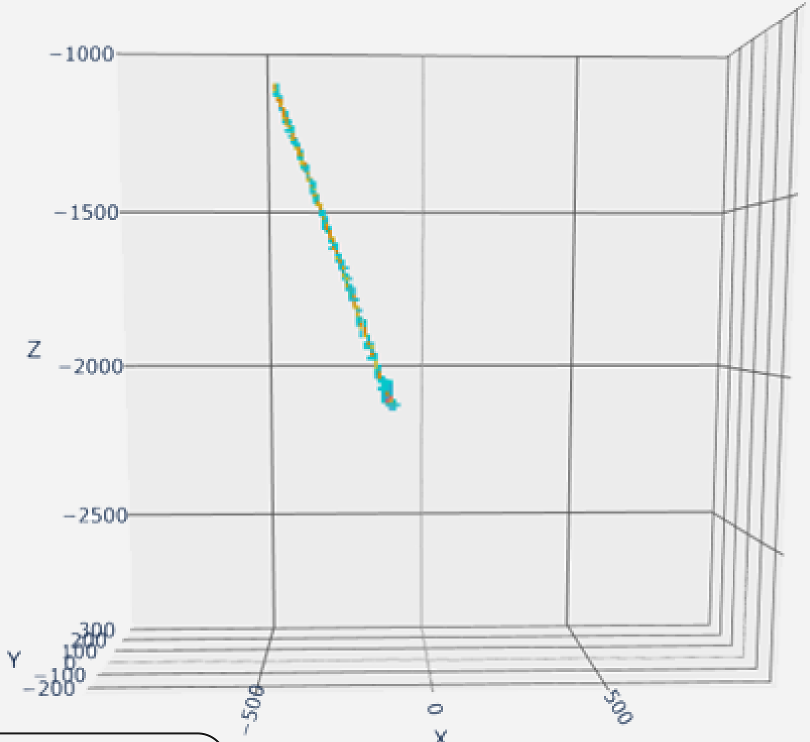


SFGD overview

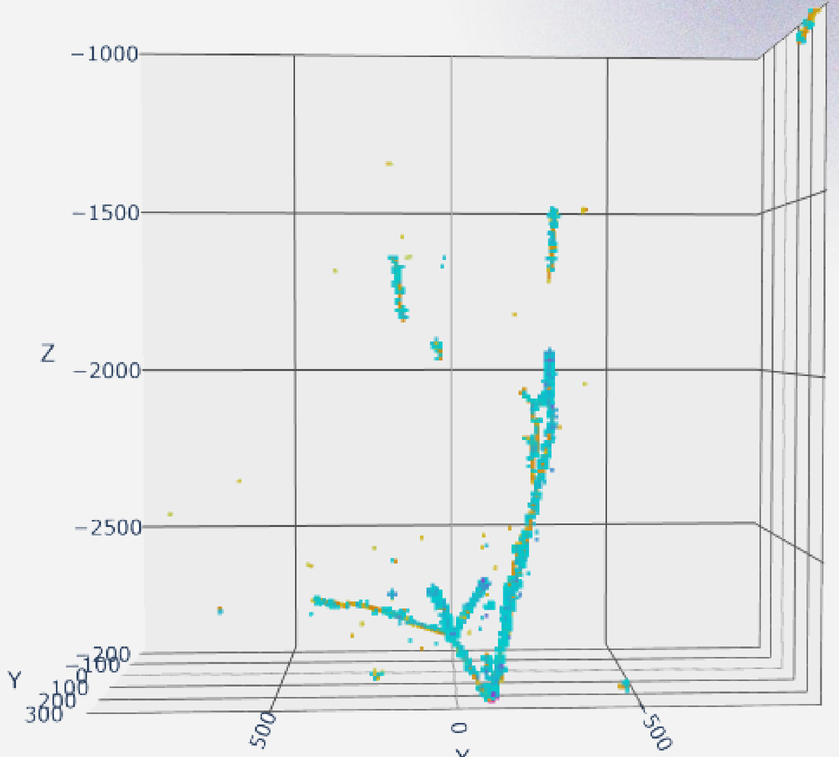
See C.Giganti's talk at Neutrino 2024

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

I.B Simulated data



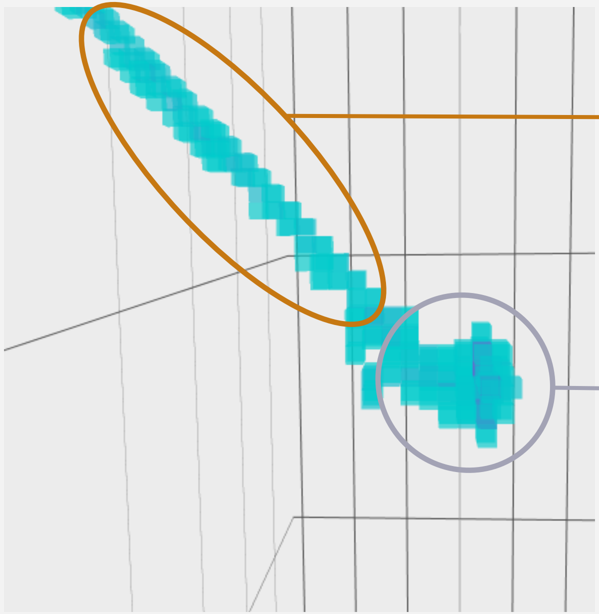
Simulated CCQE event display



Simulated CCDIS event display

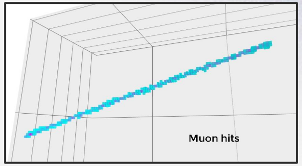
Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

I.C Strategy for our ML tools

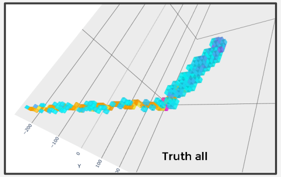


Simulated “particle bomb” event

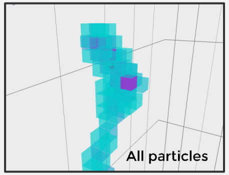
Track fitting



Hit tagging



Vertex activity



II.

Hit tagging

Identify the vertex hits with multiple particles, separate noise hits

II.A Definition of the tags

Vertex activity hit

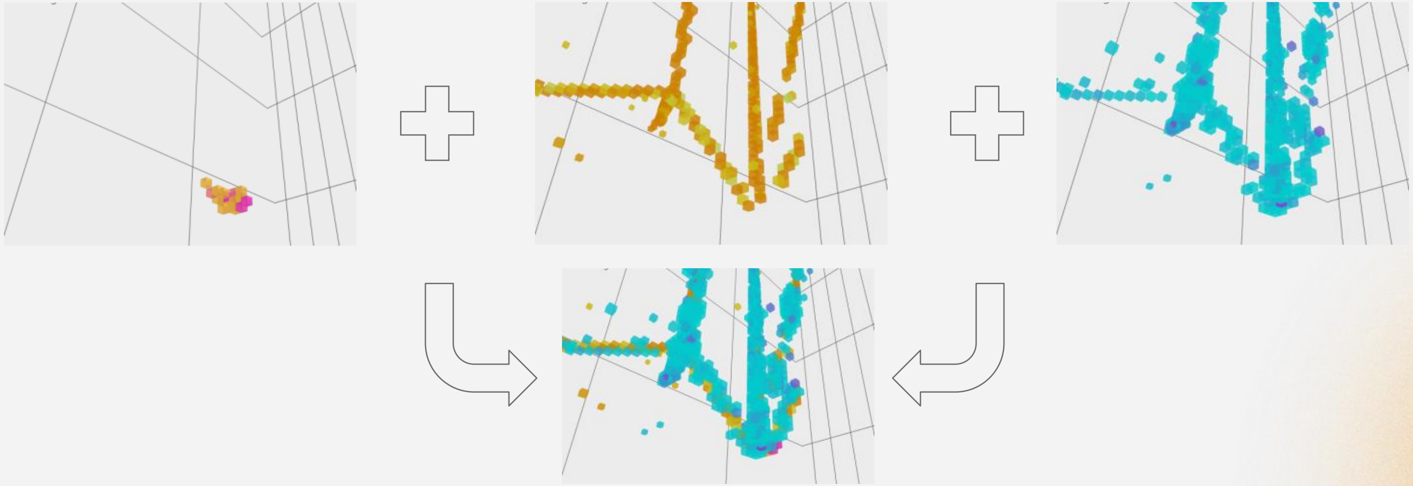
Energy contribution to the hit from two different primary particles + adjacent hits

Track hit

A non vertex activity hit with a particle crossing it

Noise hit

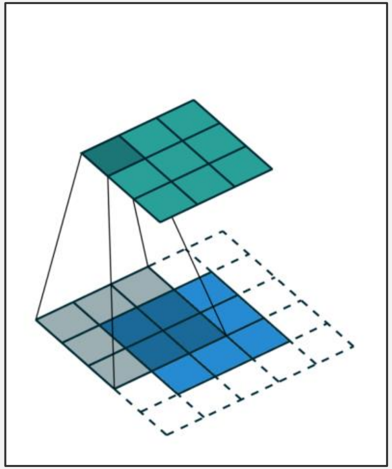
A hit with no particle crossing it



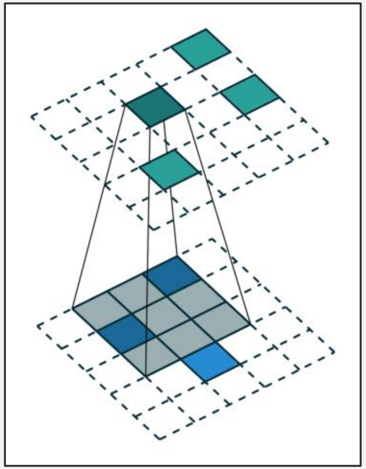
Simulated CCDIS event display viewed from below

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

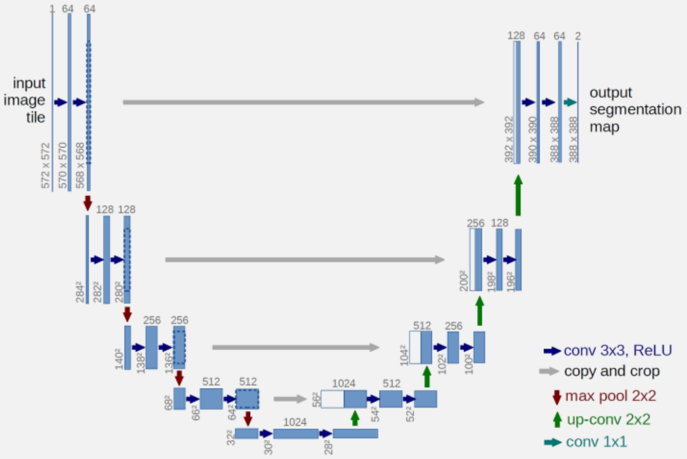
II.B Architecture chosen: Sparse CNN U-net



Convolution on a dense Tensor



Convolution on a sparse Tensor



Example of U-net

Very sparse data: *Lots of **memory** and **speed** gains with sparse convolutions!*

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

II.C Results of hit tagging

Definition of the relevant metrics

Precision

How precise is the model in its predictions?

Out of the predicted labels X, how many are true label X

$$Precision_X := \frac{True\ Positive_X}{True\ Positive_X + False\ Positive_X}$$

Recall

Does the model retrieves all true labels?

Out of true labels X, how many are also predicted as label X

$$Recall_X := \frac{True\ Positive_X}{True\ Positive_X + False\ Negative_X}$$

F1-score

Combination of precision and recall

Harmonic mean of precision and recall

$$f1_X := \frac{1}{\frac{1}{Precision_X} + \frac{1}{Recall_X}}$$

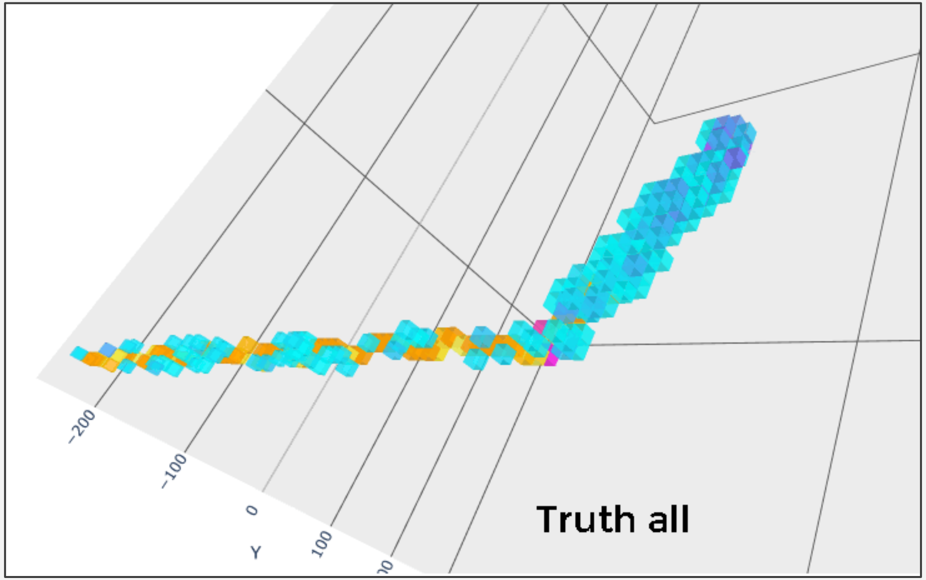
II.C Results of hit tagging

Comparison between a baseline cut, a decision tree and a Sparse CNN

Metric	Precision			Recall			Support
Model	Fixed volume	Decision tree	Sparse CNN	Fixed volume	Decision tree	Sparse CNN	
Vertex Activity	0.35	0.39	0.63	0.54	0.56	0.67	1.65%
Tracks	0.99	0.83	0.89	0.98	0.89	0.89	34.5%
Noise		0.95	0.94		0.91	0.94	63.9%
Macro average	**0.67**	**0.73**	**0.82**	**0.76**	**0.78**	**0.83**	**100%**

II.C Results of hit tagging

The model identifies the vertex activity hits



Example of a simulated CCQE event

II Hit tagging
Key takeaway

Metric	F1-score		
Model	Fixed volume	Decision tree	Sparse CNN
Macro average	**0.71**	**0.75**	**0.83**

Sparse CNN outperforms traditional methods in hit tagging and identifies precisely the vertex activity

Given the hits of a particle,
how to reconstruct its trajectory path ?

III.

Track fitting

Reconstruct the precise trajectory of particles inside the detector

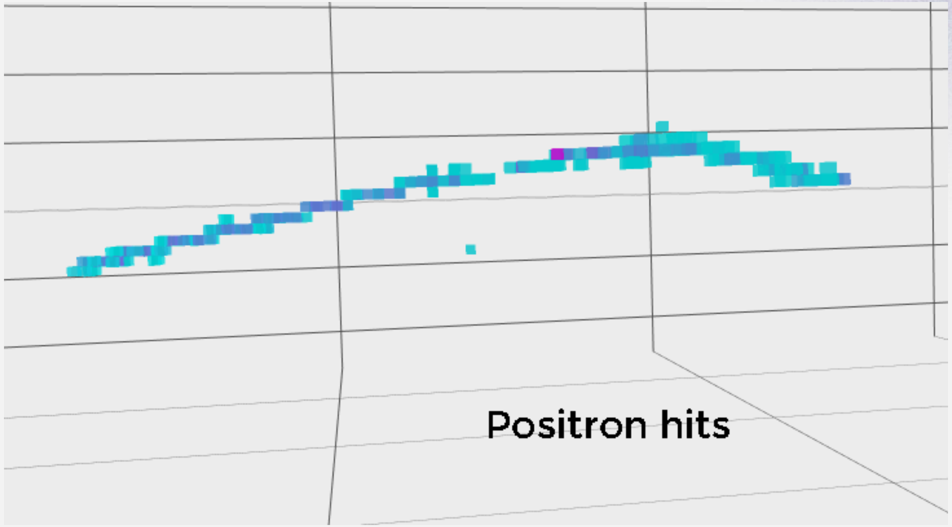
Original paper:
Artificial intelligence for improved fitting of trajectories of elementary particles in dense materials immersed in a magnetic field,
https://www.nature.com/articles/s42005-023-01239-4

III.A Tracks inside SFGD

What is the task ?

From the hits of a particle, find back its precise trajectory.

⇒ Predict the closest trajectory point of each hit.

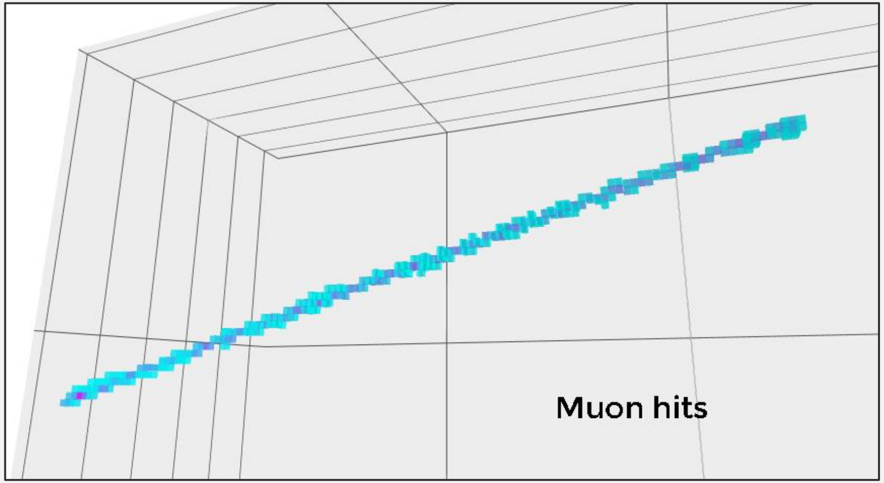


Hits and track of a simulated positron

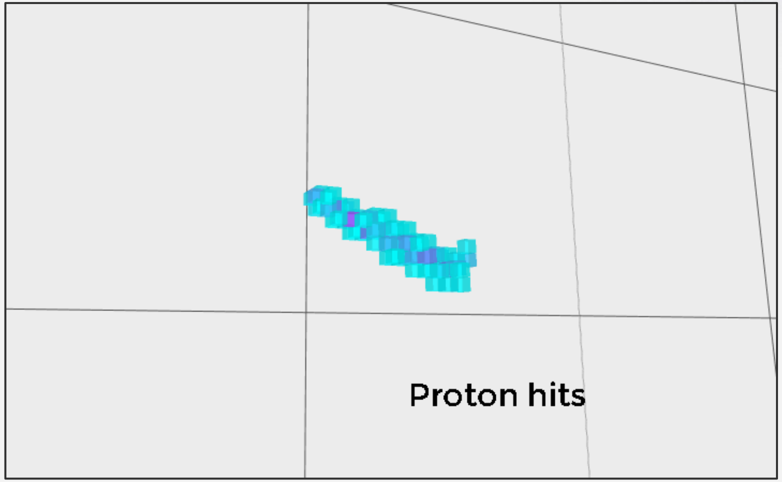
"Particle gun" dataset generated with Geant4
Agnostic of Neutrino interaction simulation

III.A Tracks inside SFGD

Long and short tracks



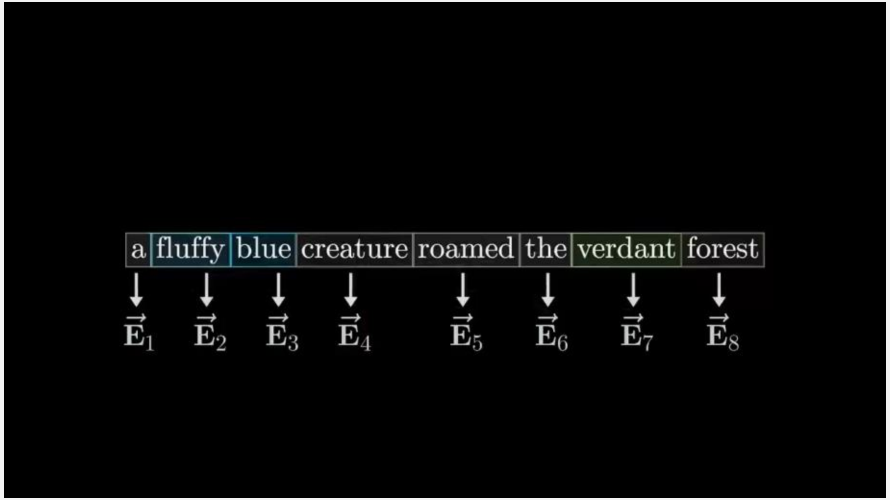
Hits and track of a simulated muon



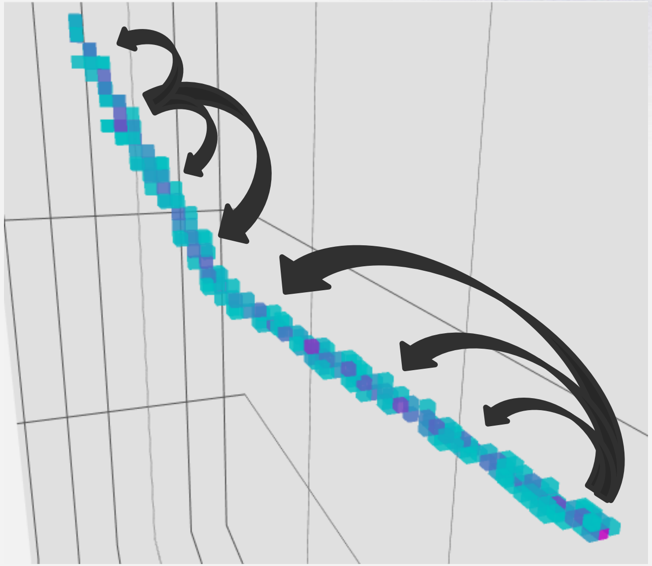
Hits and track of a simulated proton

III.B Models and results

Transformers!



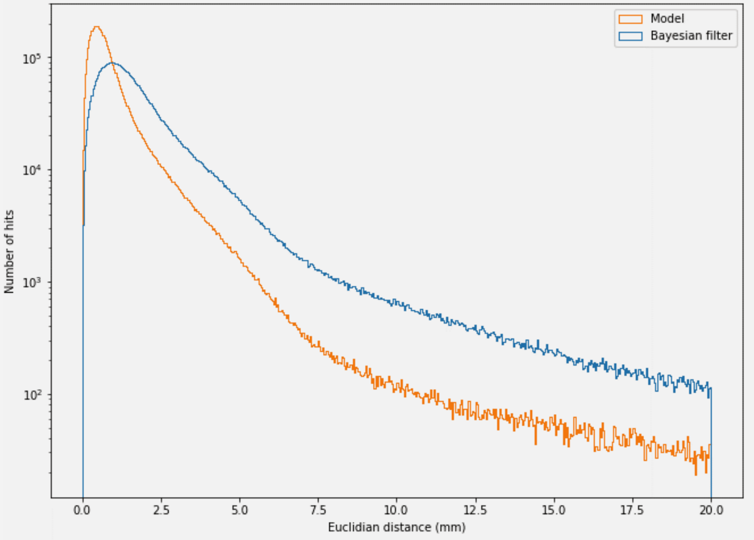
The attention mechanism explained: an accelerated extract from 3Blue1Brown Youtube video



Example of a pion track

III.B Models and results

Our model is twice better than a traditional Bayesian filter



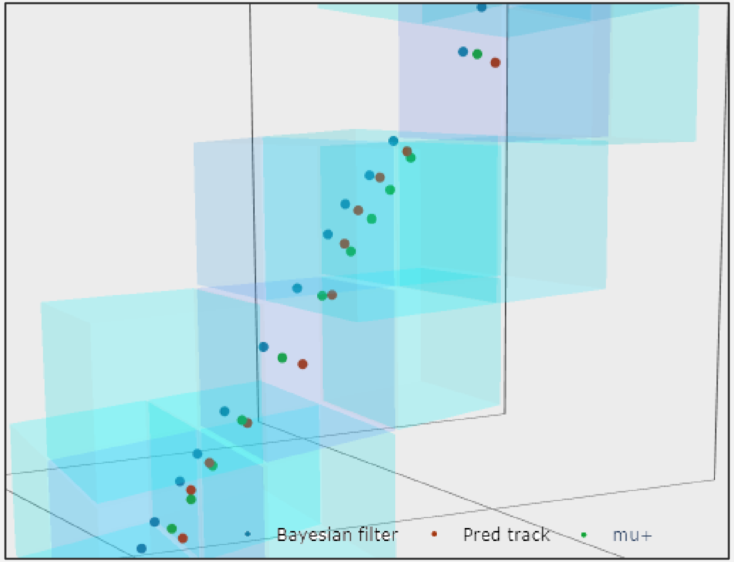
Distributions of the Euclidian 3D distances between the true and predicted points for our transformer model (blue) and a traditional Bayesian filter (orange)

Euclidian distance			
Model	$\sigma := 68\%$	Mean	$2\sigma := 95\%$
Bayesian Filter	2.04 mm	2.56 mm	5.69 mm
Transformer	0.97 mm	1.21 mm	3.15 mm

68% quantile, mean and 95% quantile of the distributions of Euclidian distances between the true and predicted points for our transformer model and a traditional Bayesian filter

III.B Models and results

Slight improvement in the direction estimation



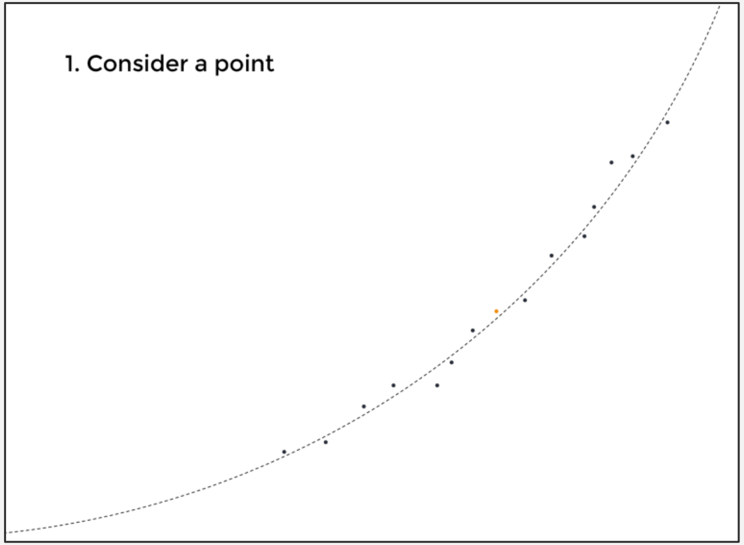
Example of predicted points for a muon track

Angular distance			
Model	σ := 68%	Mean	Wrong direction
Bayesian Filter	1.89°	5.56°	2.15%
Transformer	1.03°	5.79°	2.61%

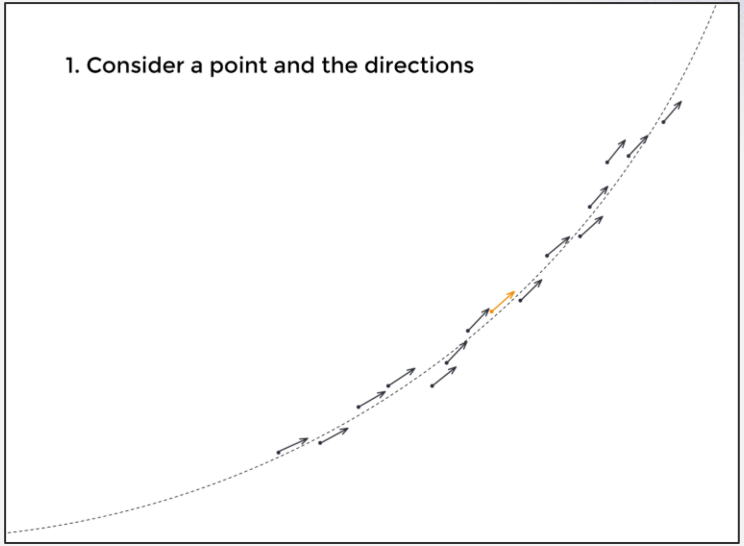
Angular distance between the reconstructed and true direction from predicted trajectory points. *Wrong direction* corresponds to reconstructed direction with opposite sign

III.C Apllication: Charge identification

Estimate direction and curvature



Direction estimation from predicted trajectory points



Curvature estimation from estimated directions

III.C Apllication: Charge identification

Slight improvement with the transformer model

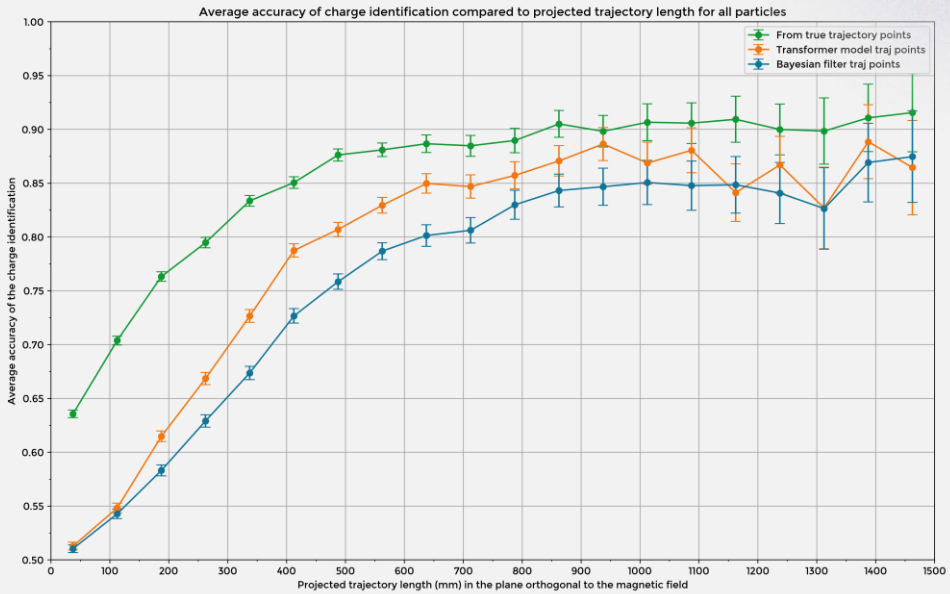
Results

The accuracy depends strongly on the trajectory length

The maximum accuracy is limited even with the true trajectory points

Trajectory points	Bayesian filter	Transformer	Truth
F1-score	0.71	0.75	0.83

Weighted f1-score over all particles of the test set



Charge identification accuracy from true trajectory points (green), transformer model predicted points (orange) and Bayesian filter points (blue) compared to the trajectory length orthogonal to the magnetic field.

III Track fitting
Key takeaway

Distance to true points		
Model	Bayesian filter	Transformer
$\sigma := 68\%$	2.04 mm	0.97 mm

Machine Learning models are twice more precise than a Bayesian filter for fitting the trajectory of particles from their hits

Given the vertex activity hits,
how to reconstruct the particles involved ?

IV.

Vertex activity

Find back the particles involved in the vertex activity region

Original paper:
Deep-learning-based decomposition of overlapping-sparse images: application at the vertex of simulated neutrino interactions
https://www.nature.com/articles/s42005-024-01669-8

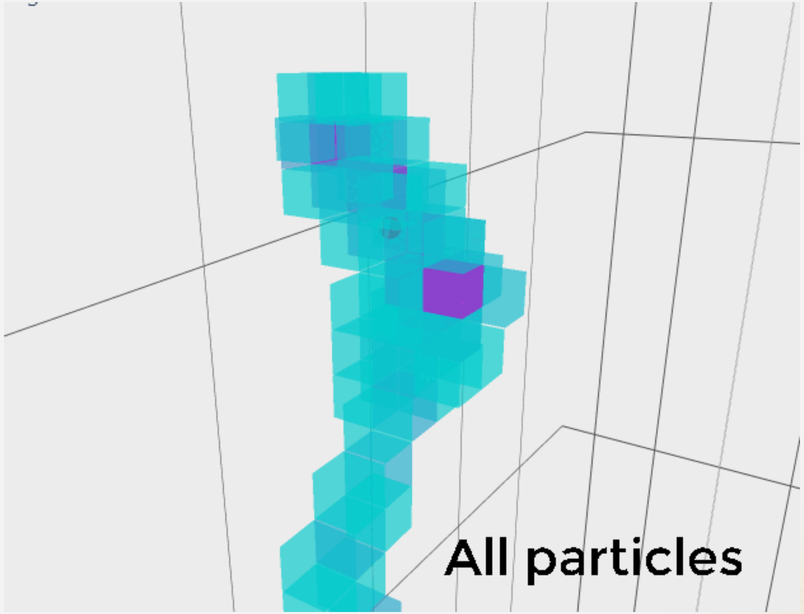
IV.A Context of the problem

Vertex activity region

Superposition of multiple particles

⇒ Difficulties to reconstruct the energy

“Unbiased” dataset



Simulated CC Pi 0 event with a muon and two proton.
The vertex position is shown with a red dot

“Particle bomb” dataset generated with Geant4

IV.A Context of the problem

Traditional approach

Approximation formula of the reconstructed energy based on the energy deposition and proton length.

Doesn't account for number, kinetic energy and type of particles

$$VisE_{\text{reco}}[\text{MeV}] = \frac{1}{1 - C_B \frac{E_{cali}}{\Delta X}} \cdot E_{cali}$$

- C_B: Birks coefficient, equal to 0.0126 cm/MeV.
- E_{cali}: represents a relation equal to $\frac{E_{\text{loss}}[p.e.]}{c_{\text{cali}}}$, where $E_{\text{loss}[p.e.]}$ is the total deposited energy (in photo-electrons), and c_{cali} is a calibration factor equal to 100 p.e./MeV.
- ΔX: approximate length of the longest proton, in millimetres.

Energy reconstruction approximation formula

IV.A Context of the problem

Differences with current study

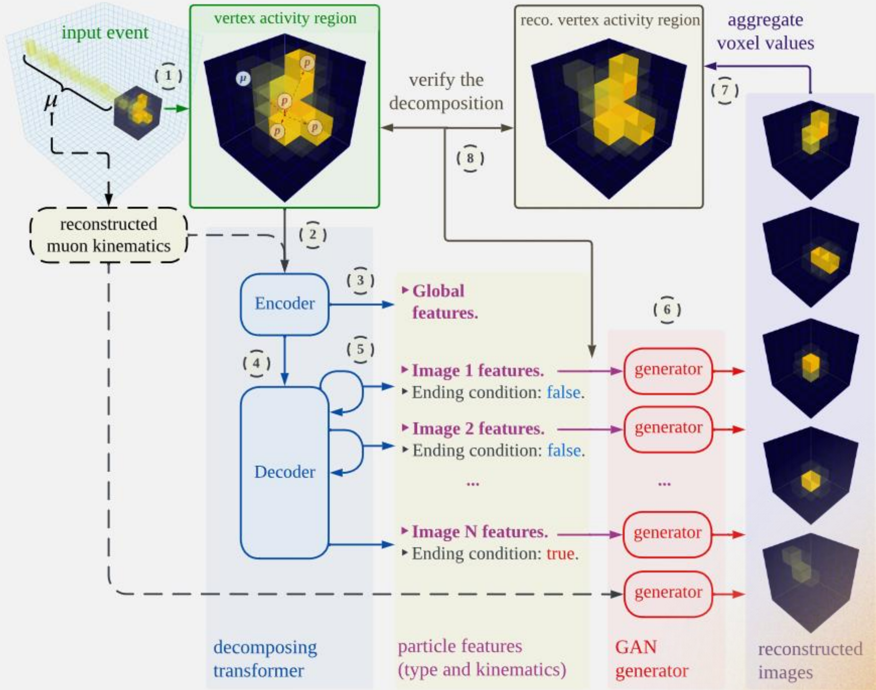
- Different simulation environment
- Lower cross talk rate
- No neutrons (only protons)
- Lower energy range (0-60 MeV instead of 0-100 MeV)

Current study in progress

IV.B Architecture chosen

Transformers!

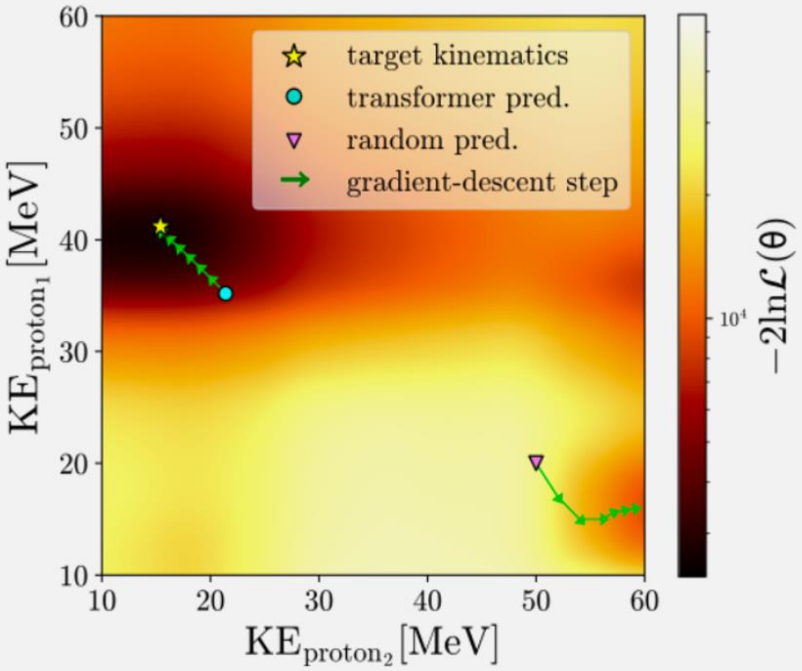
1. Transformer Encoder to analyse the event
2. Transformer Decoder to produce a sequence of particles
3. GAN to compare the prediction and refine it



Pipeline for an example neutrino interaction

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

IV.B Building the likelihood



Likelihood fit

Summing extremely fast simulations of single-particles each one obtained with the GAN:

- Validate on beam test data
- Fully differentiable, so easy to maximise
- Propagate statistics-meaningful systematics in a traditional manner

⇒ **Not possible without ML**

IV.C Previous results

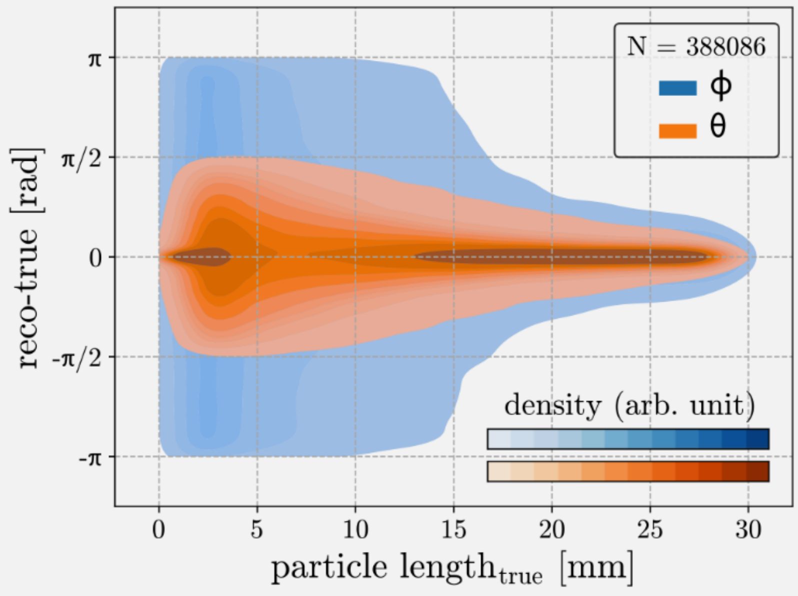
Energy resolution in the vertex activity region: ~8% better than formula

Vertex activity energy resolution (RMS)			
Number of protons	Approximation formula	Transformer	Transformer + GAN
1	33 %	27%	26%
2	22%	17%	13%
3	19%	11%	8%
4	21%	11%	9%

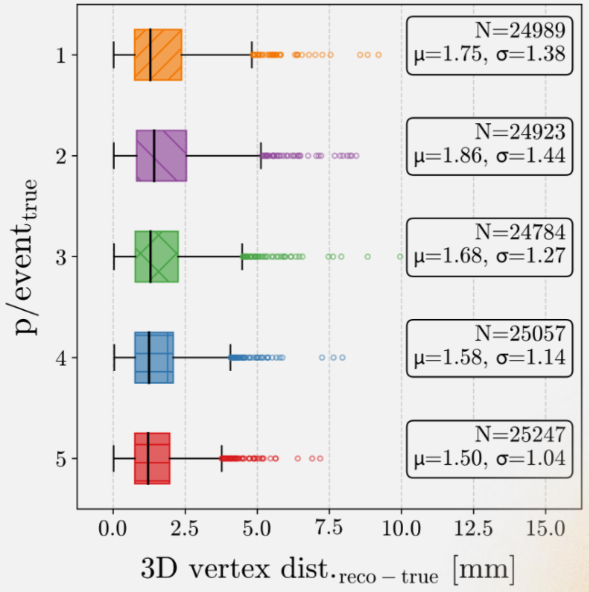
Energy resolution of the vertex activity region, depending on the number of protons of the event, for the approximation, the transformer only and the transformer coupled with the GAN refinement, tested on NEUT events.

IV.C Previous results

Good vertex position and particles directions resolutions



Angular resolution distribution depending on the particle trajectory length



Vertex resolution distribution depending on the number of protons in the event

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

IV Vertex activity
Key takeaway

Energy resolution improvement by ~8%

Also precise resolution of the vertex position and particles directions.
The GAN helps improving the precision, especially for some hard outliers.

V. Conclusion

Will Machine Learning help the data analysis of SFG detector ?

Key takeaways

*An "**unbiased**" strategy to reconstruct neutrino interactions with Deep Learning*

Hit tagging	→	Sparse CNN models surpass traditional methods	→ ***F1 score***: 0.83 vs 0.75
Track fitting	→	Transformers outperform Bayesian filter	→ ***Point resolution***: 0.97 mm vs 2.04 mm
Vertex activity	→	Transformers + GAN outdo approximation formula	→ ***Energy resolution***: 25% vs 32%

Machine Learning improves data analysis !

Key limitations

Hit tagging → Relies on the vertex reconstruction. Relies on Neutrino interaction simulator (NEUT, ...)

Track fitting → Relies on the track segmentation. True points unsufficient for charge identification.

Vertex activity → Unknown real data distribution. Limited improvement.

Data itself poses an upper bound to performances.

What’s next?

01 Complete vertex activity task on new dataset

02 Test the three models on real data

03 Integrate the models in the collaboration pipeline

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

Thanks!

Do you have any questions?

Mayeul Aubin: m.mayeul.aubin@gmail.com

Saul Alonso-Monsalve: saul.alonso.monsalve@cern.ch

Davide Sgalaberna: davide.sgalaberna@cern.ch

Appendix II.A : Hit tagging dataset

Events generated with NEUT

Training set: 135k events

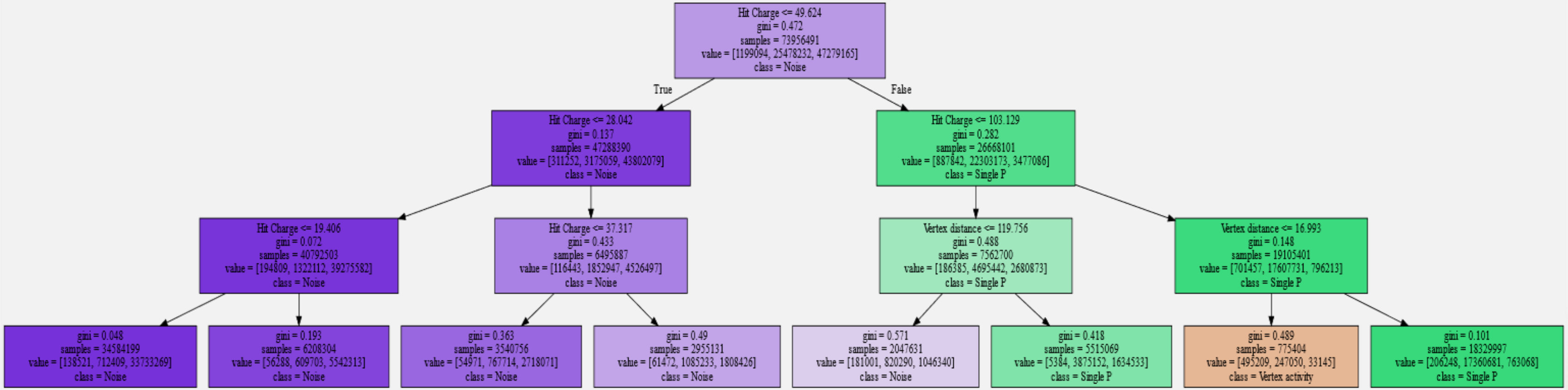
Validation set: 32k events

Test set: 50k events

Interaction	CCQE	CC p pi+	CC Multi pi	2p2h	CC DIS	NC elastic	CC n pi+	CC n pi0	NC p pi0	Other
Fraction of events	38.06%	13.51%	6.06%	5.97%	5.59%	5.38%	4.14%	3.91%	2.29%	15.1%

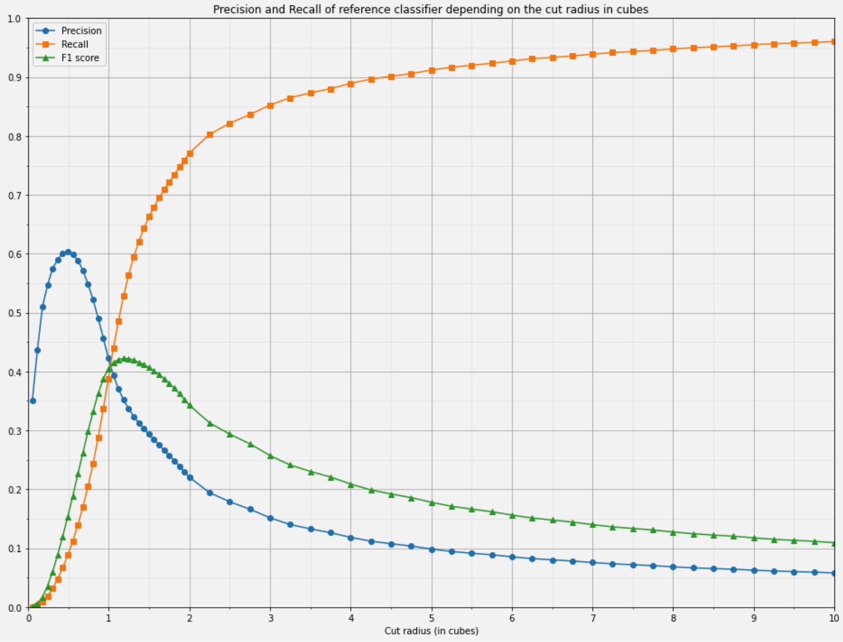
Main interaction types of the hit tagging dataset generated with NEUT sorted by decreasing fraction of events.

Appendix II.B : Hit tag decision tree



Structure and values of the optimal decision tree for hit tagging with a maximum depth of 3, and with input features hit charge and distance to the reconstructed vertex position. Purple represent noise tag, green track tag and orange vertex activity tag.

Appendix II.B : Optimal volume cut



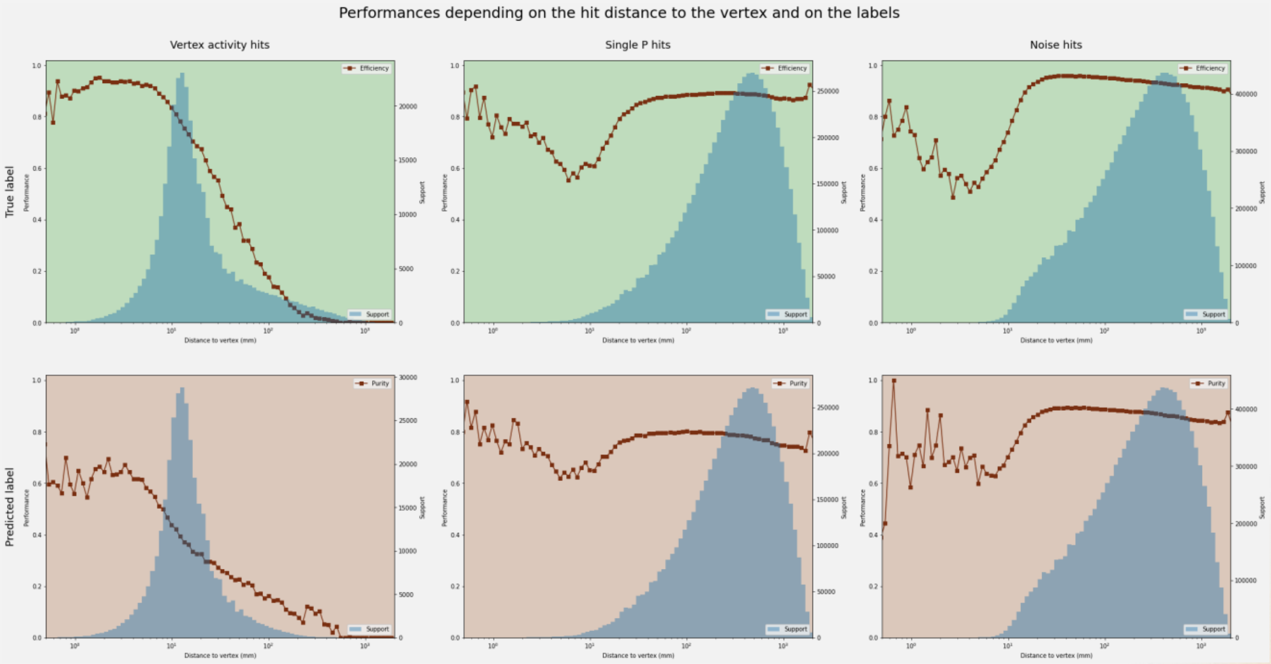
Evolution of precision, recall and f1-score for a fixed volume classifier with the cut on the distance to the vertex

Appendix II.C : Performance vs hit charge

Recall and precision (red, left y-axis) for true and predicted labels (rows) by tag (columns) depending on the hit charge (x-axis), with hit charge distribution (blue, right y-axis)

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

Appendix II.C : Performance vs vertex distance



Recall and precision (red, left y-axis) for true and predicted labels (rows) by tag (columns) depending on the distance to the vertex (x-axis), with the distance to the vertex distribution (blue, right y-axis)

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

Appendix III.A : Track fitting dataset

Events generated with Geant4

Training set: 235k particles

Validation set: 56k particles

Test set: 56k particles

Interaction	e+	e-	gamma	mu+	mu-	n	p	pi+	pi-
Fraction of events	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
Fraction of hits	15.3%	15.2%	0.02%	17.1%	16.8%	0.07%	13.6%	11.0%	10.9%
Energy range	0-3.5 GeV	0-3.5 GeV	0-1.5 GeV	0-2.5 GeV	0-2.5 GeV	0-1.5 GeV	0-1.5 GeV	0-1.5 GeV	0-1.5 GeV

Particles used in the track fitting dataset

Mayeul Aubin @ IPA ETH zürich: m.mayeul.aubin@gmail.com

Appendix III.C : All results

Best models		Primary tracks only			All data		
		Mean	68%	95%	Mean	68%	95%
Trajectory points offset only		1.21 mm	0.97 mm	3.15 mm	3.02 mm	2.38 mm	9.47 mm
Trajectory points and direction	Points	1.46 mm	1.50 mm	3.71 mm	3.14 mm	2.52 mm	9.62 mm
	Direction	0.192 rad	0.10 rad	0.98 rad	0.401 rad	0.23 rad	2.07 rad
Bayesian Filter	Points	2.56 mm	2.04 mm	5.69 mm	7.10 mm	3.81 mm	14.62 mm
	Direction	0.147 rad	0.03 rad	0.81 rad	0.393 rad	0.18 rad	2.30 rad

Table of all best models. We have best models for primary tracks only, or for all data (include secondary tracks and vertices), we have models predicting only the trajectory points, or also the trajectory directions