SYMPOSIUM ON AI FOR SCIENCE & ENGINEERING · IIT HYDERABAD

A novel use of LLM in Particle Physics

Track reconstruction as a language translation problem

Deepak Samuel

Central University of Karnataka

23 July 2026

TALK OUTLINE

01 **LLM basics for physicists**
What they are, how they are trained

02 **Tracking as a coding task**
Why numbers fail, and workaround

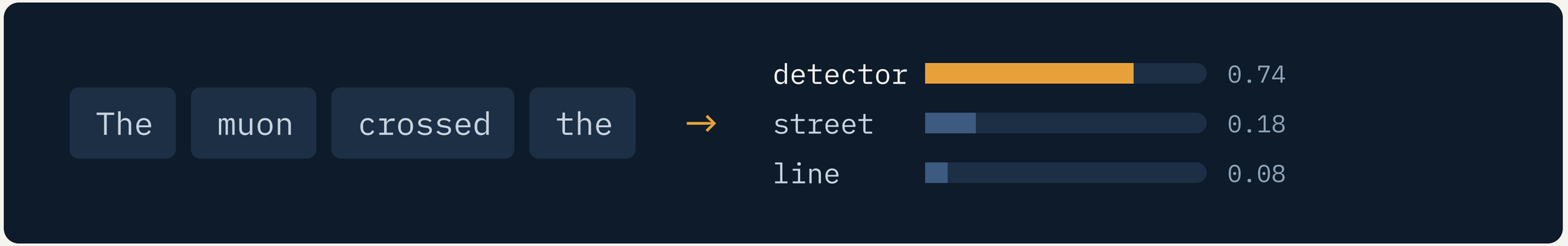
03 **The inspiration**
Compression by code generation

04 **A toy model: INO-ICAL prototype**
Methodology and training

05 **Results**
Single tracks, noise, similarity analysis

06 **Outlook**
What next

WHAT IS A LARGE LANGUAGE MODEL?



A learned probability distribution

p(next token | context) — billions of parameters
fit to trillions of words

Sample a token, append it, predict again —
essays and code emerge from this loop

HOW LLMS READ TEXT

Build a Large Language Model (From Scratch) by Sebastian Raschka

1 · TOKENIZATION

Text becomes integers

The	muon	cross	ed	the	detector
37	28372	2654	26	10	18519

2 · TOKEN EMBEDDING

Each token id is looked up as a 3D vector

The	muon	cross	ed	the	detector
0.12, -0.87, 0.34	0.42, -1.05, 0.88	-0.61, 0.29, 1.02	0.05, 0.71, -0.44	0.12, -0.87, 0.34	-0.93, 0.18, 0.56

3 · POSITION ENCODING: A DOG BIT A MAN ≠ A MAN BIT A DOG

Each position also gets its own 3D vector

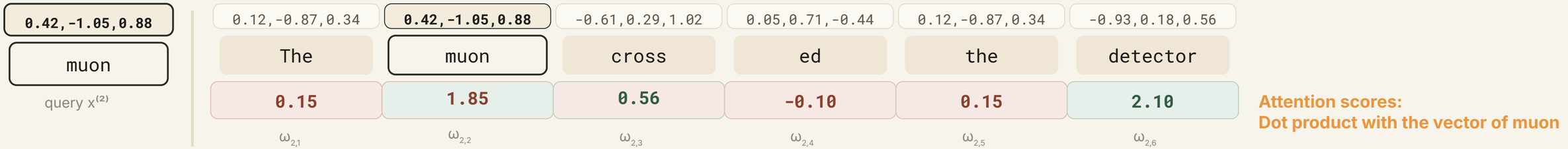
pos 0	pos 1	pos 2	pos 3	pos 4	pos 5
0.00, 1.00, 0.00	0.84, 0.54, 0.01	0.91, -0.42, 0.03	0.14, -0.99, 0.04	-0.76, -0.65, 0.05	-0.96, 0.29, 0.06

Token ID	Token String
0	<pad>
1	</s>
2	<unk>
3	
4	X
5	.
6	,
7	s
8	-
9	a
10	the
11	a
12	and
13	of
14	to
15	s
16	is
17	you
18	i
19	in
20	e
21	t
22	that
23	-

Vector embedding= Token embedding + Positional Encoding

Every word, now a 3d vector

From Attention Scores to Context Vector



$z^{(2)} = \sum_i \alpha_{2,i} \cdot x^{(i)}$ — scores normalized into weights α, each vector scaled and summed

Σ

Each input vector is multiplied by its attention weight, then all six results are summed.

Context vector:

-0.60, 0.04, 0.70

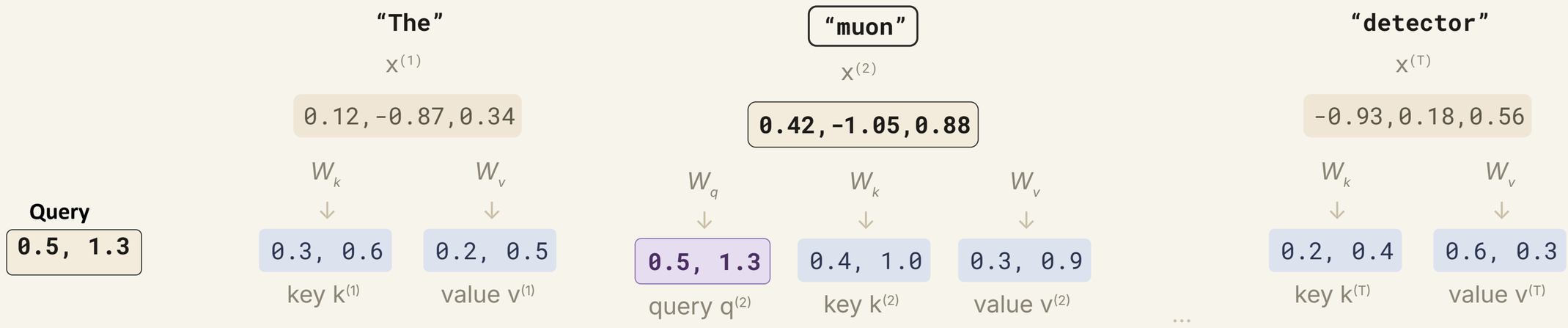
$z^{(2)}$

This is the context vector for “muon”

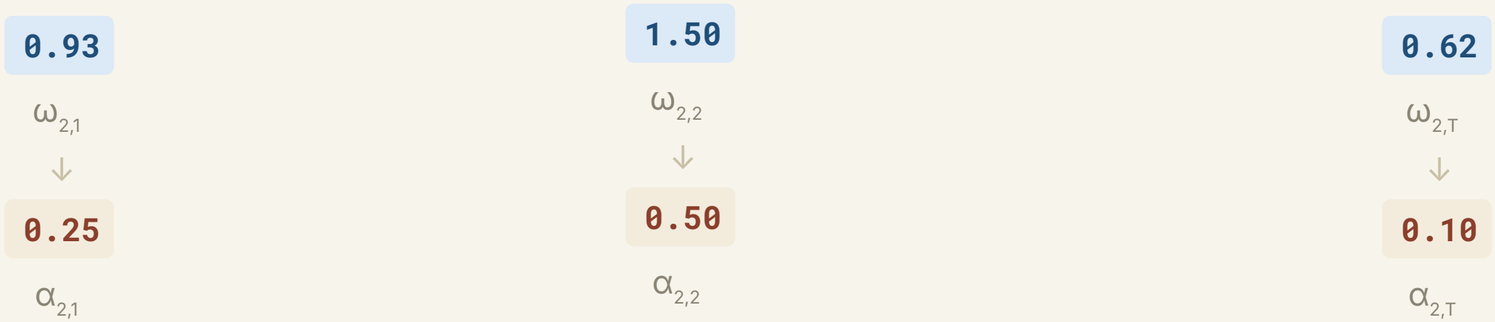
Every word, now has context vector which contains information of its position and context with respect to other words.

Context vector with trainable weights

Learned matrices W_q, W_k, W_v project each token into query, key & value vectors



Query . Key = Att. scores⇒

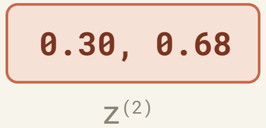


Attn. weights = normalised scores

Context vector = $\sum w_i \cdot v_i$

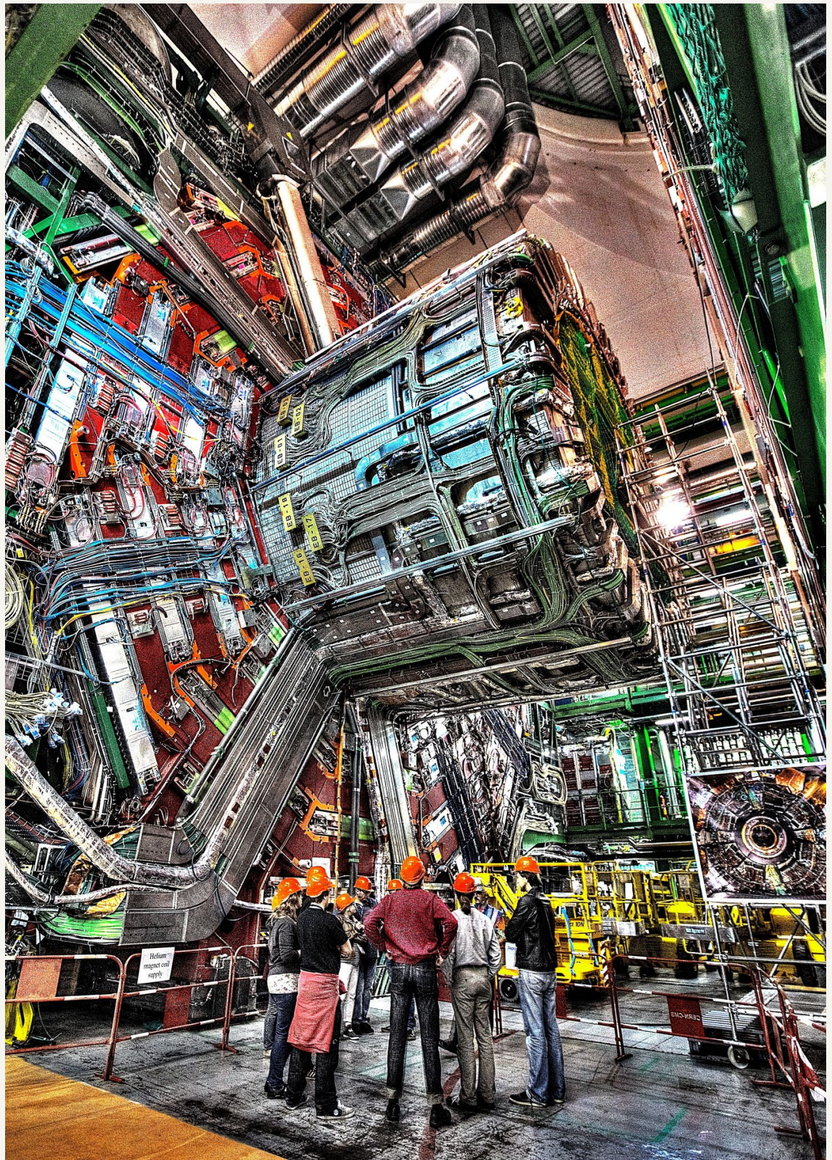
Every token produces a key and value from its own embedding; only the query token also produces a query vector, used to score relevance against every key.

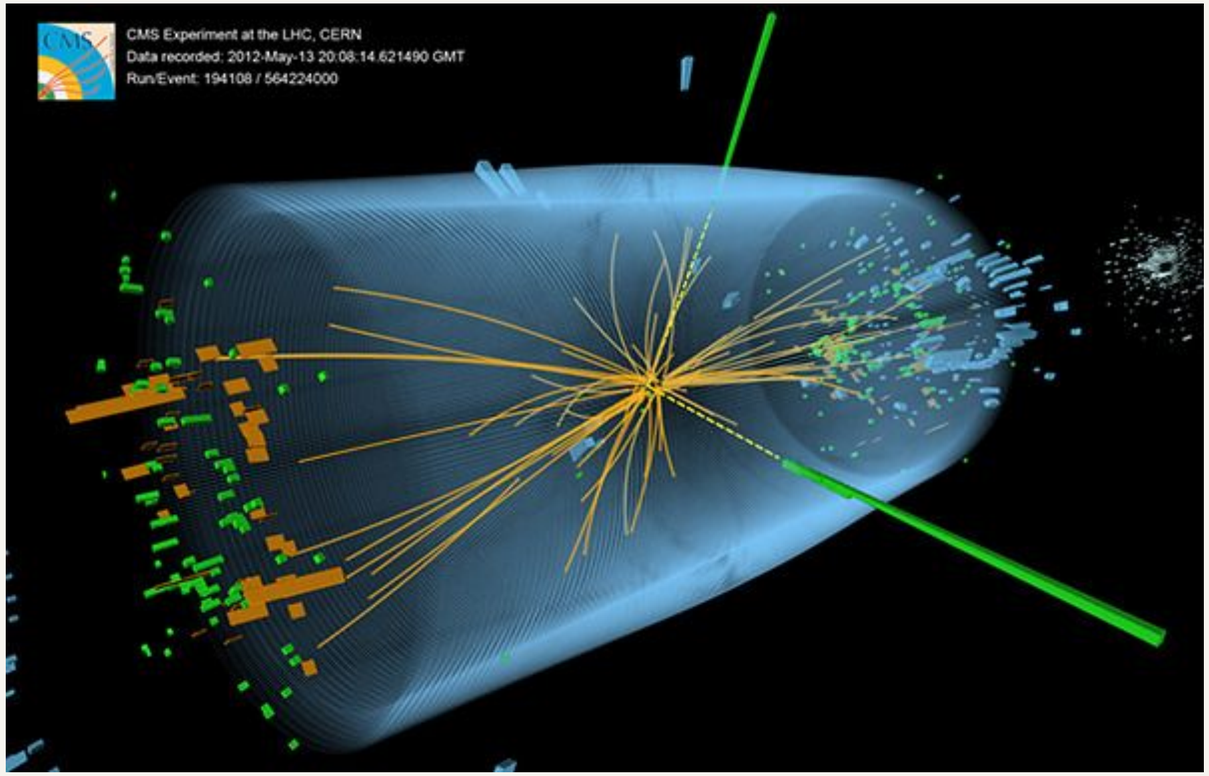
Context vector:



The last step multiplies each value vector by its attention weight α and sums them — this is the fully learned self-attention context vector for "muon".

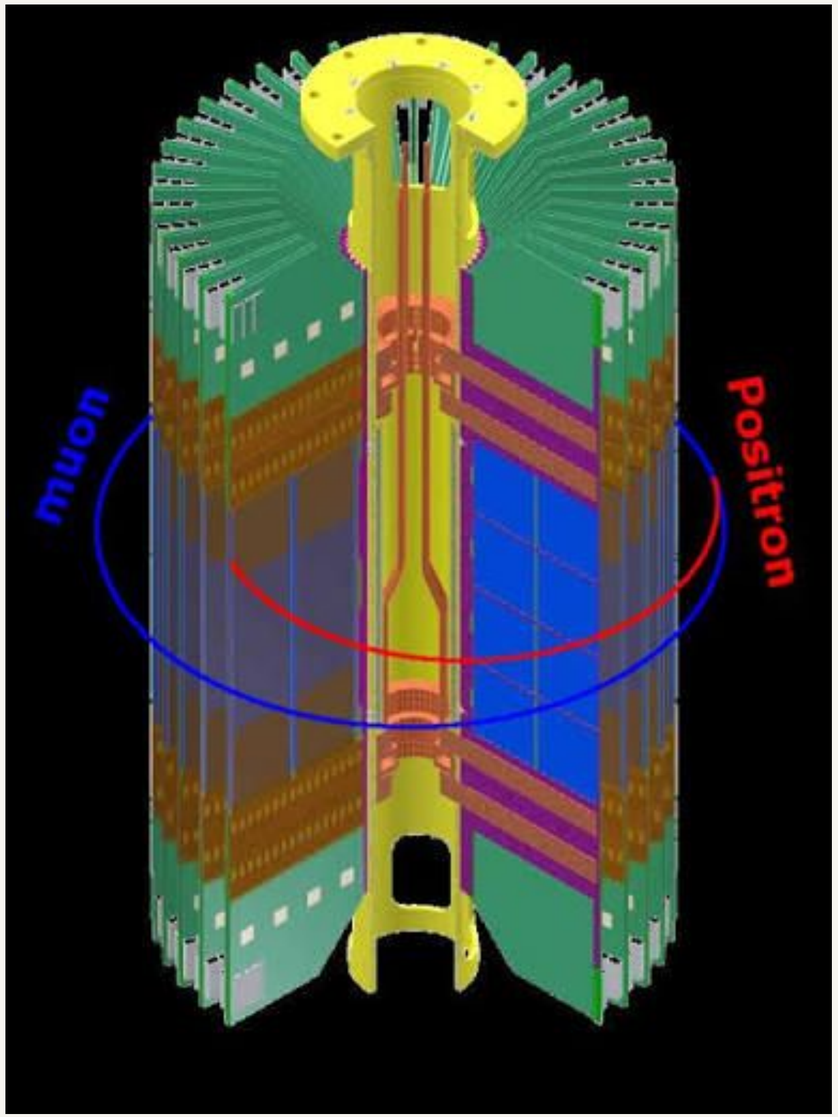
Particle physics detectors

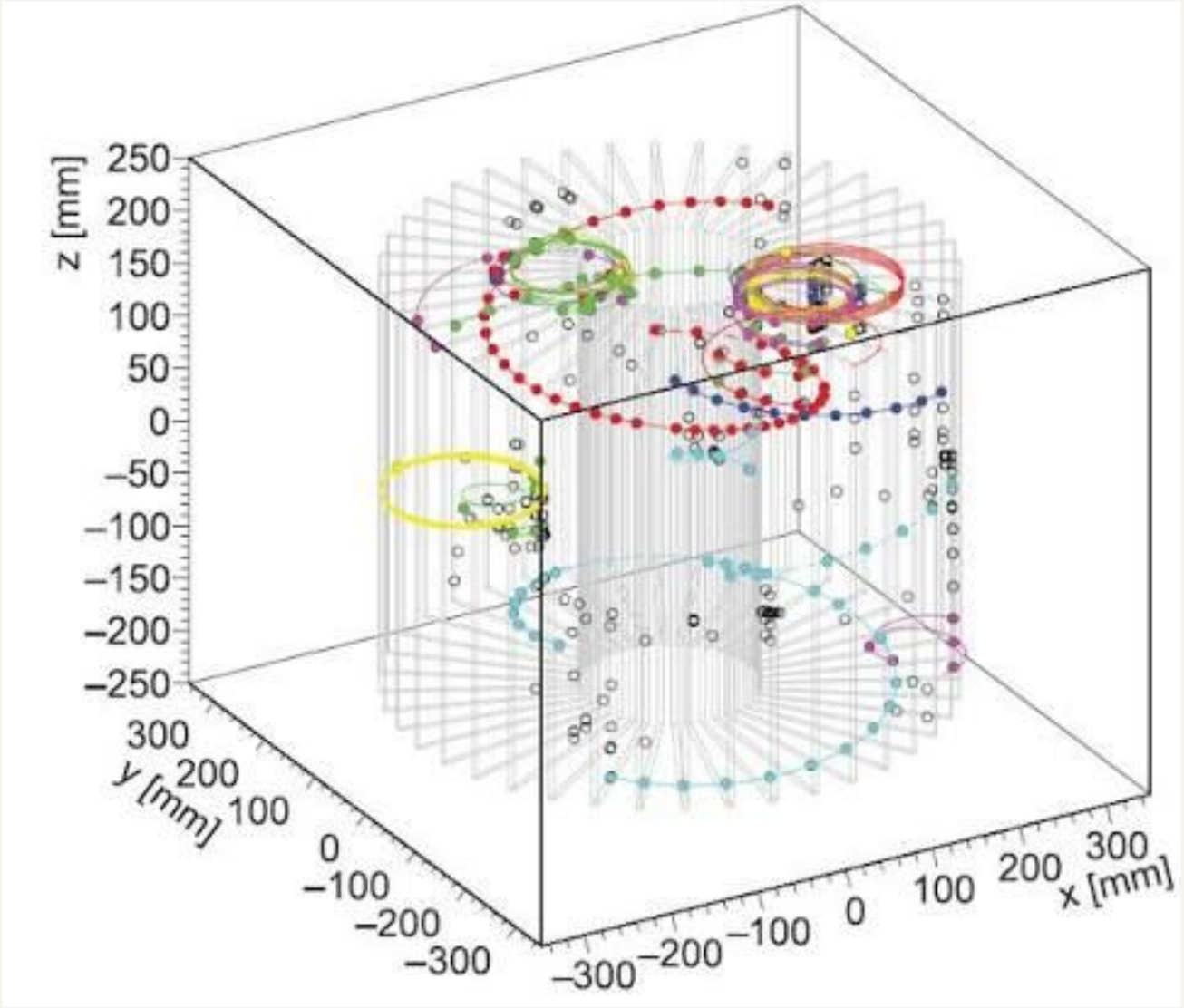




Particle physics: at the core of it is an algorithm that measure particle properties from particle track called track reconstruction.

Track reconstruction in particle physics





Track reconstruction: Given a set a space-time hits (x,y,z,t), find out how many tracks are present and their momenta

Can one use a LLM for track reconstruction?

How effective will it be?

WHAT LLMS ARE GOOD AT

Mostly text processing...

Drafting & summarising

Emails, reports, reviews

Translation

Language to language — the original seq2seq task

Coding

Code is text with rigid grammar — models excel here

Agentic workflows

cognitive backbone: planning, tool usage, reflection

Common thread: **text in, text out**. Track reconstruction is neither.

TRACK RECONSTRUCTION IS TRANSLATION

LANGUAGE OF DETECTORS

Space-time hits

```
X (cm) Y (cm) Z (cm)
23.24 44.52 0.12
2.34 34.56 0.34
22.56 54.34 -0.23
5.45 45.23 -0.32
…
```

→

track finding
+ track fitting

LANGUAGE OF PHYSICS

Track parameters

```
Track Momentum/MeV
1 200.2
2 0.9
3 50.4 …
```

Can a LLM be designed to do this job?

THE PROBLEM WITH THE DIRECT APPROACH

PROBLEM 1

Textual output is ambiguous

Digits can be interchanged, decimal points interpreted as full stops.
200.2 === 20.02

PROBLEM 2

No easy way to verify

p = 200 MeV
p = 200 mev

WHERE LLMS STRUGGLE: NUMBERS

Track information is numerical

- Numbers are encoded **digit-by-digit in base 10** — not as magnitudes (Levy & Geva, 2024)
- Benchmarks expose **fundamental numeracy gaps** even in flagship models (Li et al., ACL 2025)
- Algebraic reasoning **fails across complexity dimensions** (Patil et al., 2026; Boye & Moell, 2025)
- A tokenizer happily splits `23.24` into arbitrary fragments

132 + 238 + 324 + 139 = 833

LLM : 633 or 823 (more string similarity than value similarity like 834 or 832)

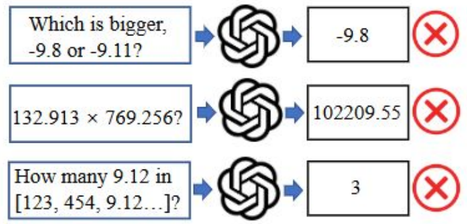
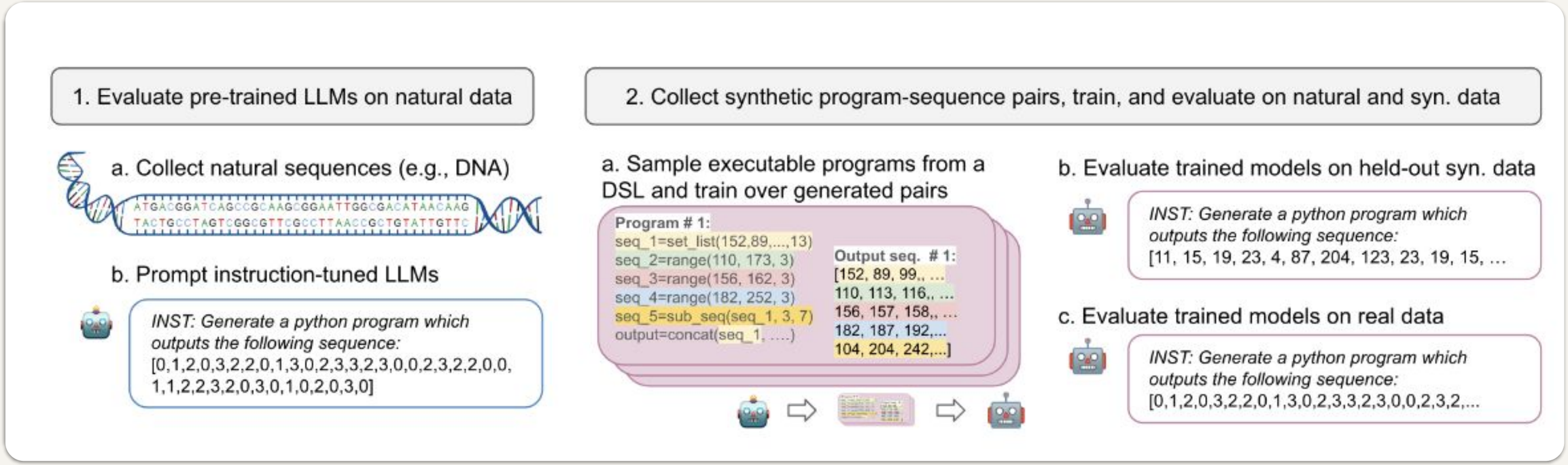


Figure 1: Numerical tasks answered incorrectly by GPT-4o. Details are in Figure 5, Figure 6, and Figure 7.

Consequently, LLMs treat numbers as discrete tokens rather than continuous magnitudes, inherently limiting their ability to understand exact numerical semantics.

THE KOLMOGOROV TEST: COMPRESSION BY CODE GENERATION

The inspiration



Yoran, Ori, et al. "The koLMogorov test: Compression by code generation." arXiv preprint arXiv:2503.13992 (2025).

THE KOLMOGOROV TEST: COMPRESSION BY CODE GENERATION

DOMAIN SPECIFIC LANGUAGE

Two types of functions:

- Sequence initiators
- Sequence modifiers: reversing, repeating, or substituting, split, merge, filter

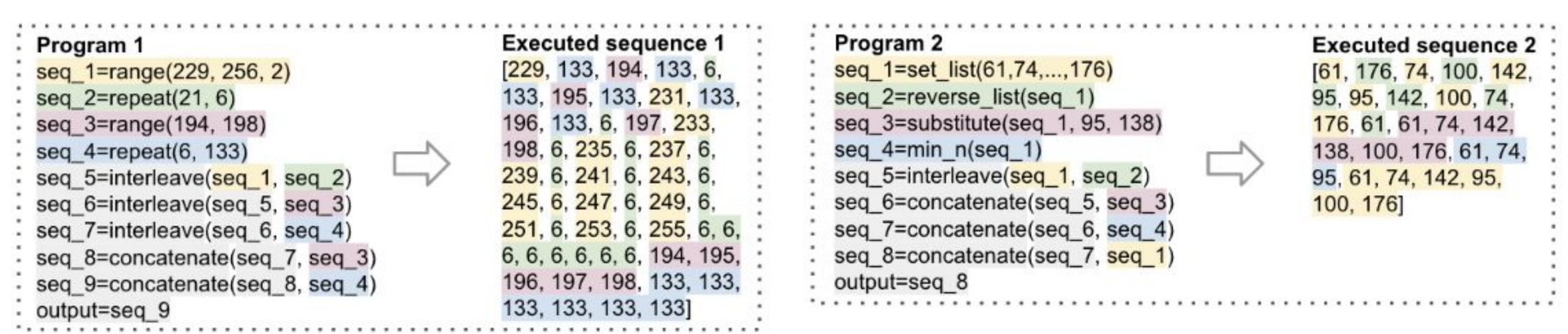


Figure 3: **Two examples of program-sequence pairs from our synthetic data generation process.**

Yoran, Ori, et al. "The koLMogorov test: Compression by code generation." arXiv preprint arXiv:2503.13992 (2025).

THE KOLMOGOROV TEST: COMPRESSION BY CODE GENERATION

The inspiration

	Audio-16-bit		Audio-8-bit		Audio-MFCC		DNA		Text		Syn.	
	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.	Acc.	Prec.
LLAMA-3.1-8B	5.9	1.54	3.9	1.74	8.8	1.51	3.7	2.34	1.4	3.12	8.5	2.48
LLAMA-3.1-70B	18.0	1.96	10.1	1.67	24.2	1.56	22.5	2.18	9.6	3.17	18.0	2.78
LLAMA-3.1-405B	35.6	1.66	15.0	1.66	29.6	1.58	24.8	2.06	6.5	3.17	33.3	2.18
GPT4-O	**69.5**	**1.34**	**36.4**	**1.43**	**83.8**	**1.33**	**50.3**	**1.73**	**54.2**	**1.94**	44.7	1.65
SEQCODER-1.5B	0.0	n/a	<0.1	n/a	0.1	n/a	0.0	n/a	0.0	n/a	84.5	0.57
SEQCODER-8B	<0.1	n/a	0.1	n/a	0.1	n/a	0.0	n/a	0.0	n/a	**92.5**	**0.56**

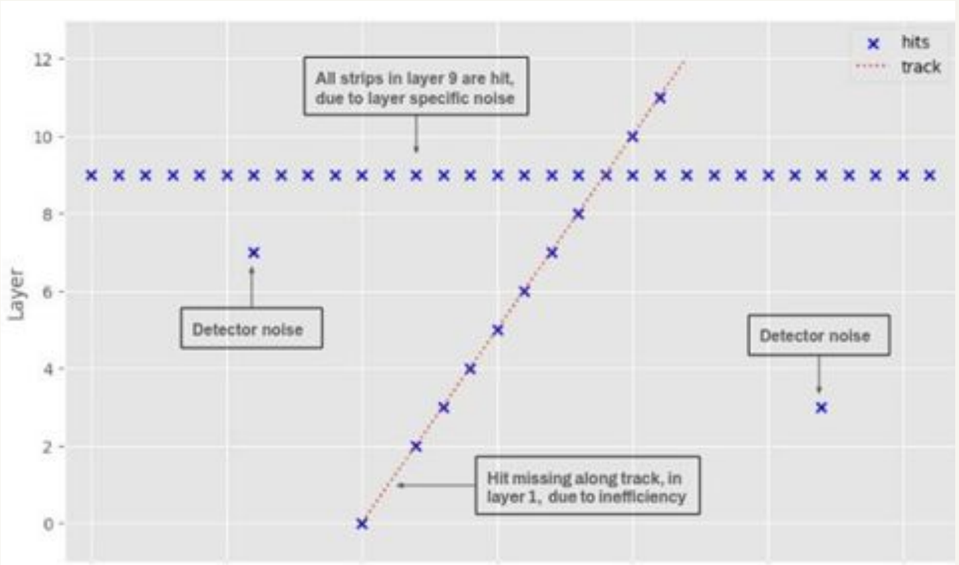
Table 1: **Accuracy and Precision for zero-shot prompted models on the KOLMOGOROV-TEST for sequences of length** 128**.** Larger models have higher Accuracy. On synthetic distributions when sampling program-sequence pairs is possible, our trained SEQCODER performs best and is the only model to achieve Precision that is lower than 1.0, i.e., it outperforms GZIP when programs are correct. However, it performs poorly on real data.

Accuracy: How accurate the program reproduces the pattern.

Precision: The size of the programme compared to the size of the pattern.

TRACK RECONSTRUCTION IN A SIMPLE DETECTOR

India-based Neutrino Observatory Cosmic Muon Detector Prototype



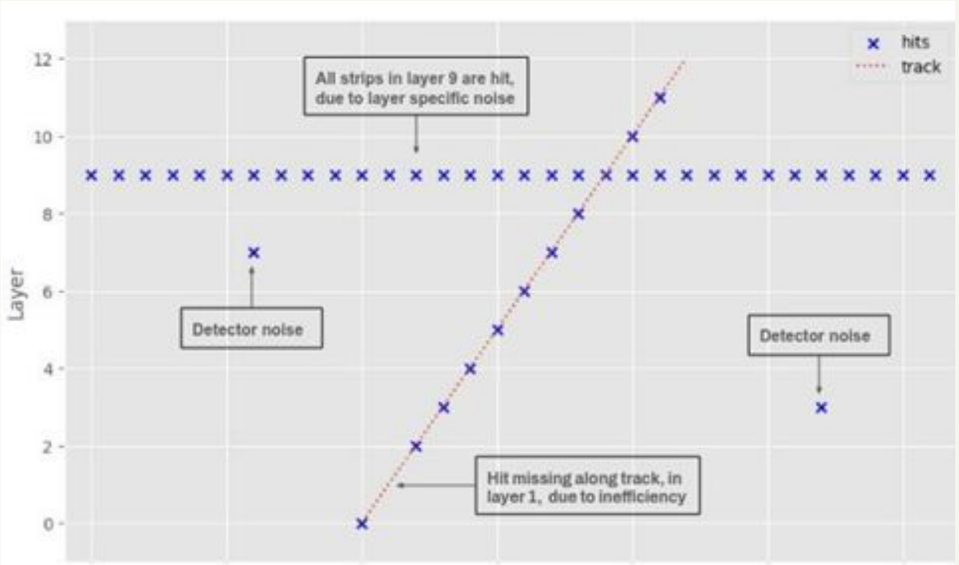
01 A particle crosses successive detector layers, leaving a **hit** in each — a position (x,y,z), and time t

02 **Track reconstruction:** Get the slope and intercept of this track.

03 **Slope and intercept:** Gets you the zenith angle (direction) of the particle.

TRACK RECONSTRUCTION IN A SIMPLE DETECTOR

The array [12,24,36,72,144,288…] displayed as a track



Noise & inefficiency

- gen_event(slp, icpt, add, rem, fill)

Track parameters

returns an array of hits: [12,24,36,72,144,288…]

- Added benefits:
  - data compression
  - seamless extension to multi-track event representation
  - direct loss computation and optimization

METHODOLOGY & TRAINING SETUP

Noise & inefficiency

- gen_event(slp, icpt, add, rem, fill)

Track parameters

- Added benefits:
  - data compression
  - seamless extension to multi-track event representation
  - direct loss computation and optimization

Base model & tokenizer	**`t5-small`**
Architecture	**text-to-text encoder–decoder**
Optimizer	**AdamW**
Clean model	**5 × 10^5 clean tracks**
Noisy model	**10^5 clean + 10^5 noisy**
Test sets	**10 k clean + 10 k noisy**

Supervised seq2seq learning: `hits → gen_event(...)` — exactly like machine translation.

[26, 58, 90, 122, 154, 186, 218, 249, 281, 313, 345, 377]	gen_event(slp=[-0.11],icpt=[26.67],add=[],rem=[],fill=[])
[21, 50, 79, 107, 136, 165, 193]	gen_event(slp=[-3.31],icpt=[21.79],add=[],rem=[],fill=[])
[16, 45, 73, 102, 131, 160]	gen_event(slp=[-3.11],icpt=[16.14],add=[],rem=[],fill=[])
[22, 50, 77, 105, 132, 160]	gen_event(slp=[-4.52],icpt=[22.73],add=[],rem=[],fill=[])
[62, 91, 119, 147, 175, 204, 232, 260, 288]	gen_event(slp=[-3.75],icpt=[34.5],add=[],rem=[],fill=[])
[3, 35, 66, 97, 128, 160, 192]	gen_event(slp=[-0.76],icpt=[3.87],add=[],rem=[],fill=[])
[28, 59, 91, 122, 154, 185, 217, 248, 280, 311, 343, 374]	gen_event(slp=[-0.51],icpt=[28.24],add=[],rem=[],fill=[])
[26, 58, 91, 123, 156, 188, 221, 253, 286, 318, 351, 383]	gen_event(slp=[0.5],icpt=[26.46],add=[],rem=[],fill=[])
[0, 37, 75, 112, 150, 188]	gen_event(slp=[5.6],icpt=[0.03],add=[],rem=[],fill=[])
[14, 45, 77, 108, 139, 171, 202, 234, 265, 296, 328, 359]	gen_event(slp=[-0.59],icpt=[14.24],add=[],rem=[],fill=[])
[15, 47, 78, 110, 141, 173, 204, 236, 268, 299, 331, 362]	gen_event(slp=[-0.45],icpt=[15.66],add=[],rem=[],fill=[])
[31, 60, 89, 118, 147, 176, 205, 234, 263, 292, 321]	gen_event(slp=[-2.99],icpt=[31.17],add=[],rem=[],fill=[])
[28, 57, 86, 116, 145, 174, 203, 233, 262, 291, 320]	gen_event(slp=[-2.75],icpt=[28.36],add=[],rem=[],fill=[])
[4, 42, 79, 117, 154, 191]	gen_event(slp=[5.39],icpt=[4.94],add=[],rem=[],fill=[])
[23, 56, 89, 122, 154, 187, 220, 253, 286, 318, 351]	gen_event(slp=[0.79],icpt=[23.83],add=[],rem=[],fill=[])
[16, 49, 81, 114, 147, 179, 212, 245, 277, 310, 343, 375]	gen_event(slp=[0.67],icpt=[16.43],add=[],rem=[],fill=[])
[23, 55, 88, 120, 153, 185, 217, 250, 282, 315, 347, 379]	gen_event(slp=[0.41],icpt=[23.39],add=[],rem=[],fill=[])
[7, 43, 79, 115, 151, 187, 223]	gen_event(slp=[3.89],icpt=[7.92],add=[],rem=[],fill=[])

THE CRUCIAL INGREDIENT

A new term in the loss function

$$L = L_{BCE} + \lambda [(1 - S) + D]$$

S — similarity

Fraction of true hits present in the predicted array

D — dissimilarity

Fraction of predicted hits not in the true array

Code that fails to run

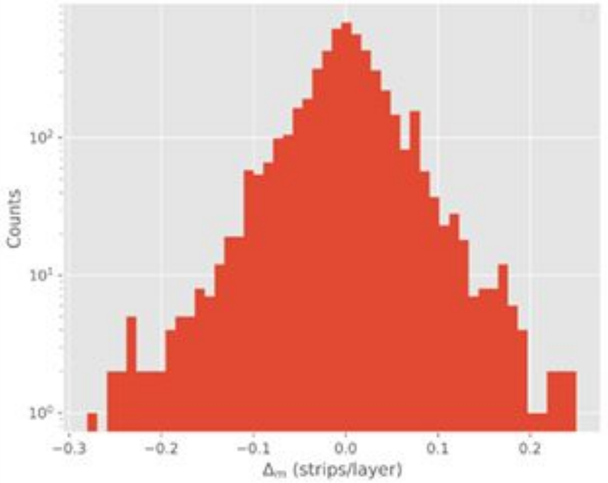
S = 0, D = 1 → maximal penalty. Syntax errors are trained away

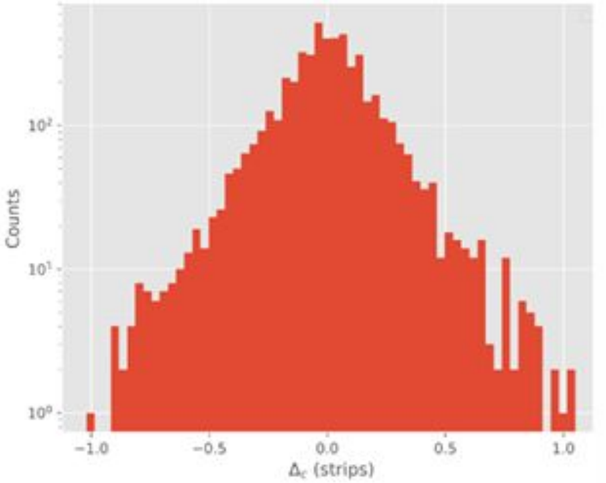
SINGLE TRACKS: RESULTS

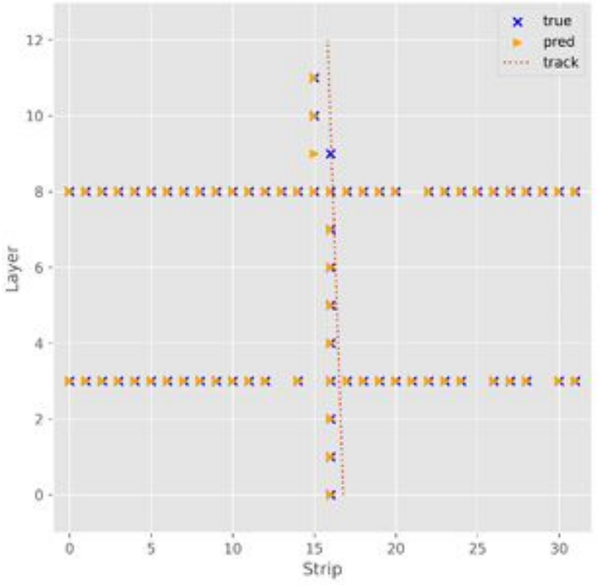
RMSD	slope	intercept
Linear fit (clean track)	0.024	0.15
LLM	0.050	0.23

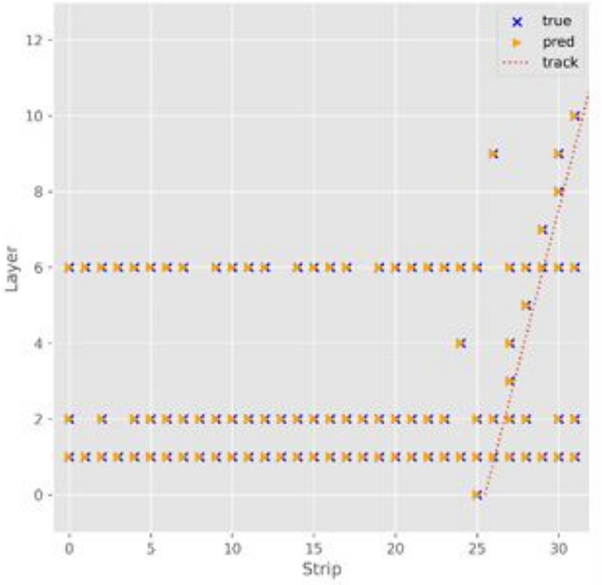
LLMs, without pre-processing were as good as linear fit results that could be obtained after noise removal

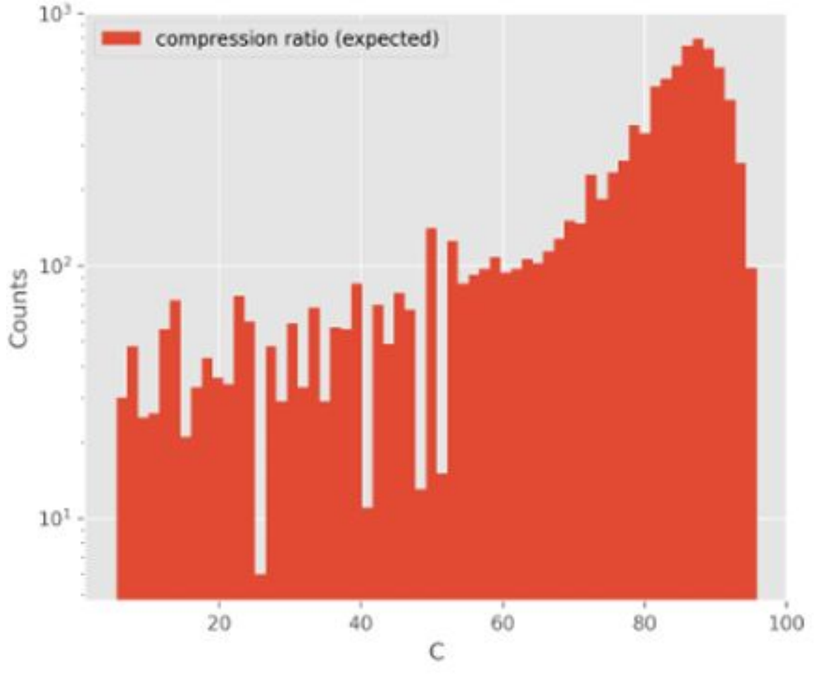
Could effectively identify noise hits, inefficiencies and fill layers

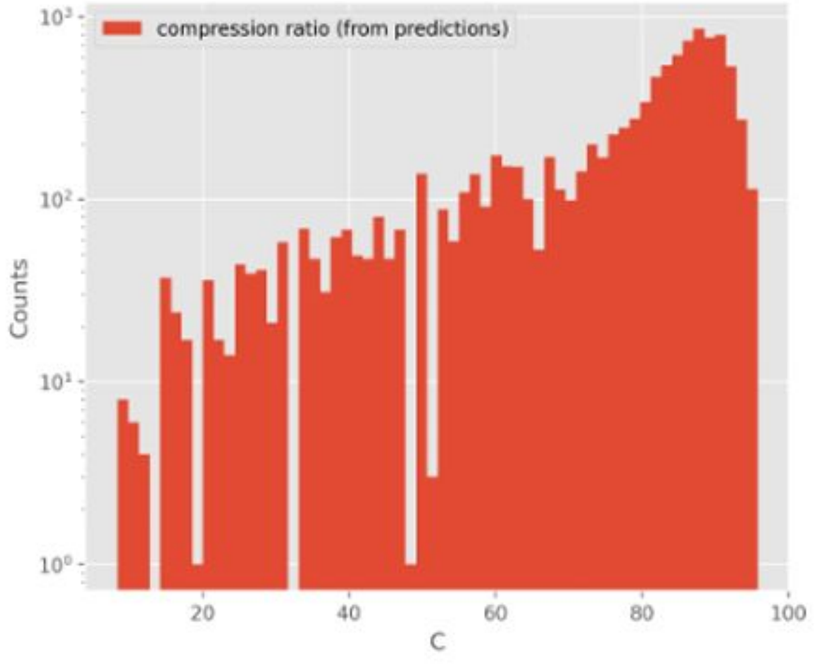


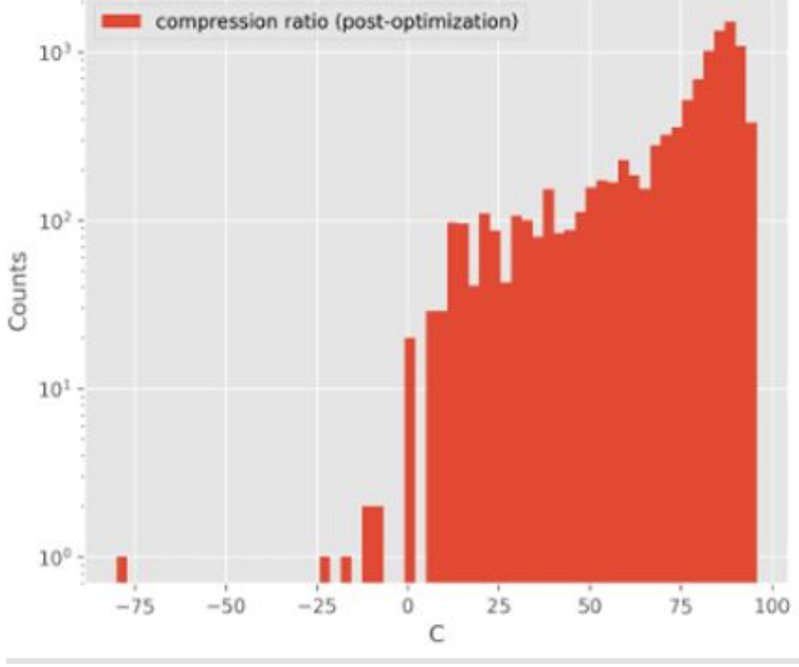


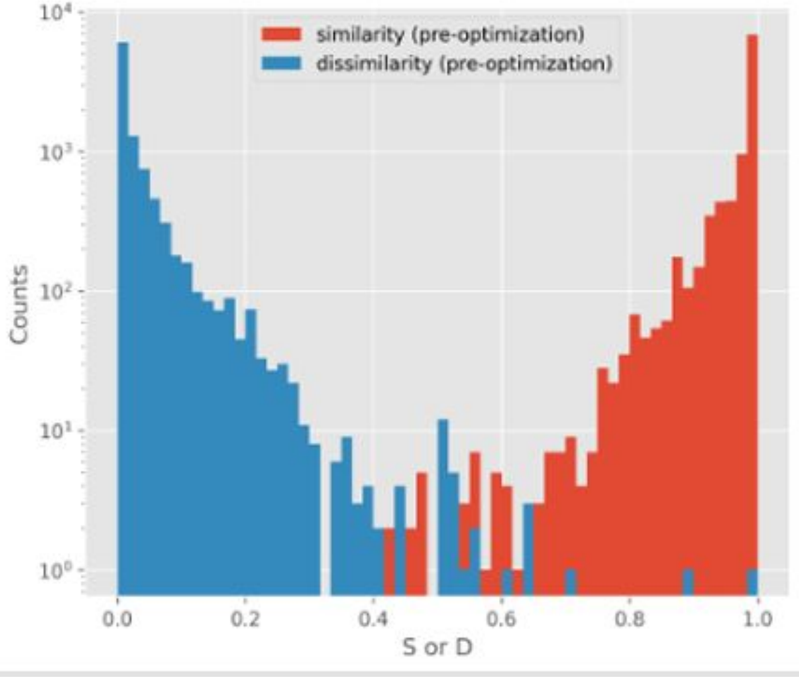








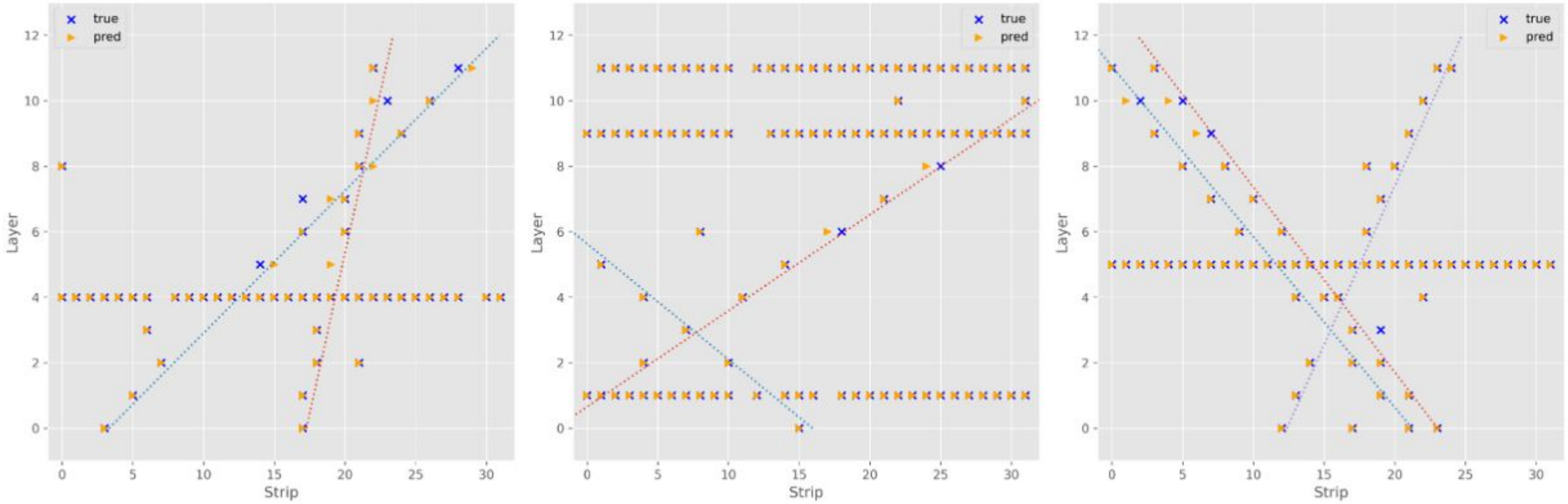




BENCHMARKING

- Compression ratio similar to theoretical expectations
- Similarity index: more than 80% show 90% similarity

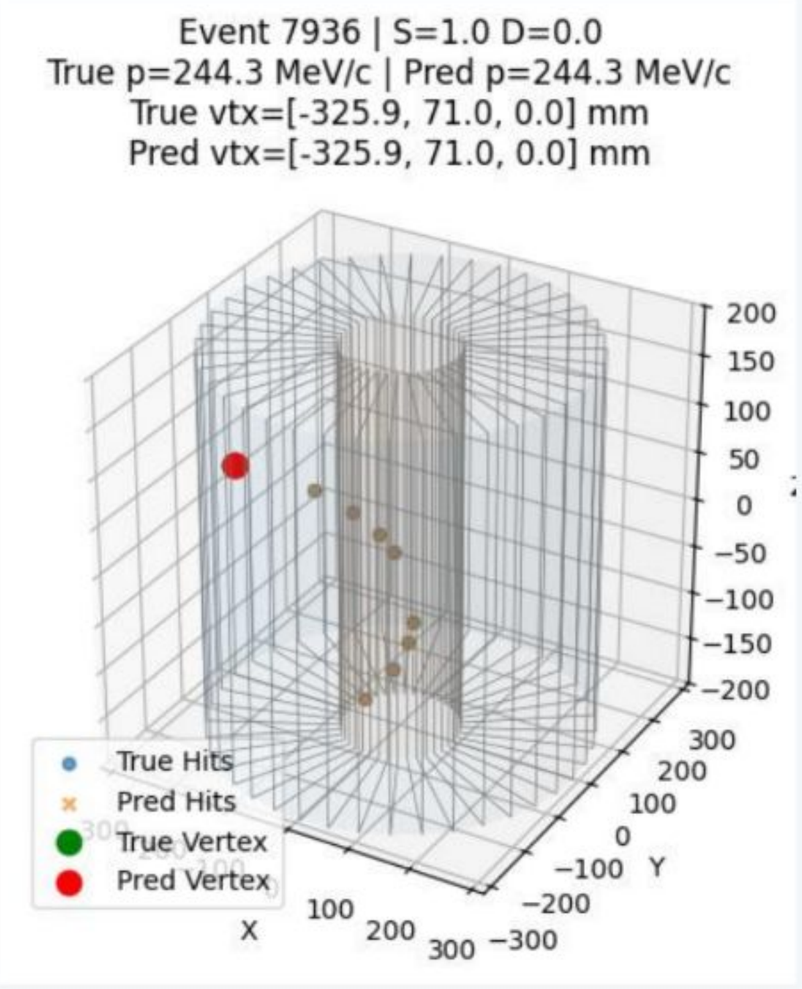
MULTI-TRACK EVENTS

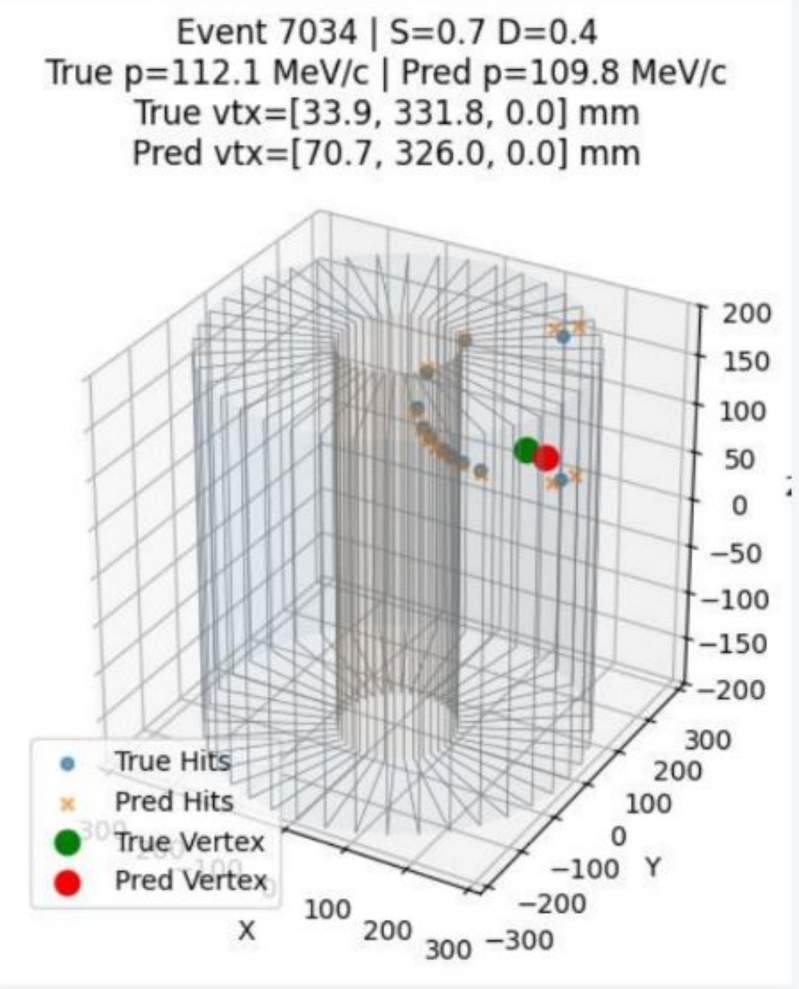


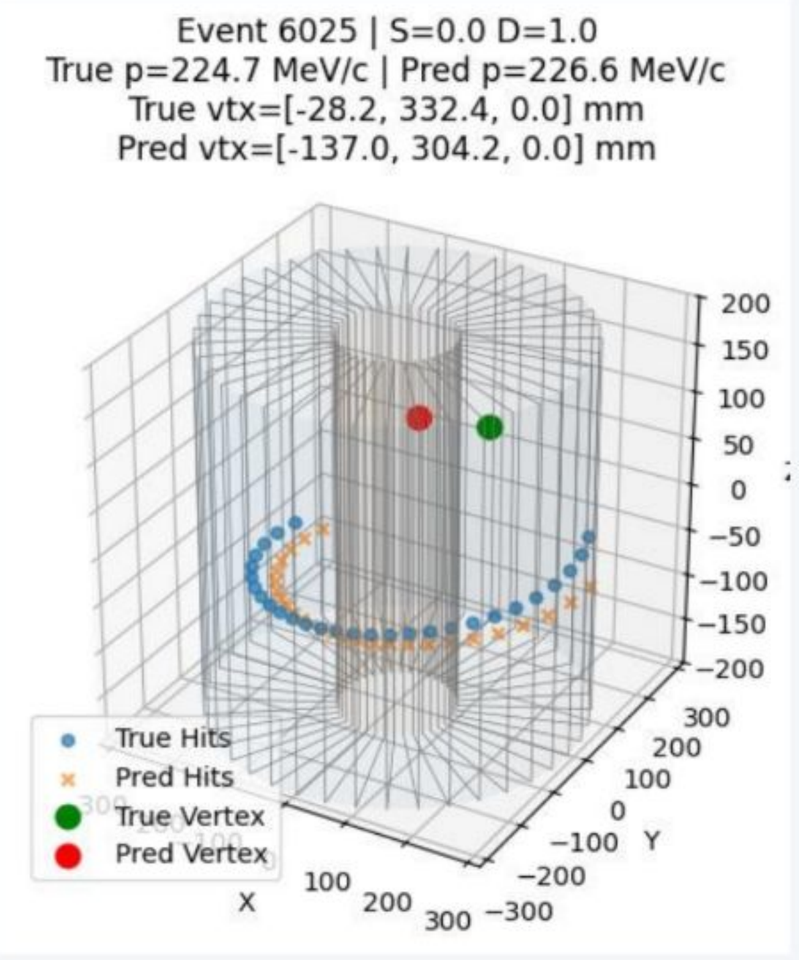
LLMs could perfectly identify multiple tracks even under noisy conditions with reasonable accuracy in the track parameters.

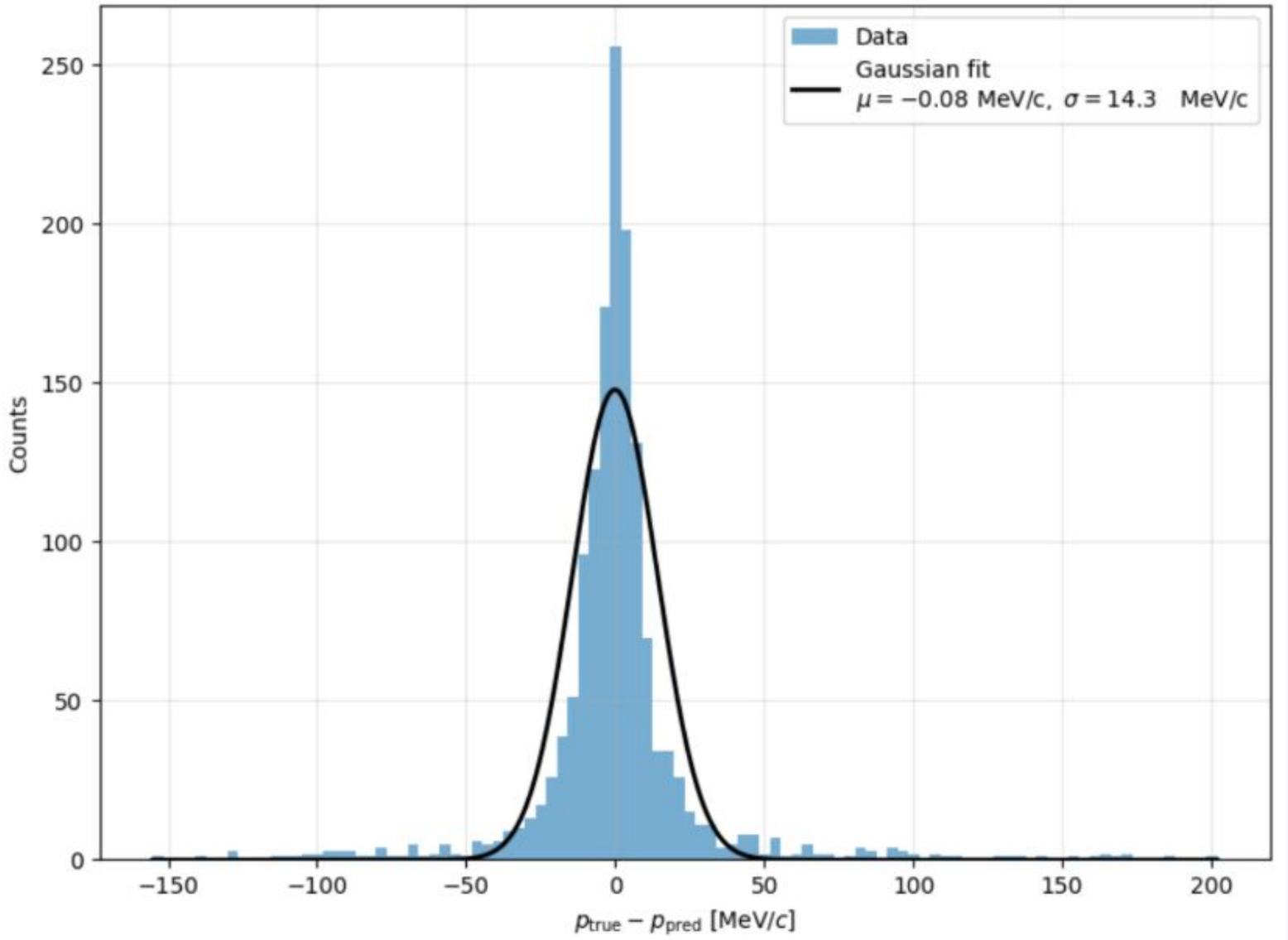
Work under progress for benchmarking performance in real cosmic data.

Sample Event Plots – Synthetic Test Dataset









Key Statistics

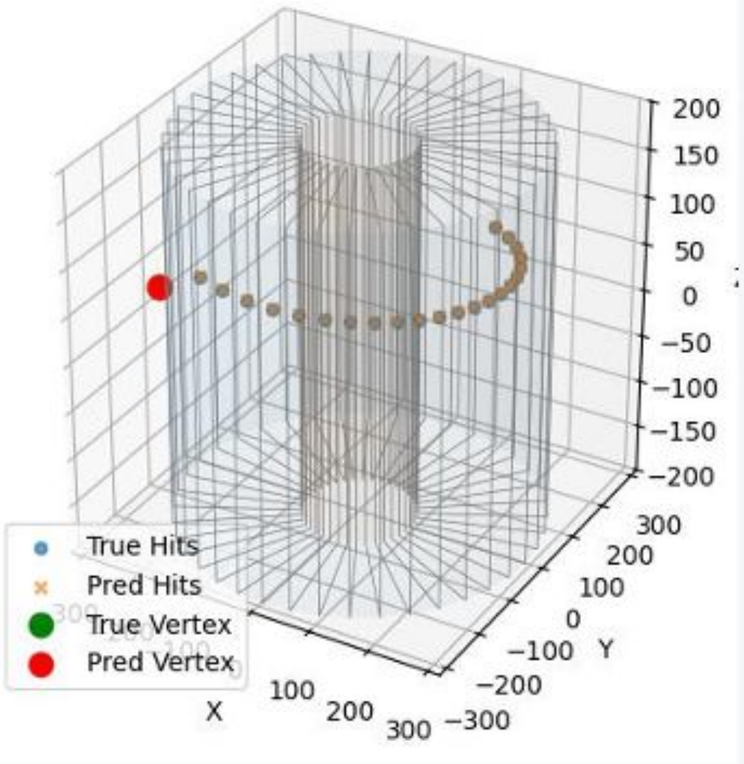
μ = −0.08 MeV/c

σ = 14.3 MeV/c

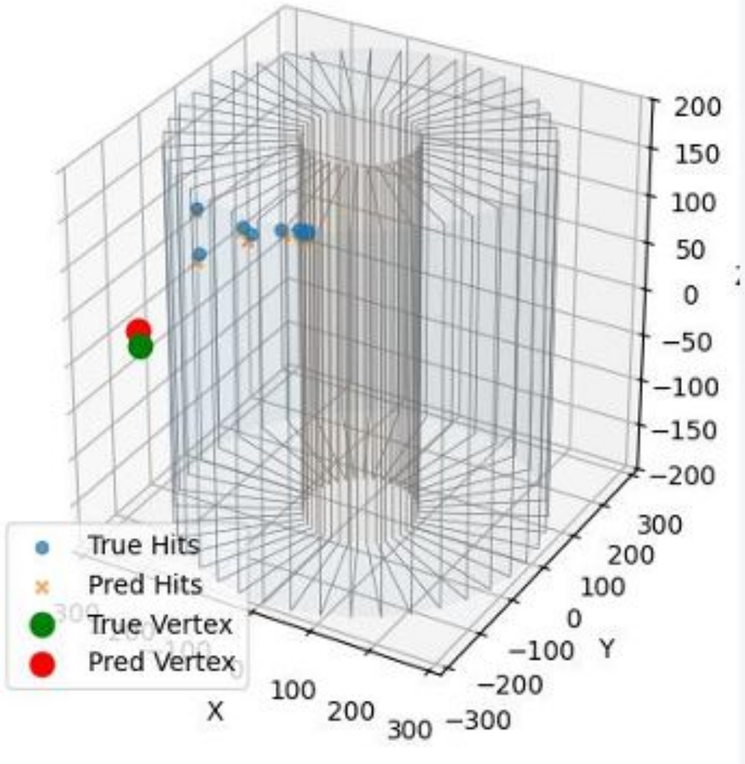
Improved resolution
(14.3 vs 31.6 MeV/c)
after Geant4 fine-tuning

Sample Event Plots – G4 Test Dataset

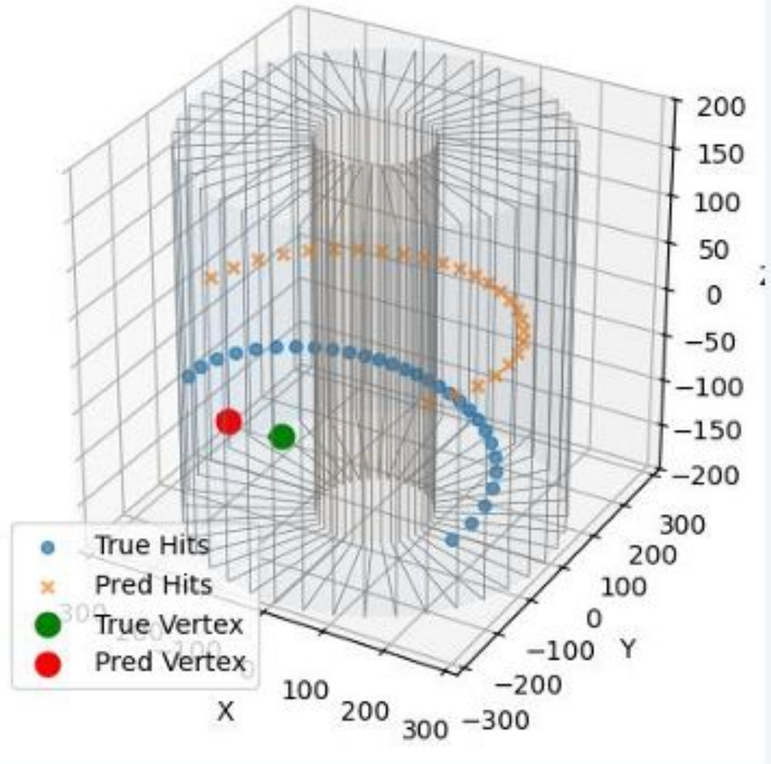
Event 20273 | S=1.0 D=0.0
True p=251.1 MeV/c | Pred p=251.2 MeV/c
True vtx=[-331.1, -40.6, 0.0] mm
Pred vtx=[-331.1, -40.6, 0.0] mm



Event 1293 | S=0.5 D=0.3
True p=118.1 MeV/c | Pred p=120.5 MeV/c
True vtx=[-261.1, -207.6, 0.0] mm
Pred vtx=[-284.1, -174.8, 0.0] mm



Event 619 | S=0.0 D=1.0
True p=206.5 MeV/c | Pred p=239.1 MeV/c
True vtx=[41.5, -331.0, 0.0] mm
Pred vtx=[-45.9, -330.4, 0.0] mm



PITFALLS & OPEN QUESTIONS

The tokenizer fights the numbers

t5-small's default tokenizer treats decimals as text, splitting them arbitrarily. A custom numerical tokenizer is in development.

The model is experiment-specific

It learned this detector's language. Experiment-agnostic training is an open research question.

Real data is the true test

All results are simulation. Benchmarking on real cosmic data from the prototype is under way.

Conclusions

LLM-based track reconstruction:

Data compression, for free

The event collapses to one function call — compression ratios close to theoretical expectations.

Multi-track by construction

One call per track. Early results: multiple tracks identified even under noisy conditions.

Loss on the physics itself

Training optimises hit-pattern agreement directly

A reasoning tool

Track, noise and inefficiency are explicit, readable arguments — the prediction explains itself.

Future work

Real cosmic data

Benchmarking on prototype data in progress

Custom tokenizer

Numerical tokenisation to replace t5 defaults

Complex geometries

Magnetic fields & pileup — early results promising

Future

J-PARC muon g-2/EDM

Full simulations

Electron-Ion Collider

CNN particle ID · dRICH simulation ·