BERKELEY LAB
Bringing Science Solutions to the World

Leveraging Language Models for Particle Tracking

Xiangyang Ju, Yash Melkani

ACAT 2024, Stony Brook University

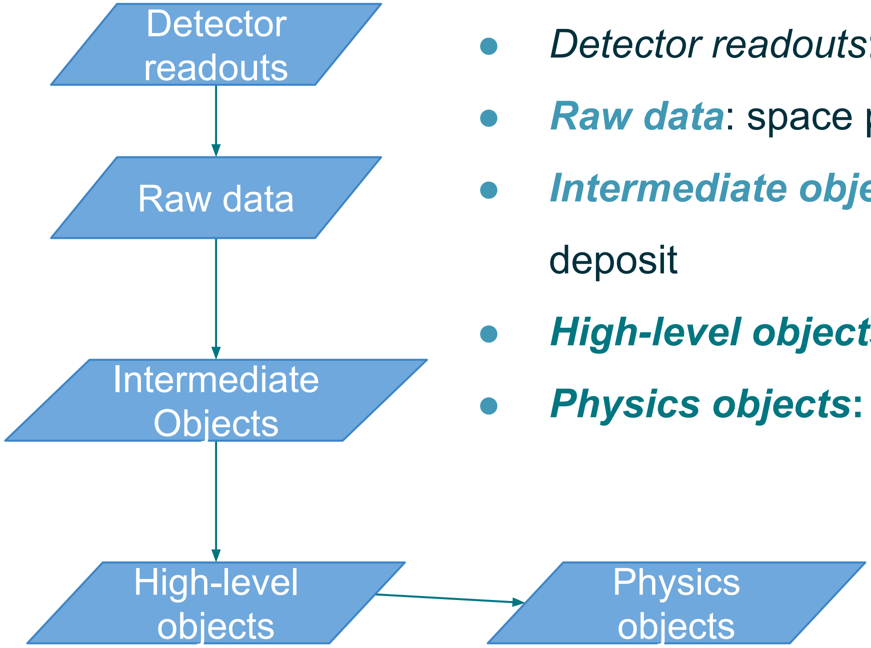
12 March, 2024

ACAT 2024
22nd International Workshop on Advanced Computing Analysis Techniques in Physics Research

Introduction

- Language models have revolutionized the machine understanding on natural languages.
    - However, they do not understand scientific data.
- Scientific data are multidimensional, continuous/discrete measurements.
    - Cannot apply LLMs directly on these data
- Reconstructing particles from the raw HEP detector data is fundamental for all physics analysis.
- We aims to leverage language models to embed detector data into a latent space that can be useful for particle reconstruction, opening new avenues for understanding detector languages.

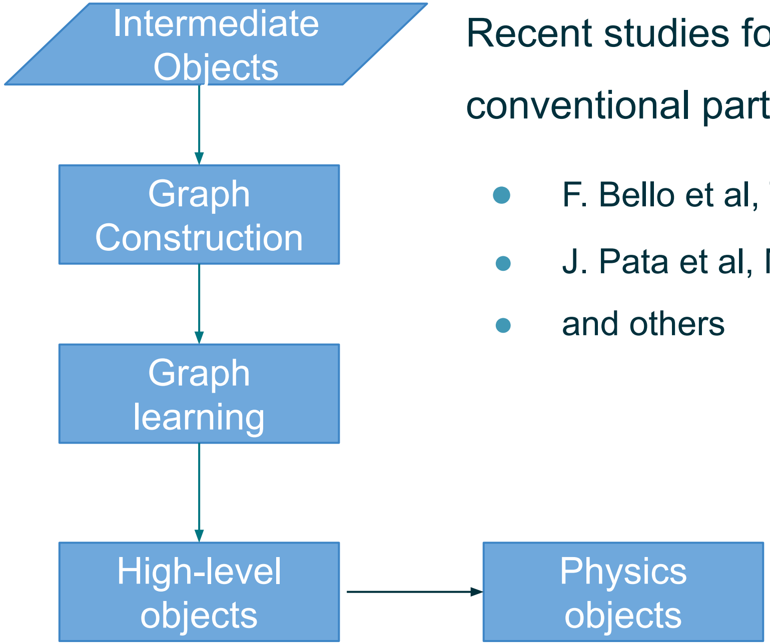
Previous work: hierarchical approach



- *Detector readouts*: analog or digital signals from the detector
- ***Raw data***: space points ID, energies in cells
- ***Intermediate objects***: particle trajectories and particle energy deposit
- ***High-level objects***: electron, muon, photon, jets, tau-lepton, et al
- ***Physics objects*****:** Higgs boson, W/Z bosons, et al

- Most reconstruction algorithms are sequential. Each level only accesses to its immediate predecessor objects.
- *Particle Flow* algorithm is global for high-level object reconstruction.
- See CMS particle flow algorithm in arXiv:1706.04965.

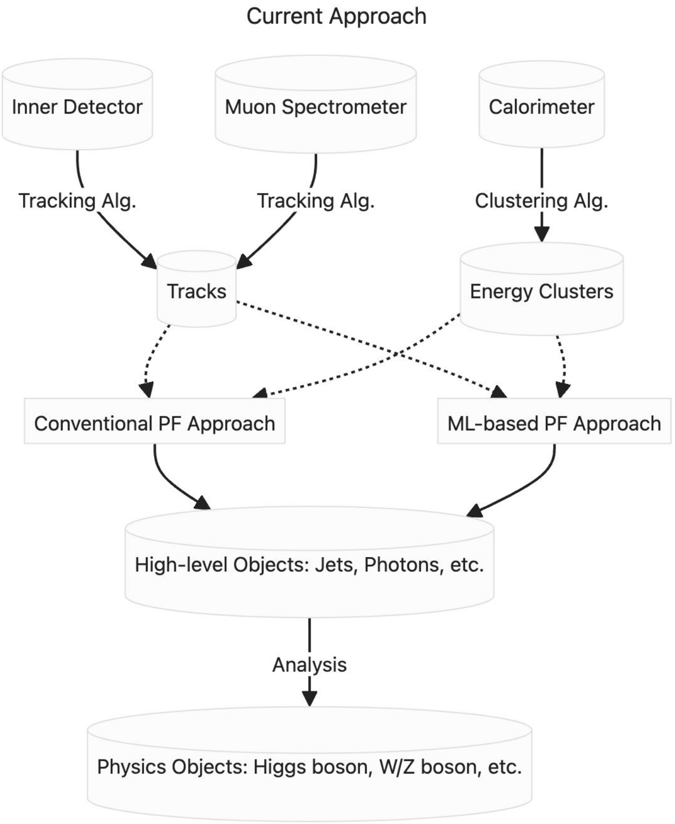
Previous work: ML approach

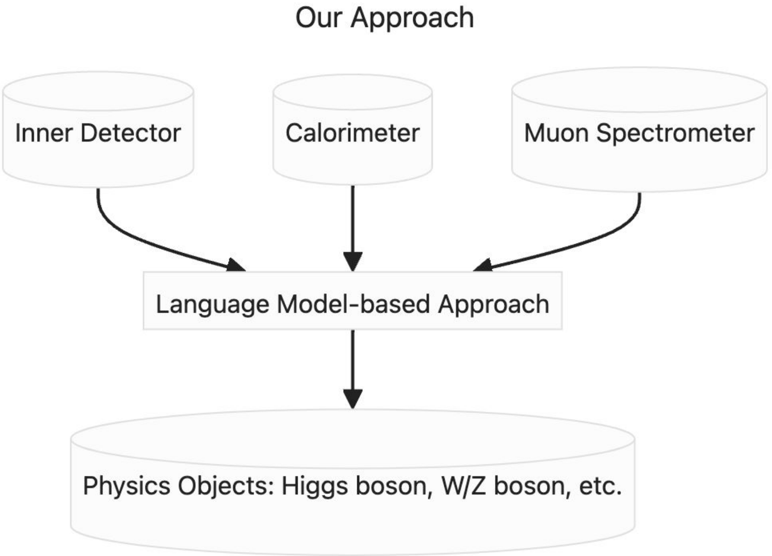


Recent studies focus on using Machine Learning Models to replace conventional particle flow algorithms.

- F. Bello et al, Towards a Computer Vision Particle Flow, arXiv:2003.08863
- J. Pata et al, MLPF: Efficient ML particle flow with GNN, arXiv:2101.08578
- and others

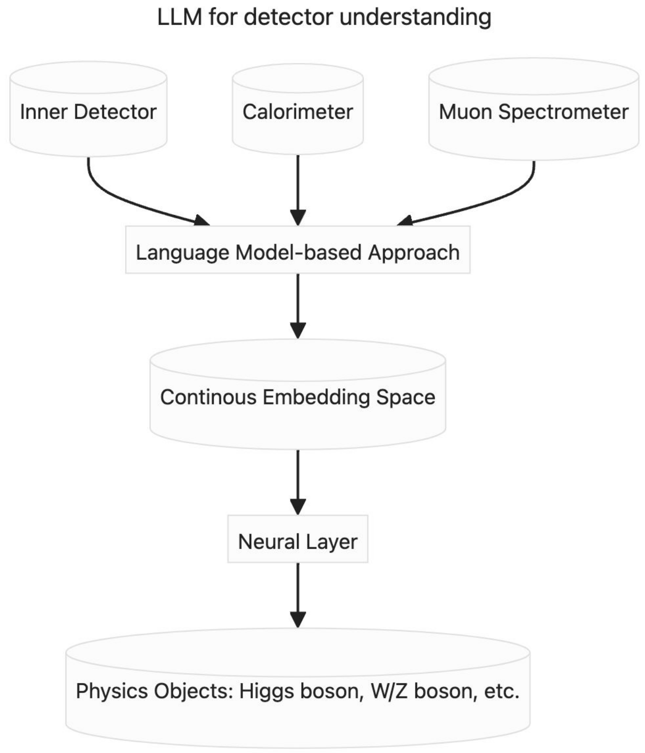
ML models improve physics performance, reduce computational requirements, and are suitable for using GPUs.





Our proposal is to train *a language model* for reconstructing physics objects with raw detector data.

LLM for detector data understanding



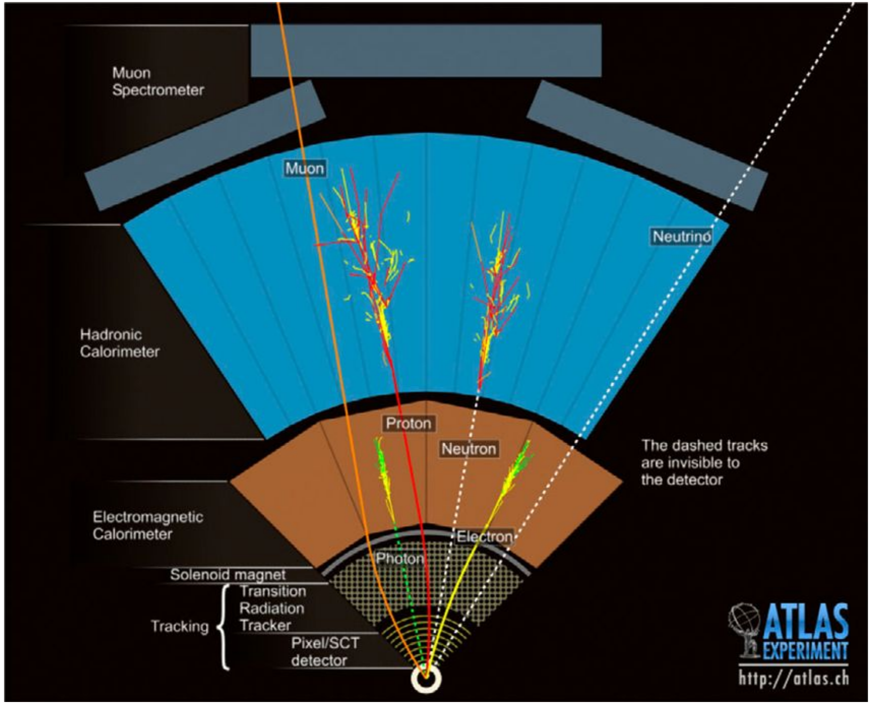
The core idea is to learn a continuous embedding space that can then be adapted or fine tuned to new problems.

Often use self-supervised learning on surrogate tasks, including Masked Data Modeling, contrastive learning, and meta learning.

- Masked Particle Modeling on Sets
- Z. Zhao Self supervised learning on jet tagging

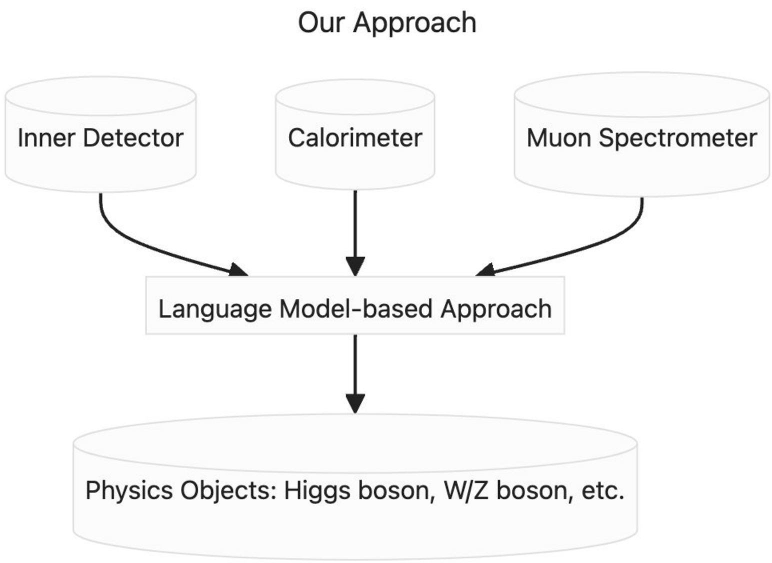
HEP detector vs NLP

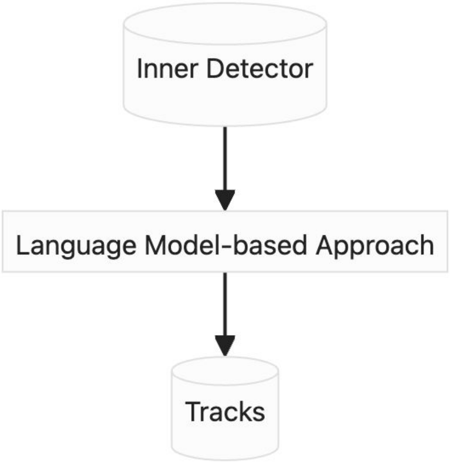
Analogy between HEP and NLP	
Detector elements	Words
All detector elements	Vocabulary
Particle trajectories or showers	Sentences
Collision Events	Paragraphs
Events from the same physics process	Sections



Language Model for Tracking

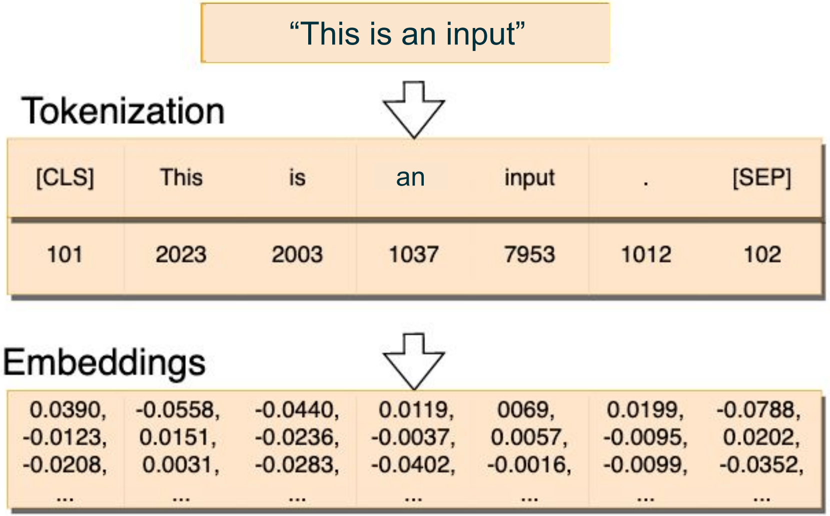
A first step





Sentence vs Tracks

Tracks are represented by a list of detector modules



[(x, y, z), …, (x, y, z)]

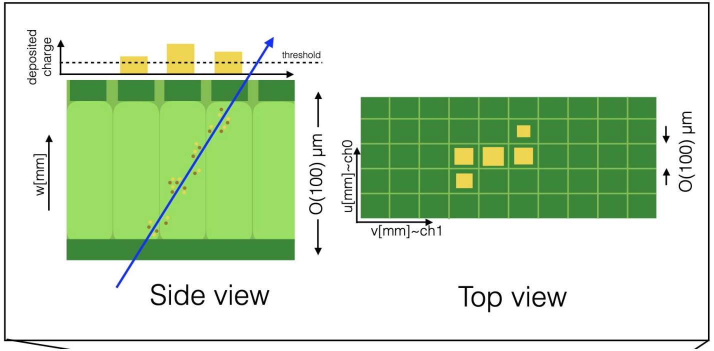
[CLS] (x, y, z) … (x, y, z) [SEP]
2 UMID … UMID 3

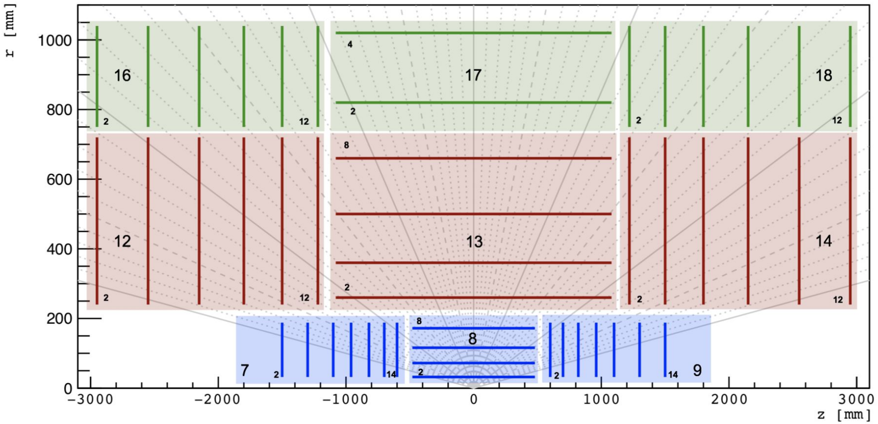
Embeddings

Input data

Use the TrackML dataset, and tokenize all *detector modules*.

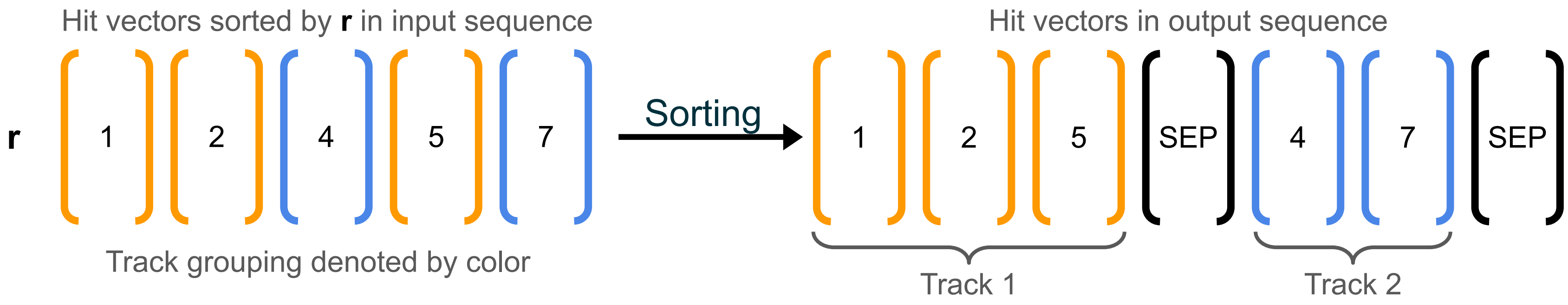
Schematic of detector modules





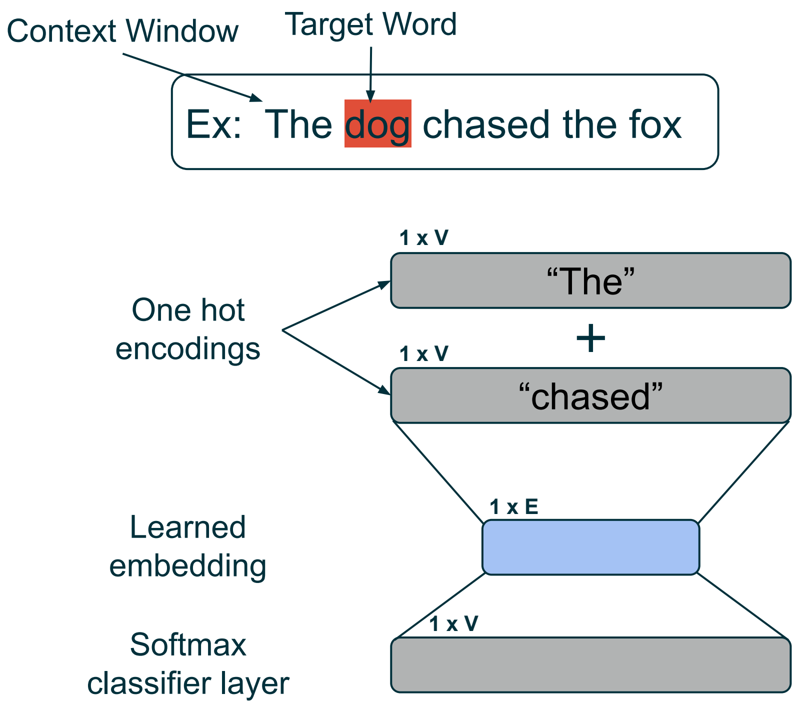
Total 18737 detector modules in the TrackML dataset. We use data from volume 8, 13, and 17, in which there are 14000 modules.

TrackSorting



- Inputs are a list of space points (represented by their associated detector modules) sorted by their distances from the collision point.
- Outputs are track candidates. *SEP* is a special token that separates tracks.
- As a starting point, we asks the model to sort detector modules from two true tracks.
  - In reality, there are ~10,000 tracks produced by HL-LHC.

Word2vec



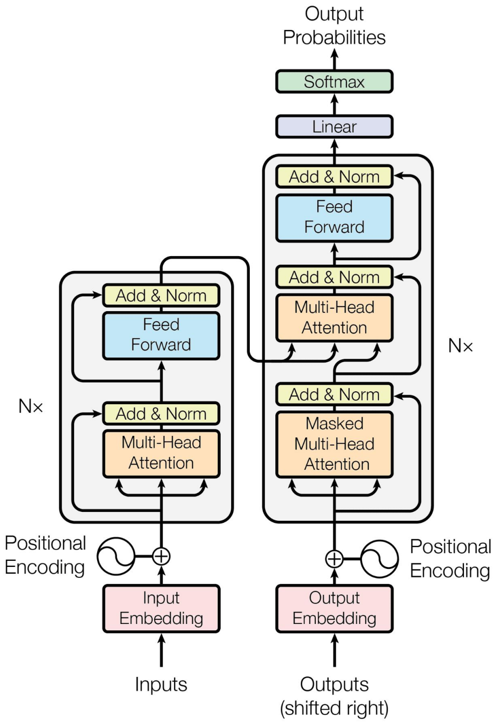
- Word2vec (arXiv:1301.3781) is used to create embedding vectors for each token in a vocabulary given a "text corpus" (a large set of sentences), especially, the continuous bag-of-words model.
- In final embedding space, words used in similar contexts are close together
  - "Dog" and "Cat" are more similar than "Dog" and "Bridge"
- We could use the TrackBERT model to embed the detector module in the future.

E - Embedding dimension

V - Size of Vocabulary (ex: number of words in English language)

Transformer Model

- Only a single attention head
- 6 encoding followed by 6 decoding layers
- The feed forward layers has dimension 256
- The output dimension is 14000 + 2
  - (number of modules + SOS and SEP)
- 1.6 M training parameters.



Track finding and evaluation

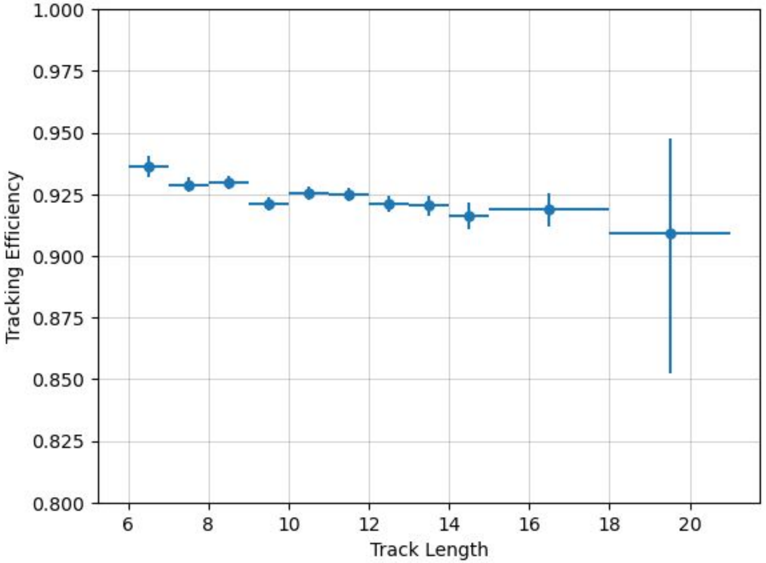
The procedure for reconstructing tracks

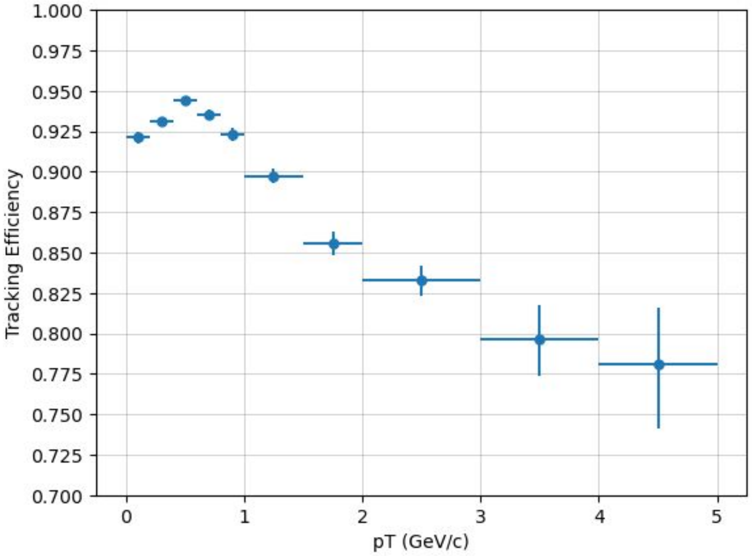
- Model predicts a probability distribution of the next module
- Choose the next module with the highest probability such that it
  - exists in the input sequence
  - has not been used in the output sequence
- Stop once all modules in the input sequence are used in the output sequence.

Matching criteria for calculating tracking efficiency. If 75% of a reconstructed track matches to a true particle, that particle is considered as identified.

Tracking Performance

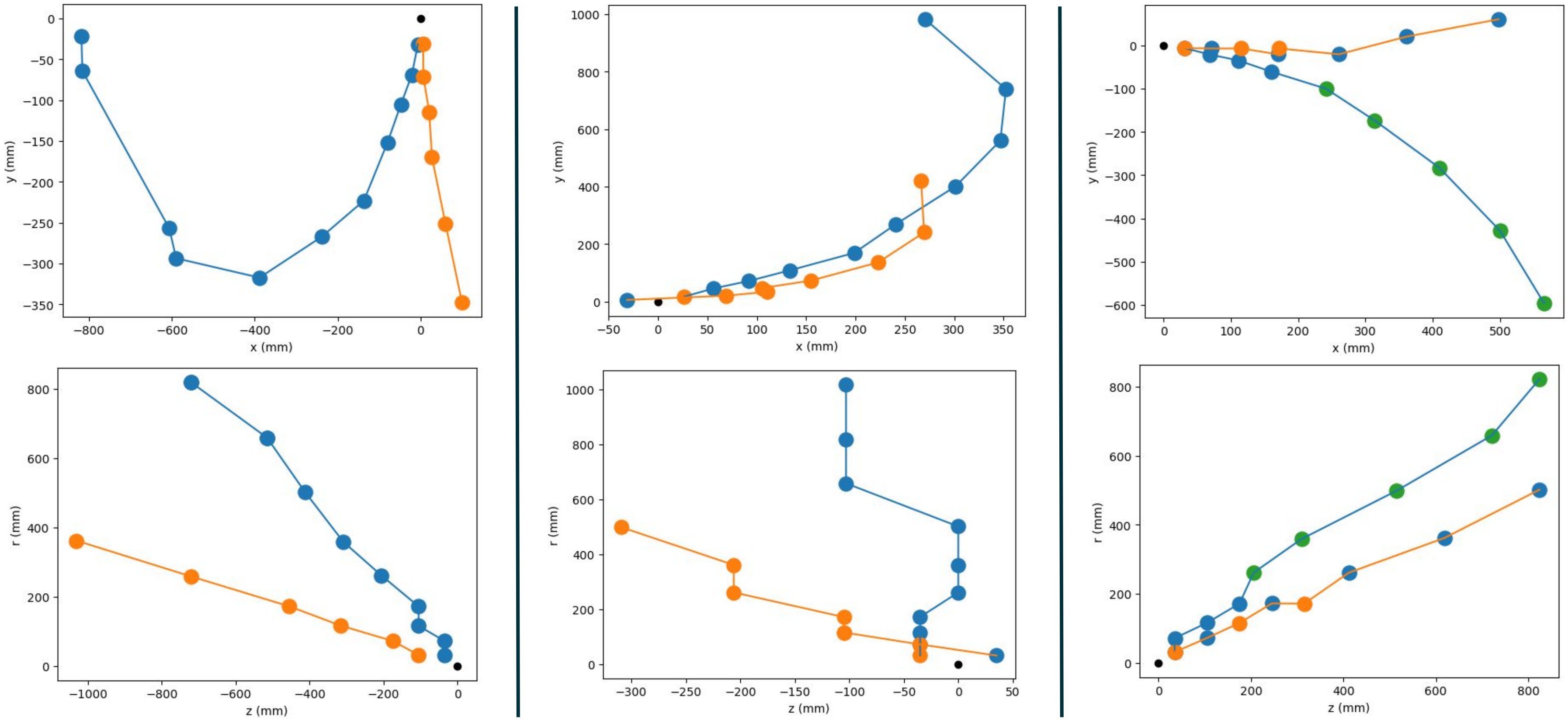
Only used data from barrel region, no noise hits, at least 6 hits per track





- Good performance for low-pT particles, but not so in high-pT.
- It is robust against the track lengths.

Visual Results



Conclusion and Outlook

- With the tokenized detector elements, we explored different approaches to leverage language models for particle tracking.
    - BERT for encoding detector modules
    - TrackSorting for regional track finding
- The *TrackingSorting* algorithm achieves good track finding performance on a dummy data.
    - Test on realistic data
- It would be interesting to teach language models others physics. E.g. particle interactions with detectors:
    - Input sequence: tokenized particle information (like codebook in arXiv:2401.13537)
    - Output sequence: a list of detector data