AI and Machine Learning Across Experimental Nuclear Physics

Yeonju Go

Stony Brook University / Brookhaven National Laboratory

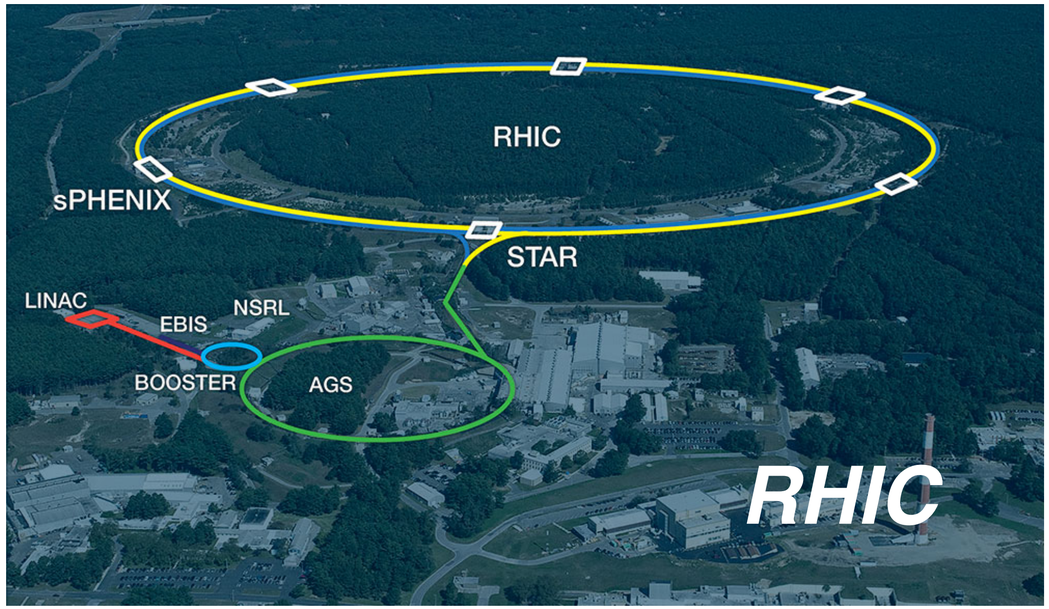
International Conference on Meson-Nucleon Physics
and the Structure of the Nucleon (MENU 2026)
Stony Brook University, NY, USA
Oct. 19 - 23, 2026

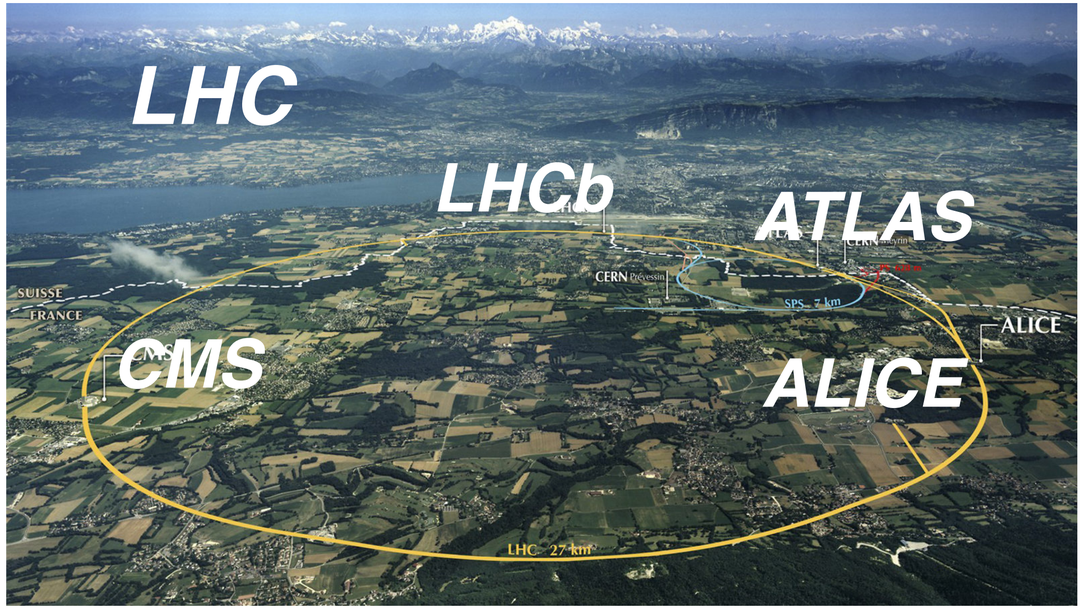
Stony Brook University

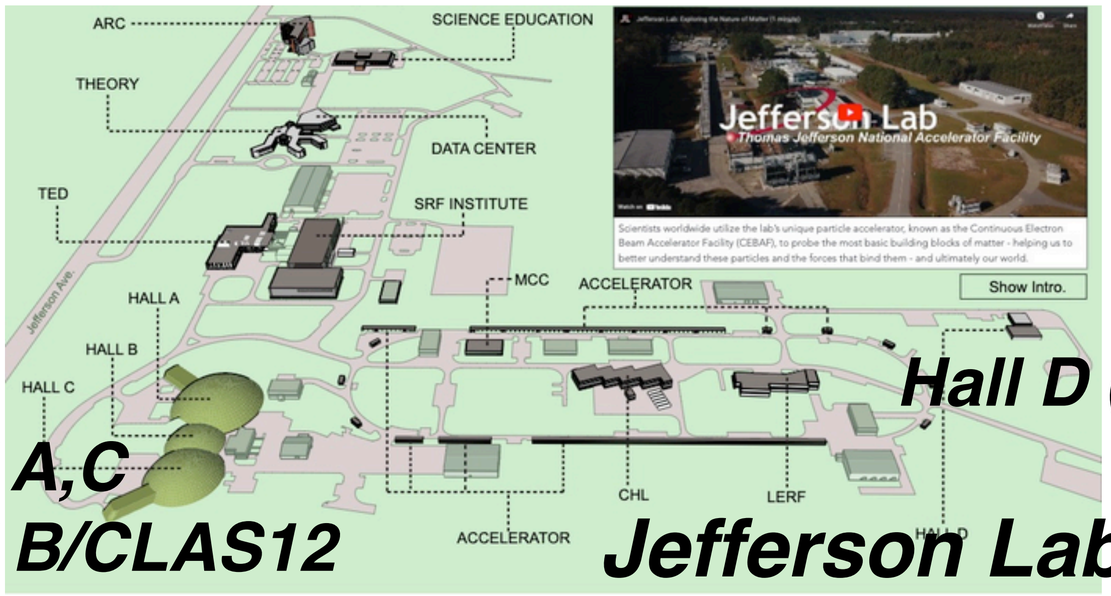
U.S. DEPARTMENT OF ENERGY

Brookhaven National Laboratory

Facilities in the QCD division





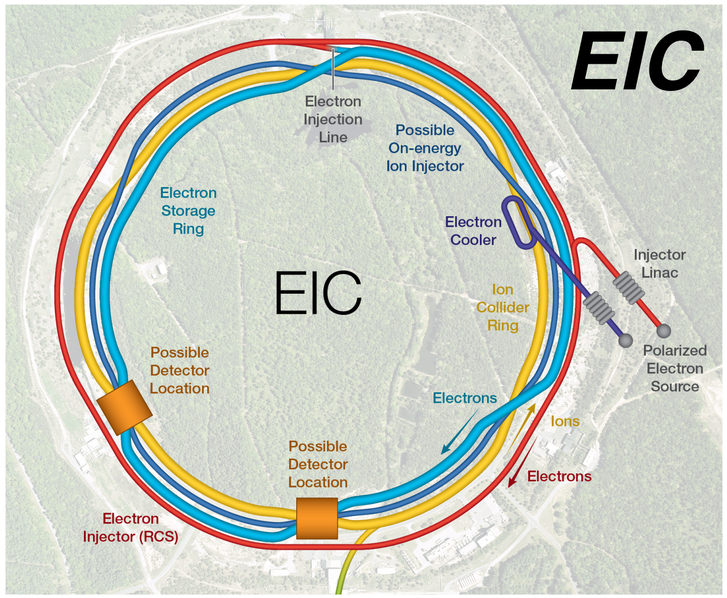


Hall A,C

Hall B/CLAS12

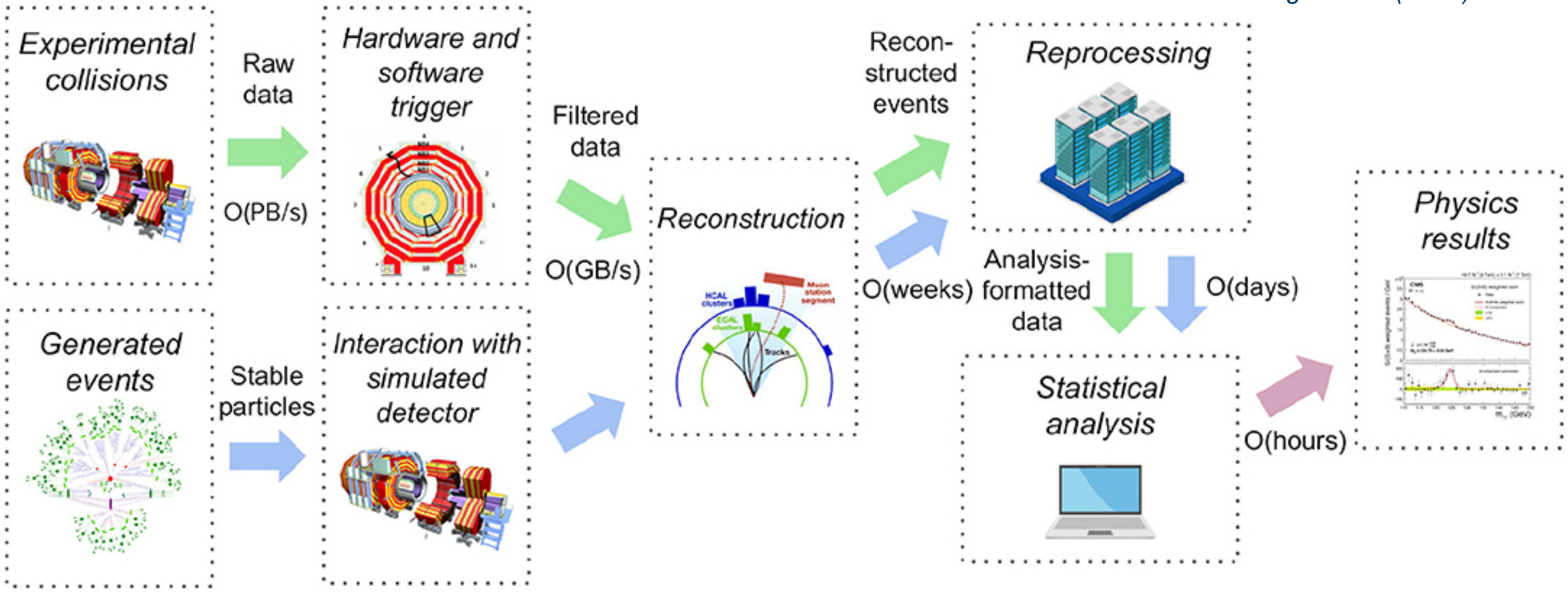
Hall D (GlueX)

Jefferson Lab



(High Energy) Nuclear Physics Analysis Workflows

Front.Big Data 4 (2021) 661501



- Each stage of analysis requires substantial human effort, computing resources, and time

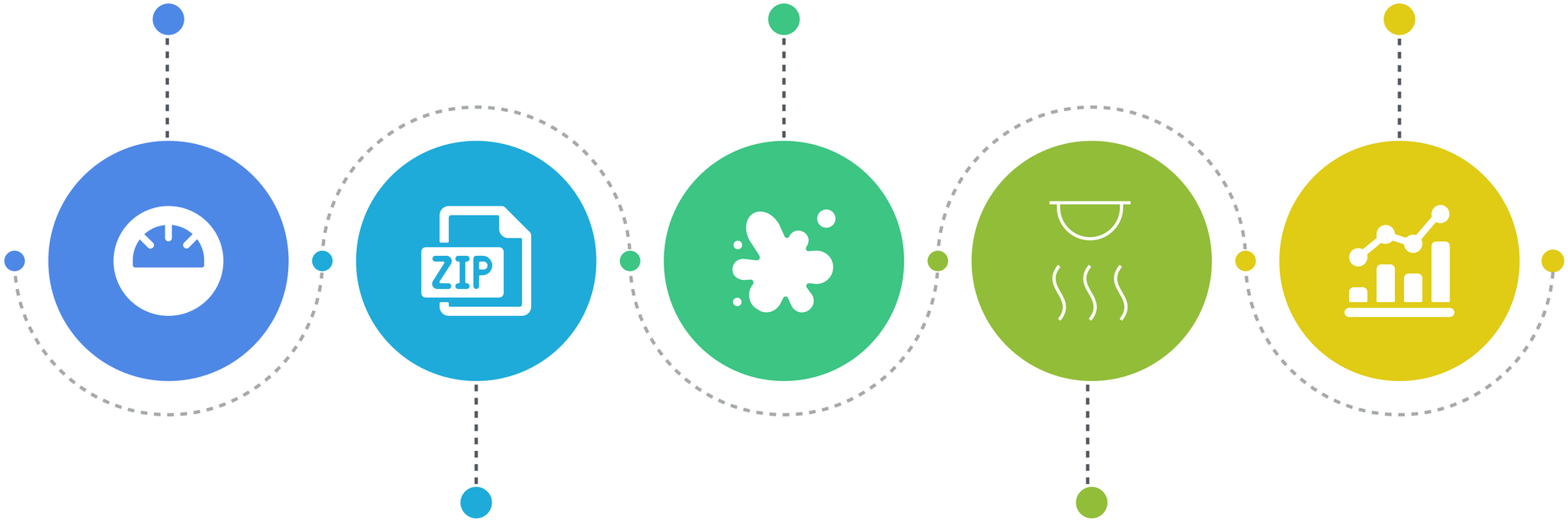
→ ***Leverage AI/ML to accelerate, automate, and improve these processes***

AI/ML Applications in the QCD Field

Detector Control, Calibration

Event Selection, Triggering

Reconstruction, Analysis

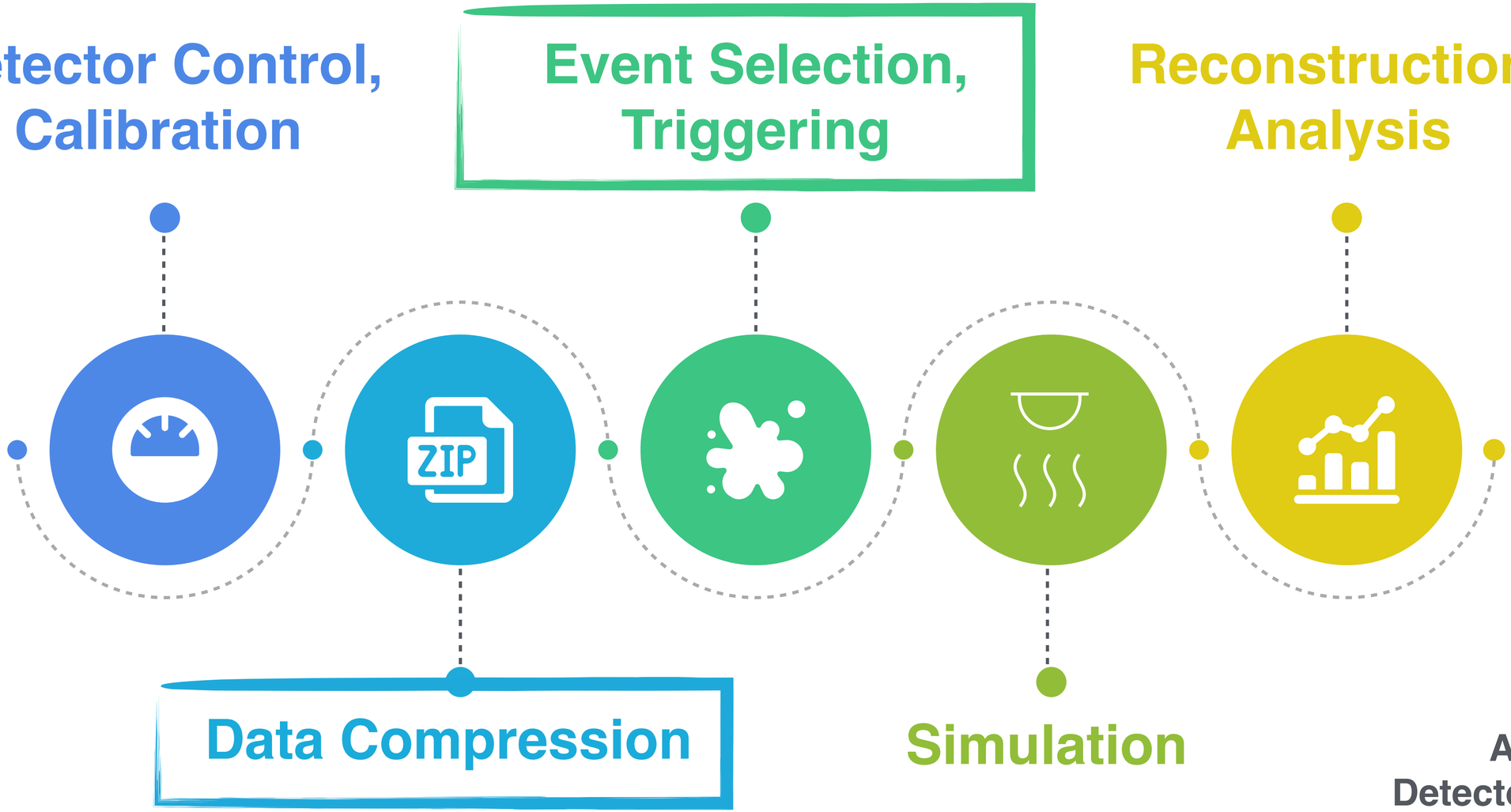


Data Compression

Simulation

Also,
Detector Design
Anomaly Detection
...

AI/ML Applications in the QCD Field

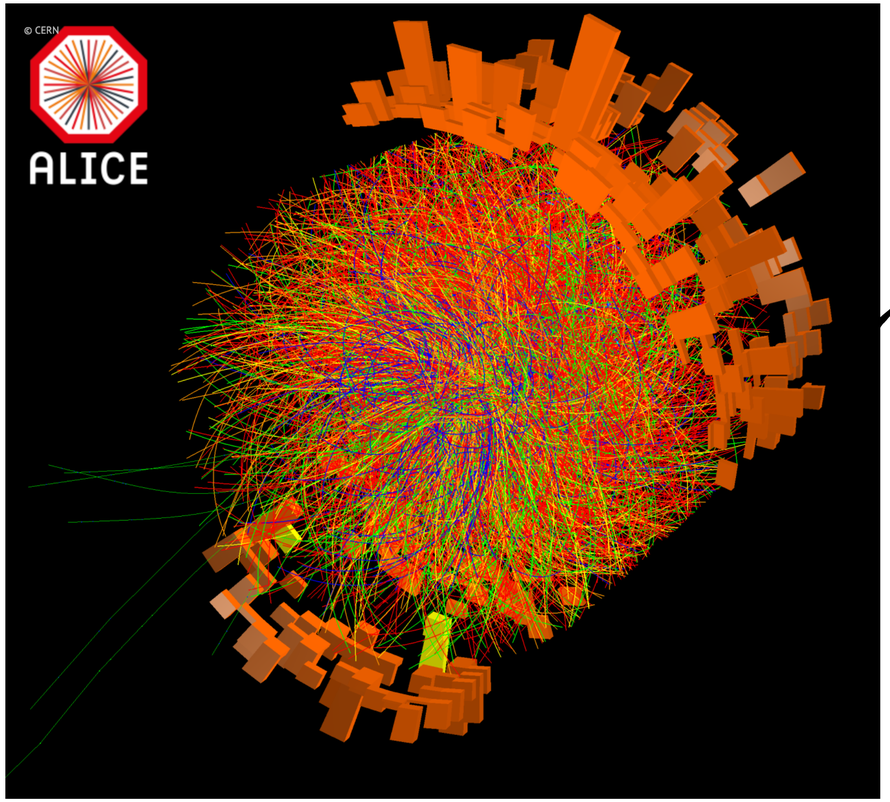


Also,
Detector Design
Anomaly Detection
...

High Energy Collision Data Taking

Collision rates ~80kHz

Raw data: O(PB/s) → 10^{15} bytes



Triggering → traditional way

Is it interesting event? → if yes, save!

(Real-time filtering)

Implement AI/ML for fast and efficient triggering !

Fast Event Selection on FPGA

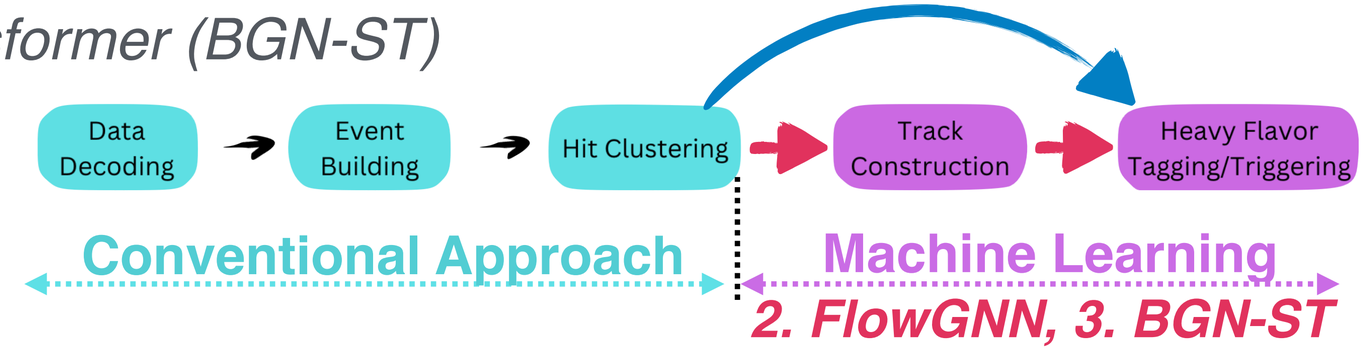
See talk by Ming Liu (Thu. 9:50am)

- Tagging heavy flavour events (rare) by ***implementing ML on FPGA for online triggering***

Three ML approaches
1. Multi Layer Perceptron (MLP) with hls4ml
2. FlowGNN
3. Bipartite Graph Network with Set Transformer (BGN-ST)

JINST 19 (2024) C02066

Other similar works
: GNN tracking on FPGA (Front.Big Data 5 (2022) 828666, arXiv:2112.02048)
: Anomaly detection using decision trees on FPGA (Nature Communications15, 3527 (2024))



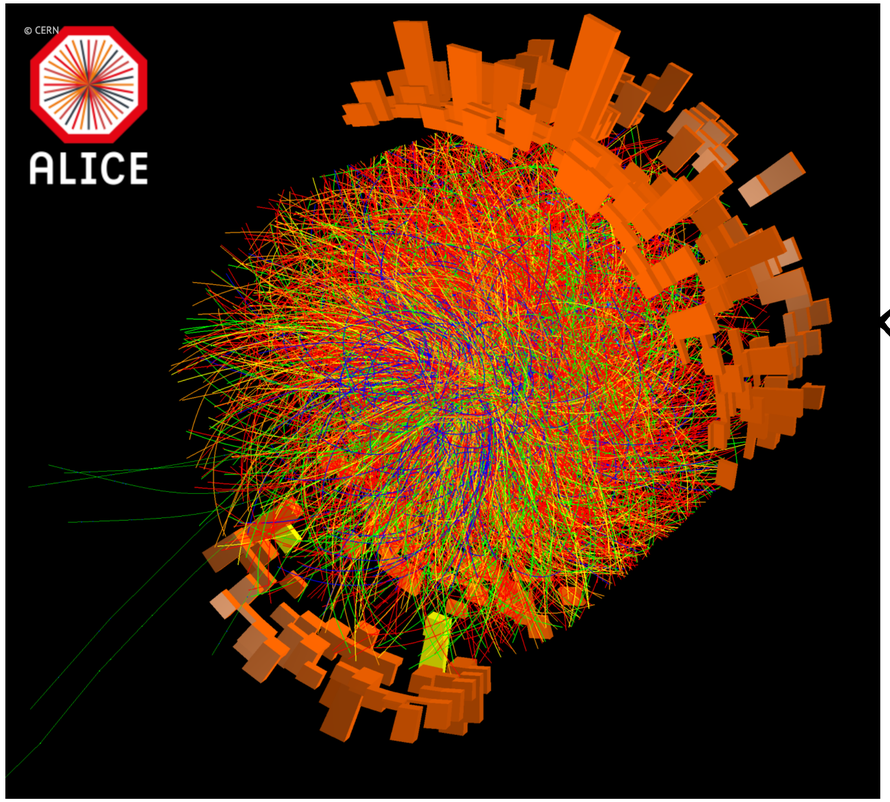
	MLP (with HLS4ML)	BGN-ST
Accuracy (Triggering on Beauty decays, 0.05% events)	88.5%	97.38%
Latency	Aggregation step (117 ns latency) prediction step (308 ns latency)	7.53 μs per graph (Freq. 180 MHz)

- **Genesis Mission Phase I** "*Agentic AI for Real-time Expedited Discovery from High-Complexity EIC Data Streams*" by PU, FNAL, LANL, MIT, NJIT for EIC beam background rejection with AI in real-time

High Energy Collision Data Taking

Collision rates ~80kHz

Raw data: O(PB/s) → 10^{15} bytes



Triggering → traditional way

Is it interesting event? → if yes, save!

(Real-time filtering)

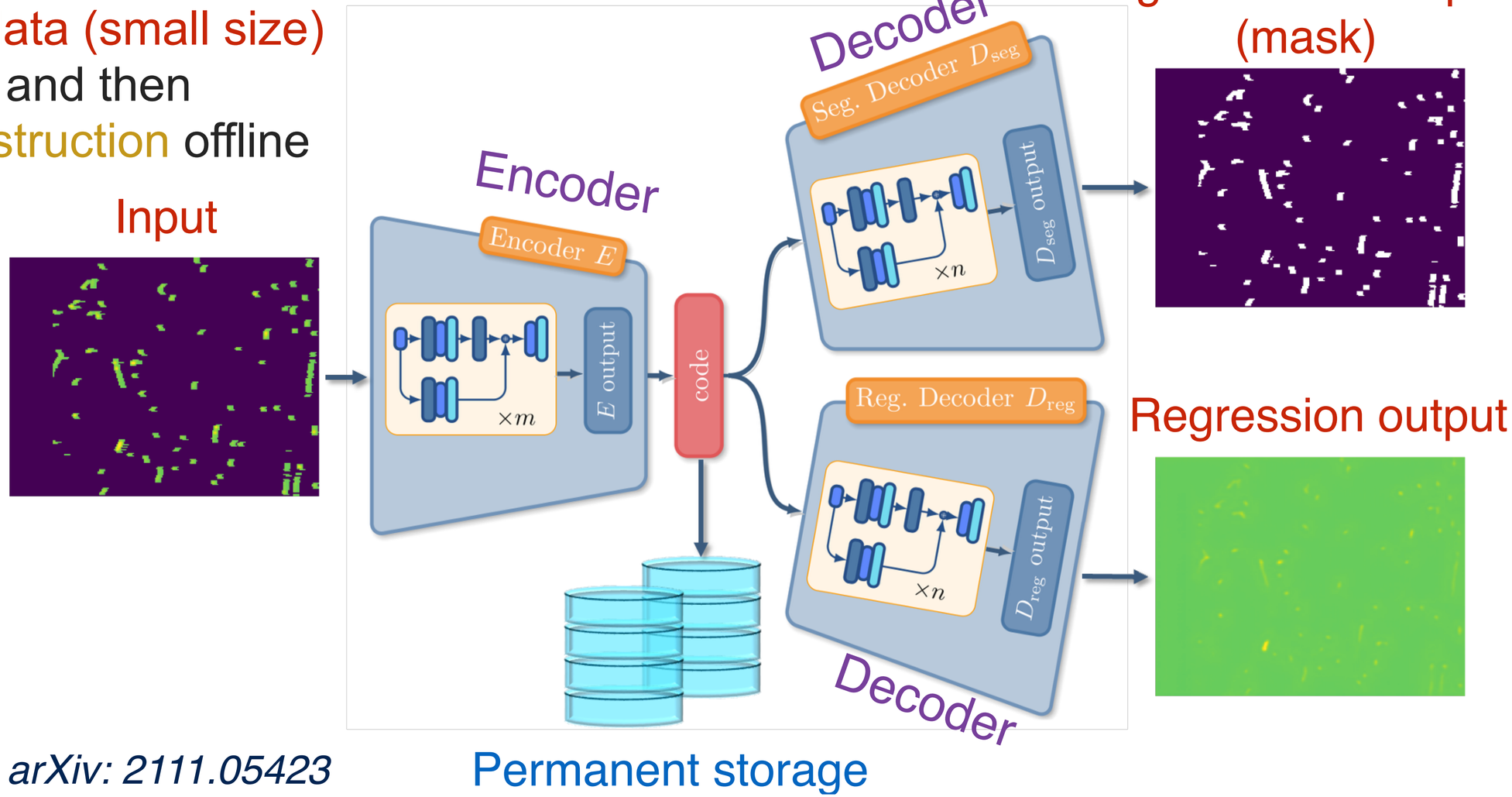
Streaming readout → next generation

Let's save everything by compressing data!

Streaming Readout / Data Compression

- ***BCAE (Bicephalous Convolutional Autoencoder) model***
- **Data Compression**
  - ➡ record encoded data (small size) when taking data and then decode for reconstruction offline

See talk by Yi Huang (Thu. 9:25am)



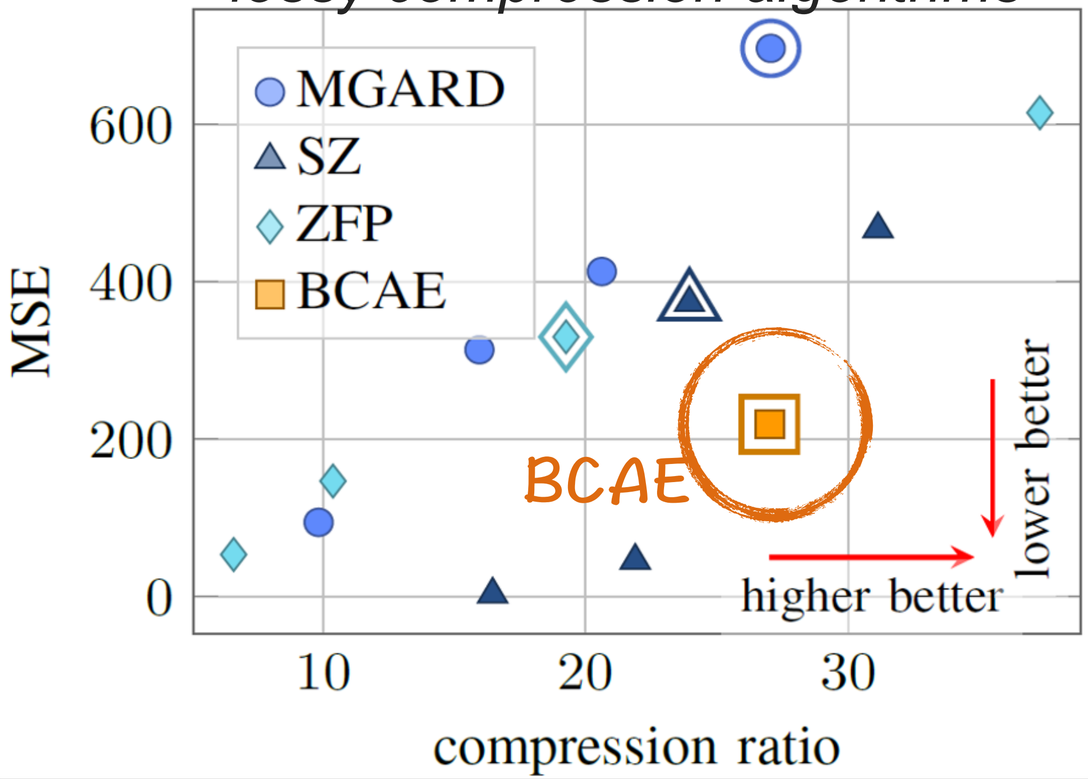
arXiv: 2111.05423

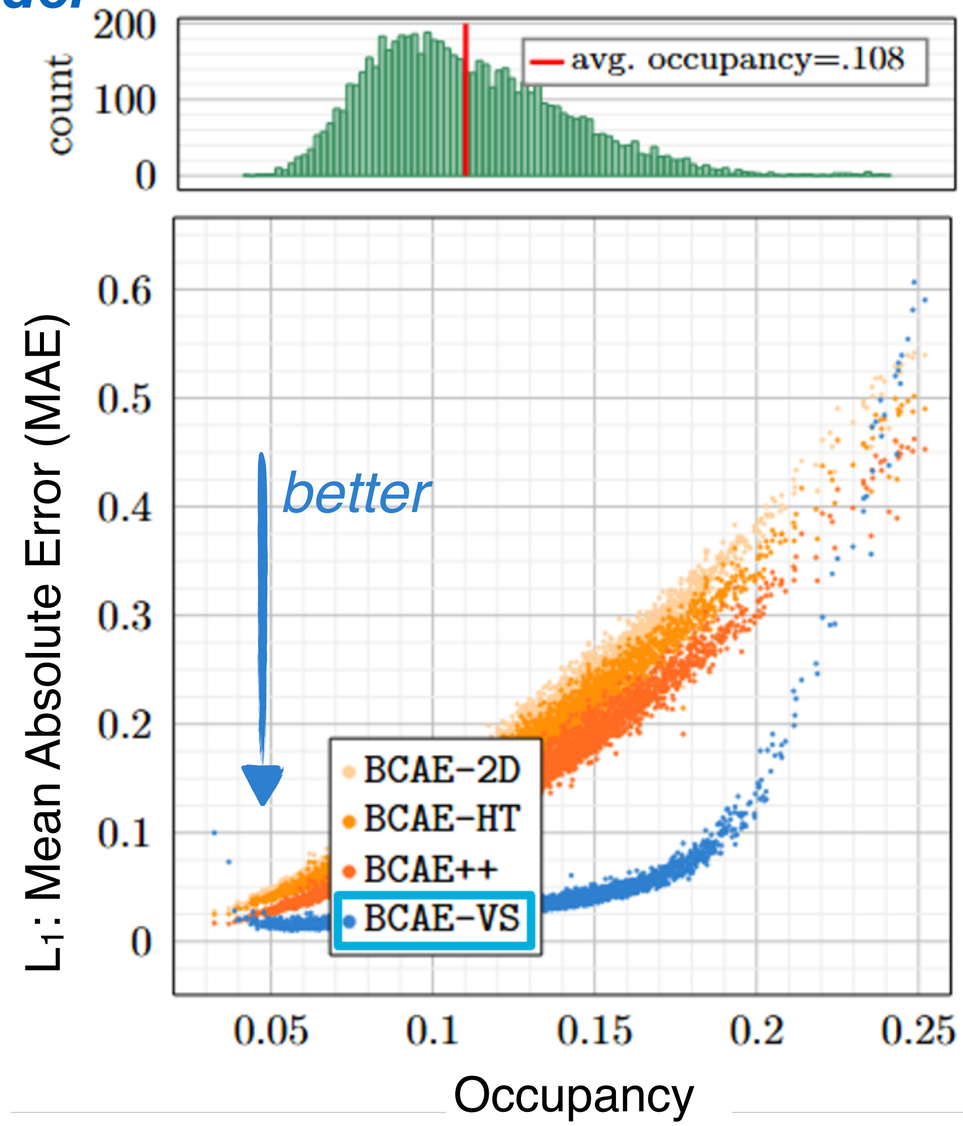
Streaming Readout / Data Compression

arXiv: 2111.05423
arXiv: 2310.15026
arXiv: 2411.11942

- ***BCAE (Bicephalous Convolutional Autoencoder) model***

BCAE outperforms other off-the-shelf lossy compression algorithms



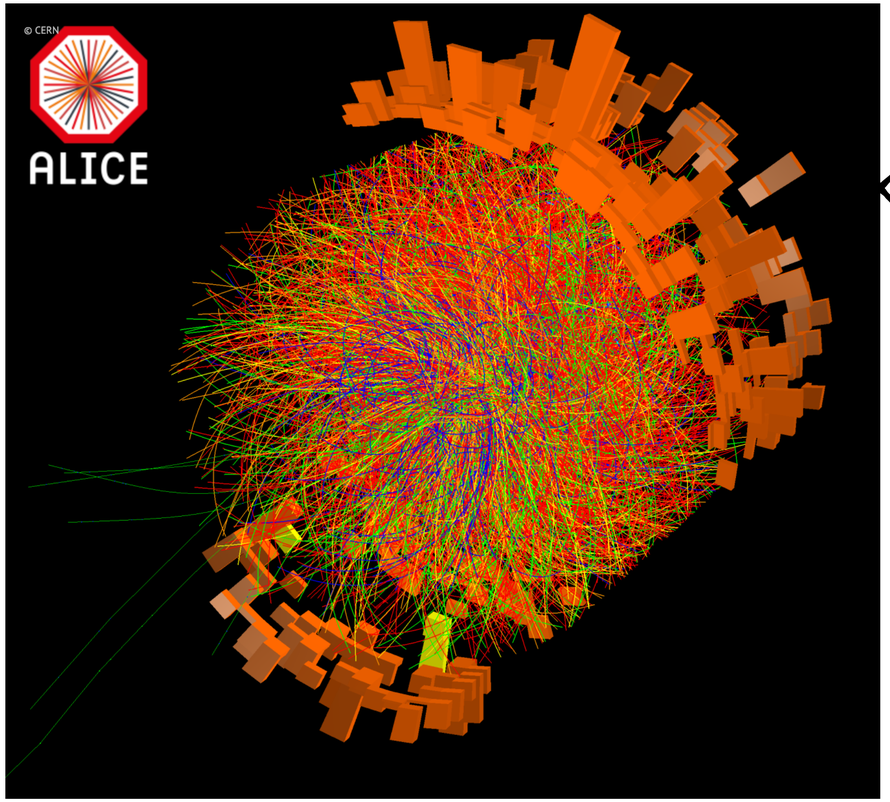


Variations of BCAE to achieve high throughput (faster compression) + high fidelity

High Energy Collision Data Taking

Collision rates ~80kHz

Raw data: O(PB/s) → 10^{15} bytes



Triggering → traditional way

Is it interesting event? → if yes, save!

(Real-time filtering)

Streaming readout → next generation

Let's save everything by compressing data!

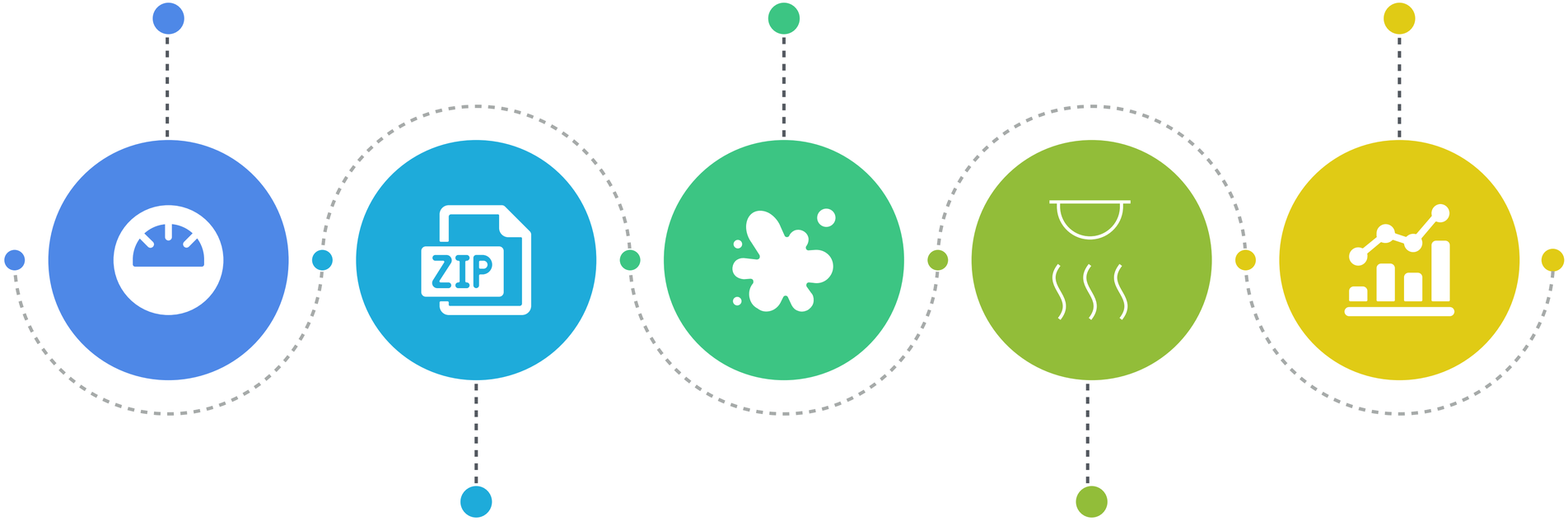
- Other approach:
  - ➡ **Genesis Mission Phase I** "*AI-Driven Petabyte-Scale Streaming Readout*" by CUA, JLab, aiming to develop a stream-native, physics-integrated AI framework; real time data processing and reconstruction

AI/ML Applications in the QCD Field

Detector Control, Calibration

Event Selection, Triggering

Reconstruction, Analysis



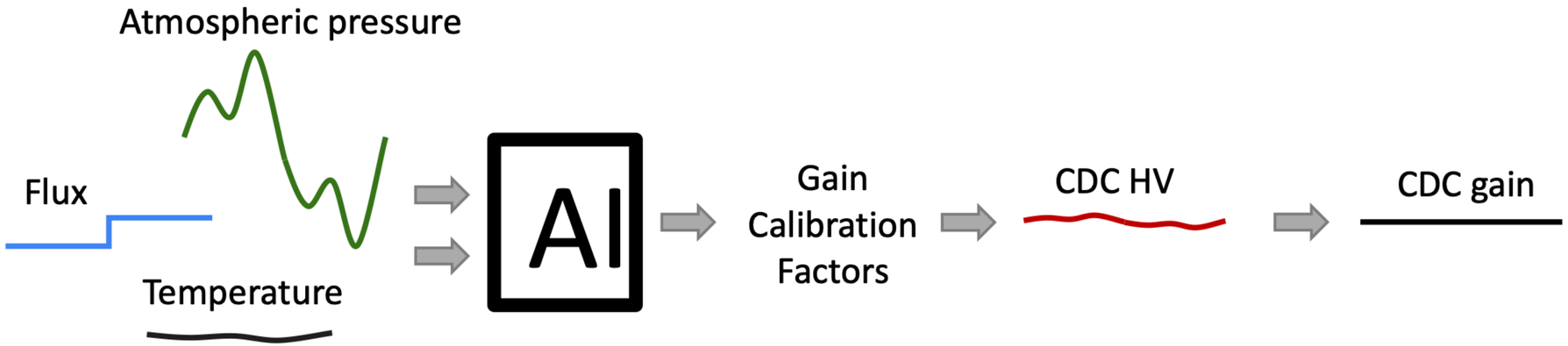
Data Compression

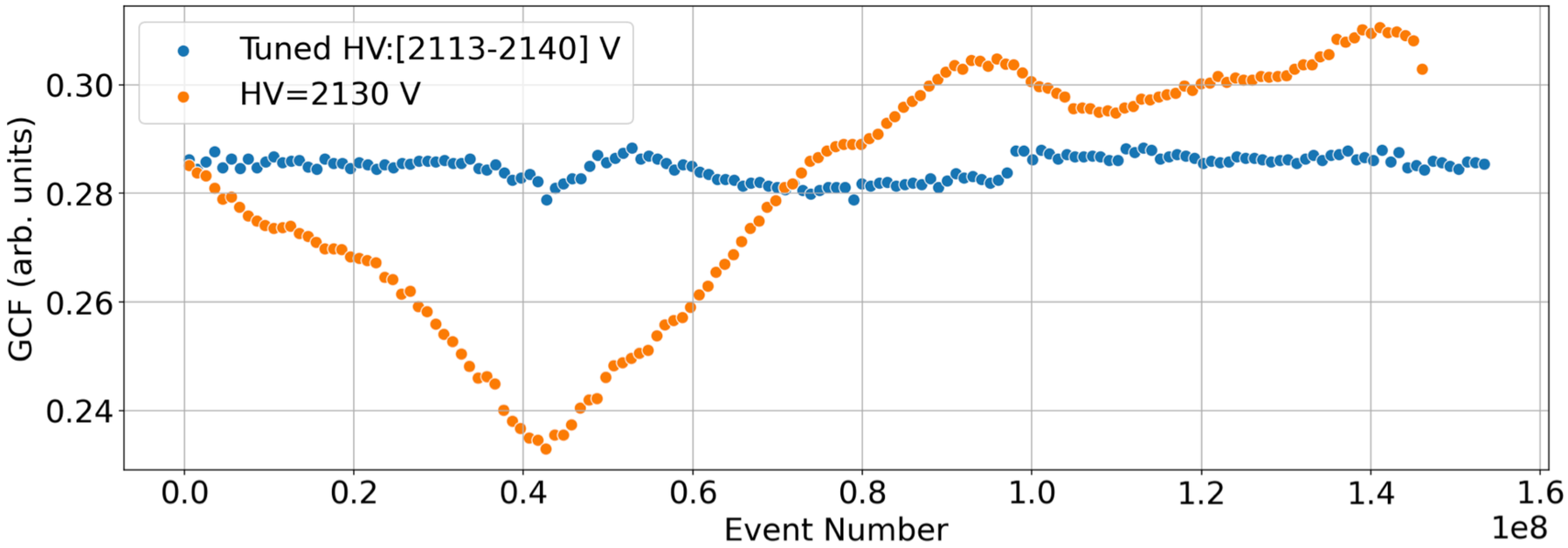
Simulation

Also,
Detector Design
Anomaly Detection
...

Detector Control

- **Automatic gain correction** on detector high voltage using *Gaussian Process Regression (GPR)* ML model
  - ➡ for central drift chamber (CDC) at GlueX



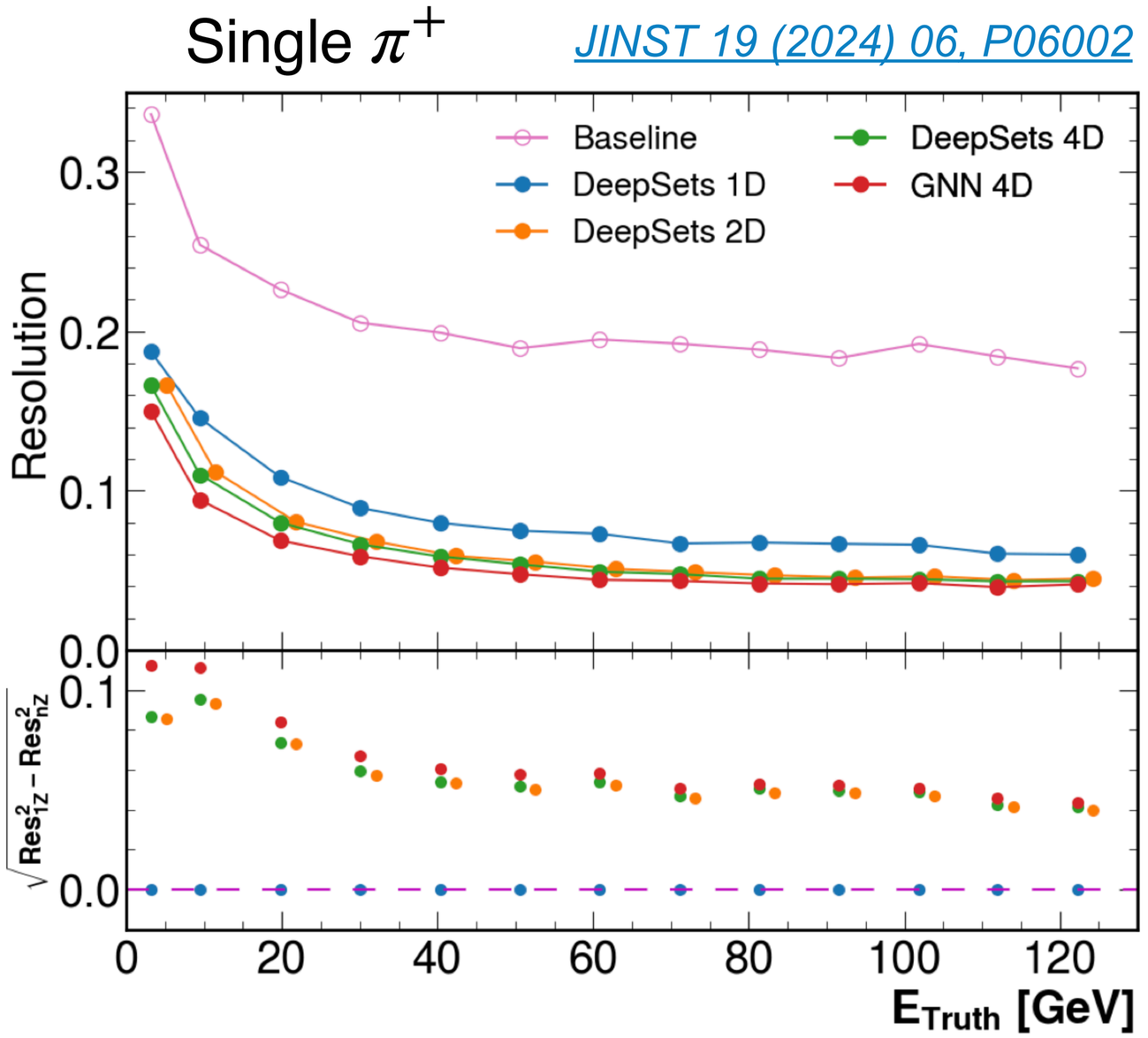


arXiv:2403.13823

Detector Calibration

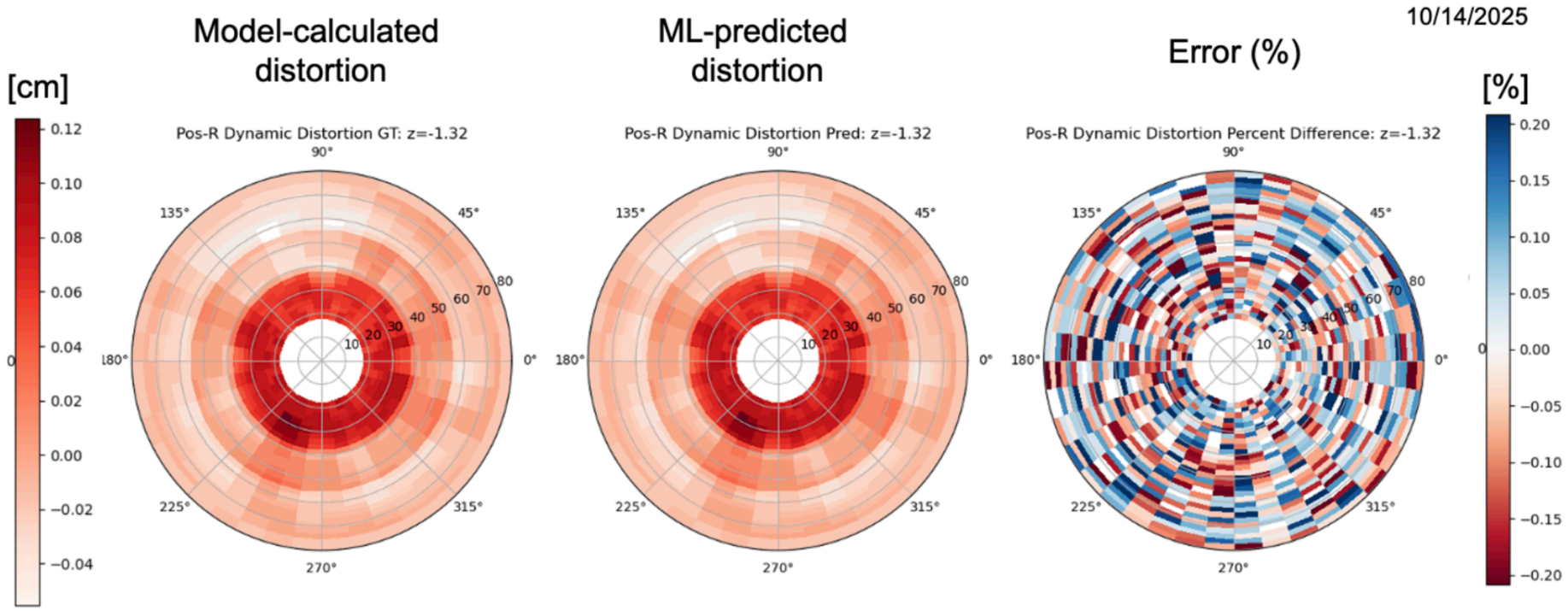
- Regression for energy calibration using DeepSets and Graph Neural Network

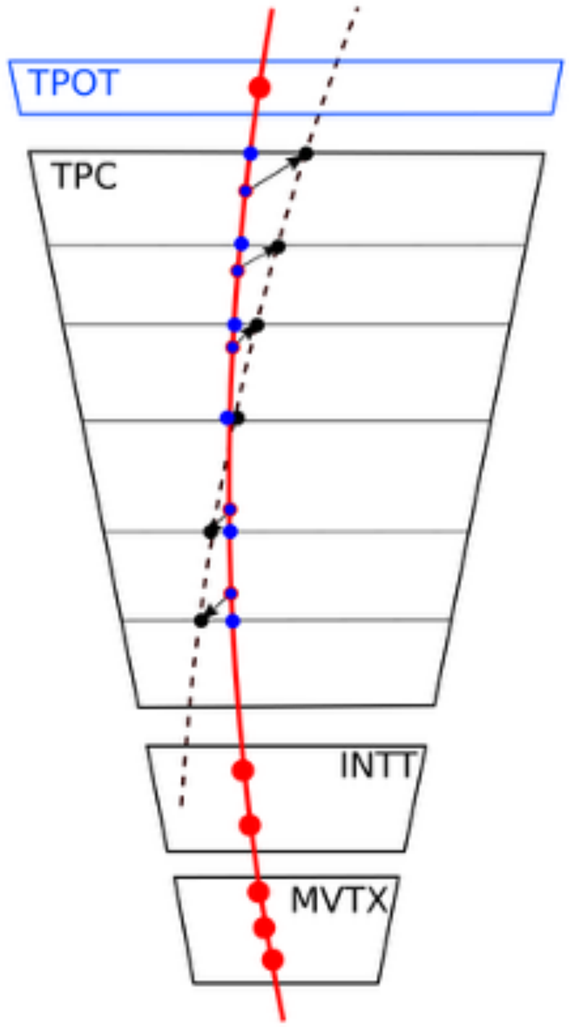
  ➡ ePIC forward calorimeter



Detector Calibration

- Time Projection Chamber Distortion correction using U-Net Convolutional Neural Network (CNN)
  - ➡ Ion back flow (space charge) distort electric field
  - ➡ reduce the significant computational cost of traditional calculation



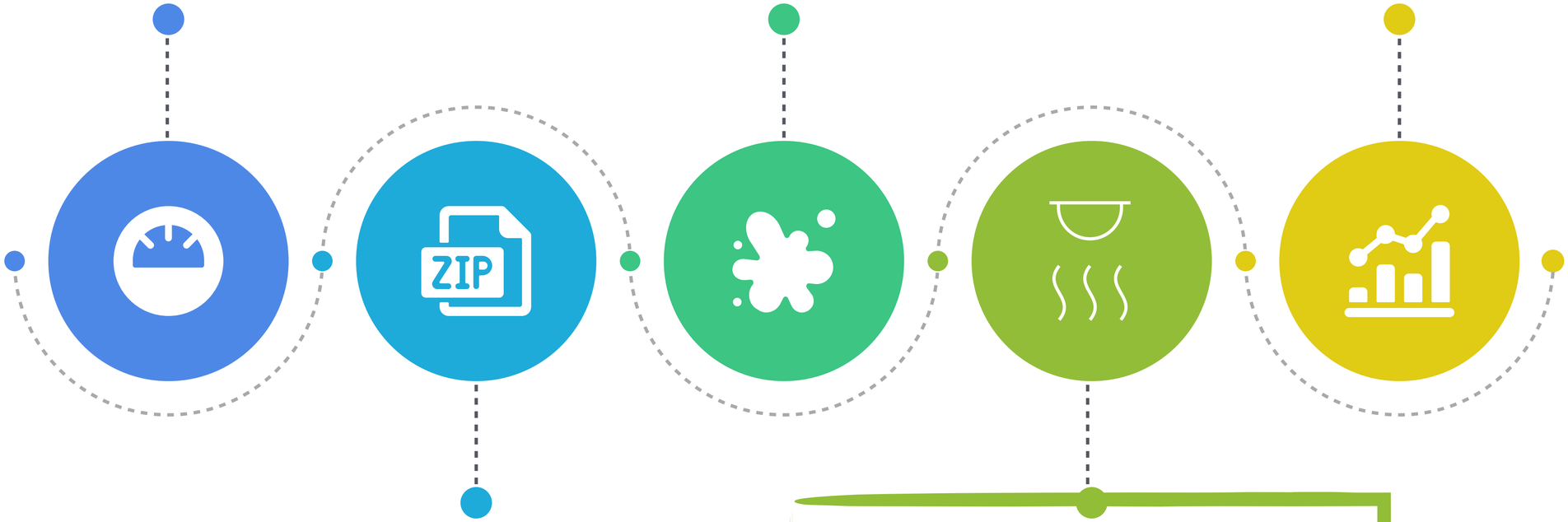


AI/ML Applications in the QCD Field

Detector Control, Calibration

Event Selection, Triggering

Reconstruction, Analysis



Data Compression

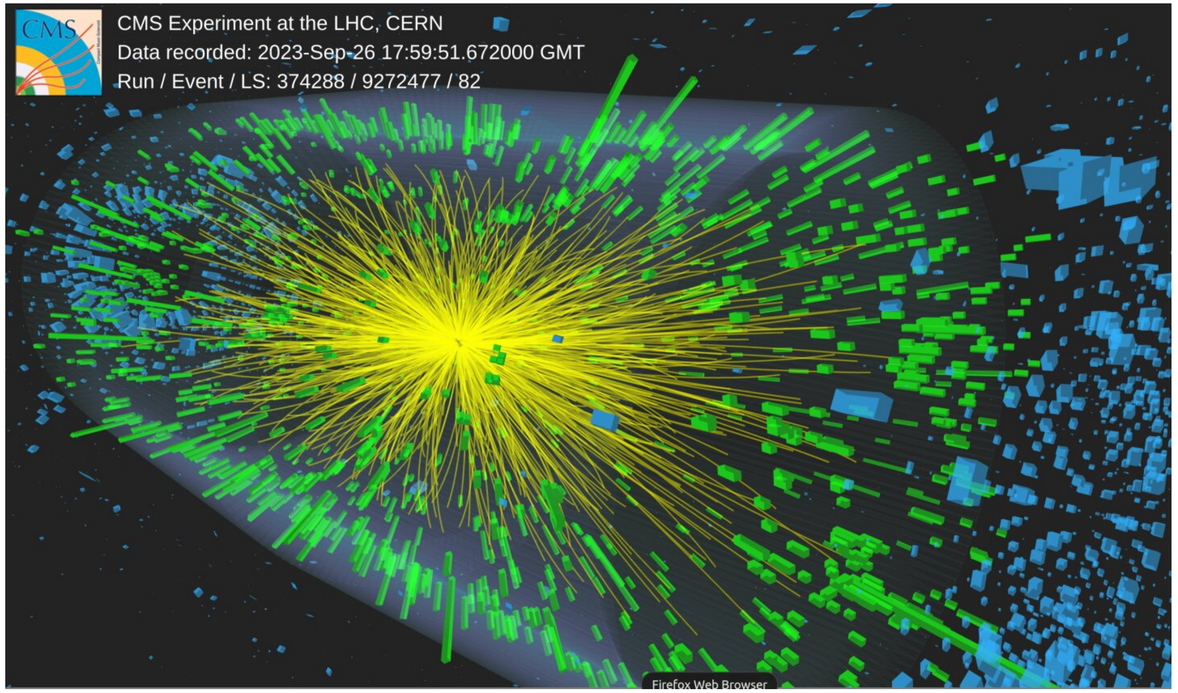
Simulation

Also,
Detector Design
Anomaly Detection
...

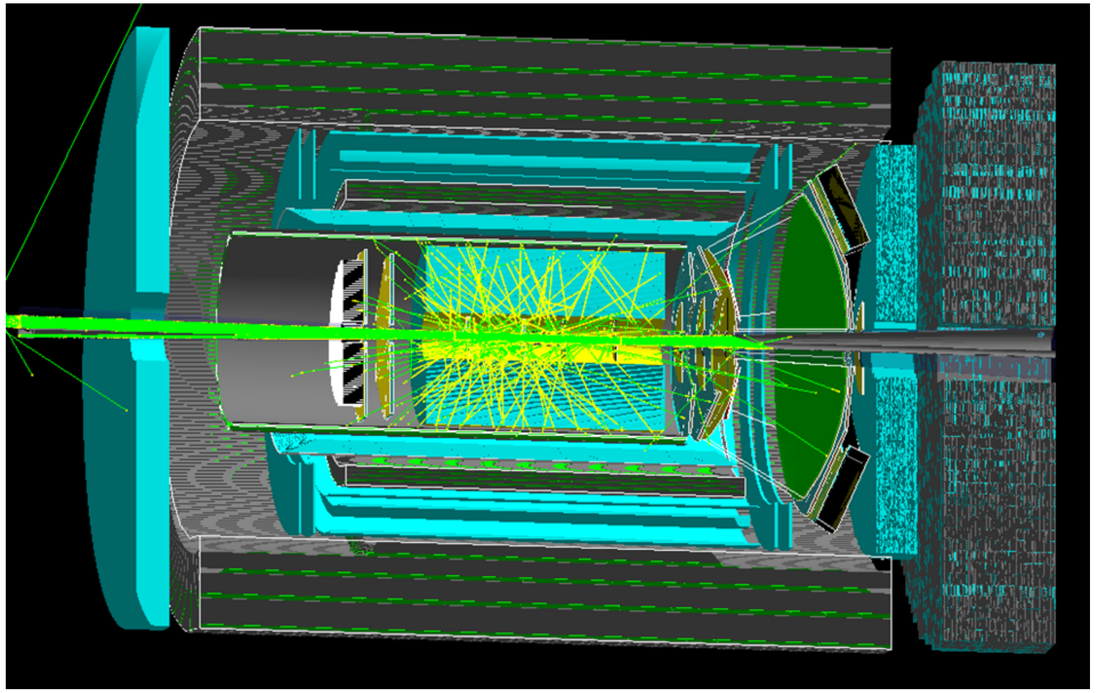
Fast Simulation for High-Occupancy Events

- High Energy Heavy Ion Collisions: ***O(1000) particles*** in one nuclear collision event + ***thousands shower steps*** per particle
  - ➡ Simulation of particle interaction with detectors is **highly complex and computationally intensive**

CMS Pb+Pb collision experimental data event display



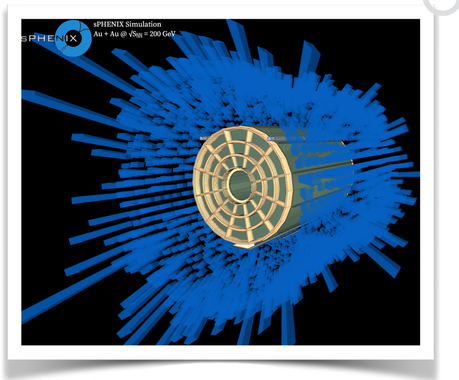
EIC collisions with beam background



ML can speed up and produce large amount of the high occupancy e.g. (heavy ion) event simulations!

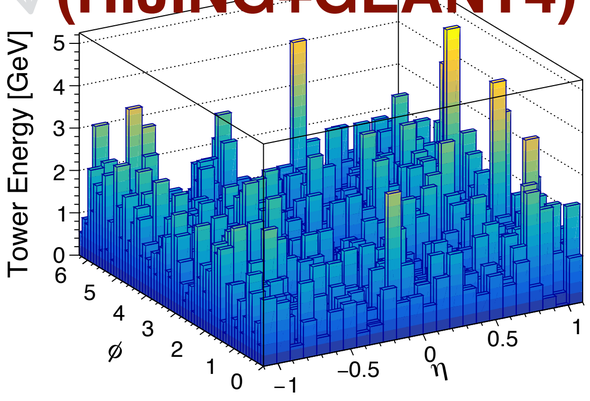
Fast Simulation: High Occupancy Full Event

- Generate high fidelity "particle production + full detector simulation" data in *central heavy ion collisions* using Denosing Diffusion Probabilistic Model (**DDPM**) *PRC 110 (2024) 034912*



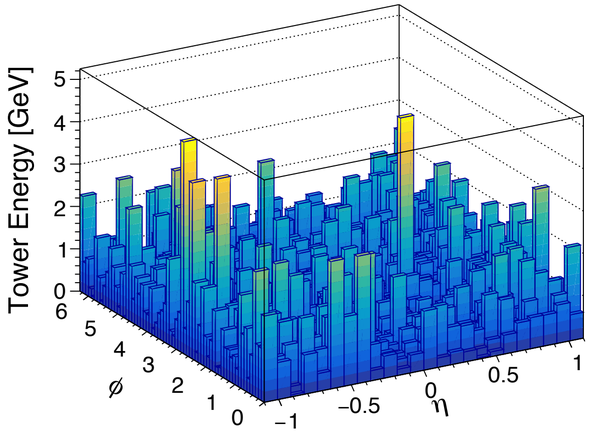
0-10% central event display

Training sample (HIJING+GEANT4)



40 minutes per event
on CPU

Generated (DDPM)

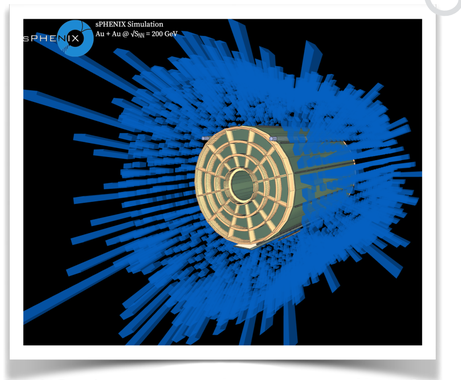


1.34 seconds per event
on GPU

- DDPM provides a speedup of **1800** times! (on GPU)
  - ➡ at least 100 times faster, considering a 32-core CPU equivalent to a GPU

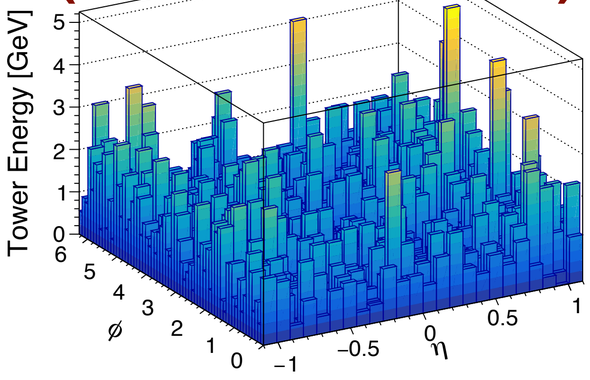
Fast Simulation: High Occupancy Full Event

- Generate high fidelity "particle production + full detector simulation" data in *central heavy ion collisions* using Denosing Diffusion Probabilistic Model (**DDPM**) *PRC 110 (2024) 034912*



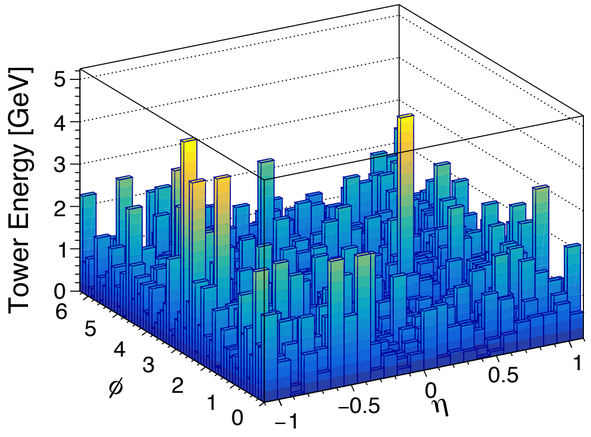
0-10% central event display

Training sample (HIJING+GEANT4)

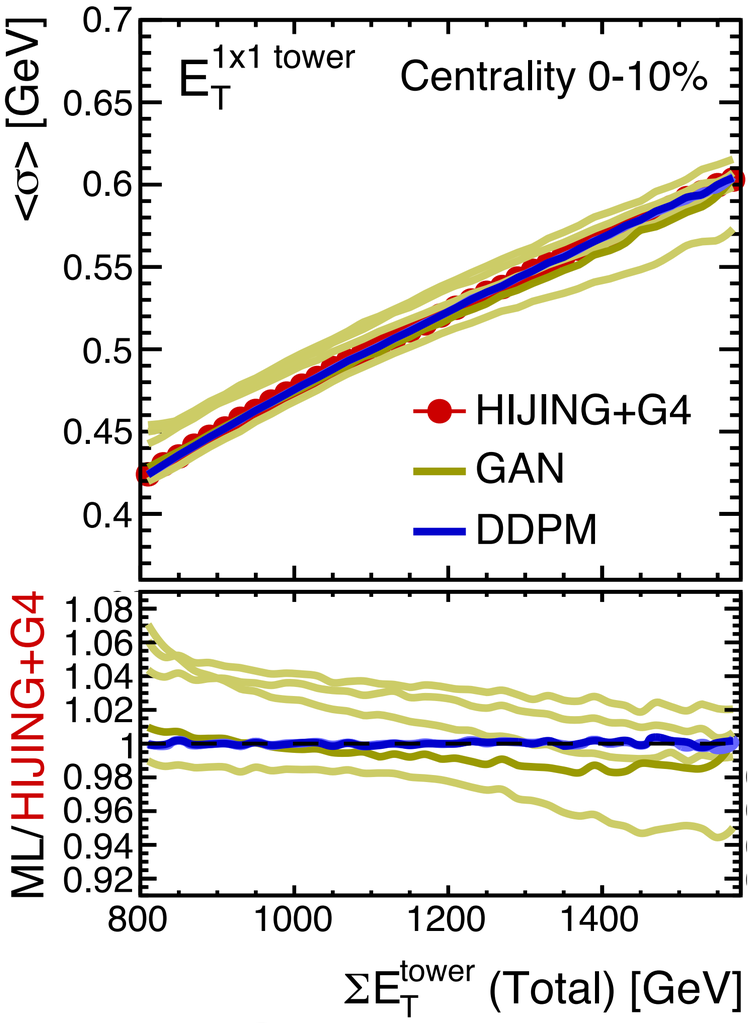


40 minutes per event *on CPU*

Generated (DDPM)



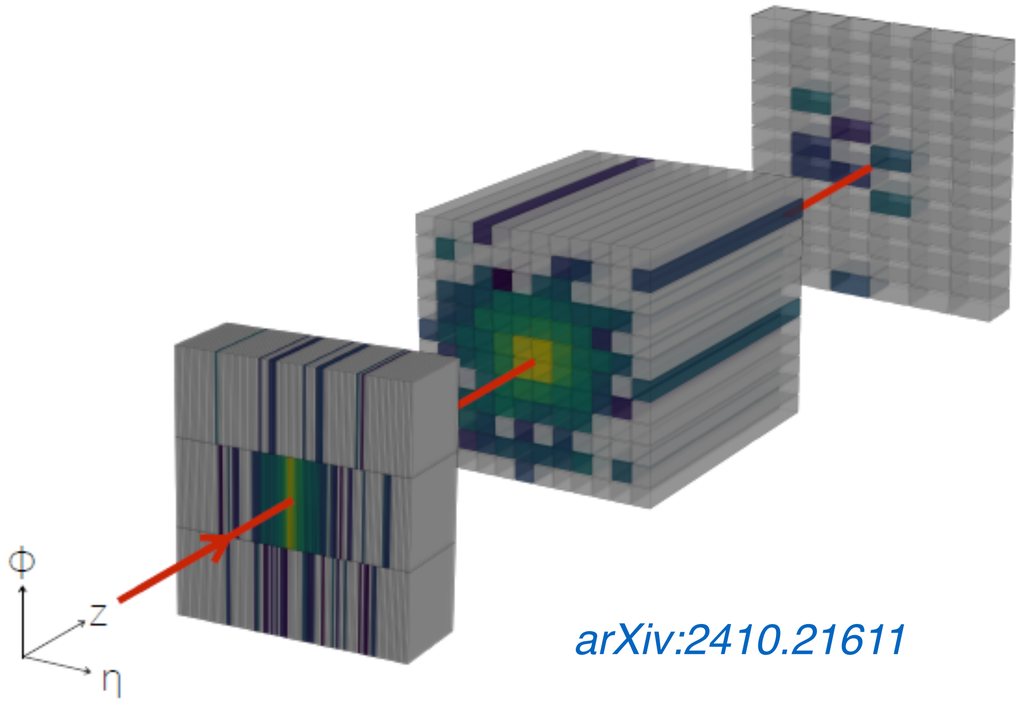
1.34 seconds per event *on GPU*



- DDPM provides a speedup of **1800** times! (on GPU)
  - ➡ at least 100 times faster, considering a 32-core CPU equivalent to a GPU
- DDPM shows great stability and accuracy within sub-percent level outperforming GAN

Fast Calorimeter Simulation

- **CaloChallenge** for High Luminosity LHC
  - ➡ Much higher number of detector channels e.g. CMS HGCAL 6.5 million channels
  - ➡ Calorimeter shower generation using various ML models for fast simulation
  - ➡ ***Trade-off between fidelity and timing***



arXiv:2410.21611

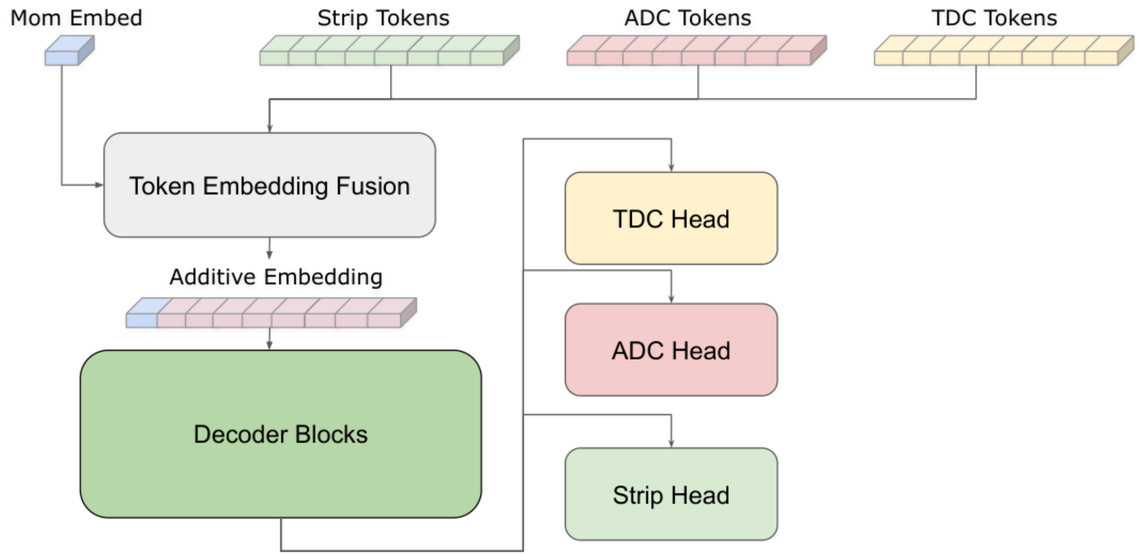
Table 1: Models submitted to the CaloChallange.

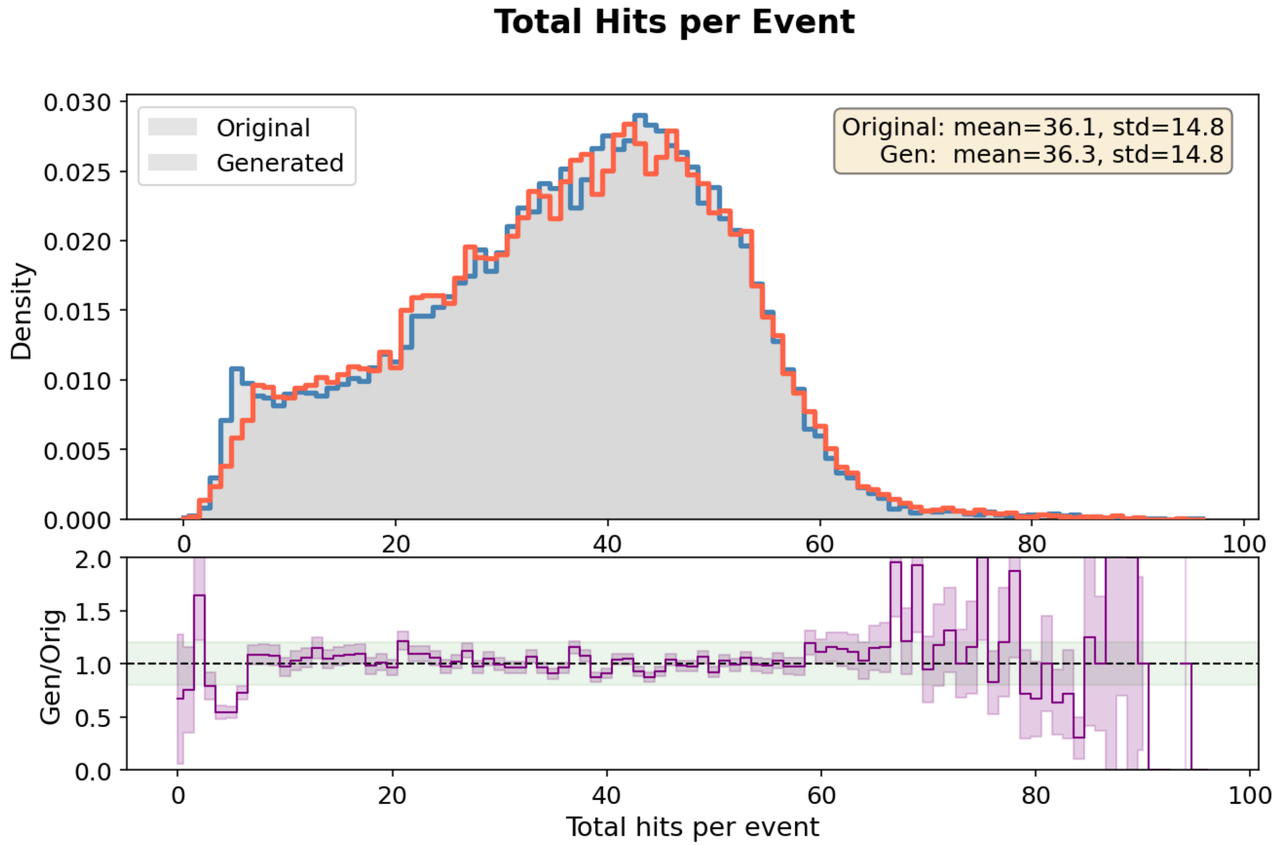
Approach	Model	Code	Dataset $1-\gamma$	Dataset $1-\pi$	Dataset 2	Dataset 3	Section
GAN	CaloShowerGAN [21]	[22]	✓	✓			3.1
	MDMA [23, 24]	[25]			✓	✓	3.2
	BoloGAN [26]	[27]	✓	✓			3.3
	DeepTree [28, 29]	[30]			✓		3.4
NF	L2LFlows [31, 32]	[33]			✓	✓	4.1
	CaloFlow [34, 35]	[36, 37]	✓	✓	✓	✓	4.2
	CaloINN [38]	[39]	✓	✓	✓		4.3
	SuperCalo [40]	[41]			✓		4.4
	CaloPointFlow [42]	[43]			✓	✓	4.5
Diffusion	CaloDiffusion [44]	[45]	✓	✓	✓	✓	5.1
	CaloClouds [46, 47]	[48, 49]				✓	5.2
	CaloScore [50, 51]	[52, 53]	✓		✓	✓	5.3
	CaloGraph [54]	[55]	✓	✓			5.4
	CaloDiT [56]	[57]			✓		5.5
VAE	Calo-VQ [58]	[59]	✓	✓	✓	✓	6.1
	CaloMan [60]	[61]	✓	✓			6.2
	DNNCaloSim [62, 63]	[64]		✓			6.3
	Geant4-Transformer [65]	[66]				✓	6.4
	CaloVAE+INN [38]	[39]	✓	✓	✓	✓	6.5
	CaloLatent [67]	[68]			✓		6.6
CFM	CaloDREAM [69]	[70]			✓	✓	7.1
	CaloForest [71]	[72]	✓	✓			7.2

Fast Calorimeter Simulation

arXiv:2606.16035

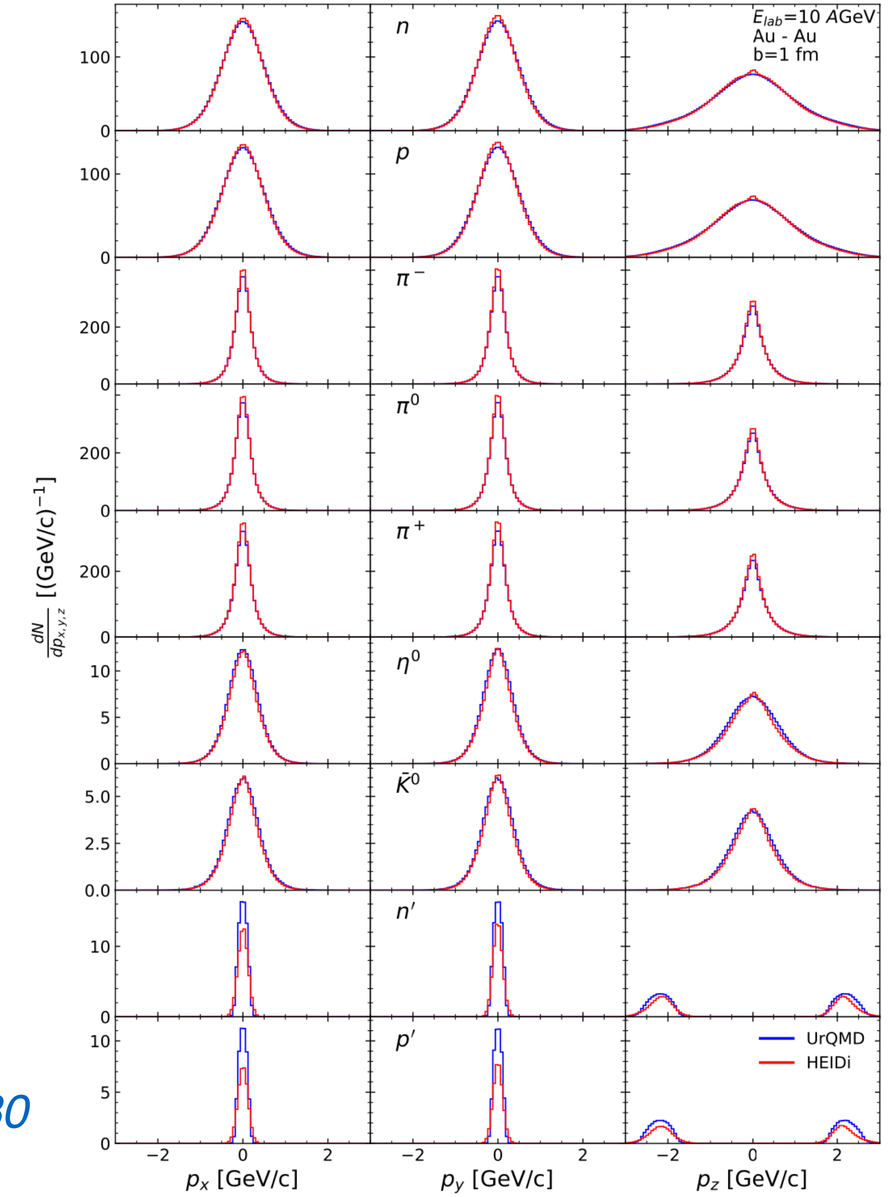
- **Calorimeter surrogate** for CLAS12
  - ➡ GPT-style Conditional Autoregressive Generation
  - ➡ ~700 events per second under single NVIDIA L40S GPU
    x10^4-10^5 speed up compared to Traditional Geant4





Fast MC Event Generation

- PointNet-based point cloud encoder and a normalizing flow-based decoder
- ***HEIDi*** generates heavy ion collision events two order of magnitude faster than the conventional UrQMD cascade computations.

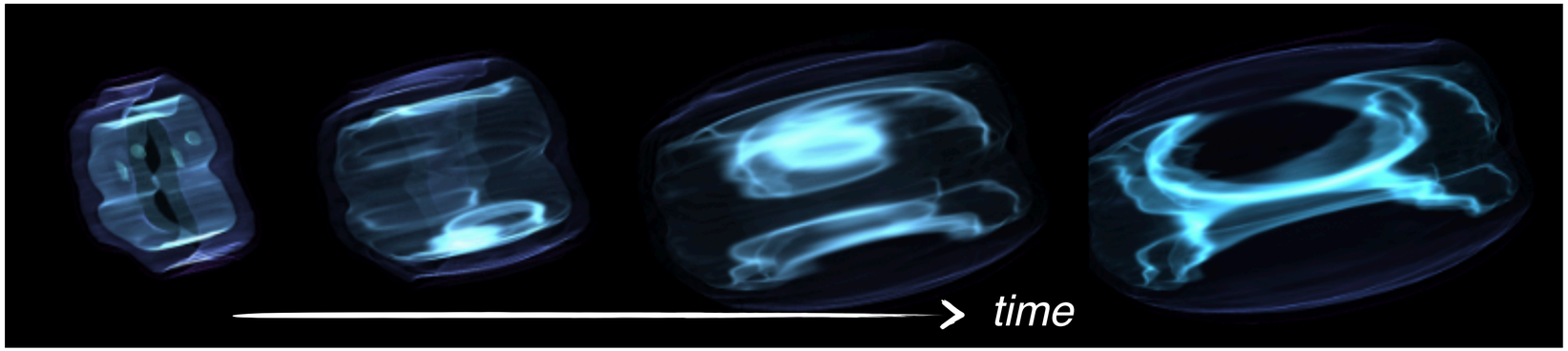


arXiv:2502.16330

Hydrodynamic Evolution Simulation

- Full *spatiotemporal hydrodynamic medium (Quark Gluon Plasma) evolution simulations* are computationally very demanding

QGP 3+1D Energy Density Evolution



- **Fourier Neural Operator (FNO)**-based model developed**: TC-QGP** to predict the full time evolution of 3D QGP fields from the initial time 3D fields within a few percent accuracy
  - **O(100,000) speedup with a few percent accuracy**

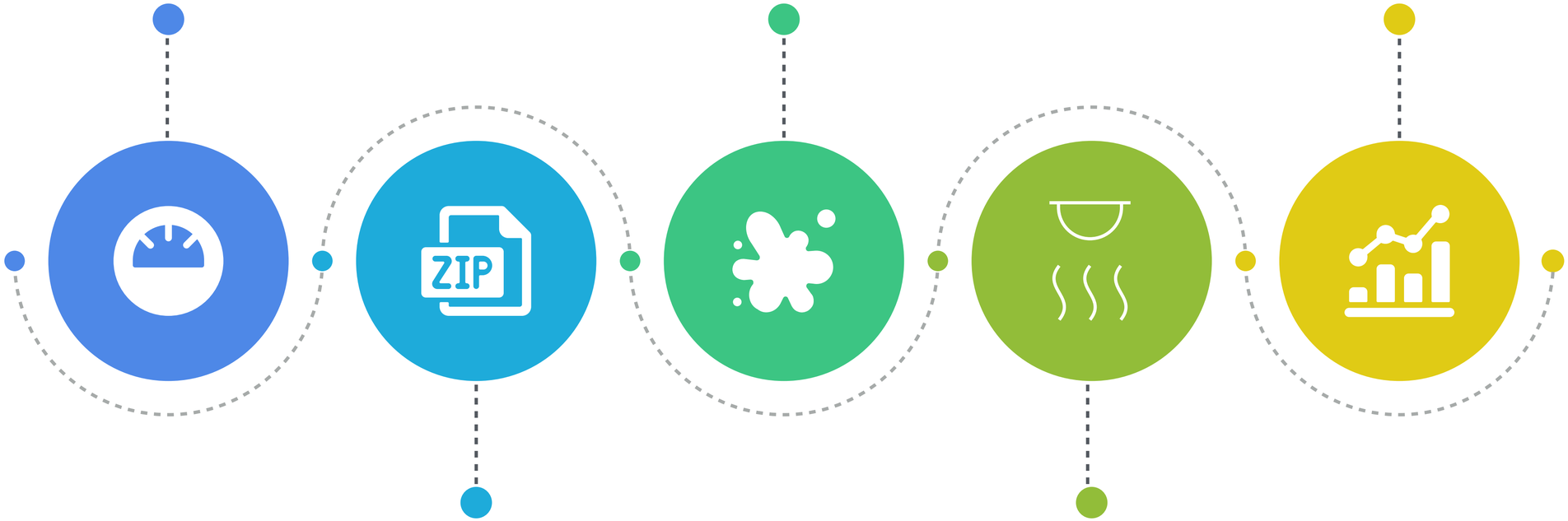
	2+1D	3+1D
Simulation time per event with 1 CPU	~10 minutes	~20 hours
TC-QGP	0.03 seconds	0.2 seconds

AI/ML Applications in the QCD Field

Detector Control, Calibration

Event Selection, Triggering

Reconstruction, Analysis



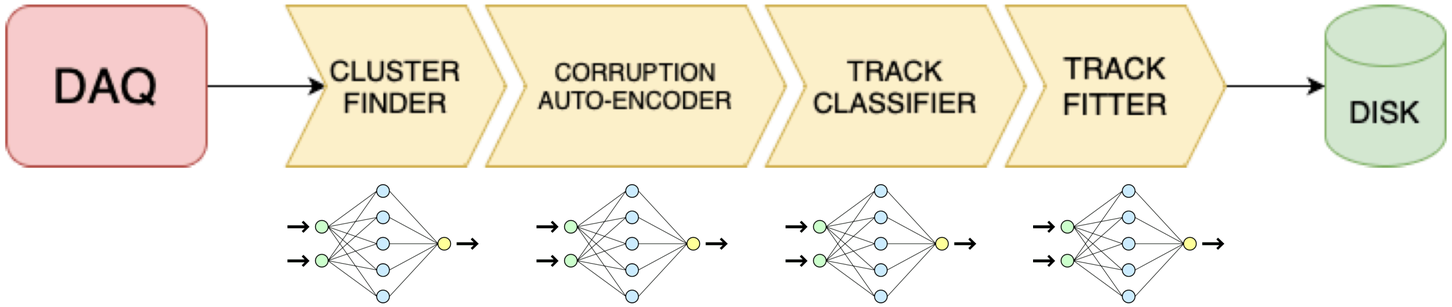
Data Compression

Simulation

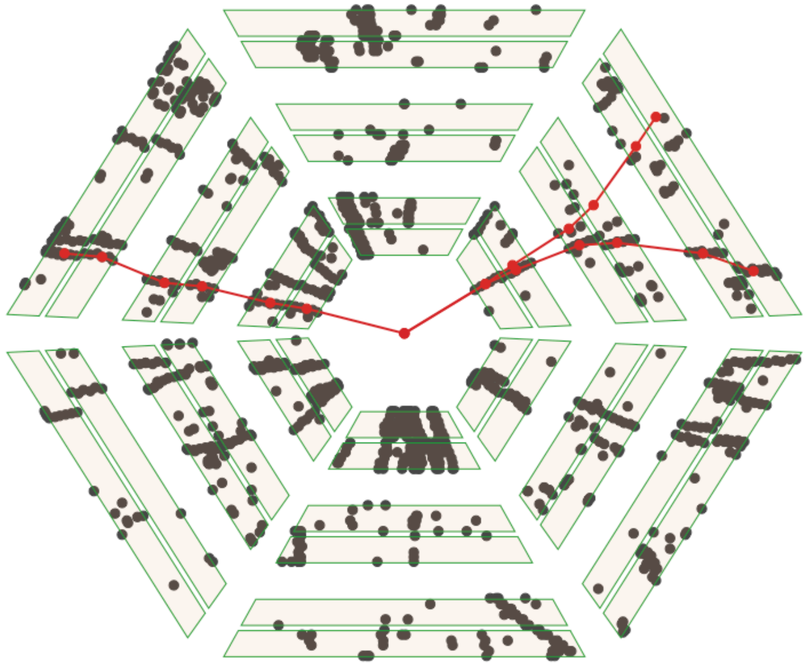
Also,
Detector Design
Anomaly Detection
...

Tracking

- Real-time (online) tracking
- Convolutional Autoencoder for noise reduction
- Track candidates using MLP



4 Neural Networks



CLAS12 Drift Chamber

Significant computing resource and time save:
~66 Hz → ~500 Hz

[Comput.Phys.Commun. 271 (2022) 108201]
[arXiv: 2205.02616]
[arXiv:2403.04020]

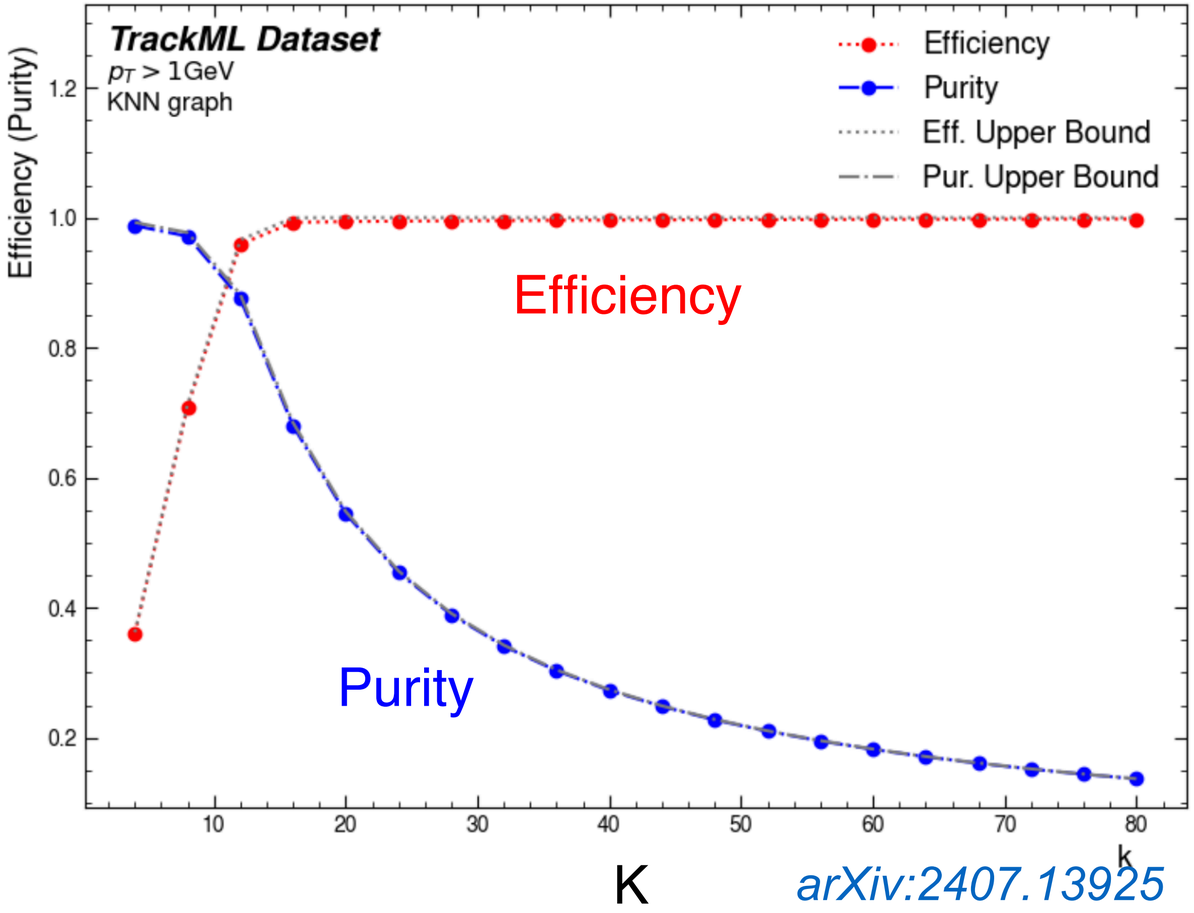
Tracking

Many Tracking Works

- State space models (SSM)-based (arXiv:2501.16237)
- Transformer-based tracking:
  - ➡ scalable sequence modeling for hits and tracks (arXiv:2411.07149)
  - ➡ TrackFormers (arXiv:2407.07179)
- EggNet: Evolving Graph-based Graph Attention Network (arXiv:2407.13925)
- Graph Neural Networks (GNNs):
  - ➡ Exa.TrkX (GNN + CKF) (arXiv:2401.16016)
  - ➡ Hierarchical GNN (arXiv:2303.01640)
  - ➡ ATLAS GNN tracking (EPJ Web Conf. 295 (2024) 03030)
  - ➡ Review (arXiv:2012.01249)
- HEP.TrkX using CNN + LSTM (*EPJ Web Conf.* 150 (2017) 00003)

TrackML Dataset for tracking development for HL-LHC

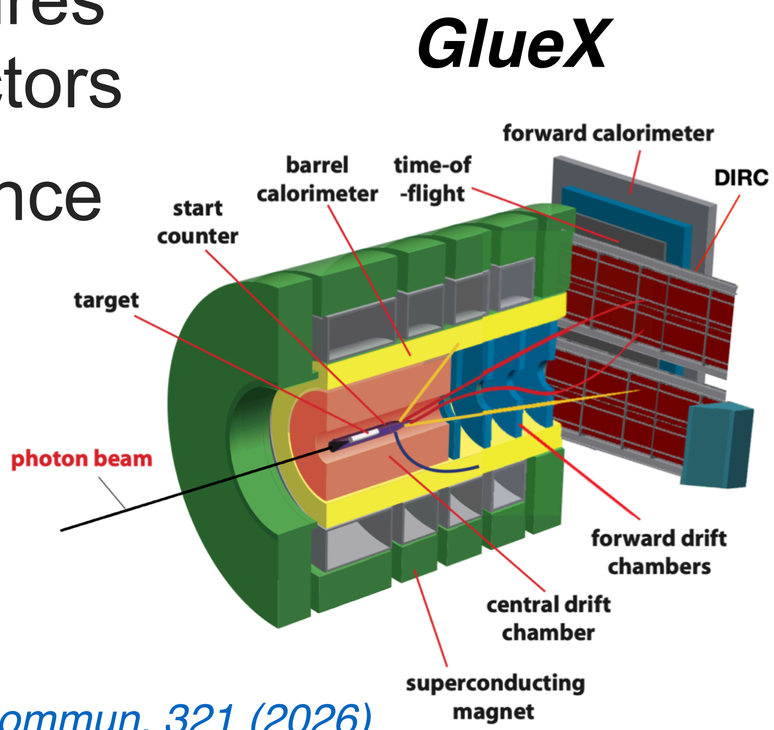
EggNet K-nearest neighbors



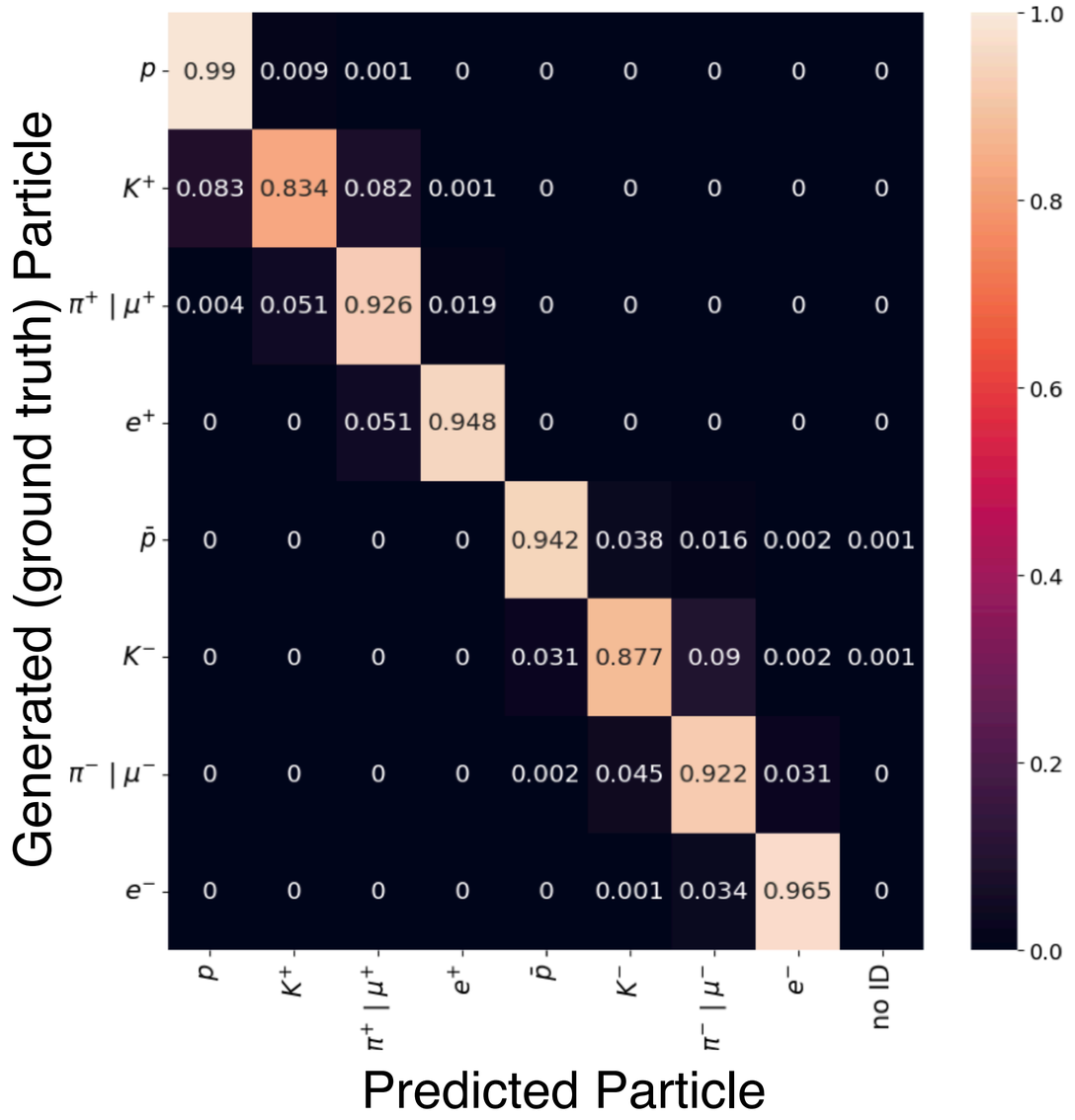
arXiv:2407.13925

Particle Identification

- MLP using features from 7 subdetectors
- Better performance than cut-based methods



JINST 20 (2025) 10, P10024

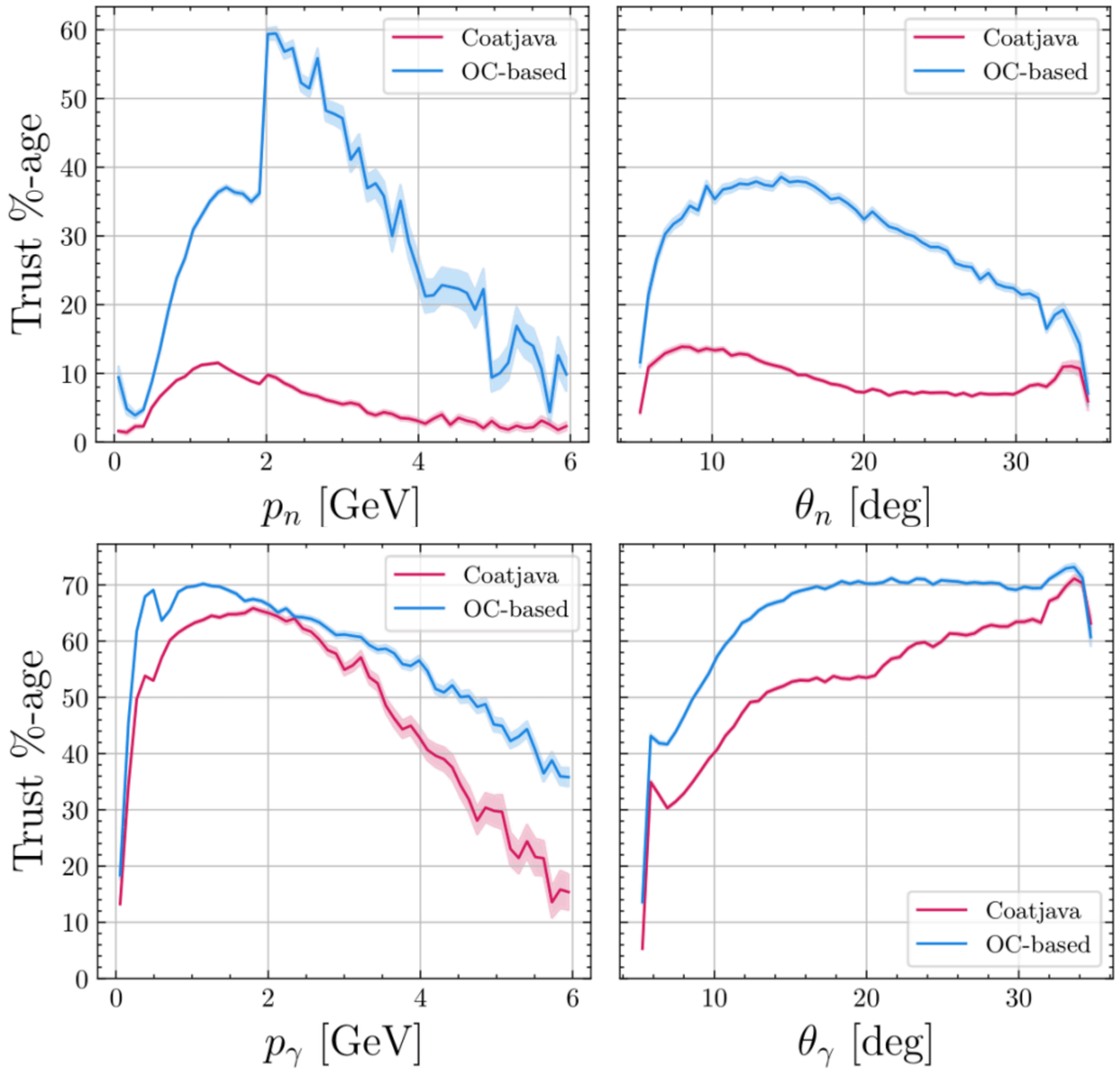


Many similar works:

CLAS12: Comput. Phys. Commun. 321 (2026) 110032 | arXiv:2507.05274
GlueX: JINST 20 (2025) 10, P10024 arXiv:2505.06732
ALICE: arXiv:2309.07768, arXiv:2204.06900
Calorimeter particle identification: arXiv:2103.16247
CNN with TPC: JINST 20 (2025) 07, P07030
…

Calorimeter Clustering

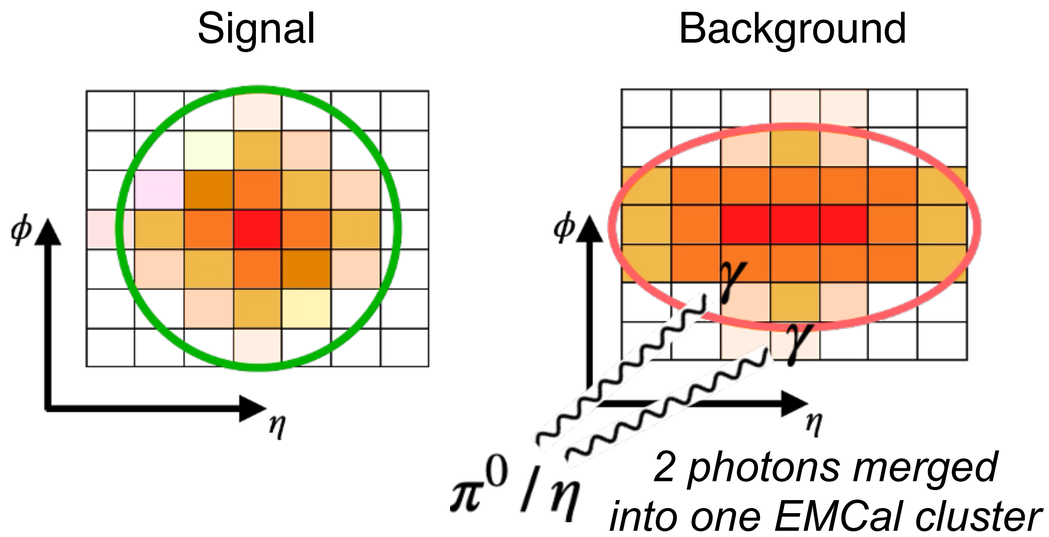
- **GravNet** (distance weighted graph network) + **Transformer** to capture local and global detector correlations
- Reconstruct **neutrons and photons** from CLAS12 Forward ECal hits: Object Condensation (OC) method
- Much better clustering accuracy achieved compared to the traditional method (Coatjava)



NIM A 1082 (2026) 170990, arXiv:2503.11277

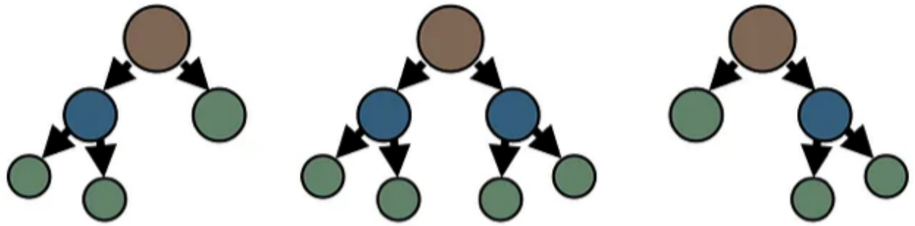
Classification: Photon vs Neutral hadrons

- **Boosted Decision Trees (BDT)**: Multivariate method for signal-background classification



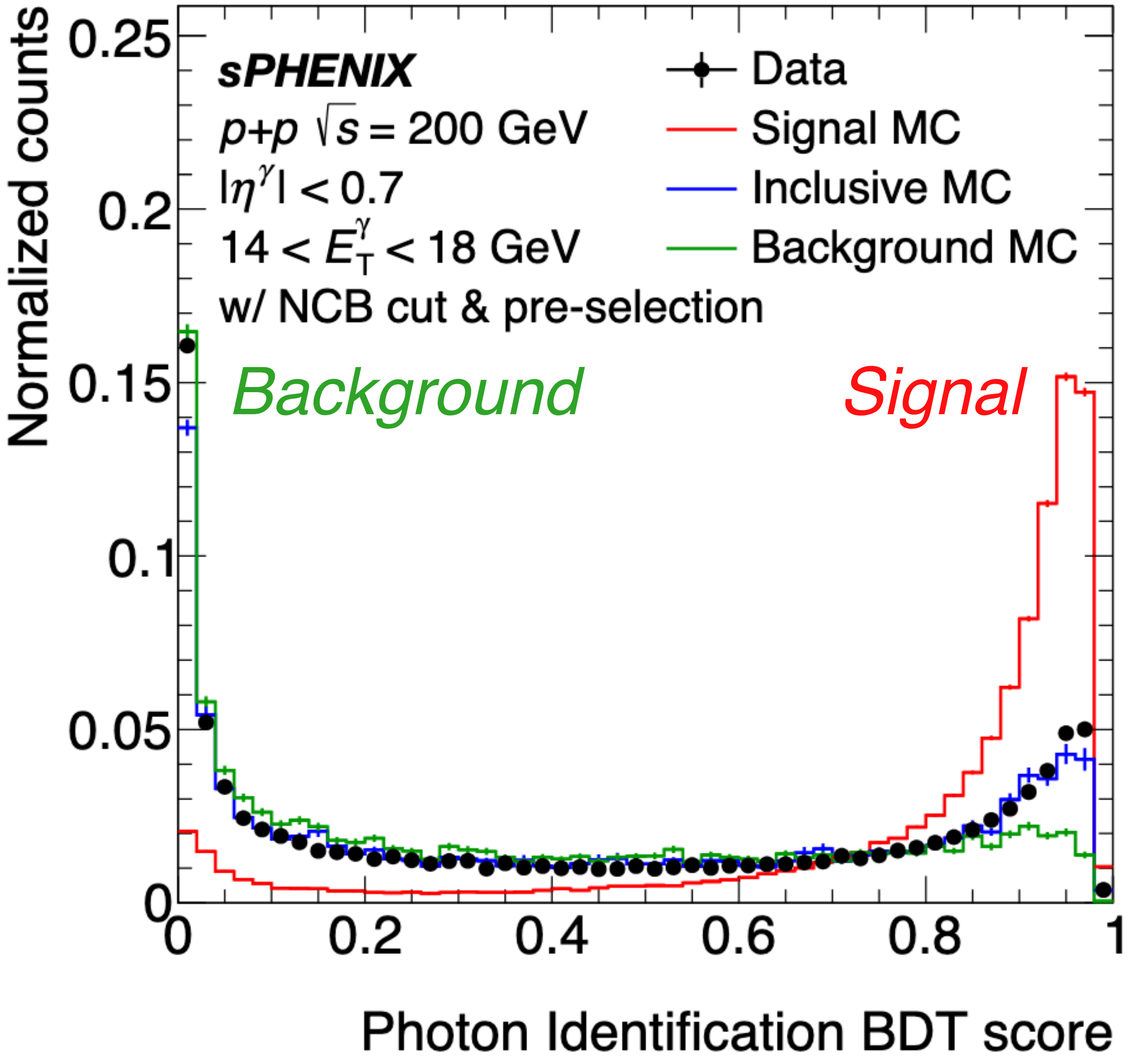
Input: Shower Shape Variables

BDT



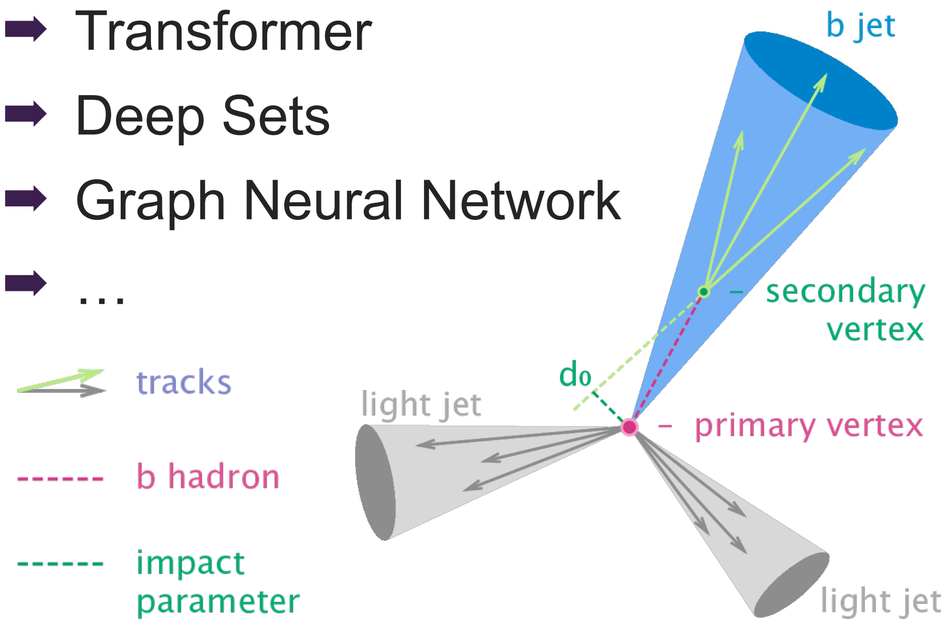
Output: Signal or Background?

arXiv:2607.03892

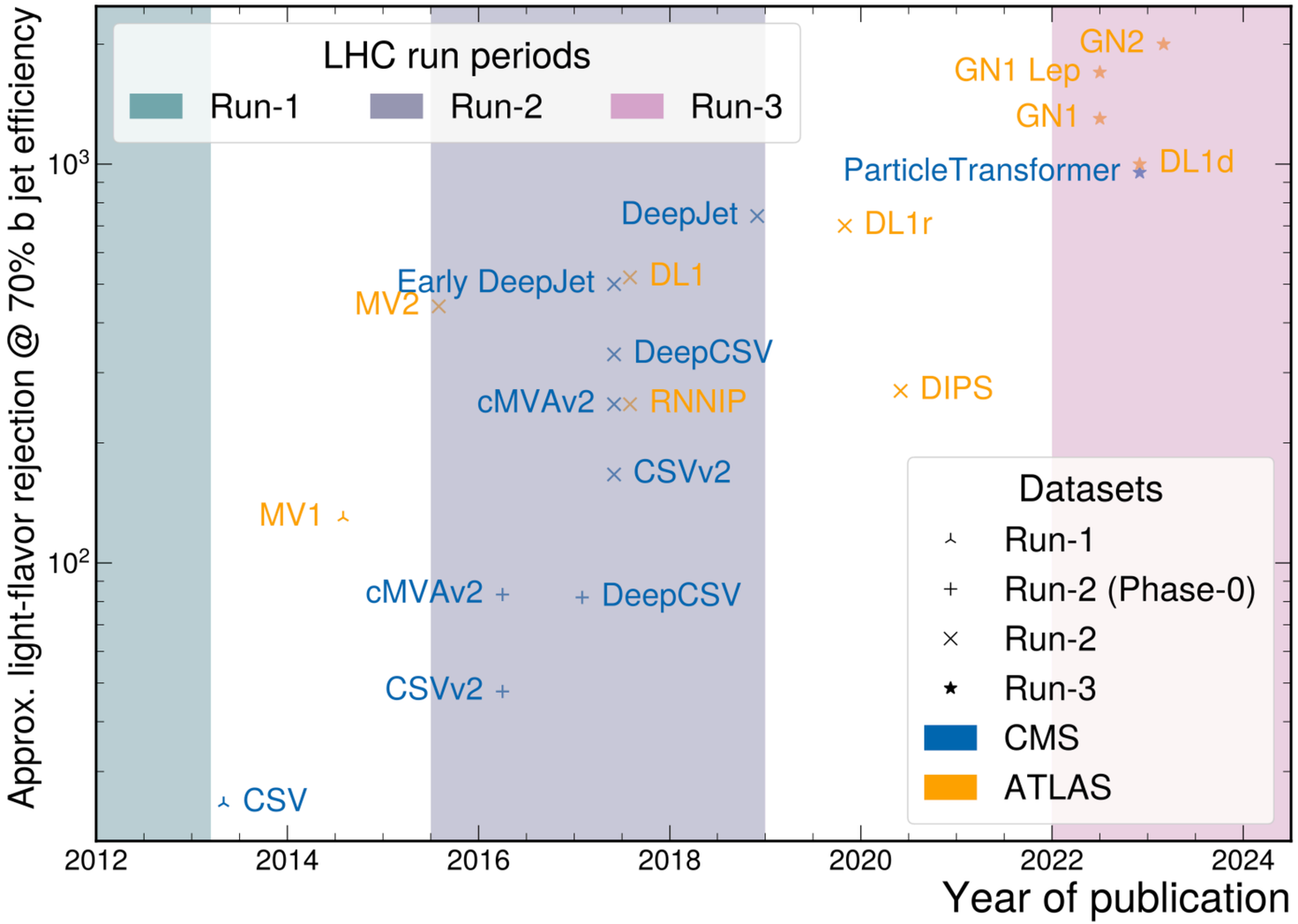


Classification: Jet Tagging

- Significant evolution on Flavor tagging using ML over the decade
  - ➡ Multi layer Perceptron
  - ➡ Convolutional Neural Networks
  - ➡ Recurrent Neural Networks
    - Long Short-Term Memory
  - ➡ Transformer
  - ➡ Deep Sets
  - ➡ Graph Neural Network
  - ➡ …



[Eur.Phys.J.ST 233 (2024) 15-16, 2657-2686]



Regression for Jet Momentum

- ***Supervised* multi-layer perceptron (MLP)**
- Regression of jet p_T using jet substructure information (jet p_T before correction, 10 constituent's p_T, mass etc) as input

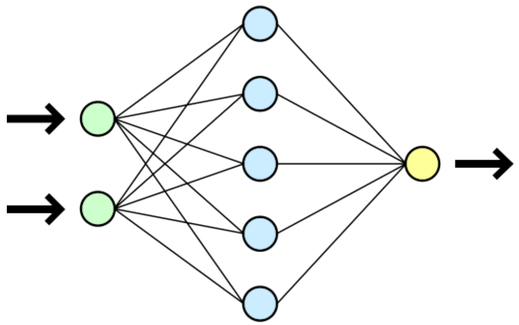
➡ *predict background-subtracted p_T*

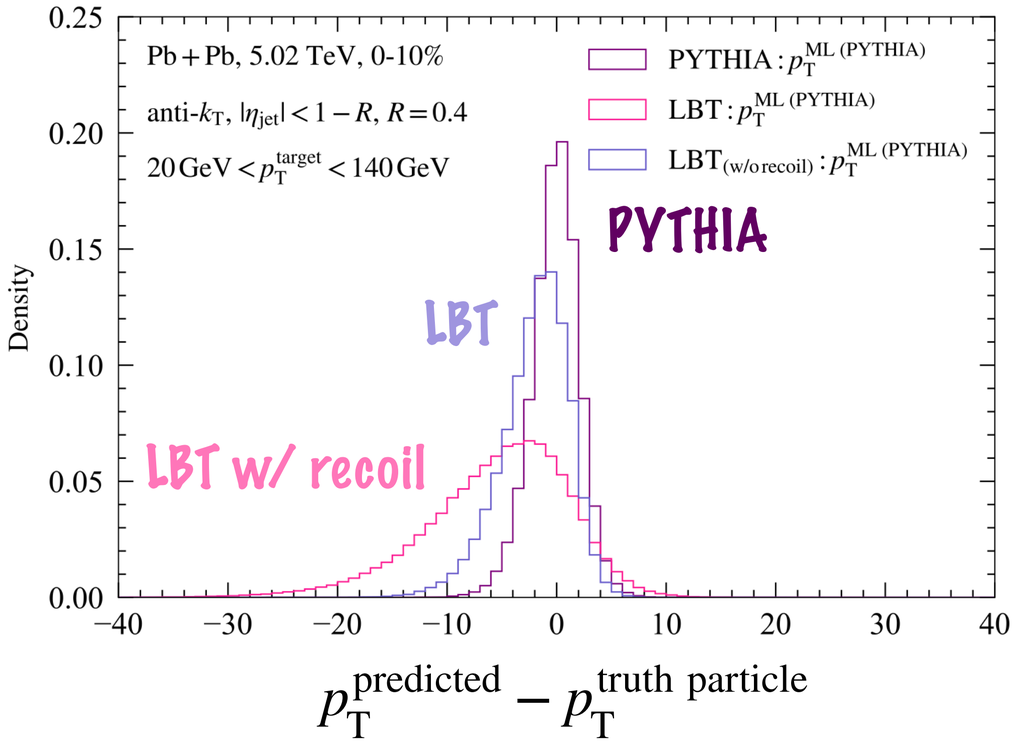
Related works:

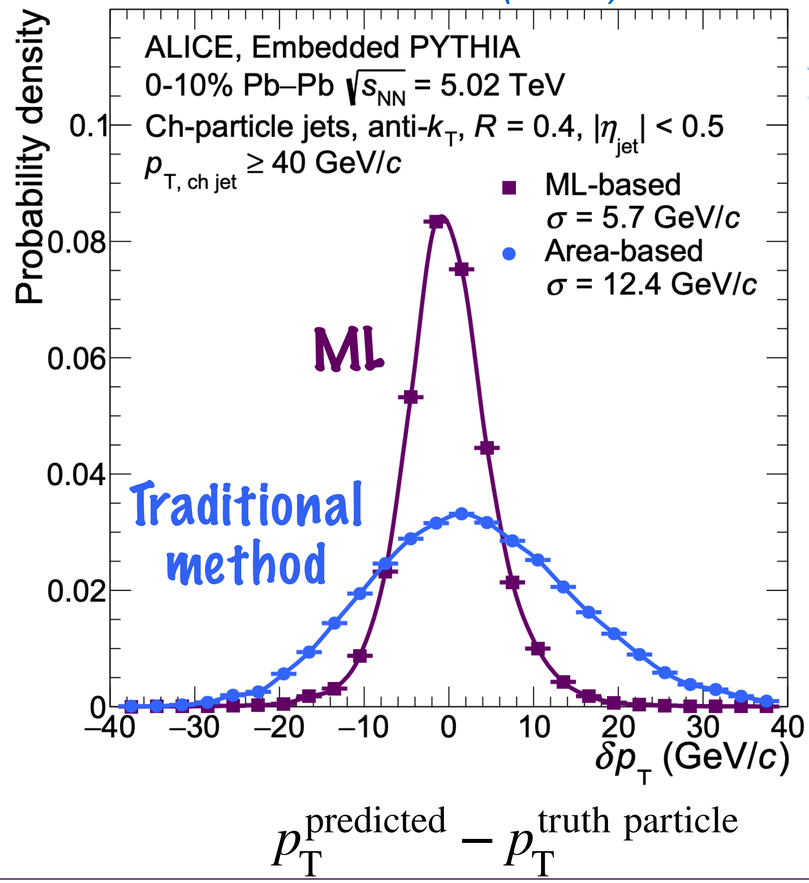
[PRC 108 (2023) 2, L021901](#)

[PRC 99 (2019) 6, 064904](#)

PLB 849 (2024) 138412

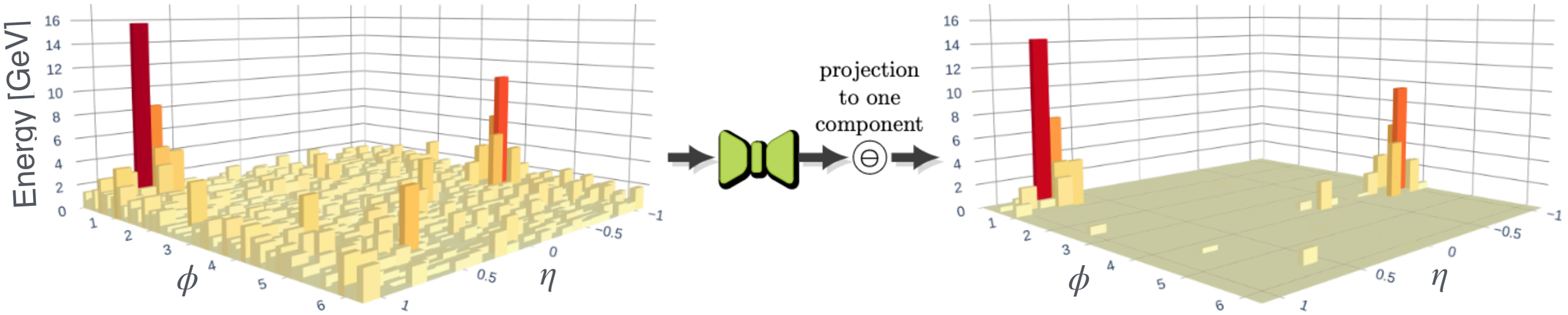






Generative AI for Jet Background Subtraction

- Translate *jet+background* calorimeter image to *jet signal-only calorimeter image*



- ***UVCGAN-S***
  - ➡ *preserves the **full dimensional information***
  - ➡ ***can use experimental data** (w/o label) for training; mitigate physics modeling-dependence*

- **Significant improvement in performance**
  - ➡ **x2** better momentum resolution
  - ➡ **~ 40%** improvement in position resolution
  - ➡ efficiency 74% → **92**%, fake rate 80% → **6**%

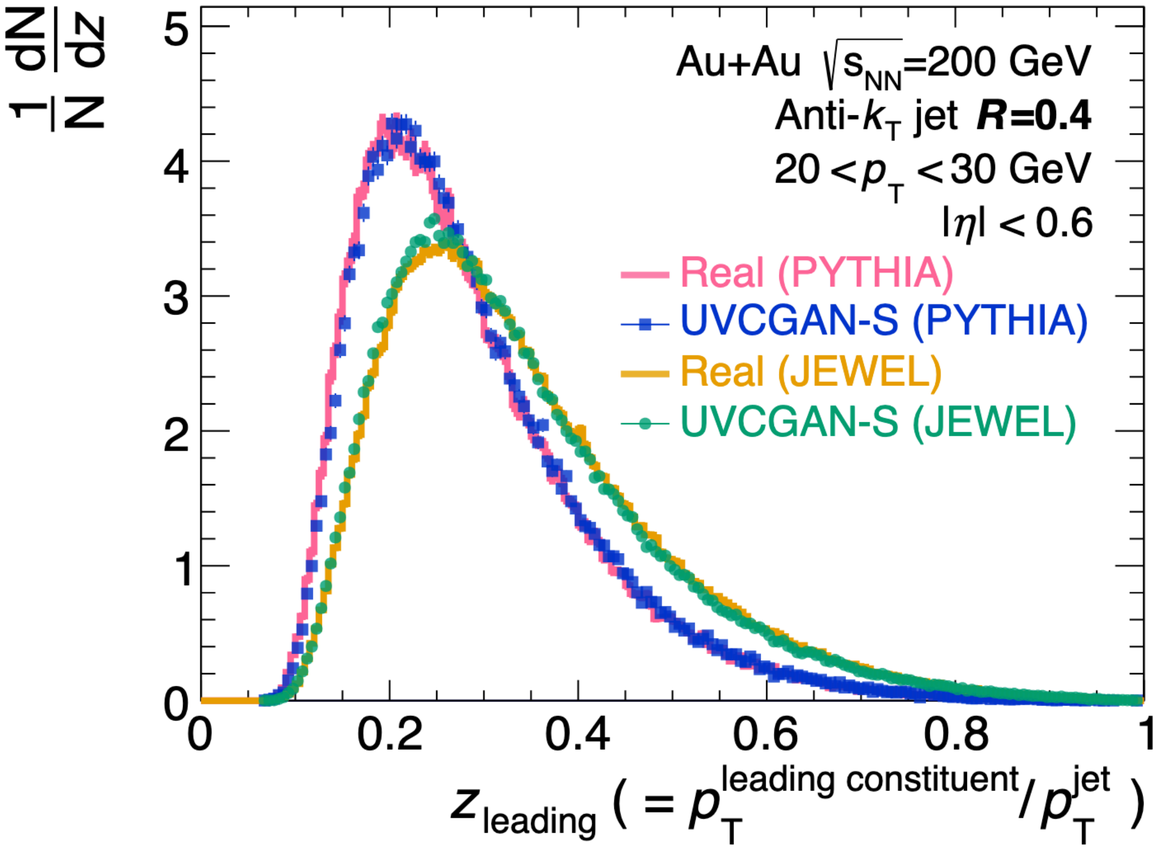
Unsupervised learning, **unpaired** image-to-image translation

arXiv:2510.23717
Nature Scientific Reports (DOI: 10.1038/s41598-026-59672-8)

Generative AI for Jet Background Subtraction

UVCGAN-S trained on PYTHIA (unmodified jets)
but tested on JEWEL (modified jets) →
the model successfully reproduce the JEWEL's distribution

Robust Generalizability of the UVCGAN-S for out-of-distribution handling!



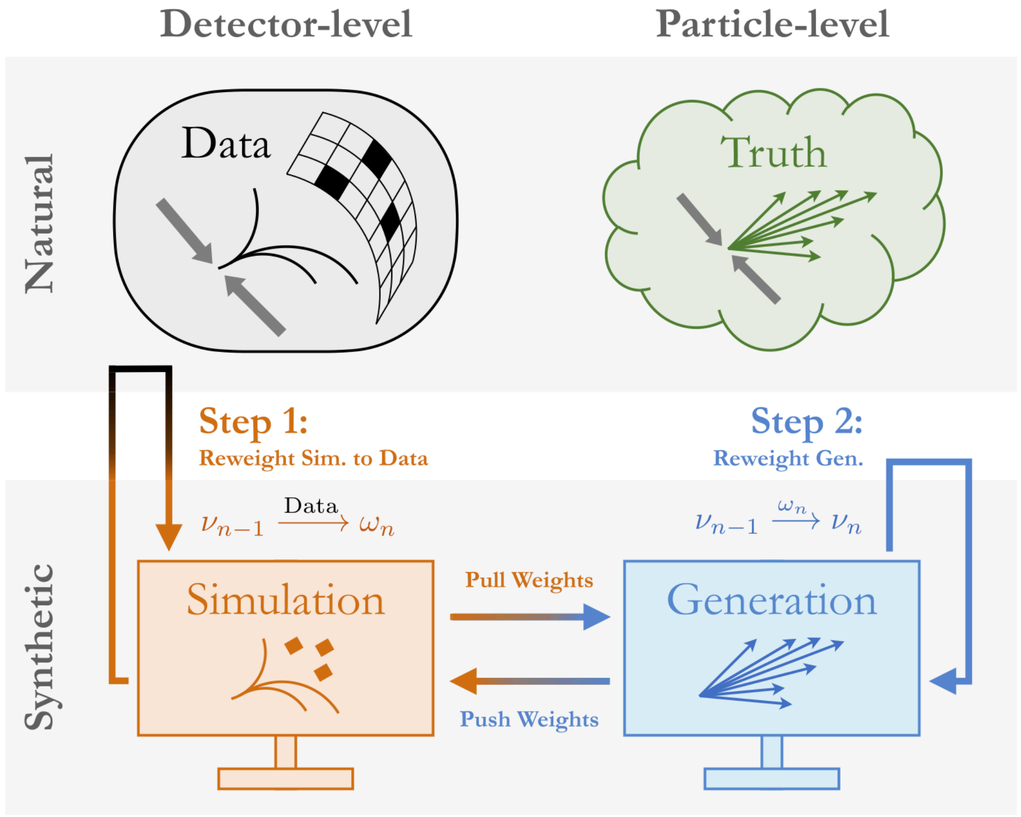
- Extension of the work:
  Genesis Mission Phase I project
  "*Characterizing Jet Modification in the Quark Gluon Plasma Using Unsupervised, Cycle-Consistent Generative Learning*"
  by Baruch College, UCB, CU, BNL

arXiv:2510.23717
Nature Scientific Reports (DOI: 10.1038/s41598-026-59672-8)

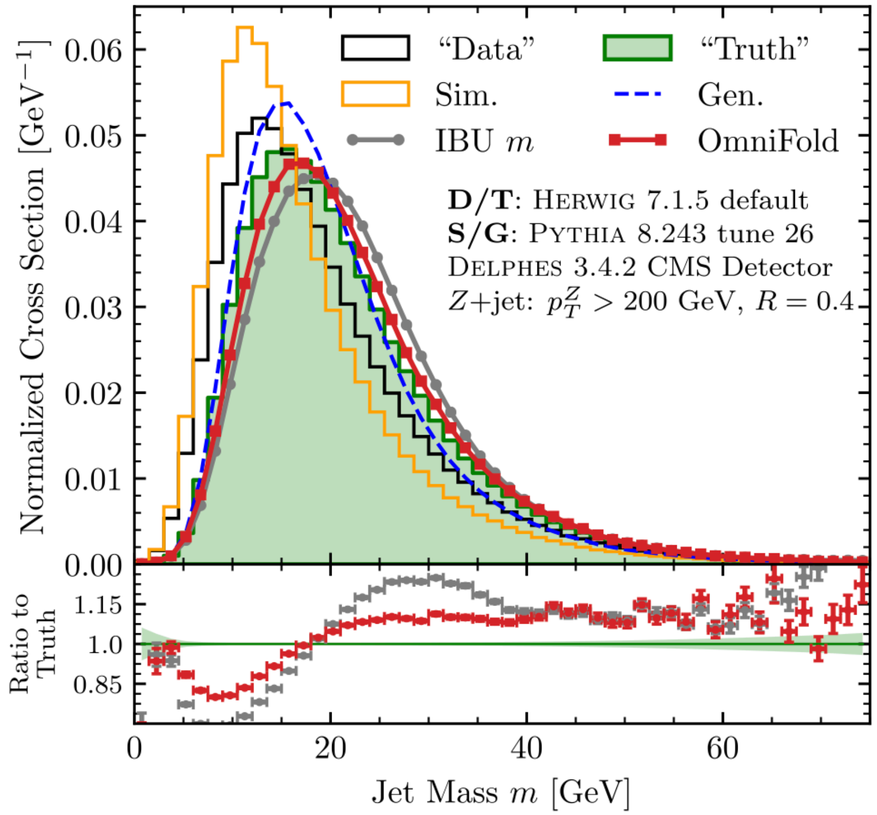
Unfolding

PRL 124, 182001 (2020)

- Unfolding: correcting for detector effect by translating reconstructed values to truth values
- **OmniFold**: using full phase space information for unfolding in an unbinned way with MLP



- Recent developments; unfolding using generative AI



More Unified AI Framework

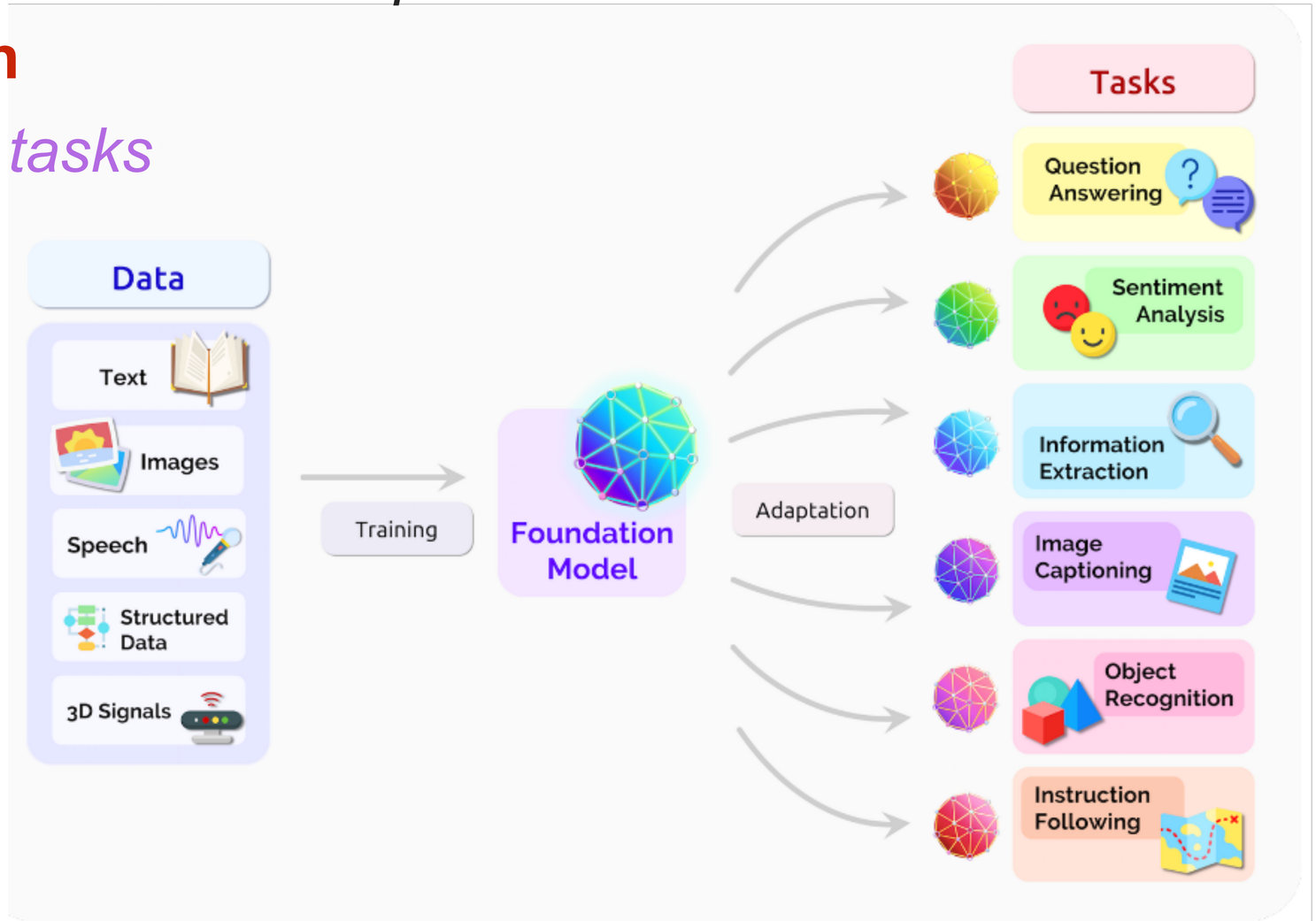
- AI/ML has been applied to and improved many different tasks (task specifically) in analysis
  - ➡ tracking
  - ➡ calibration
  - ➡ particle identification
  - ➡ particle tagging
  - ➡ jet classification
  - ➡ background subtraction
  - ➡ unfolding
  - ➡ calibration
  - ➡ …

Is there A More Unified AI Framework ?

Foundation Model

- **Foundation models** learn *broad features and patterns* from *massive datasets*: **pre-train**

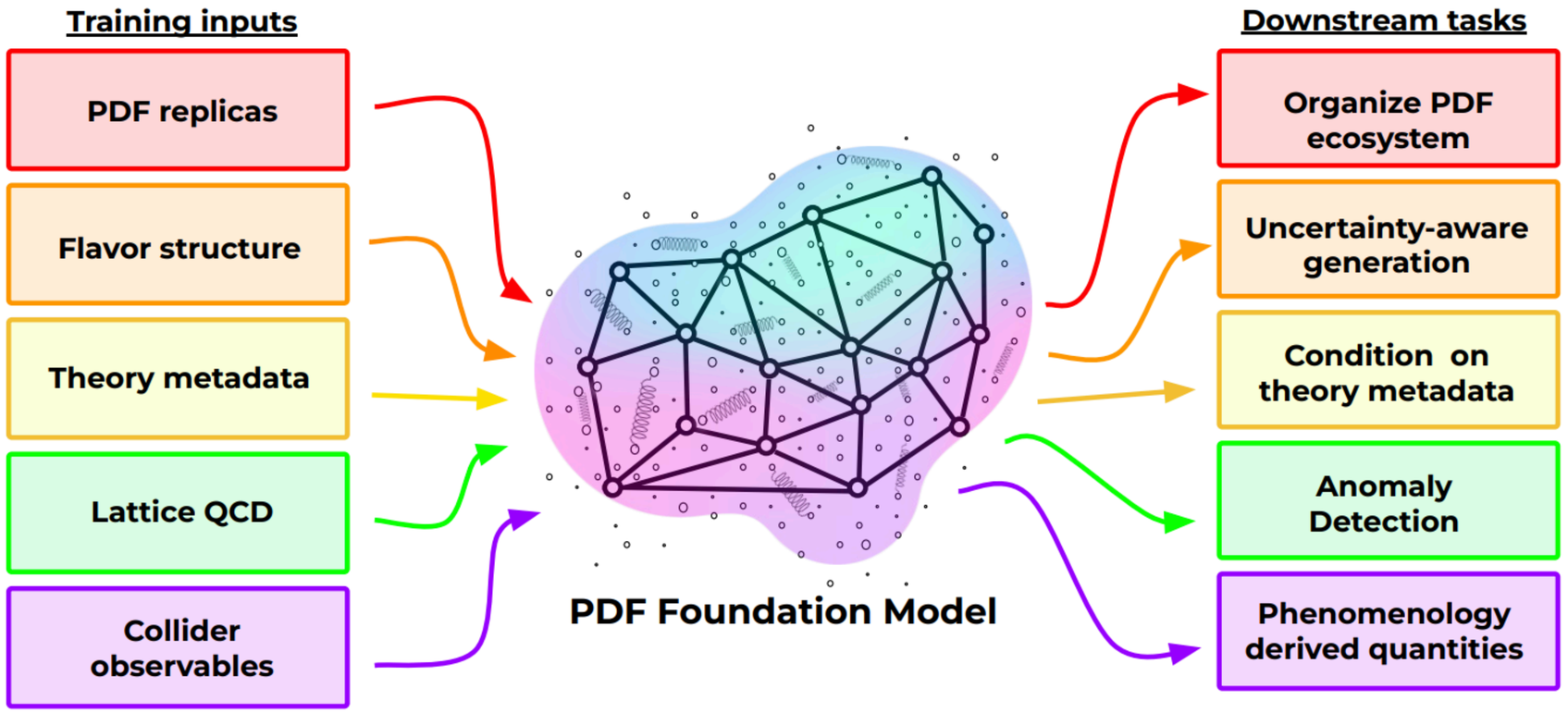
  ➡ adapted to many *downstream tasks* with minimal extra training



Foundation Models for PDFs

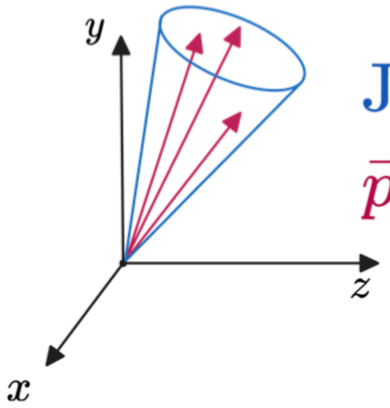
See talk by Brandon Kriesten (Thu. 9:00am)

- Discretized Tokens of (x, flavor i.e. g, u, s, d, …) for Transformer
- Multiple heads and training objectives
  - ➡ Contrastive Alignment
  - ➡ Classification
  - ➡ Masked Flavor Modeling
  - ➡ Reconstruction
  - ➡ Denoising



Foundation Model for Jet Analysis

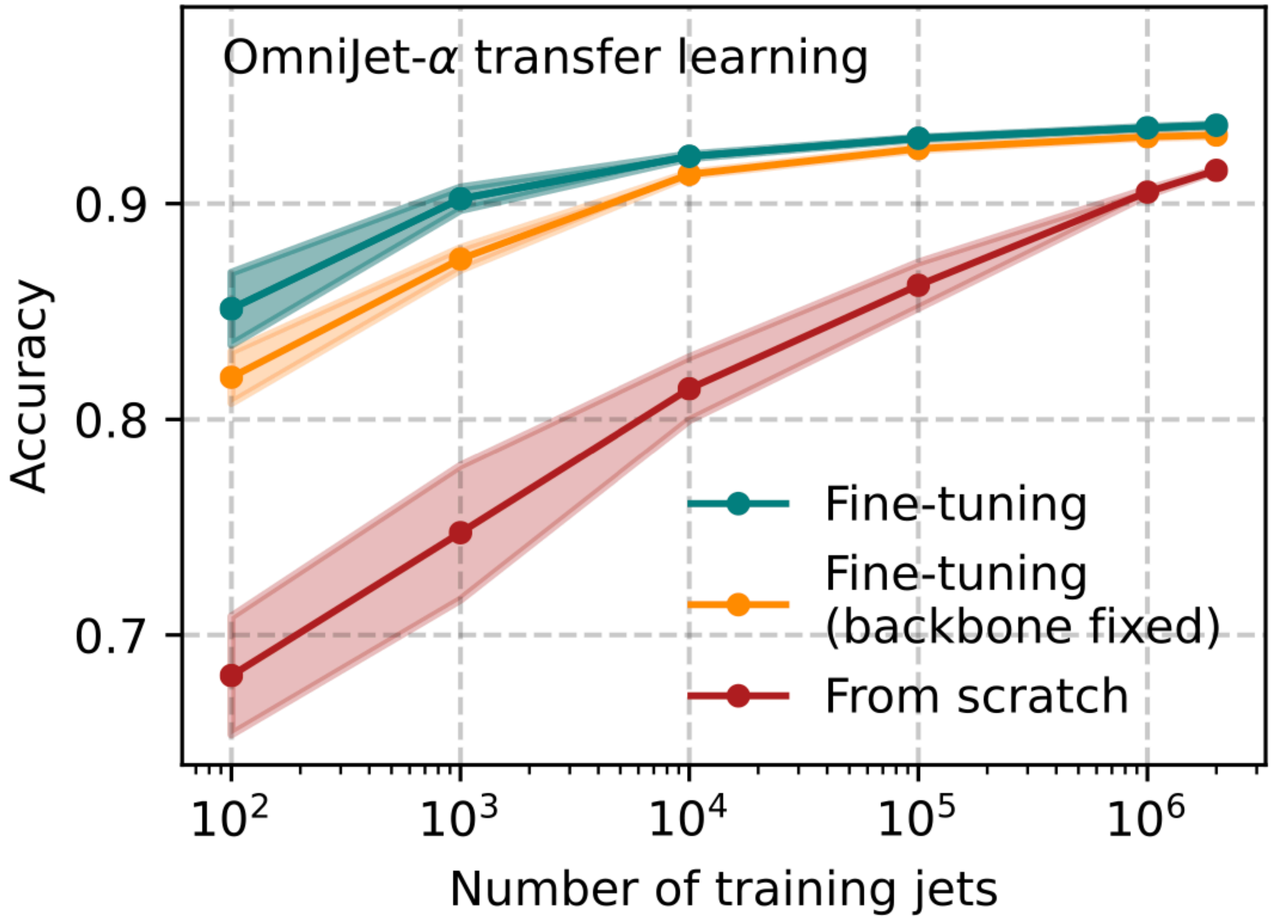
- **OmniJet-α**
  - ➡Transformer-based, next token prediction
  - ➡Input: jet constituents kinematics
  - ➡pre-train with **6.7 million** parameters
  - ➡downstream task: hadronic top quark tagging



$$\text{Jet} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$$
$$\vec{p}_i = (p_\text{T}, \Delta\eta, \Delta\phi)$$

Other works:
ParT arXiv:2202.03772
MPM arXiv:2401.13537

Mach.Learn.Sci.Tech. 5 (2024) 3, 035031

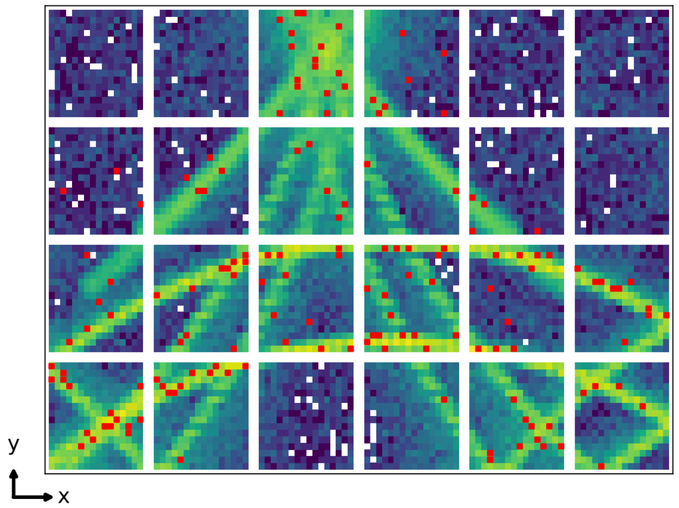


Foundation Models for low-level detector data

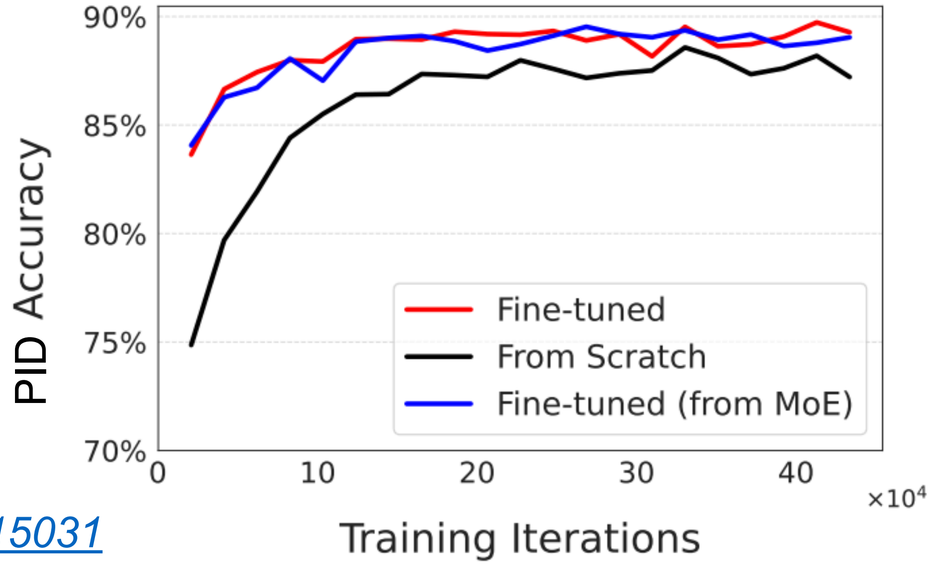
- Autoregressive Transformer
- Tokenization: **ePIC hpDIRC Cherenkov detector hits**

$$\text{spatial} \rightarrow \{|\vec{p}|, \theta, \text{SOS}_p, p_1, \ldots, p_n, \text{EOS}_p\}$$
$$\text{time} \rightarrow \{|\vec{p}|, \theta, \text{SOS}_t, t_1, \ldots, t_n, \text{EOS}_t\}$$

- Downstream tasks
  - ➡ Detector simulation
  - ➡ Noise filtering
  - ➡ Particle Identification (PID)



- Related **Genesis Mission Phase I** project "*Mixture-of-Experts Foundation Models for Scalable Holistic Reconstruction, Modeling, and Interpretation of Particle Interactions*" by W&M, JLab, BNL, SLAC



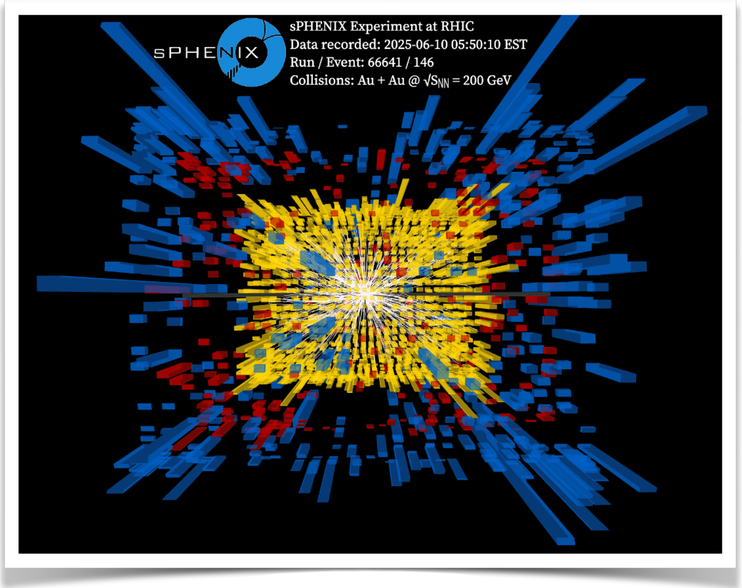
Machine Learning: Science and Technology 7.1 (2026): 015031

Foundation Model for Nuclear and Particle Physics

- Foundation Model for Nuclear and Particle Physics (**FM4NPP**): a **self-supervised** learning with inputs on ***unlabeled sparse detector-level data***

  ➡ *low-level spacepoint (E, x, y, z) in time projection chamber*

- lightweight **Adapter** for multiple downstream tasks



D. Park, S. Li et al, arXiv:2508.14087

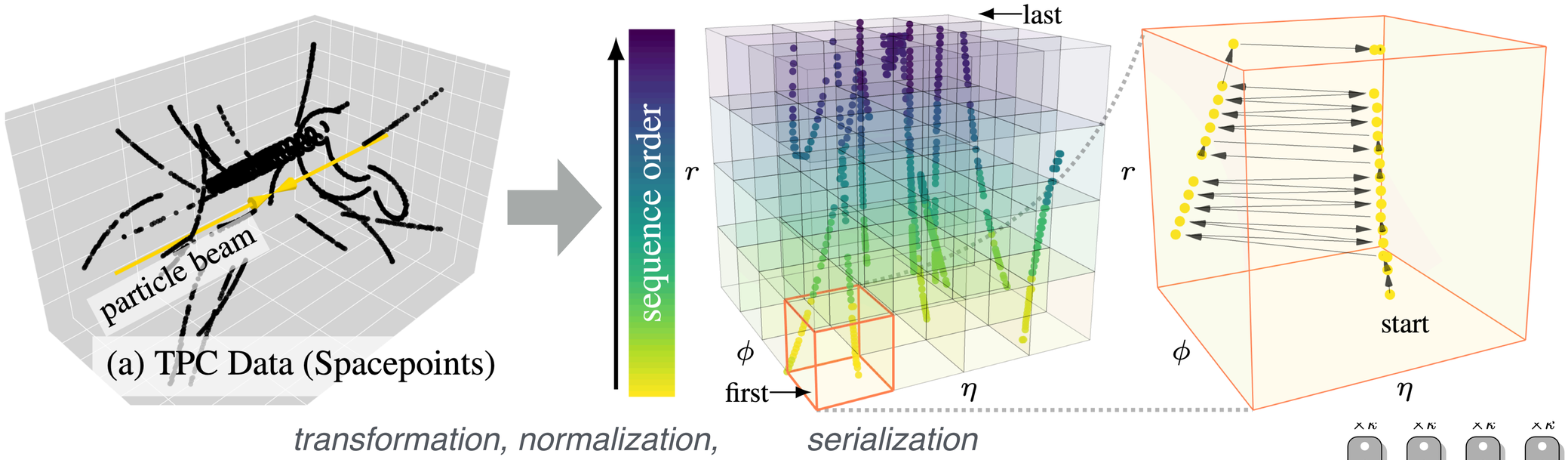
Pre-training → FM4 **NPP** → **minimal training** →

Various Downstream Tasks

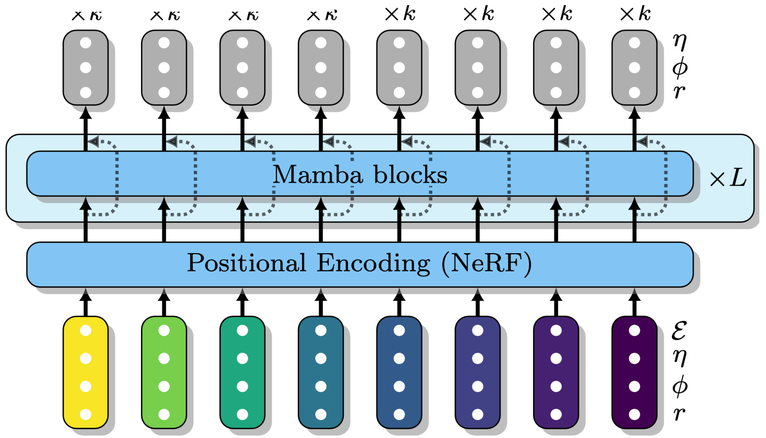
- track finding
- particle identification
- noise removal
- global reconstruction
- jet tagging
- event classification
- anomaly detection
- ⋮
- **AI-driven discovery**

FM4NPP Training

- Initial study by training with simulated Time Projection Chamber spacepoints (clusters) in 10 million *p*+*p* collision events



- Self-supervised **Next k-nearest-neighbor** prediction with Mamba model

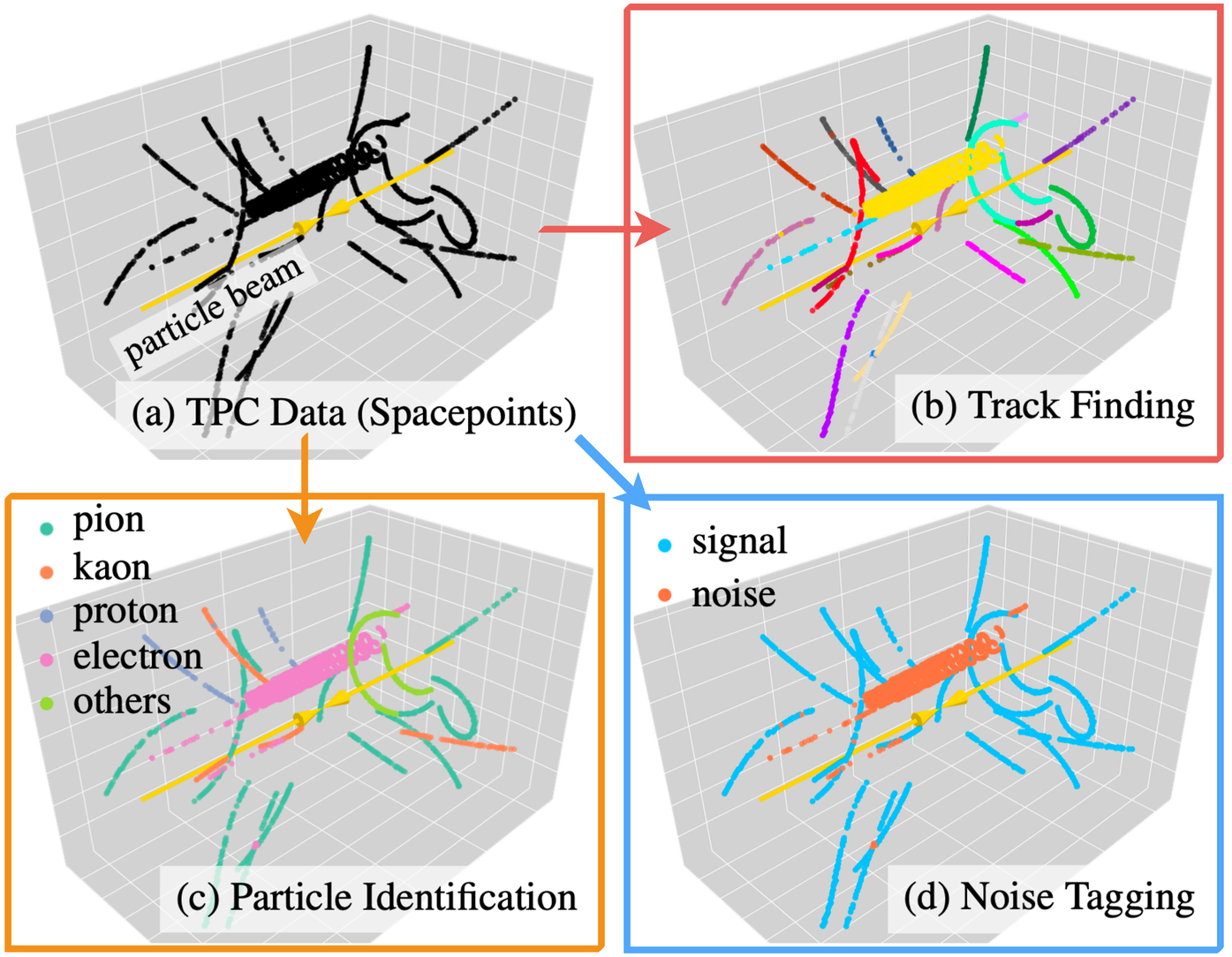


[D. Park, S. Li et al, arXiv:2508.14087]

FM4NPP: Downstream Tasks

- The FM backbone is frozen, and only *lightweight decoder* heads are trained
  - ➡ **Track Finding**: group spacepoints from the same particle trajectory
  - ➡ **Particle Identification**: classify spacepoints based on their particle species
  - ➡ **Noise Tagging**: classify spacepoints from low momentum secondary particles (“noise”)

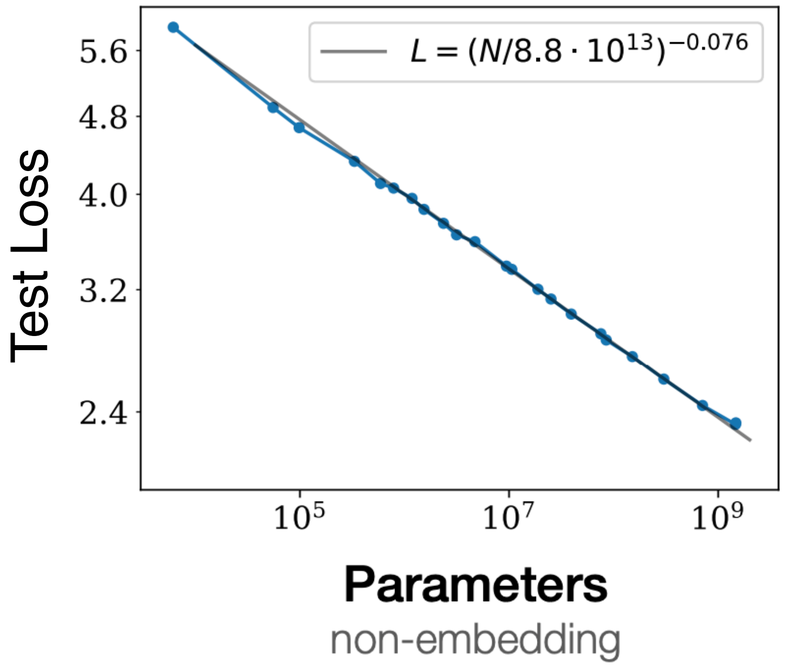
Note: a broader range of downstream tasks will become available as additional detectors are incorporated into the foundation model

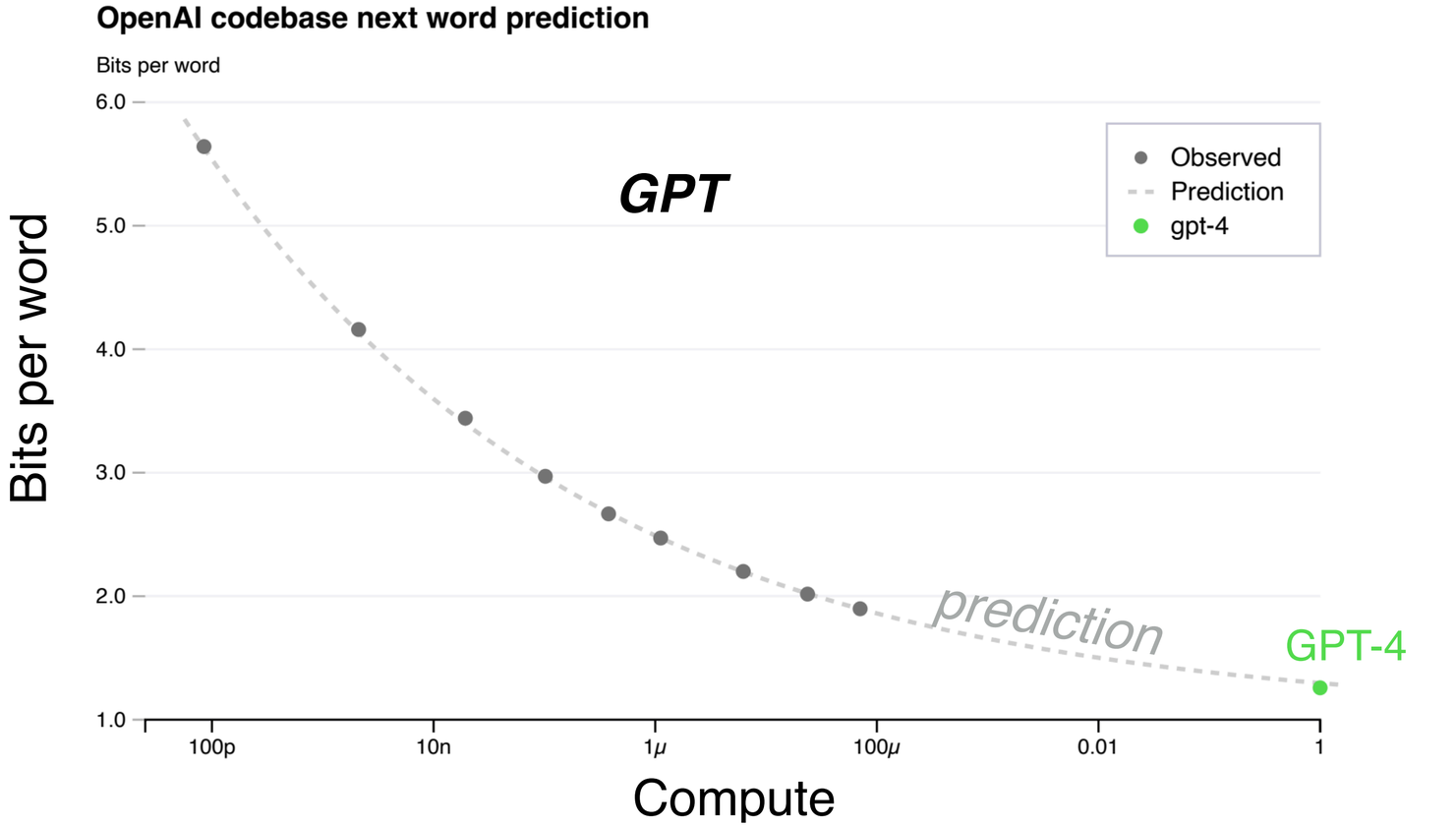


D. Park, S. Li et al, arXiv:2508.14087

Neural Scaling Laws in LLM

- Large language model studies established that performance improves predictably with model size, dataset size, and compute, often following ***power law scaling***





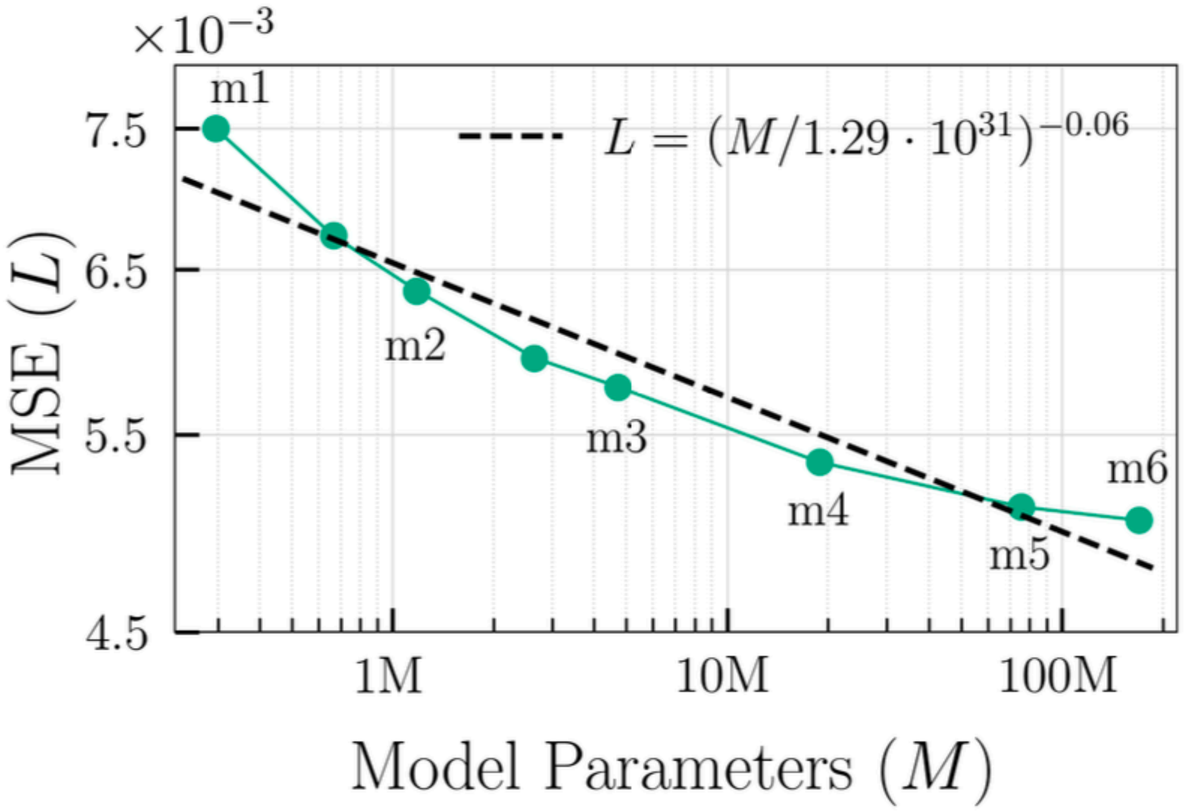
FLOPS: FLoating-point Operations Per Second

"Scaling Laws for Neural Language Models" arXiv:2001.08361

Achiam, Josh, et al. "Gpt-4 technical report." arXiv:2303.08774 (2023)

FM4NPP: Neural Scaling Behavior

D. Park, S. Li et al, arXiv:2508.14087

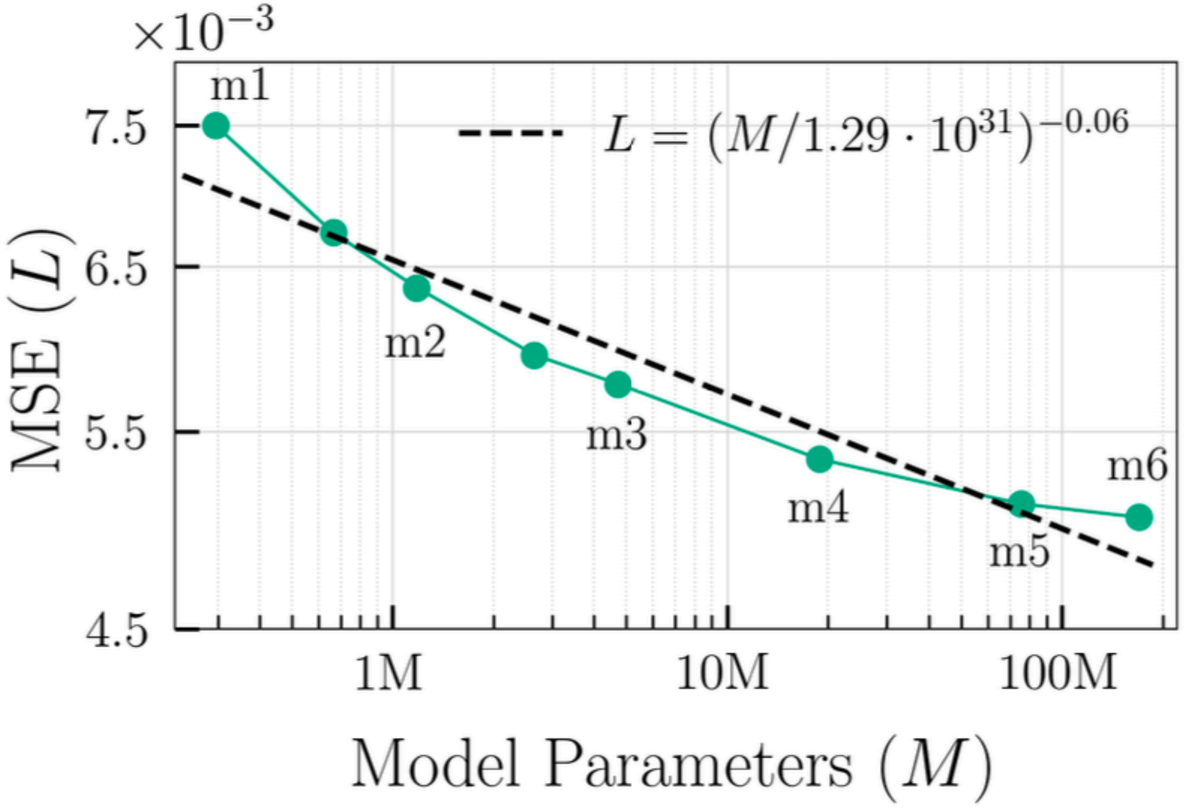


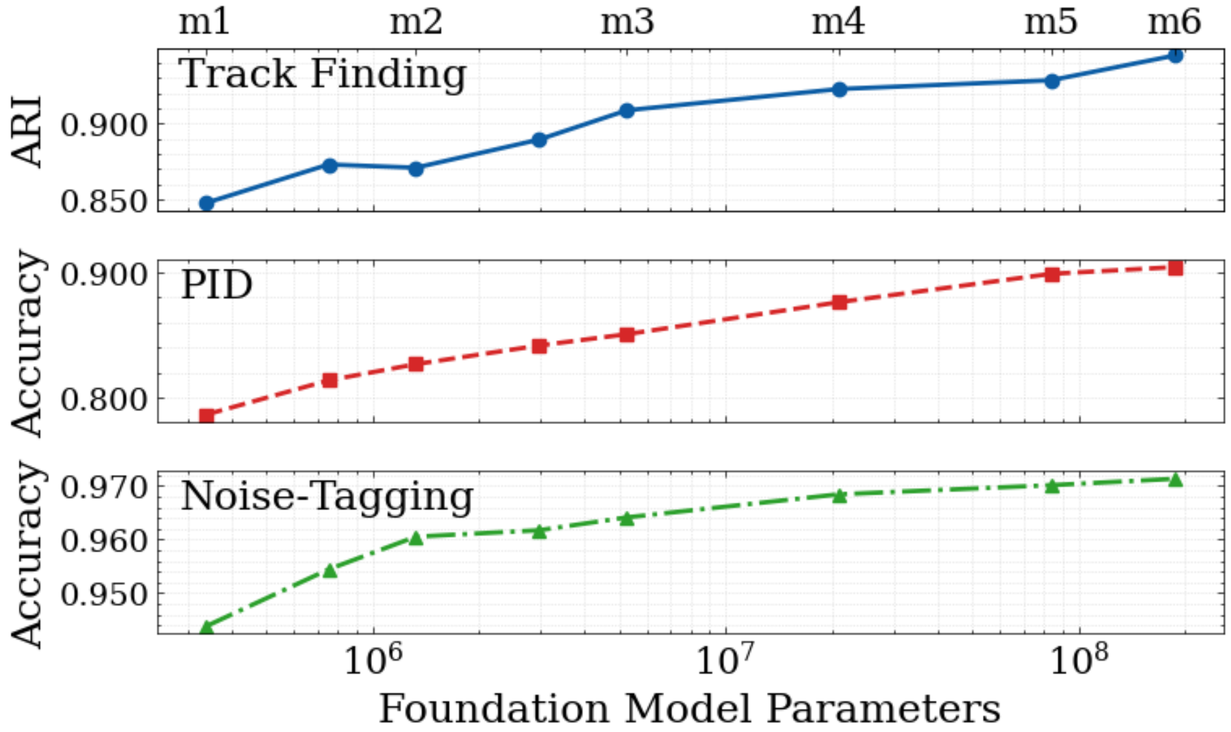
- Established ***self-supervised loss scaling behavior*** of **FM4NPP** versus ***model parameters***, ***data size***, *and* ***compute***

  ➡ *Consistent with scaling observed in LLMs*

FM4NPP: Neural Scaling Behavior

D. Park, S. Li et al, arXiv:2508.14087





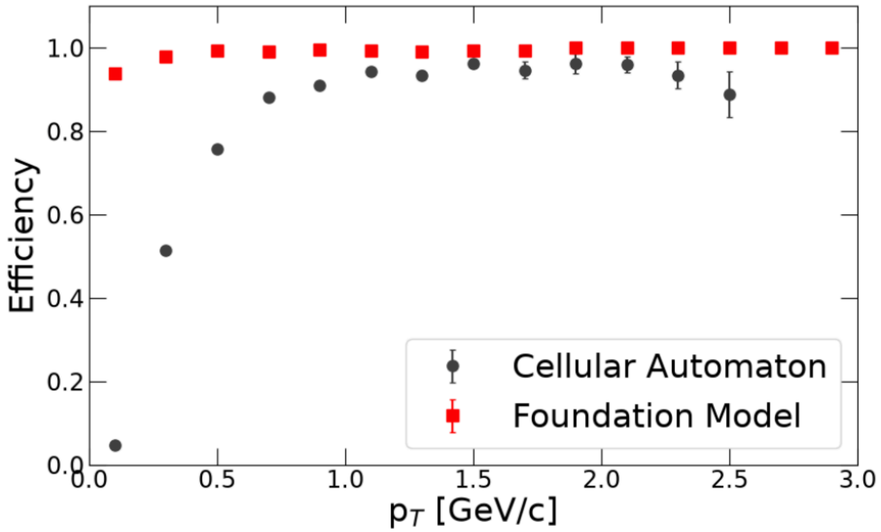
- Established ***self-supervised loss scaling behavior*** of **FM4NPP** versus ***model parameters***, ***data size***, *and* ***compute***
  - ➡ *Consistent with scaling observed in LLMs*
- Demonstrated **scaling behavior for downstream tasks** scaled with foundation model size and data size

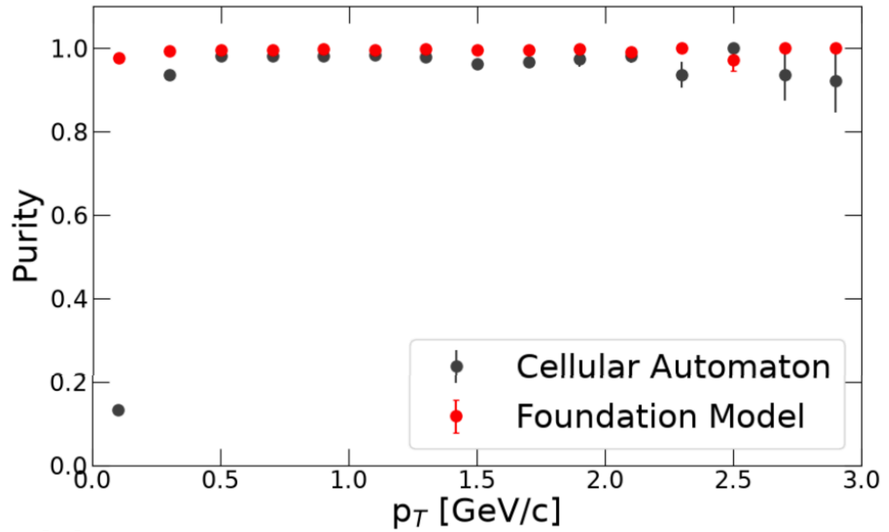
FM4NPP: Downstream Performance

arXiv:2610.31180

model	#trnbl para.	Track Finding ARI↑	efficiency↑	purity↑
EggNet	0.16M	0.7256	74.19%	75.14%
Exa.TrkX	3.86M	0.8765	91.79%	66.42%
AdapterOnly	2.39M	0.7243	78.01%	64.54%
FM4NPP (m6)	2.39M	**0.9448**	**96.08%**	**93.08%**

model	#trnbl para.	Particle Identification acc.↑	recall↑	pre.↑	Noise Tagging acc.↑	recall↑	pre.↑
SAGEConv	0.91M	0.7262	0.4563	0.6502	0.9174	0.7227	0.8165
OneFormer3D	44.95M	0.7701	0.4897	0.5767	0.9646	**0.9404**	0.8948
AdapterOnly	0.74M	0.6631	0.3387	0.6111	0.9111	0.6215	0.8359
FM4NPP (m6)	0.74M	**0.9039**	**0.7652**	**0.8782**	**0.9713**	0.9367	**0.9190**





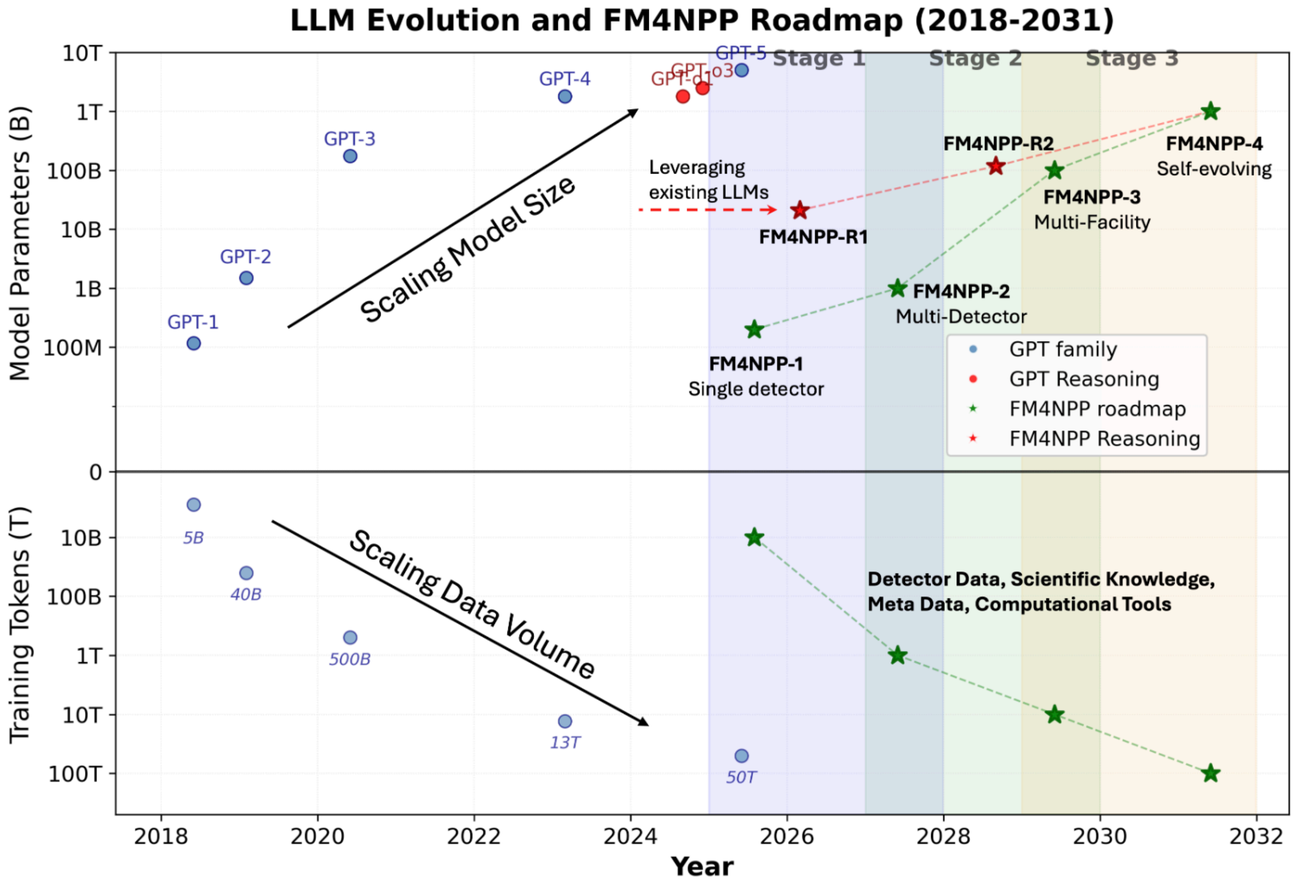
- **Superior performance** of tracking with FM4NPP compared to other state-of-art ML models
- **Outperform the traditional method**
  - ➡ Higher tracking efficiency and purity across the entire p_T range, with particularly significant improvements at p_T < 1 GeV

FM4NPP: Future Directions

- **Current stage**
  - ➡ initial stage (FM4NPP-1) utilized single detector (sPHENIX TPC simulation) with **180M** parameters
  - ➡ Similar scaling behavior, which was first observed in ChatGPT-2

- **3 Genesis Mission Phase I projects**
  - ➡ **SBU&BNL&ANL**: Transferable tracking directly from raw detector data, with expansion to multiple facilities, including JLab experiments such as MUSE and DarkLight
  - ➡ **LLNL&BNL&UCB&CU**: Multimodal extensions integrating tracking and calorimetry
  - ➡ **MIT&BNL&ISU**: Extension to silicon tracking detectors and applications to legacy e^+e^- collider data, such as DELPHI and ALEPH

Roadmap Whitepaper
IJMPA 41, 15n16, 2650086 (2026)
[ssrn.5467666]



Agentic AI

- AI has the potential to **increase scientific productivity** across the full workflow
- This includes coding, analysis design, analysis execution, validation, and scientific writing

Resummation of the C-Parameter Sudakov Shoulder Using Effective Field Theory

Matthew D. Schwartz[1,2]

[1]*Department of Physics, Harvard University, Cambridge, MA 02138, USA*
[2]*Institute for Artificial Intelligence and Fundamental Interactions (IAIFI)*

schwartz@g.harvard.edu

AI Research Assistant: Claude Opus 4.5 (Anthropic)

January 7, 2026

Abstract

The C-parameter distribution in e^+e^- annihilation exhibits a kinematic shoulder at $C = 3/4$, where three-parton final states reach their maximum and a fourth parton is required to exceed it. This boundary generates large logarithms that must be resummed. Using soft-collinear effective theory, we derive a factorization theorem involving new jet and soft functions specific to the C-parameter measurement, in which soft radiation contributes quadratically in transverse momentum. This quadratic structure explains the step discontinuity at leading order. We compute all ingredients at one loop, validate against Monte Carlo, and present matched NLL+NLO results. Unlike thrust and heavy jet mass, the C-parameter has no Sudakov–Landau pole, making momentum-space resummation straightforward. All calculations, numerical analysis, and manuscript preparation were performed by Claude, an AI assistant developed by Anthropic, working under physicist supervision.

[M. Schwartz arXiv:2601.02484]

Agentic AI – Physicist Collaboration in Experimental Particle Physics: A Proof-of-Concept Measurement with LEP Open Data

Anthony Badea,[a] Yi Chen,[b] Marcello Maggi,[c] Yen-Jie Lee,[d] Electron-Positron Alliance

[a]*University of Chicago, Enrico Fermi Institute, 60637 IL, USA*
[b]*Vanderbilt University, Nashville, Tennessee, USA*
[c]*Istituto Nazionale di Fisica Nucleare, Bari Division, BA 70126, Italy*
[d]*Laboratory for Nuclear Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

E-mail: anthony.badea@cern.ch

Abstract: We present an AI agentic measurement of the thrust distribution in e^+e^- collisions at $\sqrt{s} = 91.2$ GeV using archived ALEPH data. The analysis and all note writing is carried out entirely by AI agents (OpenAI Codex and Anthropic Claude) under expert physicist direction. A fully corrected spectrum is obtained via Iterative Bayesian Unfolding and Monte Carlo based corrections. This work represents a step toward a theory–experiment loop in which AI agents assist with experimental measurements and theoretical calculations, and synthesize insights by comparing the results, thereby accelerating the cycle that drives discovery in fundamental physics. Our work suggests that precision physics, leveraging the open LEP data and advanced theoretical landscape, provides an ideal testing ground for developing advanced AI systems for scientific applications.

[A. Badea, Y. Chen et al, arXiv:2603.05735]

Automating High Energy Physics Data Analysis with LLM-Powered Agents

Eli Gendreau-Distler,[1,2,*] Joshua Ho,[1,2,†] Dongwon Kim,[1,2,‡] Luc Tomas Le Pottier,[1,2,§] Haichen Wang,[1,2,¶] and Chengxi Yang[1,2,**]

[1]*Department of Physics, University of California, Berkeley, Berkeley, CA 94720, USA*
[2]*Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*

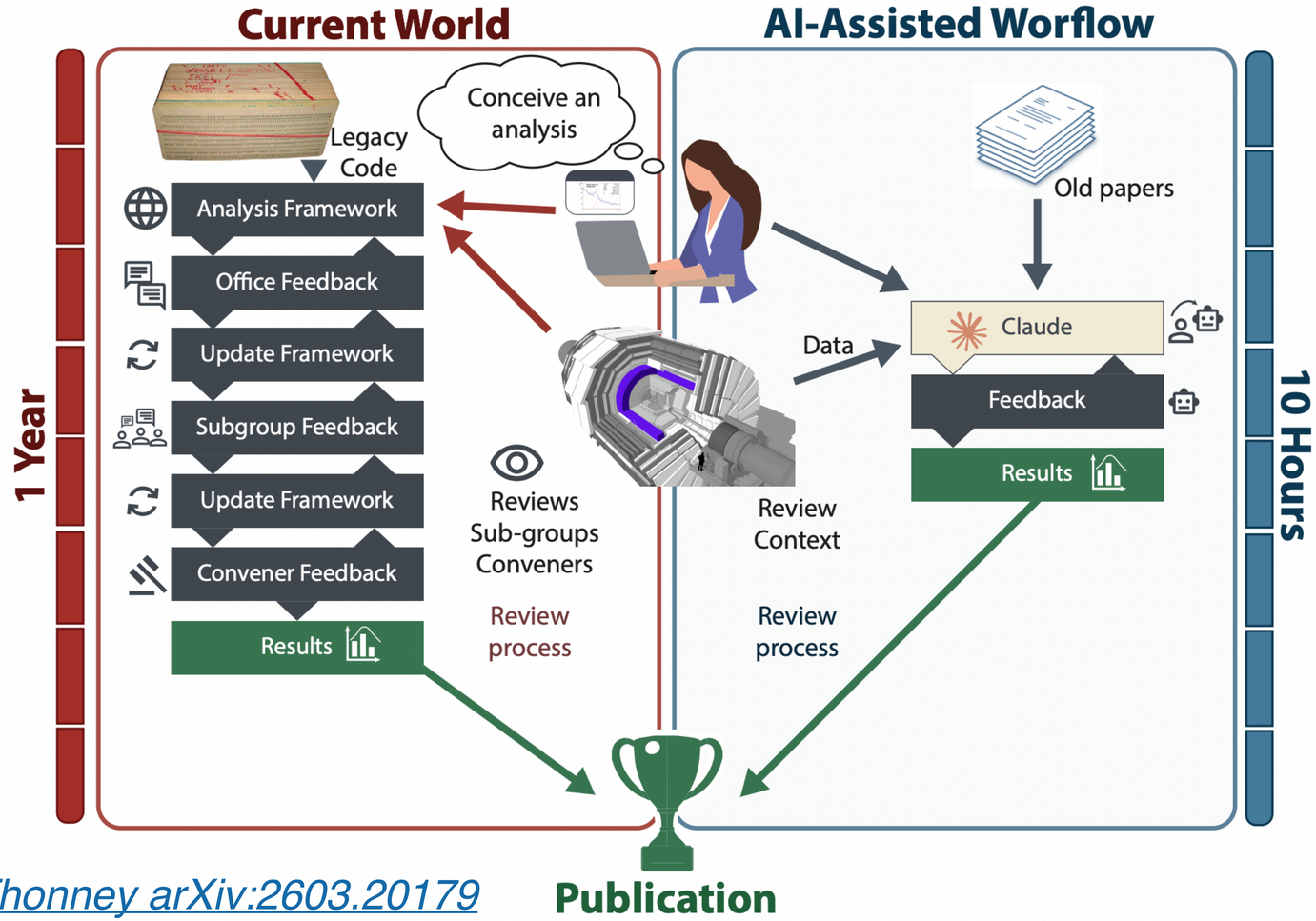
We present a proof-of-principle study demonstrating the use of large language model (LLM) agents to automate a representative high energy physics (HEP) analysis. Using the Higgs boson diphoton cross-section measurement as a case study with ATLAS Open Data, we design a hybrid system that combines an LLM-based supervisor–coder agent with the `Snakemake` workflow manager. In this architecture, the workflow manager enforces reproducibility and determinism, while the agent autonomously generates, executes, and iteratively corrects analysis code in response to user instructions. We define quantitative evaluation metrics – success rate, error distribution, costs per specific task, and average number of API call for the task – to assess agent performance across multi-stage workflows. To characterize variability across architectures, we benchmark a representative selection of state-of-the-art LLMs, spanning the *Gemini* and *GPT-5* series, the *Claude* family, and leading open-weight models. While the workflow manager ensures deterministic execution of all analysis steps, the final outputs still show stochastic variation. Although we set the temperature to zero, other sampling parameters (e.g., top-p, top-k) remained at their defaults, and some reasoning-oriented models internally adjust these settings. Consequently, the models do not produce fully deterministic results. This study establishes the first LLM-agent-driven automated data-analysis framework in HEP, enabling systematic benchmarking of model capabilities, stability, and limitations in real-world scientific computing environments. The baseline code used in this work is available at https://huggingface.co/HWresearch/LLM4HEP. This work was accepted as a poster at the Machine Learning and the Physical Sciences (ML4PS) workshop of the 39th Conference on Neural Information Processing Systems (NeurIPS 2025). The initial submission was made on August 30, 2025.

[E. Gendreau-Distler et al, arXiv:2512.07785]

Agentic AI for Data Analysis

- AI has the potential to **increase scientific productivity** across the full workflow
- This includes coding, analysis design, analysis execution, validation, and scientific writing

- ***Key consideration:***
  - ➡ How do we identify and control biases from AI models?
  - ➡ How do we reliably maintain scientific quality?
  - ➡ How do we review and assess a rapidly growing volume of AI-assisted work?
  - ➡ …

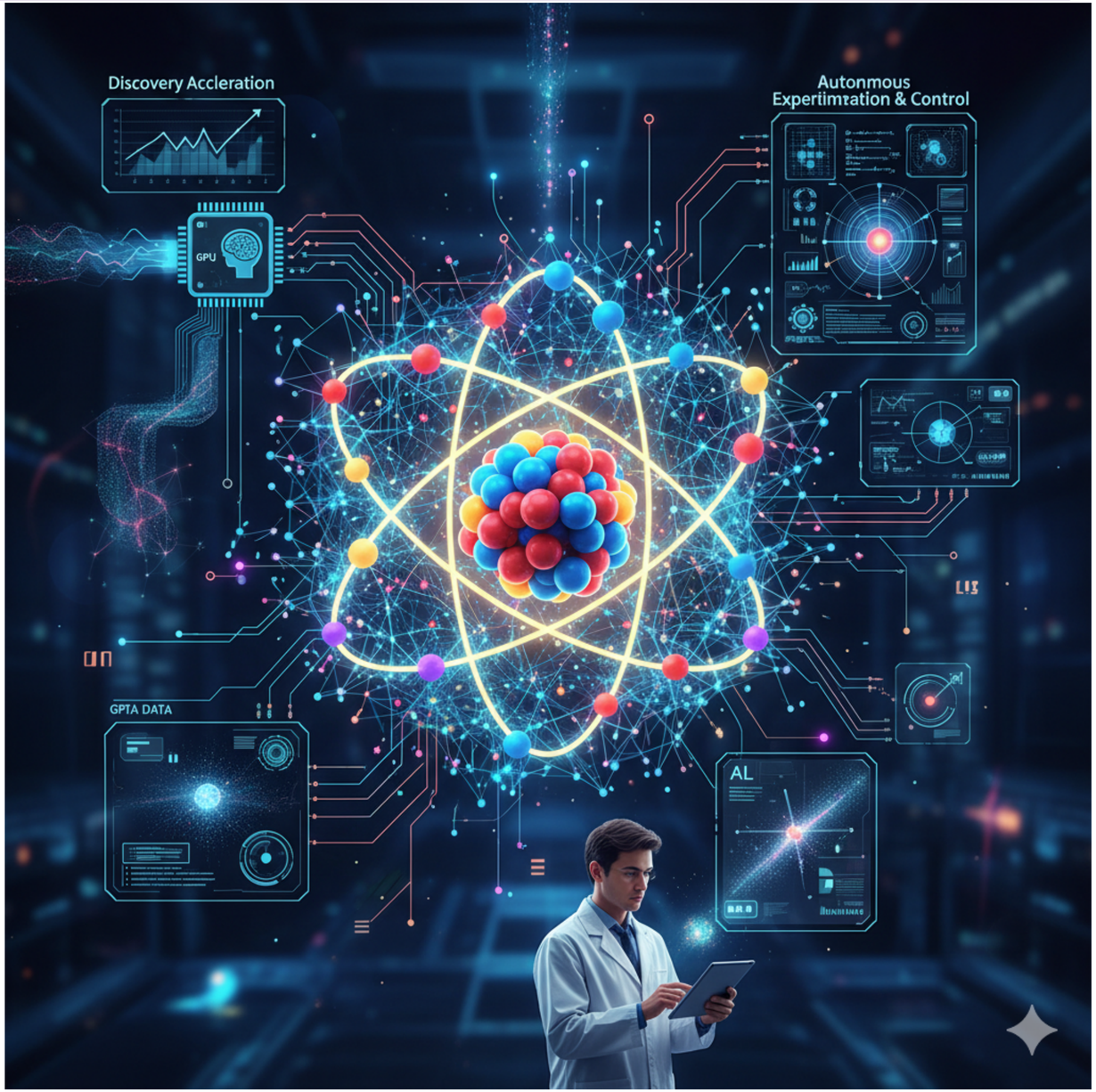


E. Moreno, S. Thonney arXiv:2603.20179

AI/ML in Nuclear Physics

AI/ML is transforming QCD experiments !

*With careful validation and responsible use, AI/ML will open a **new paradigm** for experimental discovery, offering **unprecedented opportunities** to explore physics !*

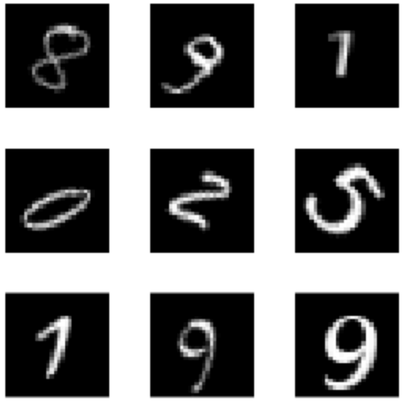


Backup

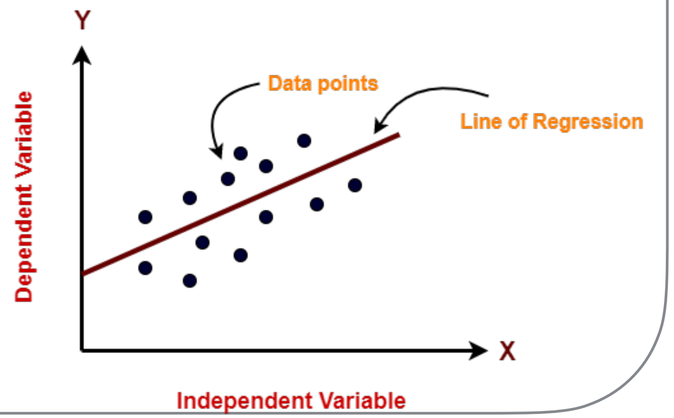
Types of Machine Learning

Supervised

- Use **labelled** data
- **Input-output values** are given by human
- Forecast outcomes
- Classification ……

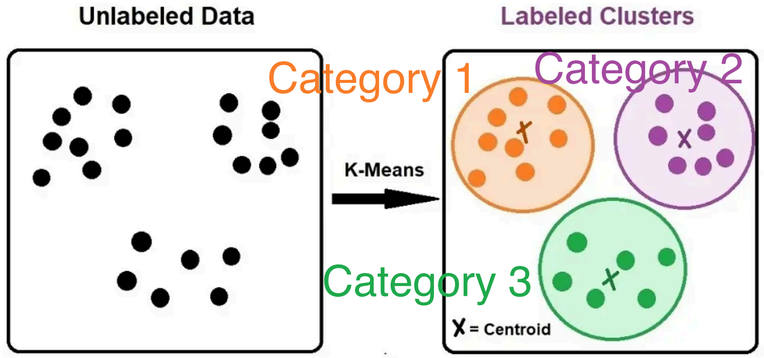


- Regression ……



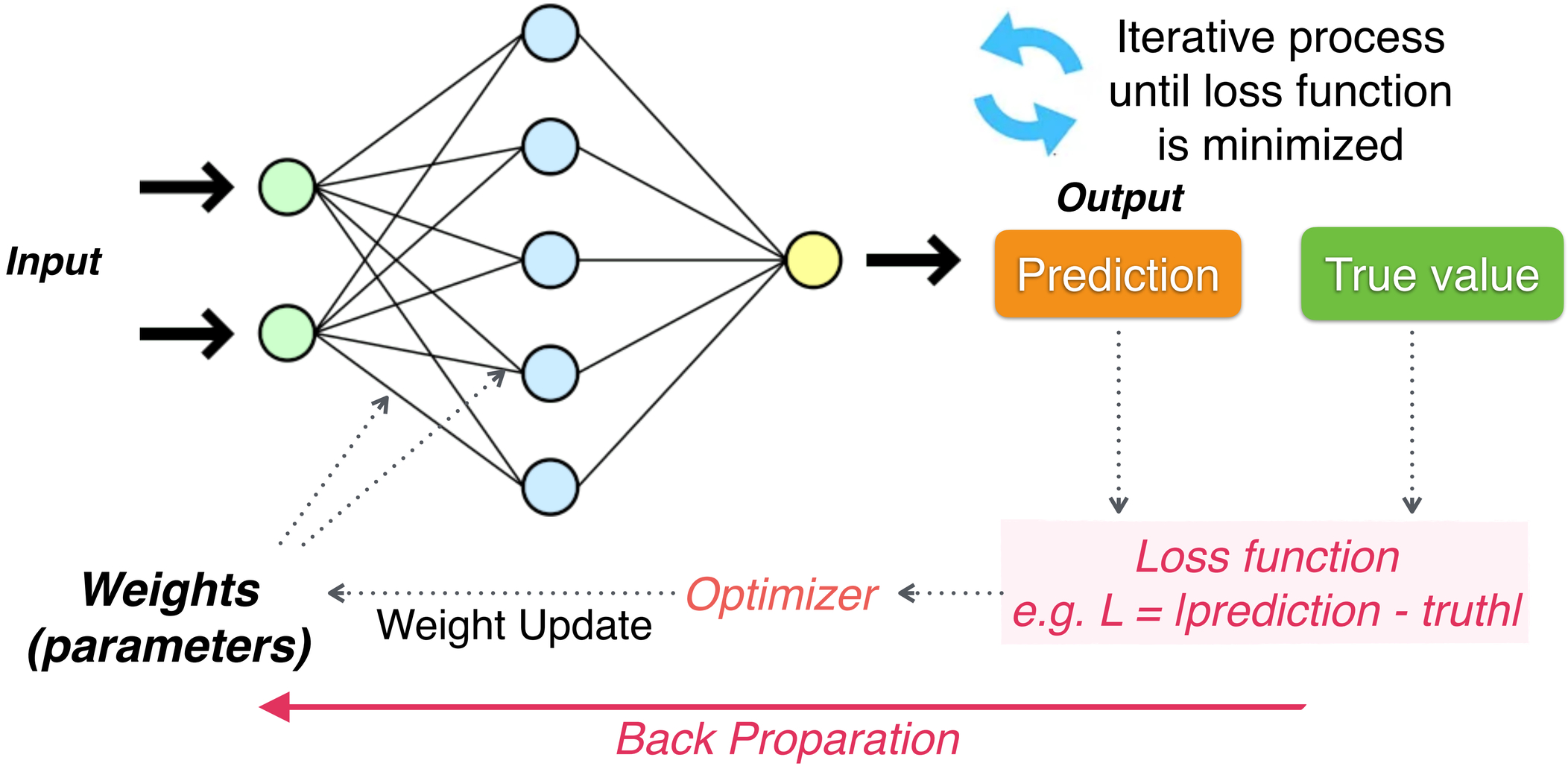
Unsupervised

- Use **unlabelled** data
- Find **hidden patterns** w/o human intervention
- Clustering



- Association
- Dimension Reduction
- Anomaly detection

Neural Network



Many layers: Multi Layer Perceptron (MLP), Deep Neural Network (DNN)

Various Neural Network Architecture

The different structure (architectures) based on how they organize, connect, and process inputs before or between MLPs

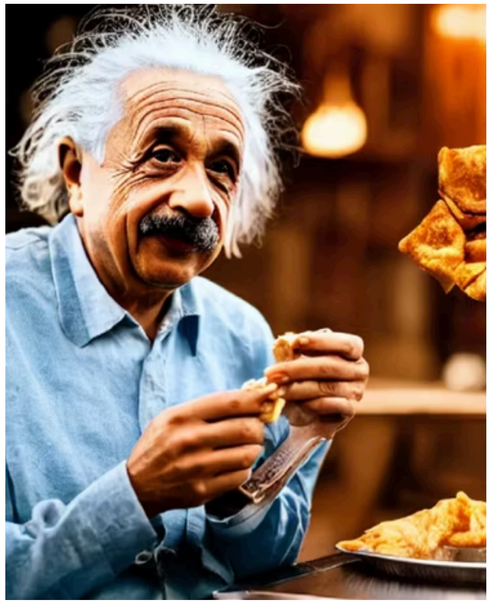
- **Convolutional Neural Network (CNN)** : good for image processing
- **Graph Neural Network (GNN)** : good for irregular, relational data
- **Recurrent Neural Network (RNN)** : good for sequential data
- **Transformer** : learns “attention”
- **Autoencoder** : encode → latent space → decode
- …

Generative AI

- ***Generates syntactic data e.g. text, image, audio, video, …***
- *Generally more advanced and complex AI*
  - ➡ combines representation learning (understanding) with generation (creation)
- *Models*
  - ➡ Variational Autoencoder (VAE)
  - ➡ Generative Adversarial Network (GAN)
  - ➡ Diffusion models
  - ➡ Transformer-based Models, e.g. Large Language Model
  - ➡ …

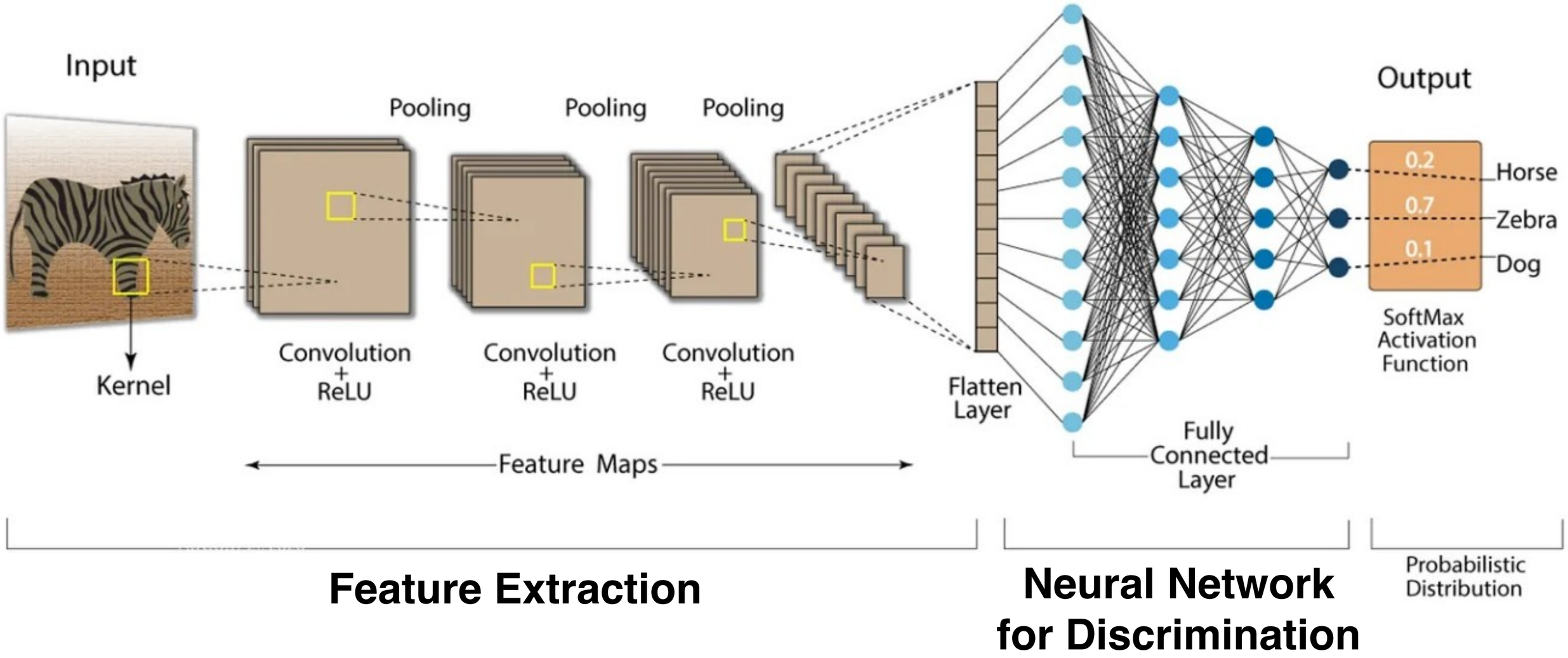
Images from Dalle-2 (Diffusion Model)





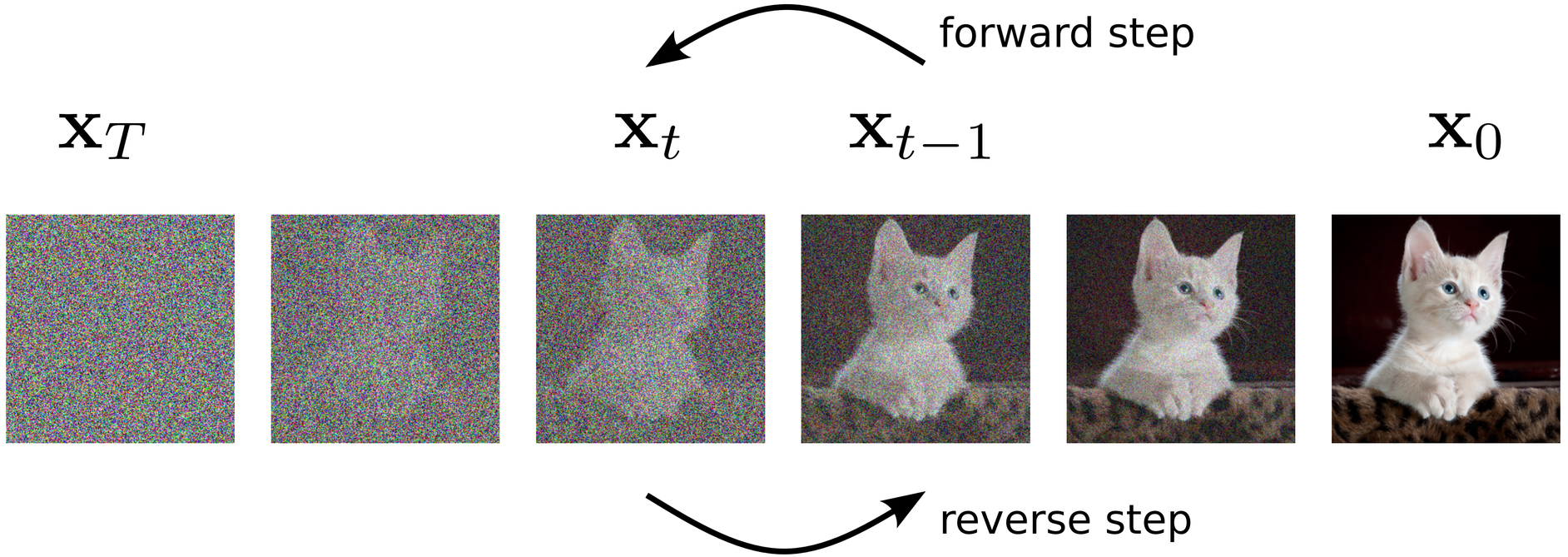
Convolutional Neural Network (CNN)

- Convolutional Neural Network (**CNN**) → good for image processing

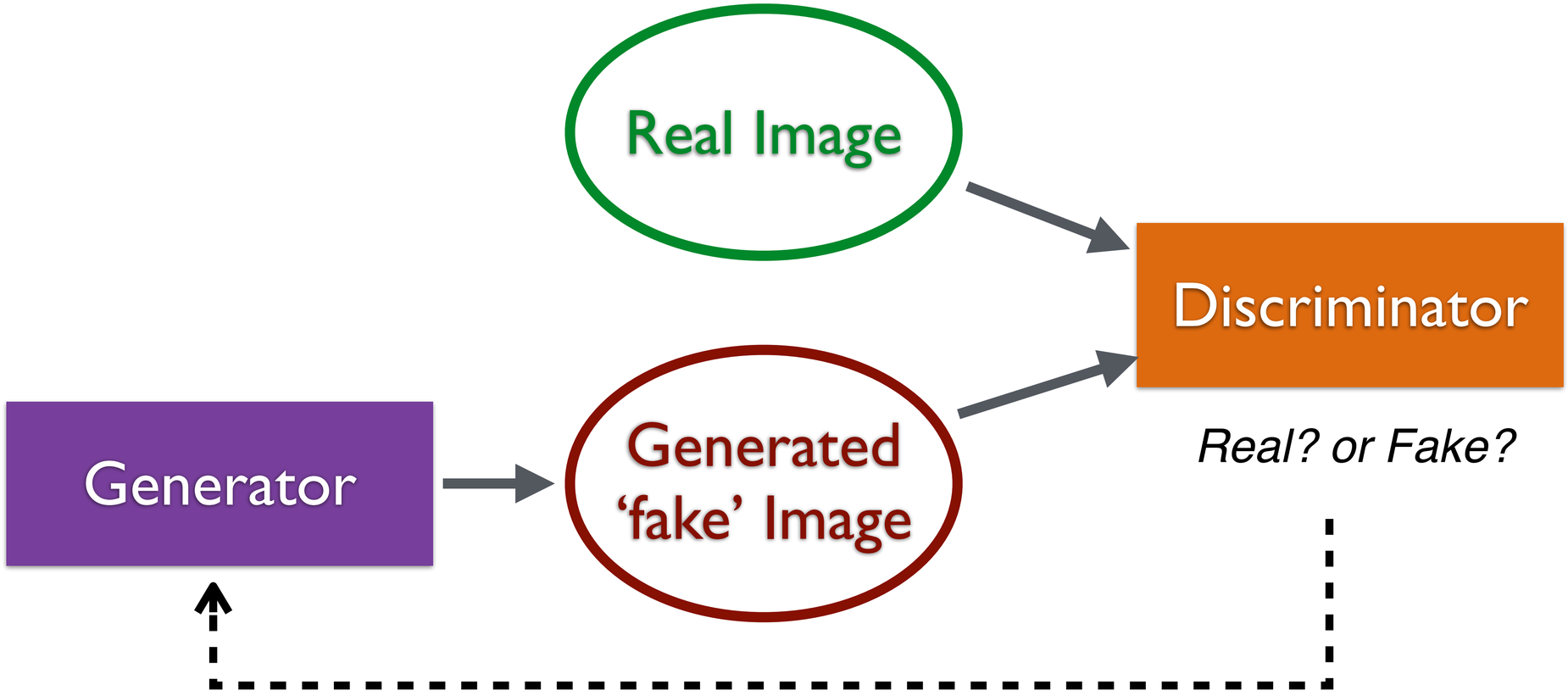


Denoising Diffusion Probabilistic Model (DDPM)

- **Forward** process: gradually adds Gaussian noise to the data
- **Reverse** process: a neural network learns to reverse this process; start from pure noise, gradually remove noise using the learned denoising model; generate data similar to the original images
- In real application, O(1,000) steps are used for high quality



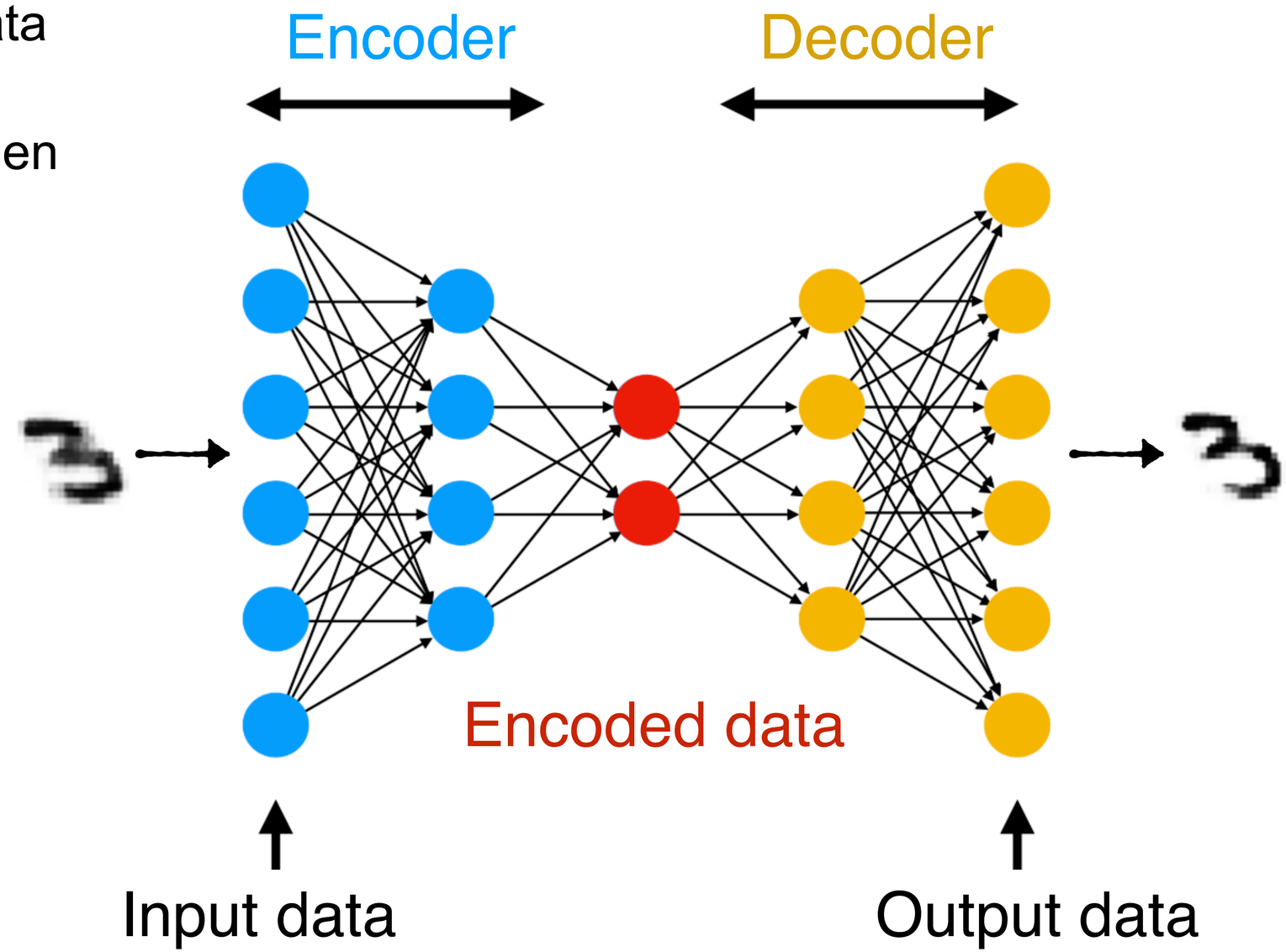
Generative Adversarial Network (GAN)



- Two neural networks (**Generator** and **Discriminator**) *compete* against each other
  - ➡ ***feedback*** to each other to improve
  - ➡ the process is repeated until **Discriminator** can no longer distinguish fake generated by **Generator** from real

AutoEncoder

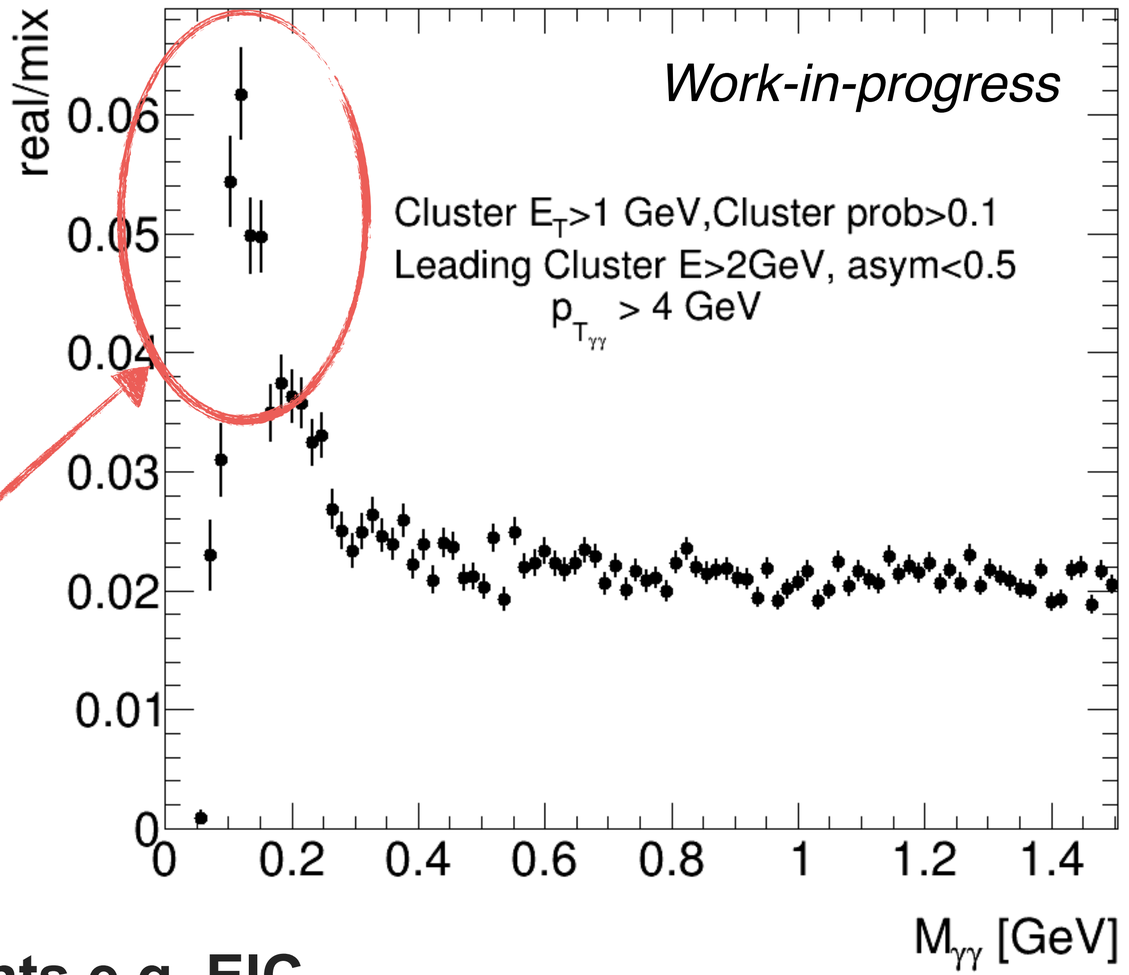
- **Autoencoder**: compress input data into a lower-dimensional representation (**encoding**) and then reconstruct it (**decoding**)
- Application:
  - *data compression*
  - *feature learning*
  - *anomaly detection*
  - *dimensionality reduction*
  - *image denoising*
  - *…*



Emerging Phenomena Captured by AI

- Can DDPM describe more complex features of heavy ion collisions?
  - ➡ **Resonance, flow** can be reproduced by DDPM!

π^0 peak reconstructed using simulation samples generated by DDPM



- **Similar approach applicable to other experiments e.g. EIC**

Jet Background Subtraction Using ML

- Previous works on jet bkg. subtraction using **ML** showed improved jet energy resolution, however,
  - ➡ ***supervised*** *learning*: regression task trained using jet substructure;
    trained on *unmodified jet* substructure, applied on *modified* jet substructure → **bias**
  - ➡ map to low-dimensional distributions quantity, i.e. predicting background-subtracted p_T,
    → don't preserve the full substructure information (e.g. substructure)

  - Multi-layer perceptron, 3 layers with 100 and 50 nodes

Feature	Score	Feature	Score
Jet p_T (no corr.)	**0.1355**	$p^1_{T,\,const}$	0.0012
Jet mass	0.0007	$p^2_{T,\,const}$	**0.0039**
Jet area	0.0005	$p^3_{T,\,const}$	0.0015
Jet p_T (area-based corr.)	**0.7876**	$p^4_{T,\,const}$	0.0011
LeSub	0.0004	$p^5_{T,\,const}$	0.0009
Radial moment	0.0005	$p^6_{T,\,const}$	0.0009
Momentum dispersion	0.0007	$p^7_{T,\,const}$	0.0008
Number of constituents	0.0008	$p^8_{T,\,const}$	0.0007
Mean of const. p_T	**0.0585**	$p^9_{T,\,const}$	0.0006
Median of const. p_T	0.0023	$p^{10}_{T,\,const}$	0.0007

UVCGAN

- **UVCGAN** (U-net Vision transformer Cycle-consistent Generative Adversarial Network)
  - ➡ **unpaired** image-to-image (I2I) translation; bridging gap between simulation and data reference
  - ➡ arXiv:2303.16280 [cs.CV]



arXiv:2303.16280

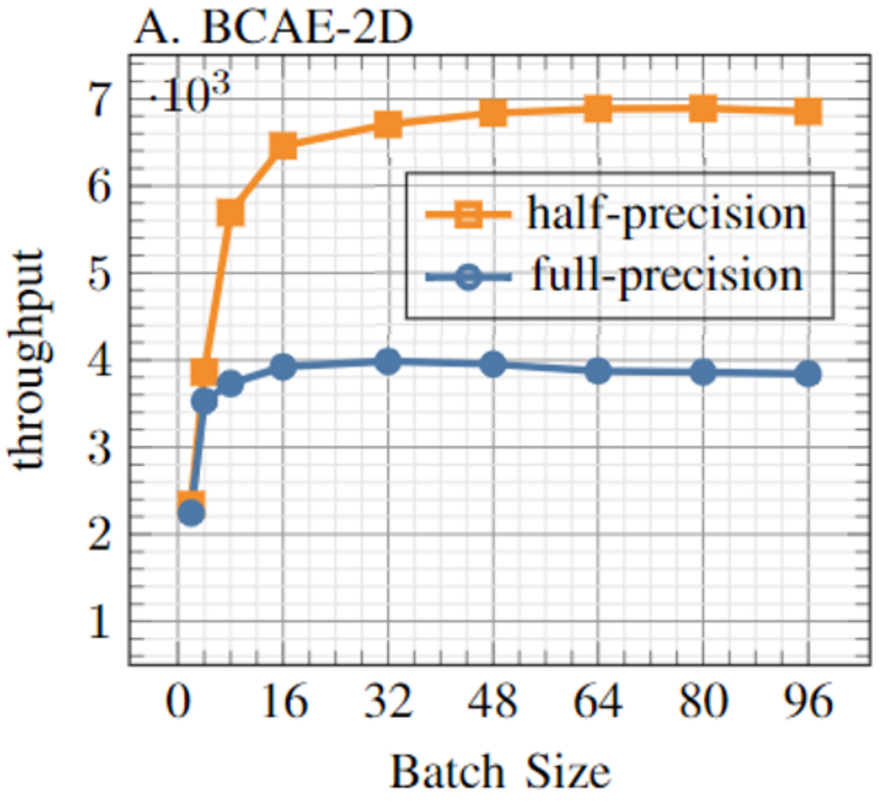
How Fast Can We Compress?

arXiv: 2310.15026

model	MAE ↓	PSNR ↑	precision ↑	recall ↑	encoder size ↓	throughput ↑
BCAE-2D	0.152	11.726	0.906	0.907	169.0k	~ 6.9k
BCAE[10]	0.198	9.923	0.878	0.861	201.7k	~ 2.4*k*

- **BCAE-2D**: treat the radial dimension as the feature dimension and use 2D convolutions for higher throughput → better throughput than 3D

- **Half-precision (16-bit floats)** for encoder weights and input affords more than 70% improvement in throughput compared to full-precision (32-bit floats).



Performance of BCAE-VS

arXiv: 2411.11942

Table 1: Comparing compression algorithm with dense and sparse convolutions.

		reconstruction performance					efficiency	
model	comp. ratio ↑	L_1 ↓	L_2 ↓	PSNR ↑	recall ↑	precision ↑	encoder size	throughput ↑
BCAE-2D	31.1	.152	.862	20.6	.907	.906	169k	**9.6k**
BCAE-HT (3D)	31.1	.138	.781	20.8	.916	.915	9.8k	**9.6k**
BCAE++ (3D)	31.1	.112	.617	21.4	.936	.934	226k	3.2k
BCAE-VS	**34.3**	**.030**	**.101**	**25.6**	**.986**	**.997**	**382**	5.6k

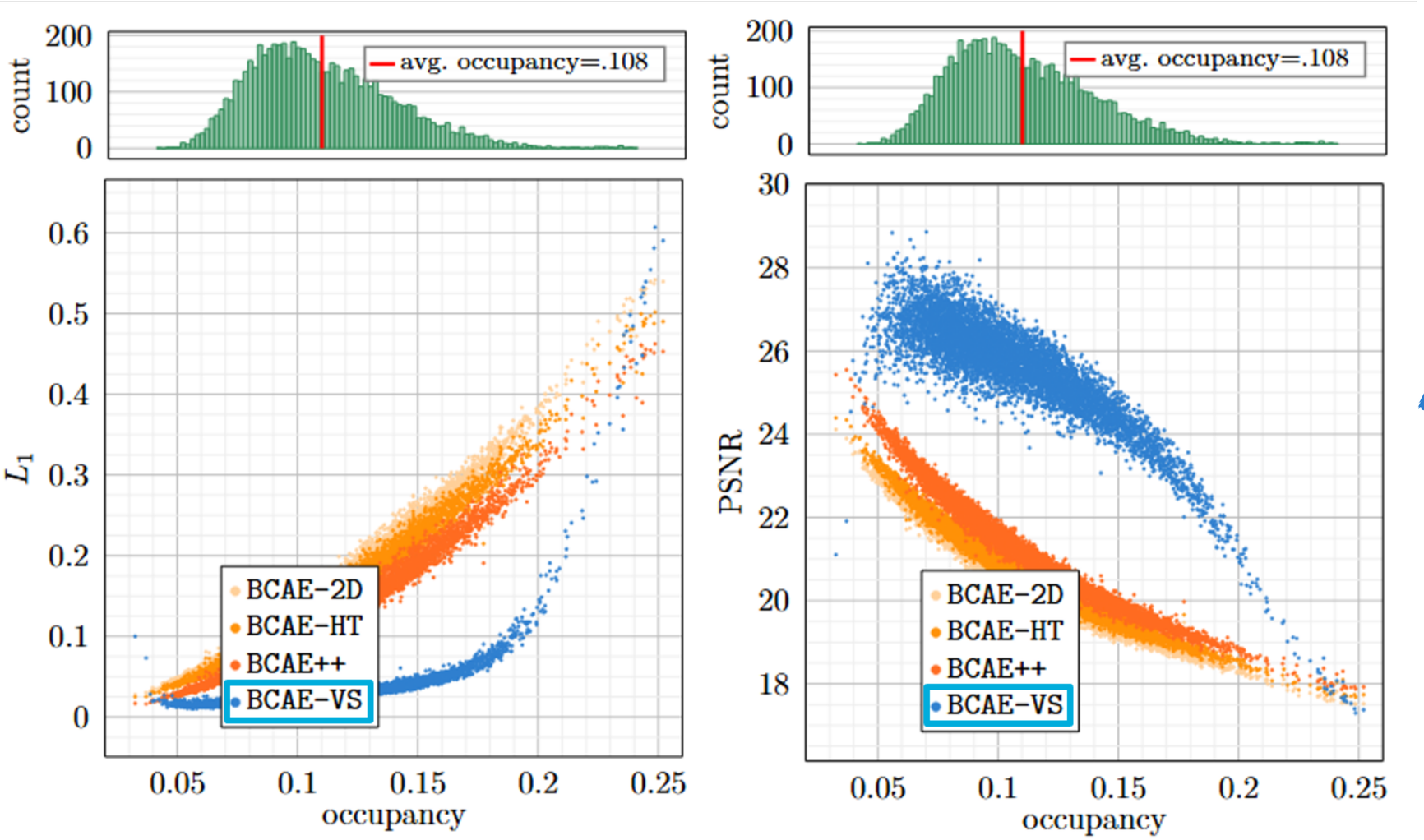
peak signal-to-noise ratio (PSNR)

$$\text{recoll} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positives}} = \frac{TP}{TP + FP}$$

Reconstruction Accuracy of BCAE-VS; Regression

arXiv: 2411.11942



better

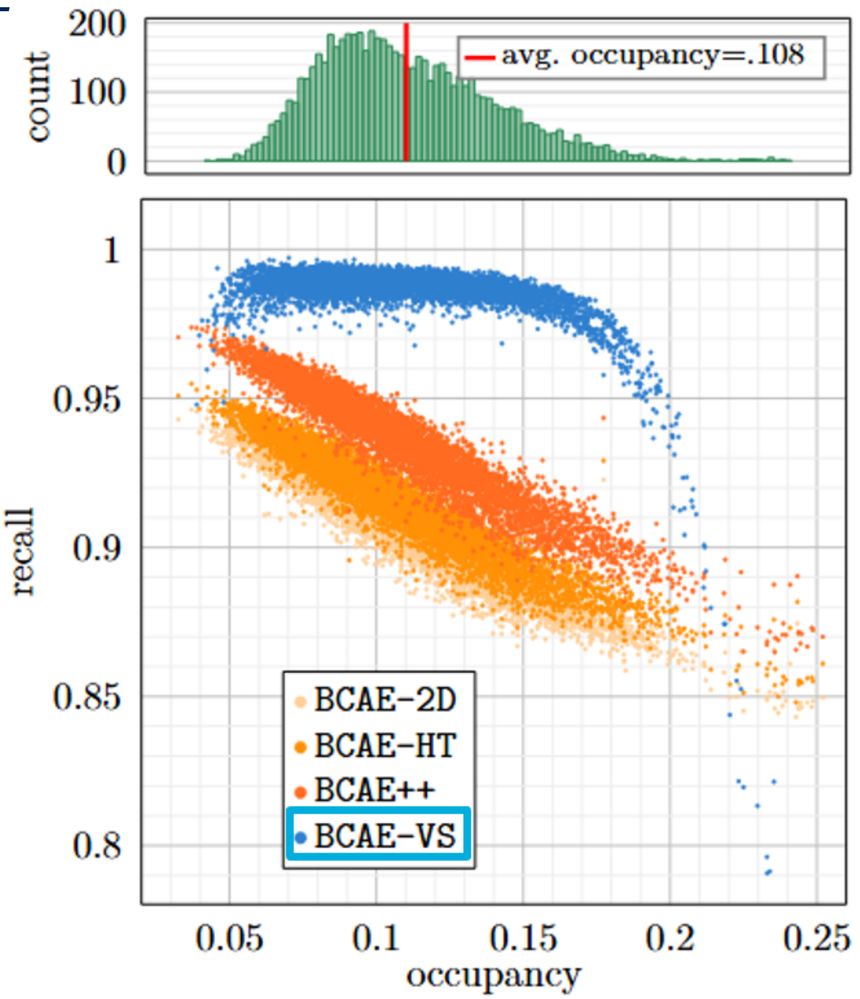
better

L_1: Mean Absolute Error (MAE)

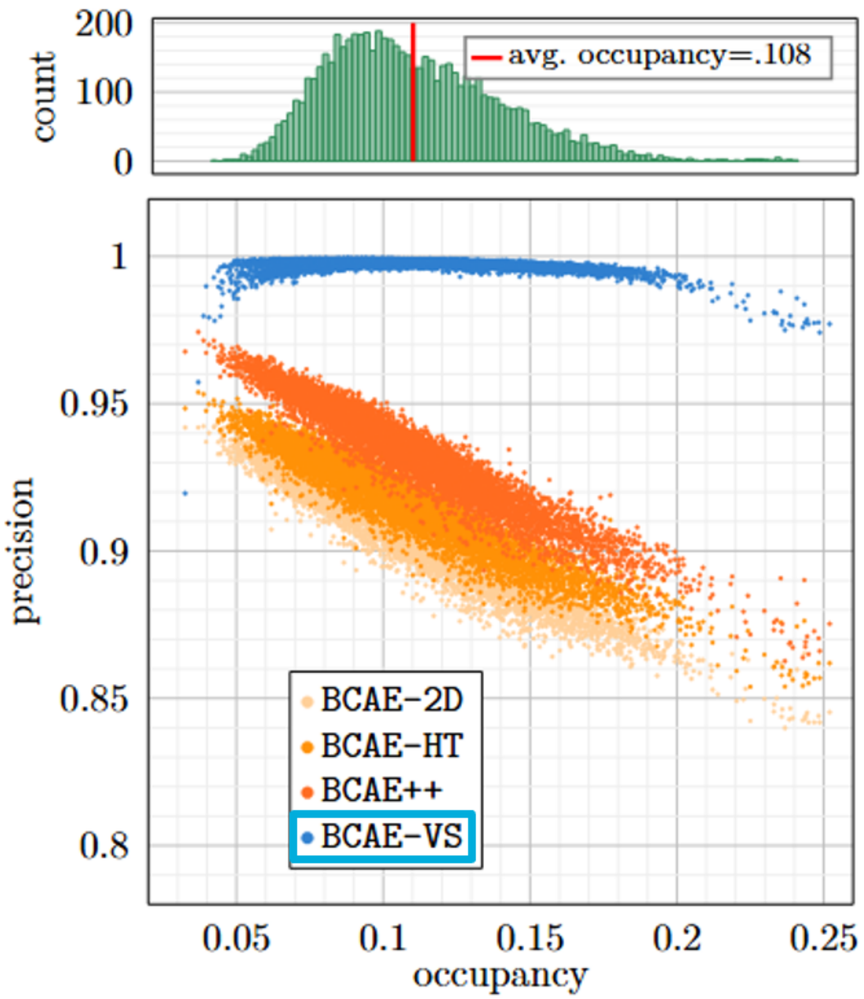
peak signal-to-noise ratio (PSNR)

Reconstruction Accuracy of BCAE-VS; Classification

arXiv: 2411.11942



better

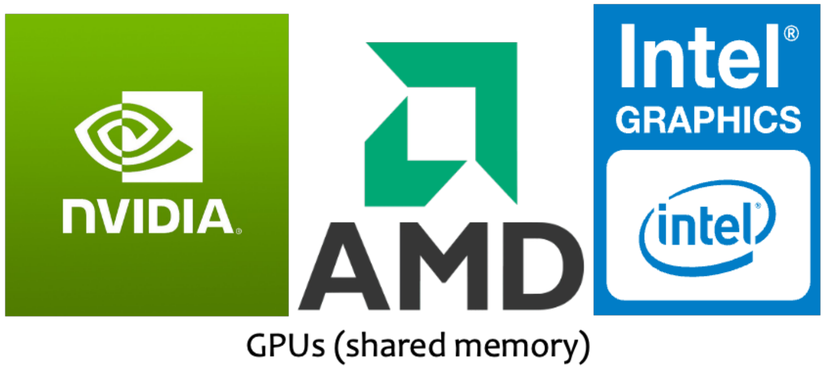


better

$$\text{recoll} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positives}} = \frac{TP}{TP + FP}$$

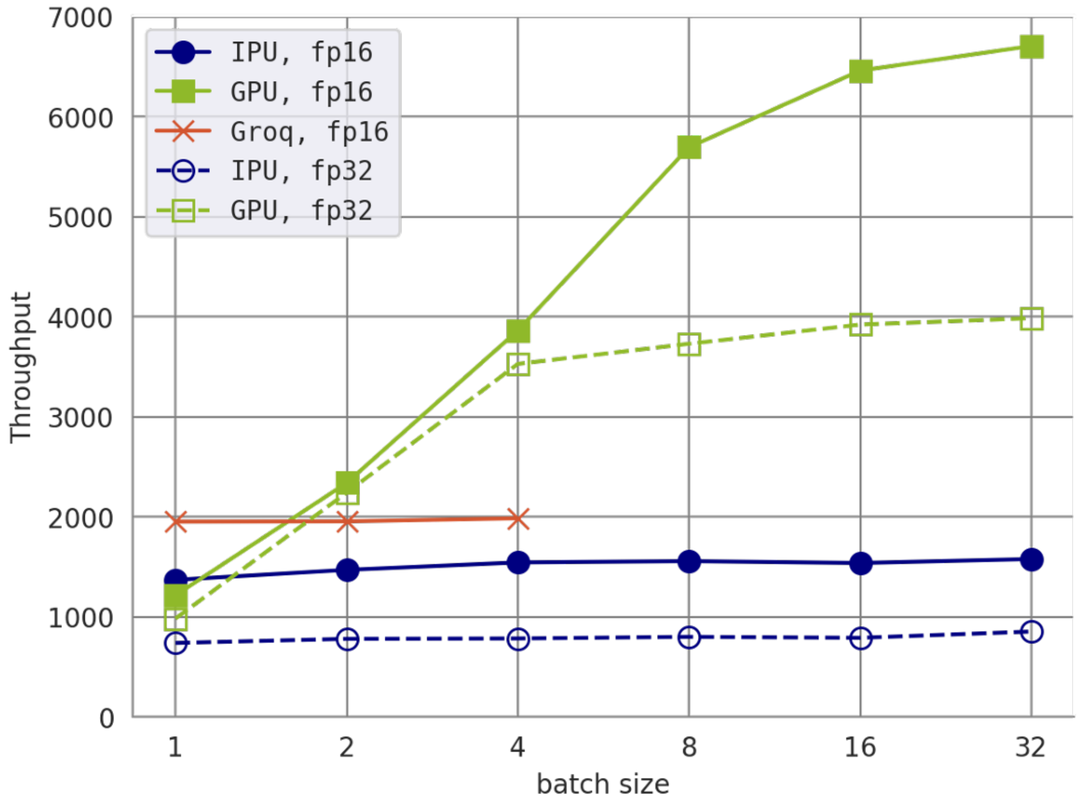
Future (On-going) Development



GPUs (shared memory)



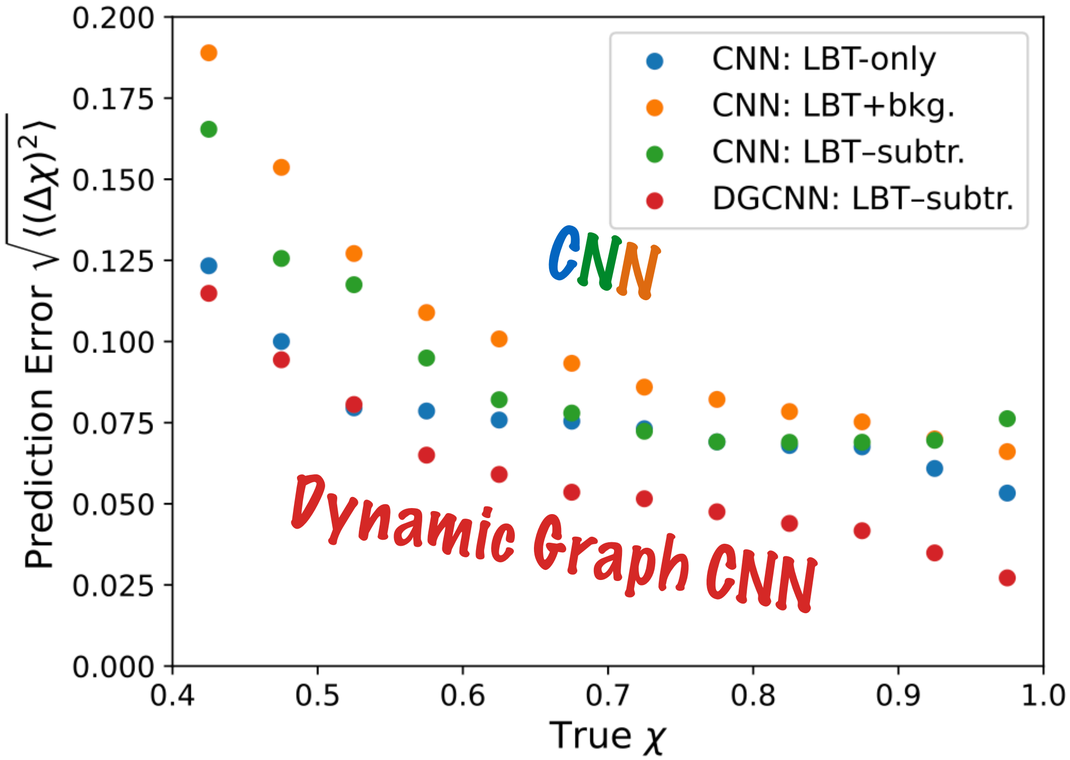
GraphCore and Groq (on-chip memory)



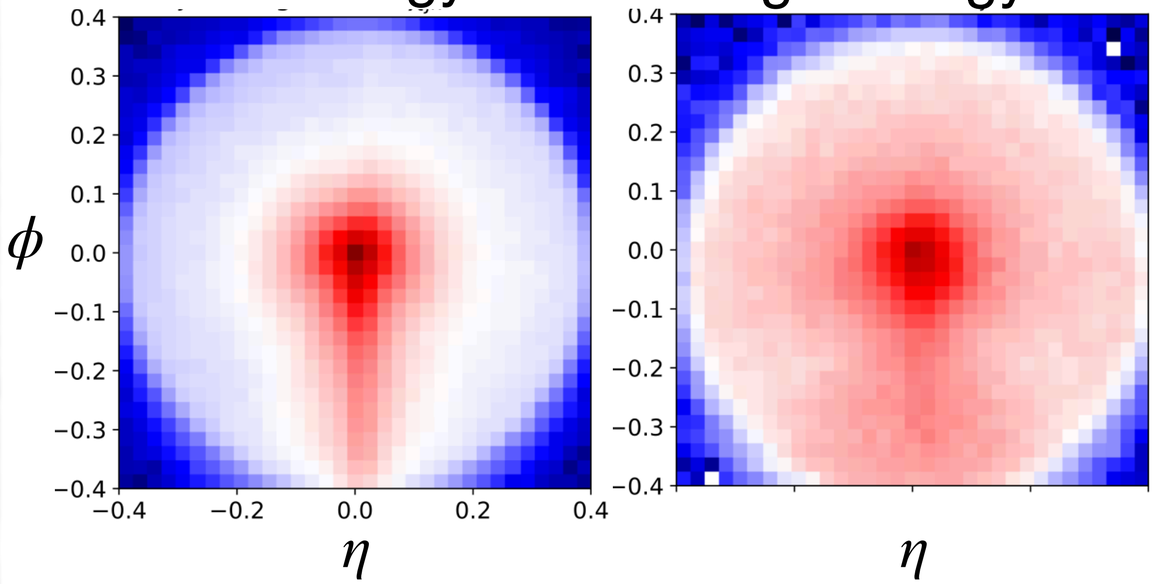
- Faster compression using AI hardware instead of GPU
- Harder to optimize AI hardwares

Regression for Jet Energy Loss

- Convolutional Neural Network (**CNN**) → good for image processing
- **Supervised Learning**
  - ➡ Trained with *Hybrid* [JHEP 03 (2021) 206] or *LBT* [arXiv:2508.20856] theoretical models
  - ➡ Predict amount of jet quenching



Jet Image (on a jet-by-jet basis)



JHEP 03 (2021) 206

arXiv:2508.20856

Classification for Jet Quenching

- Sequential **Attention**-based NN → iteratively passing important features with learned weights

- **Supervised Learning**
  - ➡ Trained with multiple theoretical models (JEWEL+CoLBT and PYTHIA+HERWIG) to reduce model-dependence
  - ➡ Predict amount of jet quenching

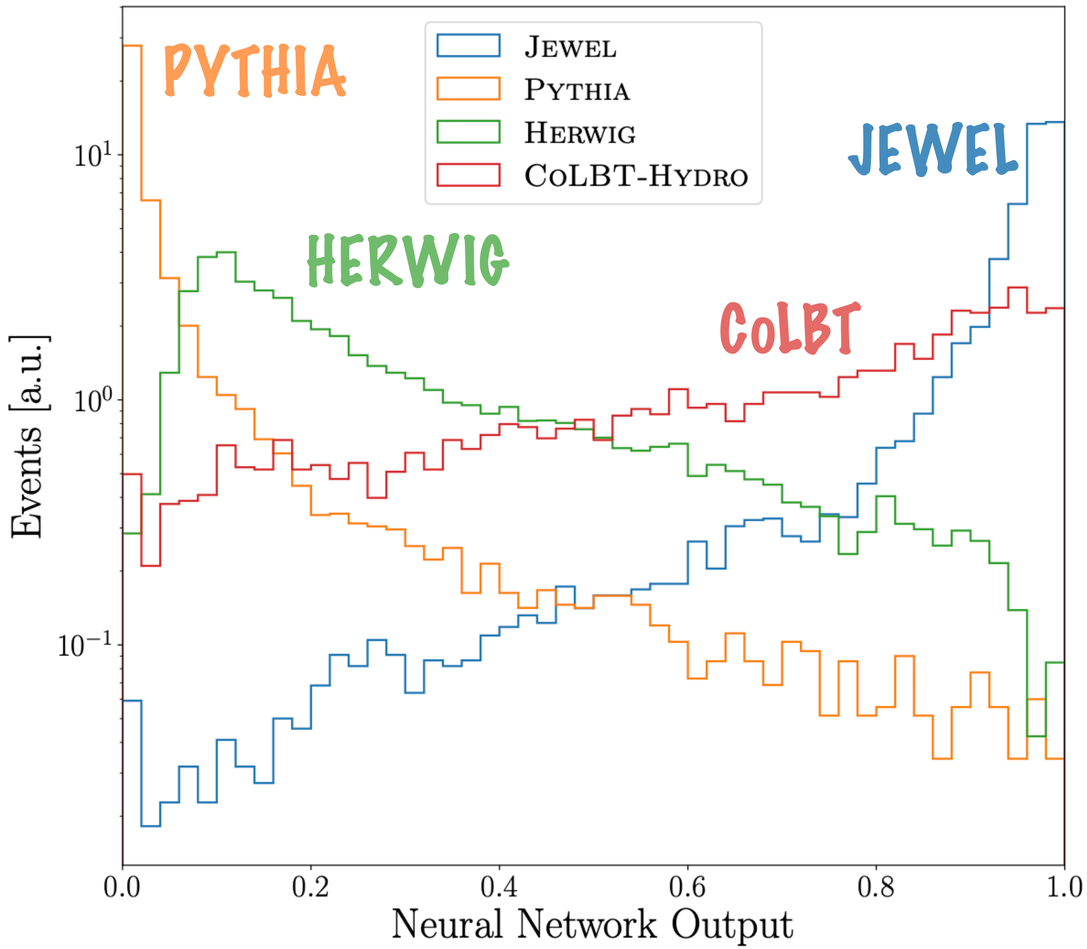
Trained for

JEWEL CoLBT = 1

PYTHIA HERWIG = 0

Therefore, output close to 1 → quenched-like jet
output close to 0 → unquenched-like jet

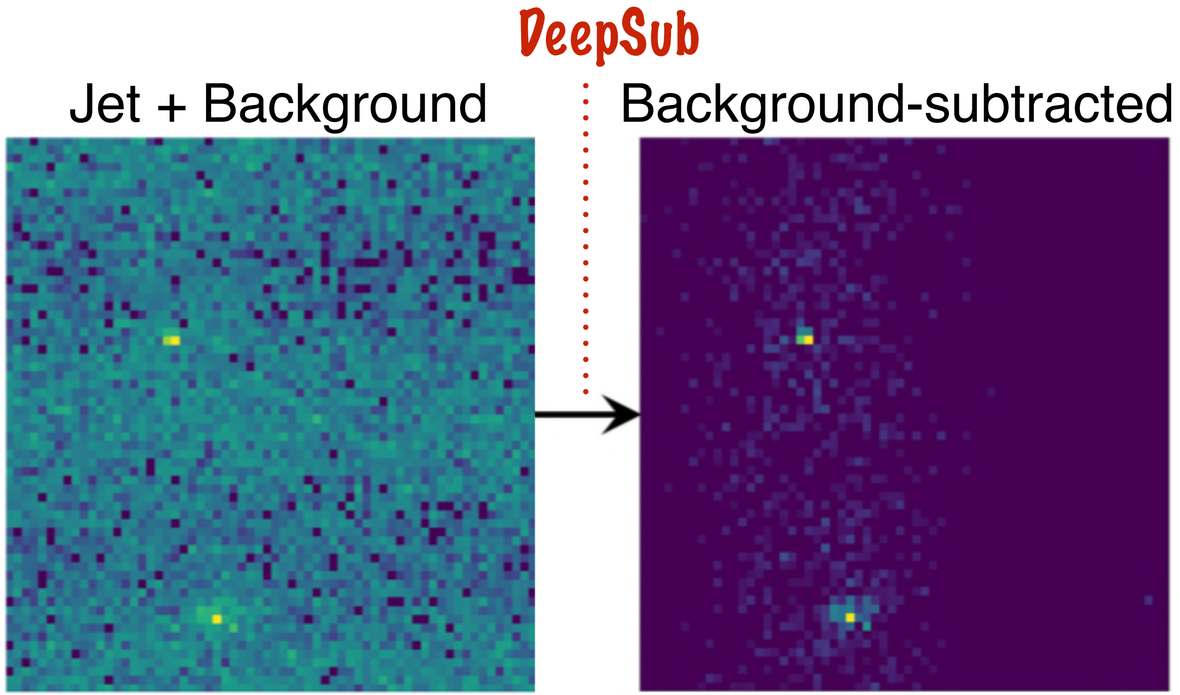
arXiv:2411.19389

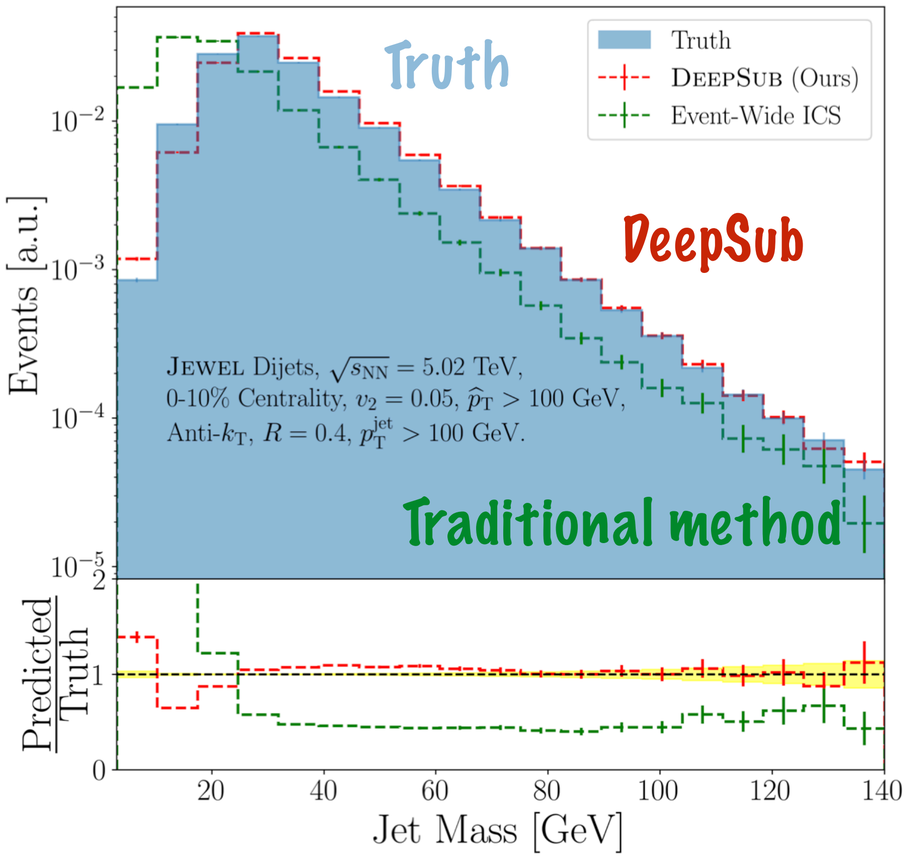


Generative AI for Jet Background Subtraction

- Jet background subtraction as an “image” using Swin Transformer in a ***supervised*** **way**
  - ➡ trained with JEWEL **DeepSub model**
- Rather than a single value, “image” preserves full dimensional information
  - ➡ From the image, one can extract any substructure observables

arXiv:2507.14036





Discussion: Cautions in Using AI/ML

- Model-dependent bias
  - ➡ ML learns what it's trained on (simulation priors, detector effects)
  - ➡ especially more in supervised learning;
    use data-driven, unsupervised method when available
  - ➡ Cross-check with independent models and real data
- Uncertainty quantification
  - ➡ Need calibrated, interpretable uncertainty estimates e.g. Bayesian NNs
- Interpretability & Trust
  - ➡ Use physics-informed architectures and feature-level validation
- Reproducibility
  - ➡ Document data, training, and hyperparameters for long-term use