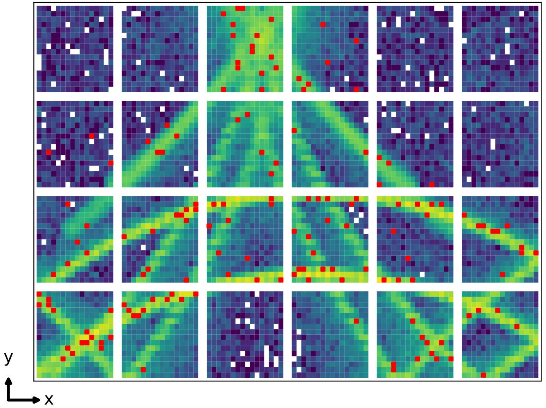
Fast High Fidelity Simulation of Cherenkov Detectors at EIC



James Giroux

Digital Twins for Nuclear and Particle physics

10/07/2025

William & Mary
Chartered 1693
Data Science

Outline

- The High Performance DIRC (hpDIRC) at EIC
- Generative Models for Fast Simulation of Cherenkov Detectors at EIC (arXiv:2504.19042, MLST)
  - Hit-level learning for generative AI
  - A collection of SOTA generative models
  - Holistic Simulation pipeline - Generating the photon yield
  - Takeaways
- Towards Foundation Model for Readout Systems combining Discrete and Continuous Data (arXiv:2505.08736)
  - Towards FM in physics
    - Potential Issues with Tokenization
    - Combining Continuous and Discrete Data
  - Conditional Generation through prepended context
  - Class Conditional Generation through conditional computing - Mixture of Experts
  - Takeaways

hpDIRC at EIC

- Barrel geometry
  - 16-sided polygonal barrel around the beam line (R=1m)
  - Divided into optically isolated sectors - a bar box and a readout box
  - Each bar box contains eleven fused silica radiator bars (~ 4m in length) - mirrored ends for photon reflection
  - Exiting photons are focused by a 3-layer spherical lens
- Pixelated Detector Plane
  - 4x6 PMTs - 16x16 pixels per PMT
  - Provide spatial and timing information (100ps)
- Operation Requirements
  - 3σ separation for π/K at 6 GeV/c

EICDIRC

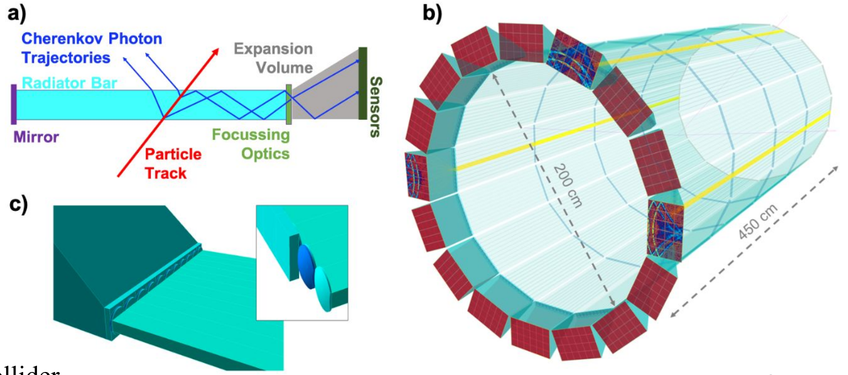
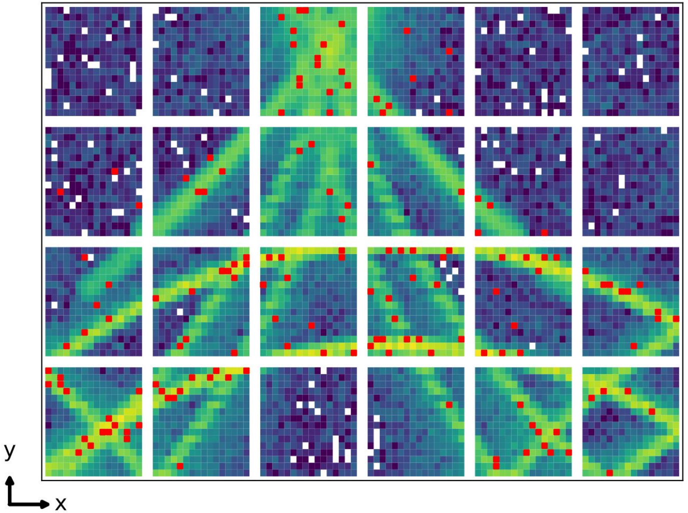


Image taken from [1]

[1] Kalicy G 2022 Developing high-performance DIRC detector for the Future Electron Ion Collider Experiment (arXiv:2202.06457) URL https: //arxiv.org/abs/2202.06457

Generative Models for Fast Simulation of Cherenkov Detectors at the Electron Ion-Collider



[1] Giroux, James, Michael Martinez, and Cristiano Fanelli. "Generative Models for Fast Simulation of Cherenkov Detectors at the Electron-Ion Collider." *arXiv:2504.19042* (2025).
(Accepted into IOP - Machine Learning Science and Technology)

Learning at the Hit Level

- Difficulties in working with Cherenkov detectors for Generative AI
  - Lack of fixed input sizes - dynamic photon yield dependent on kinematic parameters
  - Pixelated (discrete) spatial readout system - algorithms are designed for continuous spaces

Learning at the Hit Level

- Difficulties in working with Cherenkov detectors for Generative AI
  - Lack of fixed input sizes - dynamic photon yield dependent on kinematic parameters
  - Pixelated (discrete) spatial readout system - algorithms are designed for continuous spaces
- Abstract away from fixed input sizes
  - Remain agnostic to the photon yield
- Learning at the hit level, conditional on $< |\boldsymbol{p}| , \boldsymbol{\theta} >$
  - Treating individual Cherenkov photons in a track as ~ independent

Learning at the Hit Level

- Difficulties in working with Cherenkov detectors for Generative AI
  - Lack of fixed input sizes - dynamic photon yield dependent on kinematic parameters
  - Pixelated (discrete) spatial readout system - algorithms are designed for continuous spaces
- Abstract away from fixed input sizes
  - Remain agnostic to the photon yield
- Learning at the hit level, conditional on < |***p***| , ***θ*** >
  - Treating individual Cherenkov photons in a track as ~ independent
- Use physical sensor dimensions to remove discrete representation in space
  - DIRC readout has a fixed “row,col” coordinate system (1)
  - Transform to x,y coordinate system (mm) (2)
  - Smear uniformly over individual PMT pixels

(1)

$$D_{i,j} = \begin{cases} \lfloor M_{PMT.}/6 \rfloor \cdot 16 + \lfloor N_{pixel.}/16 \rfloor \\ (M_{PMT.}\ \%\ 6) \cdot 16 + (N_{pixel.}\ \%\ 16) \end{cases}$$

(2)

$$x = 2 + D_j \cdot p_{width.} + (M_{PMT.}\ \%\ 6) \cdot \mathrm{gap}_x + \frac{1}{2} p_{width.}$$

$$y = 2 + D_i \cdot p_{height.} + \lfloor M_{PMT.}\ /\ 6 \rfloor \cdot \mathrm{gap}_y + \frac{1}{2} p_{height.}$$

Learning at the Hit Level Cont’d…

- What does this look like during training?

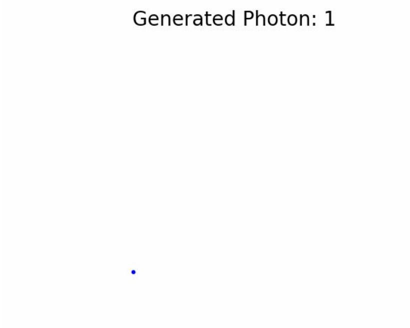
ID	x (mm)	y (mm)	t (ns)	$\lvert \boldsymbol{p} \rvert$	$\boldsymbol{\theta}$
1				3.0	5.0
1				3.0	5.0
…	…	…	…	…	…
N				4.0	7.0
N				4.0	7.0

Learning at the Hit Level Cont'd…

- What does this look like during training?

ID	x (mm)	y (mm)	t (ns)	$\lvert \boldsymbol{p} \rvert$	$\boldsymbol{\theta}$
1				3.0	5.0
1				3.0	5.0
…	…	…	…	…	…
N				4.0	7.0
N				4.0	7.0

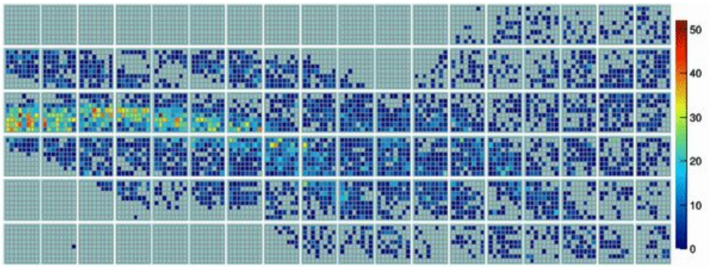
- What does this look like at inference (generation)?
  - Our models are trained to generate **individual** photons
  - We aggregate multiple forward calls to generate **tracks**
  - These are **not** sequential - batch processing of N_γ



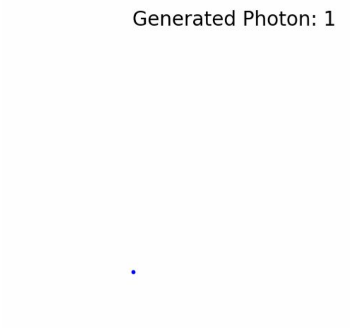
Learning at the Hit Level Cont'd…

- What does this look like during training?

ID	x (mm)	y (mm)	t (ns)	$\|\boldsymbol{p}\|$	$\boldsymbol{\theta}$
1				3.0	5.0
1				3.0	5.0
…	…	…	…	…	…
N				4.0	7.0
N				4.0	7.0



- What does this look like at inference (generation)?
  - Our models are trained to generate **individual** photons
  - We aggregate multiple forward calls to generate **tracks**
  - These are **not** sequential - batch processing of N_γ



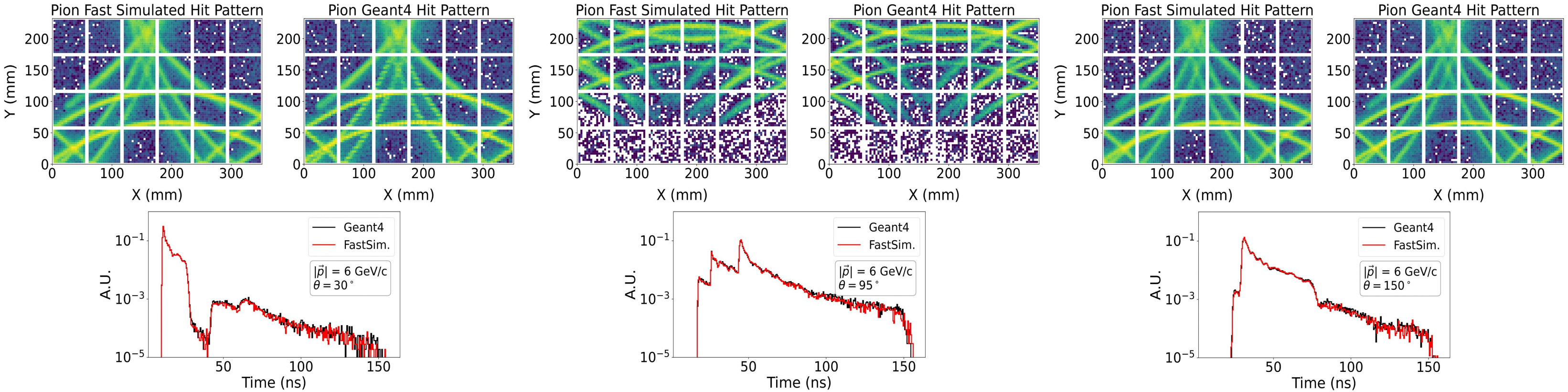
Integrate over tracks to create PDF

A collection of SOTA Generative Models

Method	Transformation	Objective
Discrete Normalizing Flows	$x_k = f_\theta(z,k) = f_{\theta_N} \circ f_{\theta_{N-1}} \circ ... f_{\theta_1}(z_0,k)$	$min_\theta \; \mathbb{E}_{x \sim p_{data}(x)}[-\log p_\theta(x\|k)]$
Continuous Normalizing Flows	$\frac{\mathrm{d}z}{\mathrm{d}t} = v_t(z_t,k;\theta)$ $z_{t_0} = x$	$min_\theta \; \mathbb{E}_{x \sim p_{data}(x)}[-\log p_\theta(x\|k)]$
Conditional Flow Matching	$\frac{\mathrm{d}z}{\mathrm{d}t} = v_t(z_t,k;\theta)$ $z_{t_0} = x$	$min_\theta \; \mathbb{E}_{t,p(x_{t_1}\|k),q_t(x\|x_{t_1},k)}\|\|v_t(x,k) - u_t(x\|x_{t_1},k)\|\|^2$
Denoising Diffusion Probabilistic Models	$p_\theta(x_{0:T}\|k) := p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}\|x_t,k)$	$min_\theta \; \mathbb{E}_{t,x_0,\epsilon}[\|\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, k)\|\|^2]$
Score Based Generative Models	$\mathrm{d}x = [f(x,t) - g(t)^2\nabla_x \log p_t(x)]\mathrm{d}t + g(t)\mathrm{d}\bar{w}$	$min_\theta \; \frac{1}{2}\mathbb{E}_{t,x,\epsilon}[\|\|s_\theta(\tilde{x},t,k) - \frac{\epsilon}{\sigma_t}\|\|^2]$

A collection of SOTA Generative Models

Discrete Normalizing Flows

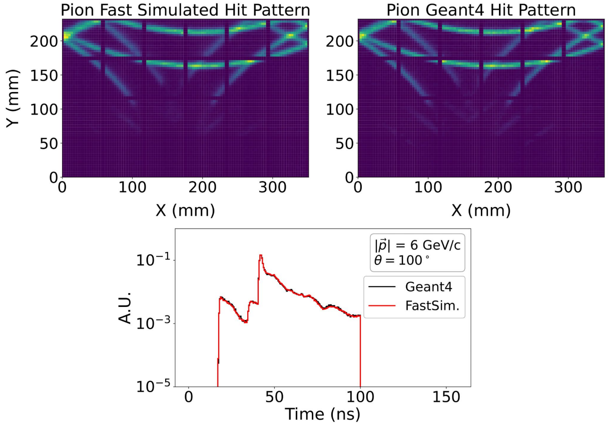


Visualization of Generations at 6 GeV/c and various polar angle

WILLIAM & MARY | DATA SCIENCE
CHARTERED 1693

Quantitative Evaluation through PID Metrics

- Translate simulation quality metrics to a more meaningful representation
    - Separation power through KDE based PID method (FastDIRC)

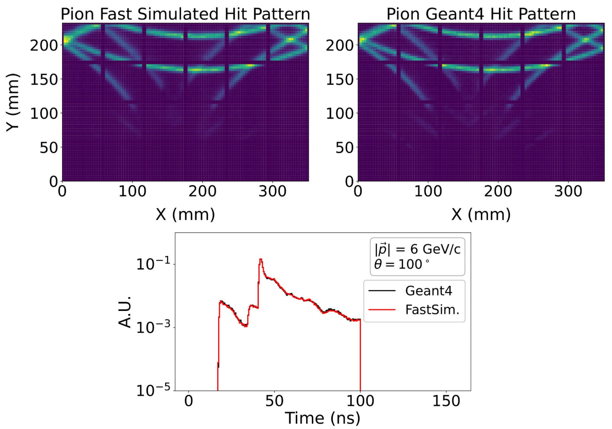


[3] Hardin, John, and Mike Williams. "FastDIRC: a fast Monte Carlo and reconstruction algorithm for DIRC detectors." *Journal of Instrumentation* 11.10 (2016): P10007.

Quantitative Evaluation through PID Metrics

- Translate simulation quality metrics to a more meaningful representation
  - Separation power through KDE based PID method (FastDIRC)

Create large reference populations of π / K (support PDF)



[3] Hardin, John, and Mike Williams. "FastDIRC: a fast Monte Carlo and reconstruction algorithm for DIRC detectors." *Journal of Instrumentation* 11.10 (2016): P10007.

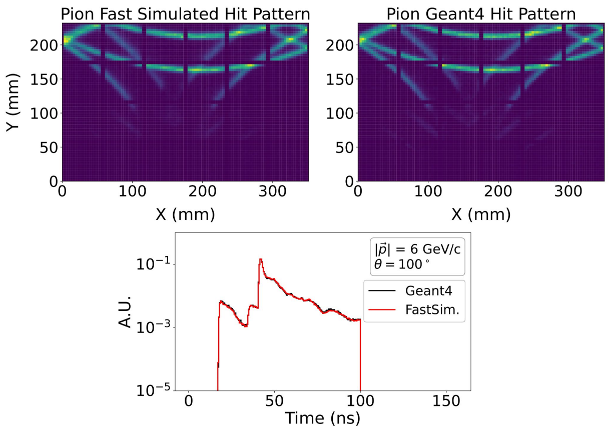
Quantitative Evaluation through PID Metrics

- Translate simulation quality metrics to a more meaningful representation
  - Separation power through KDE based PID method (FastDIRC)

Create large reference populations of $\boldsymbol{\pi}$ / K (support PDF)

For each photon in a track, calculate likelihood:

$$log\ p(\vec{x}_i|\vec{k})_{K\pi} \propto (\vec{x}_i - \vec{\mu}_{K\pi})^T \boldsymbol{\Sigma}^{-1} (\vec{x}_i - \vec{\mu}_{K\pi})\ , \boldsymbol{\Sigma} = \begin{bmatrix} p_{width}^2 & 0 & 0 \\ 0 & p_{height}^2 & 0 \\ 0 & 0 & \sigma_t^2 \end{bmatrix}$$

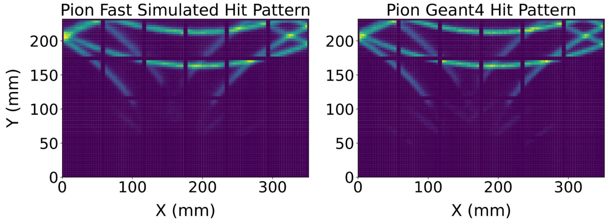


[3] Hardin, John, and Mike Williams. "FastDIRC: a fast Monte Carlo and reconstruction algorithm for DIRC detectors." *Journal of Instrumentation* 11.10 (2016): P10007.

Quantitative Evaluation through PID Metrics

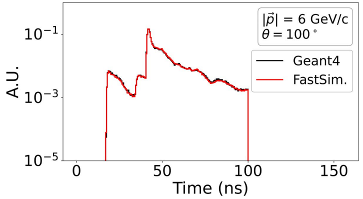
- Translate simulation quality metrics to a more meaningful representation
  - Separation power through KDE based PID method (FastDIRC)

Create large reference populations of $\boldsymbol{\pi}$ / K (support PDF)



For each photon in a track, calculate likelihood:

$$log\ p(\vec{x}_i|\vec{k})_{\mathcal{K}\pi} \propto (\vec{x}_i - \vec{\mu}_{\mathcal{K}\pi})^T \mathbf{\Sigma}^{-1} (\vec{x}_i - \vec{\mu}_{\mathcal{K}\pi}) \ , \ \mathbf{\Sigma} = \begin{bmatrix} p^2_{width} & 0 & 0 \\ 0 & p^2_{height} & 0 \\ 0 & 0 & \sigma^2_t \end{bmatrix}$$



Perform DLL:

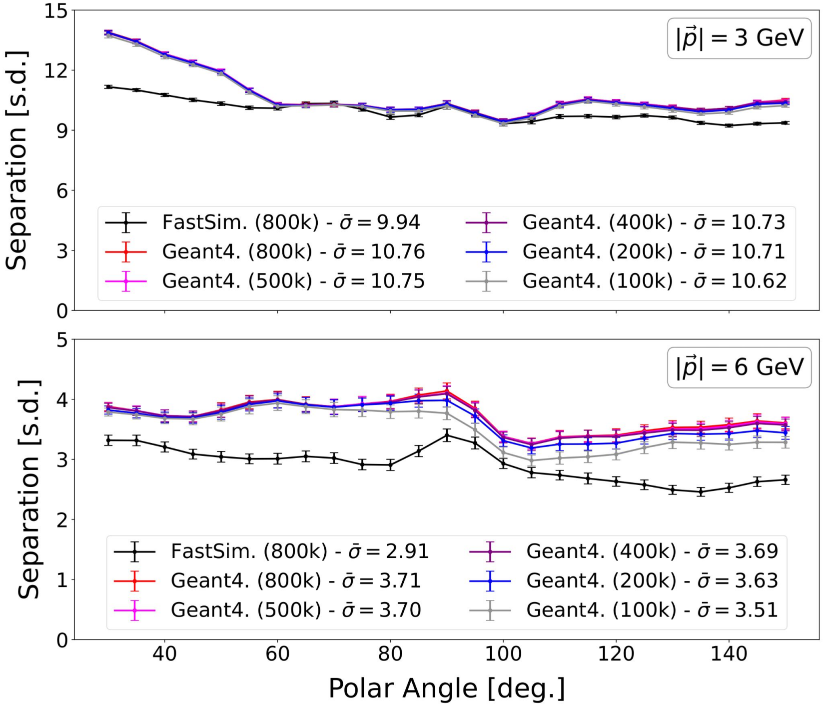
$$\Delta \log \mathcal{L}_{\mathcal{K}\pi} = \sum_i^N \log p(\vec{x}_i|\vec{k}, \mathcal{K}) - \sum_i^N \log p(\vec{x}_i|\vec{k}, \pi)$$

[3] Hardin, John, and Mike Williams. "FastDIRC: a fast Monte Carlo and reconstruction algorithm for DIRC detectors." *Journal of Instrumentation* 11.10 (2016): P10007.

Quantitative Evaluation through PID Metrics

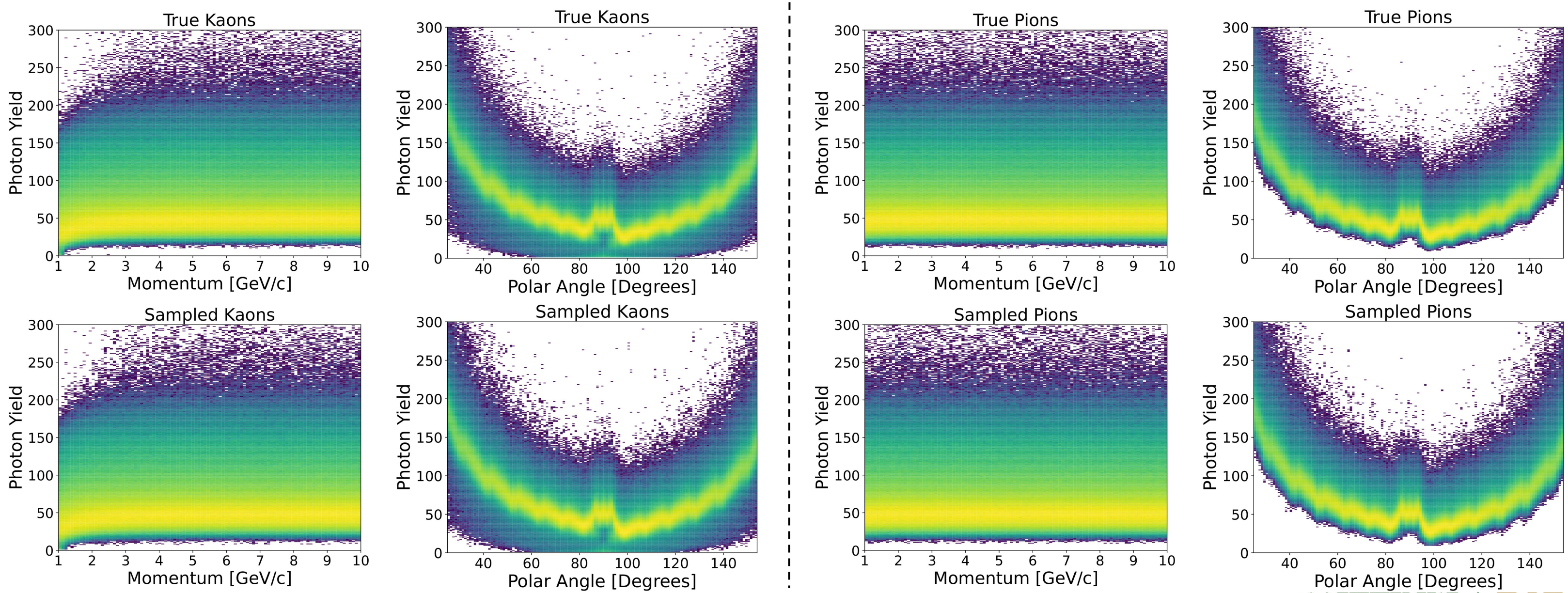
- Evaluate at fixed kinematics
  - 3,6 GeV/c - 5 degree bins over acceptance range
- For each bin, fit a Gaussian distribution to both PID's in the DLL space
- Calculate the separation between distributions

Discrete Normalizing Flows



Holistic Simulation - Photon Yield Generation

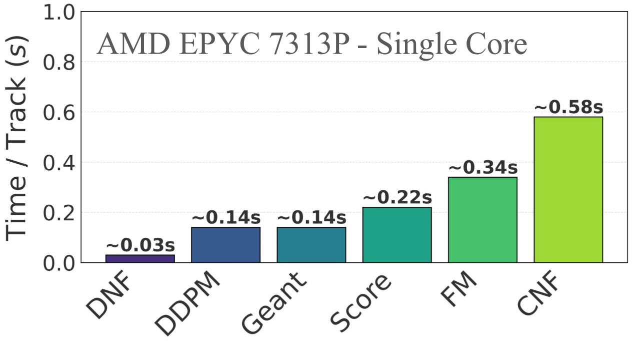
- Low dimensional problem $f : \mathbb{R}^2 \rightarrow \mathbb{Z}_+$
- Must be fast - approximately zero overhead, preferably CPU bound
- A simple Look-Up-Table (LUT) does the trick - 100 MeV/c, 1 degree bins



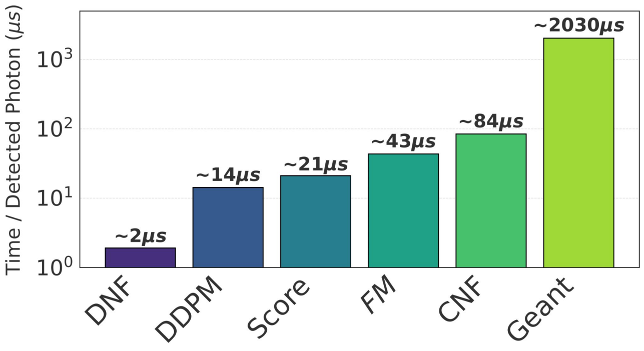
Key Takeaways

All code is open source and pre-trained models are provided. Github

- All Cherenkov Rings (underlying PDF's) generated by our models are "correct"
  - Ring and time structures follow correct kinematic dependencies for both PIDs
  - We incur a smoothing effect - can cause different PIDs to appear more similar
- Beyond usage in Physics environments
  - We have created an open source suite of SOTA algorithms for the hpDIRC (easily adapted to other detectors)
  - Our fast simulation is self contained, fast and capable of being run on CPU or GPU

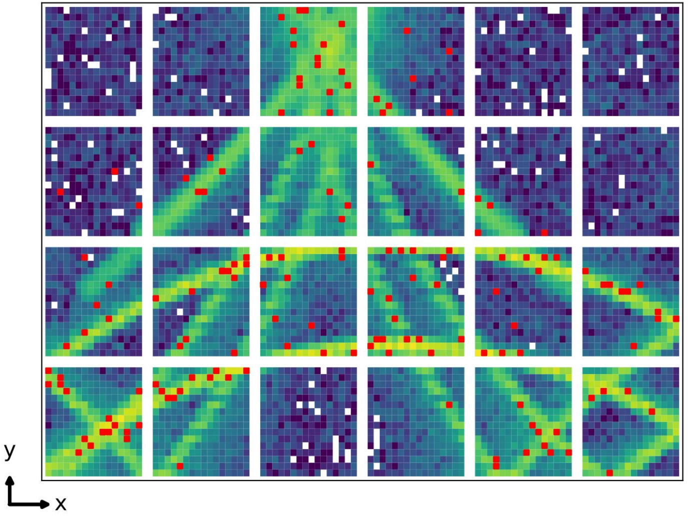


Track Generation (CPU)



PDF Generation (GPU)

Towards Foundation Models for Experimental Readout Systems Combining Discrete and Continuous Data



[4] Giroux, James, and Cristiano Fanelli. "Towards Foundation Models for Experimental Readout Systems Combining Discrete and Continuous Data." *arXiv preprint arXiv:2505.08736* (2025).

Foundation Models in Physics

- Foundation Models (FM) are becoming increasingly popular in the Physics community
    - Relatively large pre-trained models
    - Capable of supporting multiple downstream tasks - e.g., **fast simulation**, reconstruction, etc.
- Different approaches have emerged (focusing on two recent ones)
    - Diffusion Transformer (DiT) style (see [3])
    - GPT style (see [4])
- In both cases, they work with relatively high level features
    - Facilitating generation and classification of Jets through their constituents (4-vector like quantities)

[3] Mikuni, Vinicius, and Benjamin Nachman. "OmniLearn: A method to simultaneously facilitate all jet physics tasks." *arXiv preprint arXiv:2404.16091* (2024).

[4] Birk, Joschka, Anna Hallin, and Gregor Kasieczka. "OmniJet-α: the first cross-task foundation model for particle physics." *Machine Learning: Science and Technology* 5.3 (2024): 035031.

Foundation Models in Physics cont'd…

- More recently [5] has shown very nice, and promising results using GPT style models to generate point clouds in calorimeters
  - Treat cells and energy as **tokenized** representations
  - Generate the shower forward in time, predicting the **next token** given the previous (**context**)
  - This is akin to modern LLMs such as ChatGPT

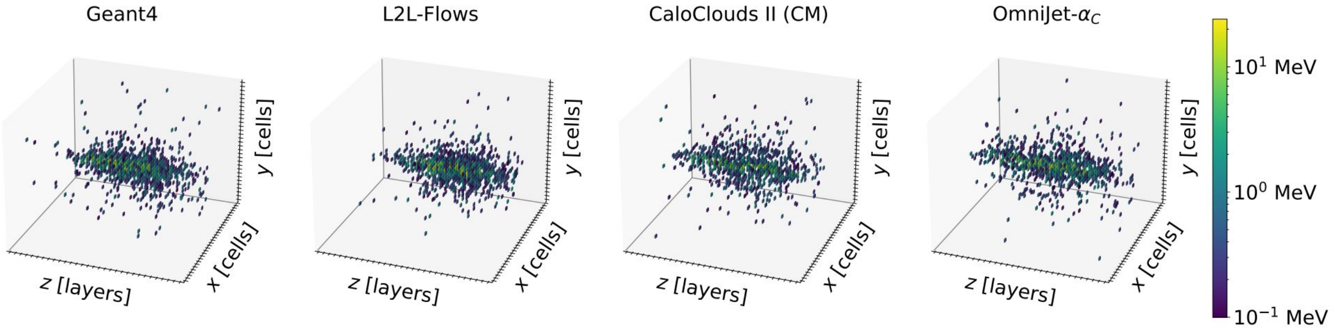


Figure from [5]

[5] Birk, Joschka, et al. "OmniJet-{\alpha_ {C}}: Learning point cloud calorimeter simulations using generative transformers." *arXiv preprint arXiv:2501.05534* (2025).

Potential Issues with Tokenization

- Tokens in the context of LLM are discrete integers representing a "word"
  - The vocabulary of the LLM is then the discrete set of tokens it is able to learn and generate
- What if we want to use **next token** prediction in continuous domains?
  - For example to generate images, or detector response (location and some value)
  - We can first **learn** the **discretized codebook** through some external model e.g., a **Vector Quantized Variational Autoencoder (VQ-VAE)**

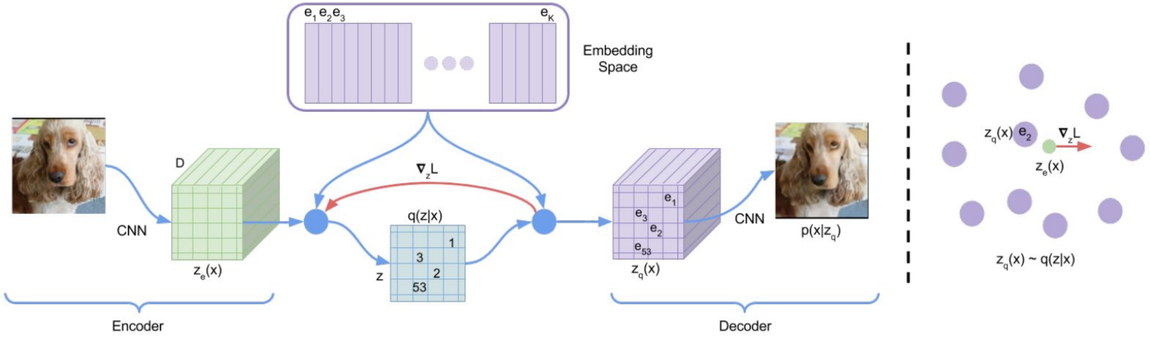


Figure from [6]

[6] Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." *Advances in neural information processing systems* 30 (2017).

Potential Issues with Tokenization cont’d

- A VQ-VAE like model solves the issue of allowing **next token** prediction style models to operate in continuous domains
- But it does come with drawbacks
  - There is inherent information loss in the encoding procedure
  - The reconstruction is limited by the granularity of the codebook
  - Potential inconsistencies or artifacts - crucial in high precision applications

Combining Discrete and Continuous Data

- In attempt to circumvent these potential issues, we devise an alternative strategy
  - In our Cherenkov data - we have a pixel (**discrete** integer) and a **continuous** time associated with each hit
- We utilize two vocabularies - two prediction heads
  - A discrete set of pixels
  - A discrete set of time bins - a linear binning at ¼ the timing resolution of the readout system
- As a result, our data structure for a given track is of the form

$$\begin{aligned} \text{spatial} \rightarrow \{ \quad & \mathrm{SOS}_p, p_1, \ldots, p_n, \mathrm{EOS}_p\} \\ \text{time} \rightarrow \{ \quad & \mathrm{SOS}_t, t_1, \ldots, t_n, \mathrm{EOS}_t\} \end{aligned}$$

- We have still discretized the continuous variate - but in a controlled manner

Conditional Generation

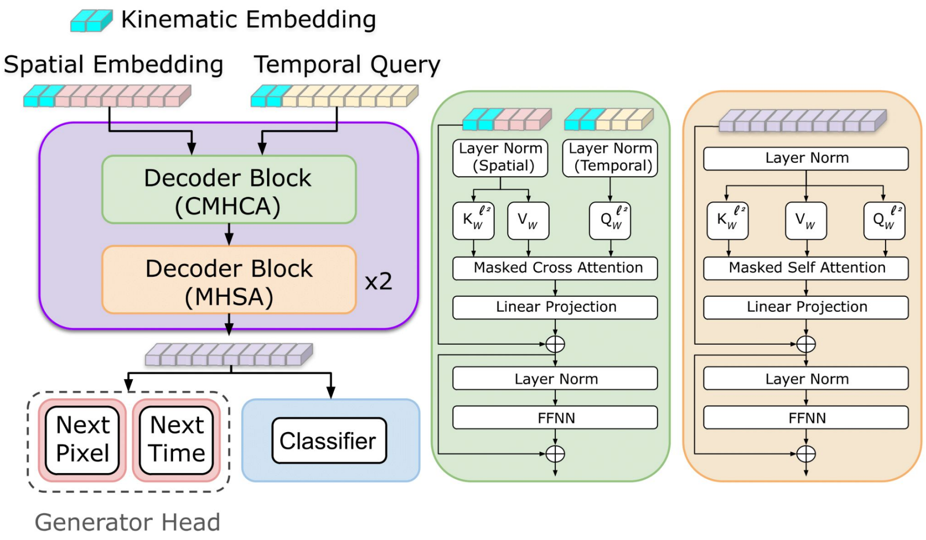
- Cherenkov hits in particular are highly dependent on external kinematic parameters
  - Spatial location (ring structures in PDFs), and time distributions are highly variable
- While these variates are also continuous, we do not need to **tokenize** (discretize them)
- We instead embed them through linear projections and **prepended as context**

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

- The prepending strategy allows the kinematics to *guide* sequence generation forward in time

Towards FM for Pixelated Readout Systems

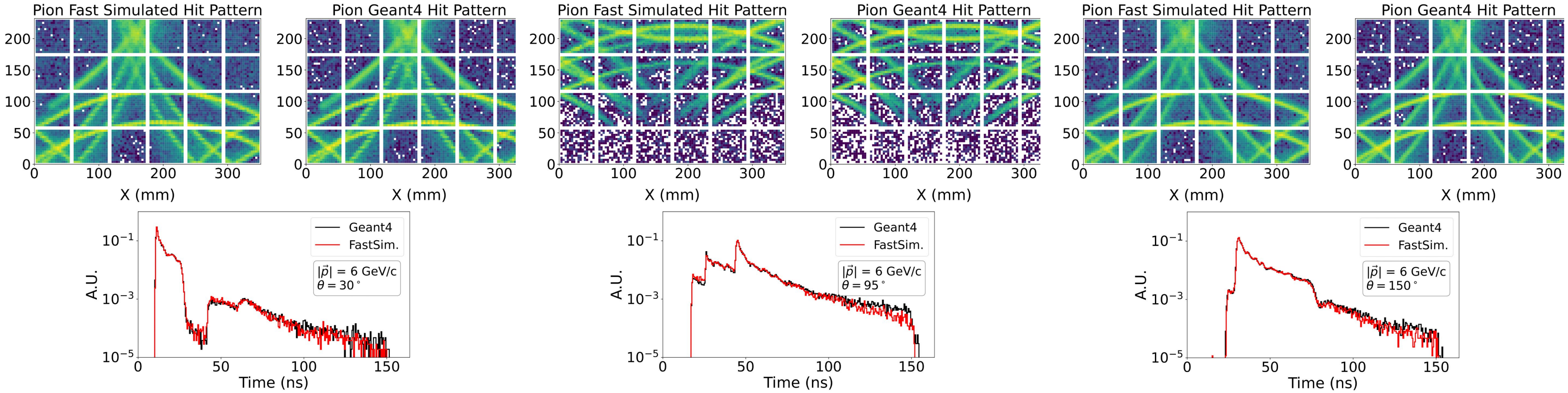
- Our vocabularies operate adjacent to one another - providing next **pixel** and **time** through independent prediction heads
- We combine information through **Causal Cross Attention**
  - Time drives the sequence - at a given time, query the pixel space for possible locations



Causal implies we apply a **mask** to prevent seeing forward in time

Example Generations

Autoregressive (Next Token)



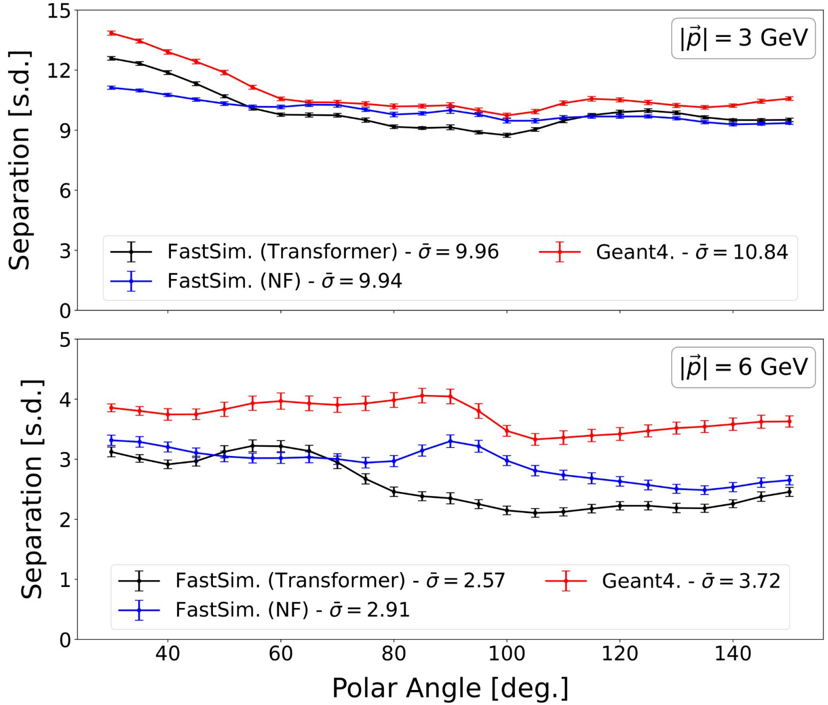
Visualization of Generations at 6 GeV/c and various polar angle

WILLIAM & MARY | DATA SCIENCE
CHARTERED 1693

Quantitative Evaluation through PID Metrics

- Evaluate at fixed kinematics
  - 3,6 GeV/c - 5 degree bins over acceptance range
- For each bin, fit a Gaussian distribution to both PID's in the DLL space
- Calculate the separation between distributions

Autoregressive (Next Token)



Class Conditional Generation

- Generations shown prior are from independent models ($\boldsymbol{\pi}$/K)
  - How do we combine multiple classes under a single model?

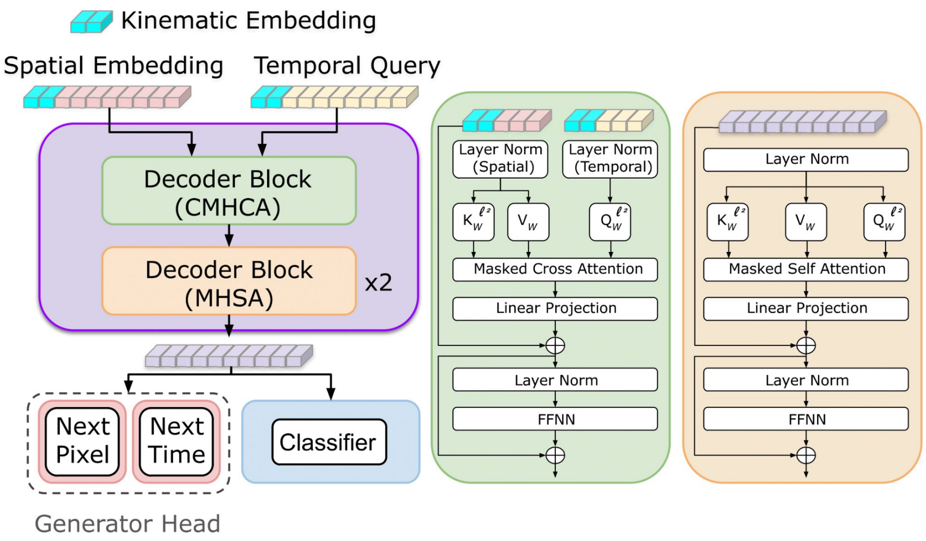
Class Conditional Generation

- Generations shown prior are from independent models (***π***/K)
  - How do we combine multiple classes under a single model?
- Standard Gen. AI
  - Build conditional probability distribution $p(x|k,c)$
  - Difficult given merging of π/K PDFs as momentum increases
  - Modes collapse together - fidelity decreases

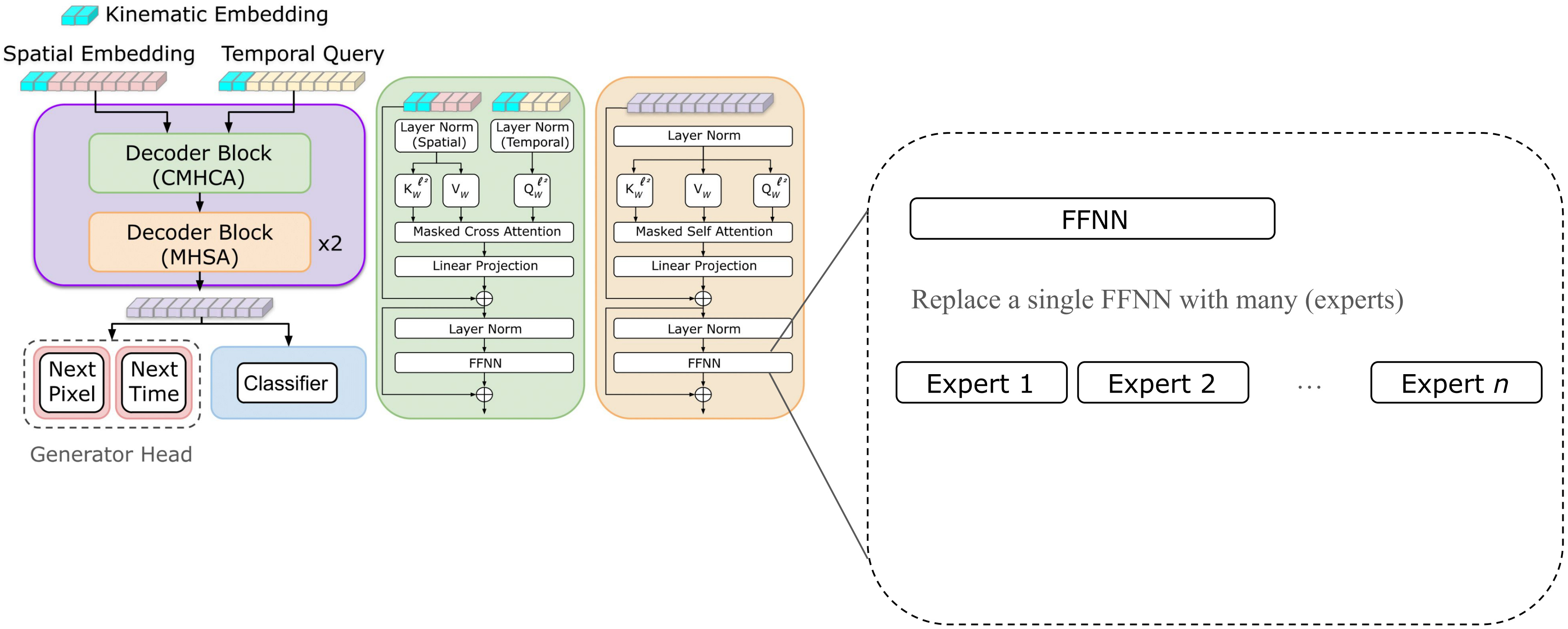
Class Conditional Generation

- Generations shown prior are from independent models (**π**/K)
  - How do we combine multiple classes under a single model?
- Standard Gen. AI
  - Build conditional probability distribution $p(x|k,c)$
  - Difficult given merging of π/K PDFs as momentum increases
  - Modes collapse together - fidelity decreases
- Autoregressive (next token)
  - We can prepend additional context (class label)
  - Same issues as before - modes collapse together

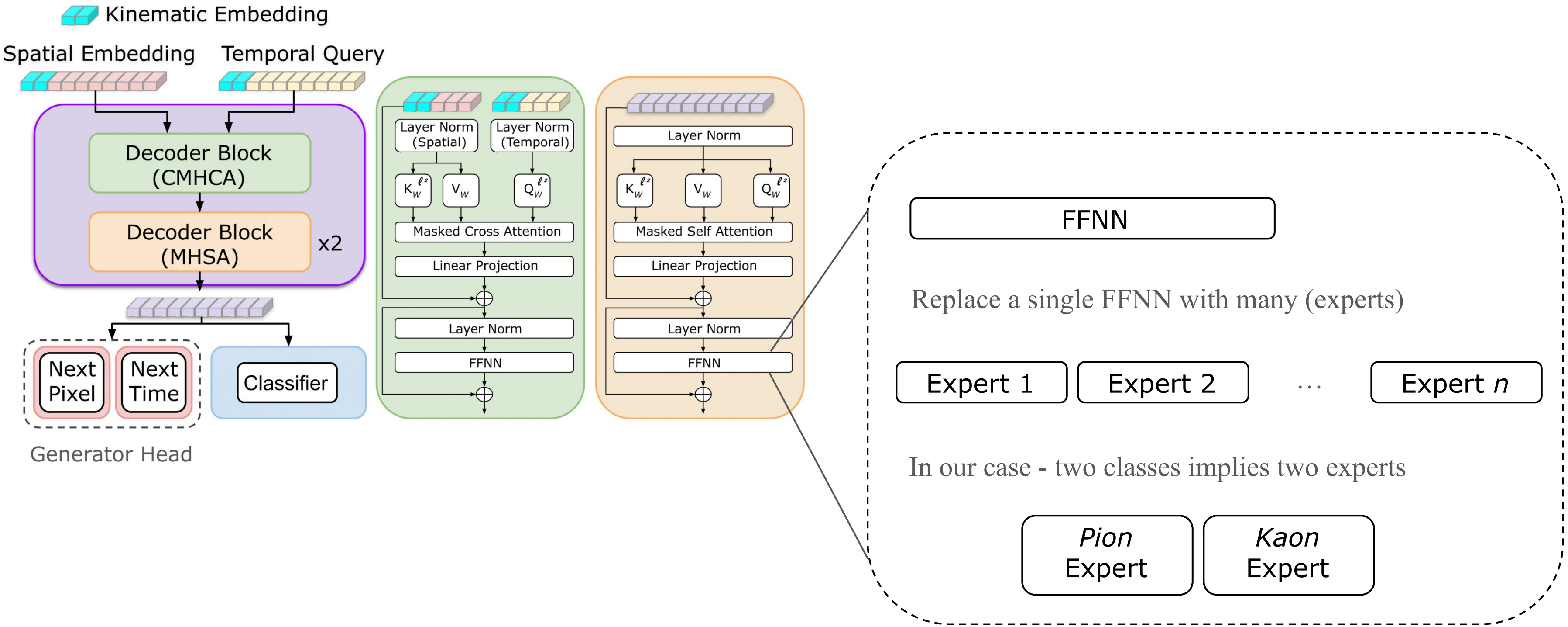
A Mixture of Experts - Conditional Computing



A Mixture of Experts - Conditional Computing



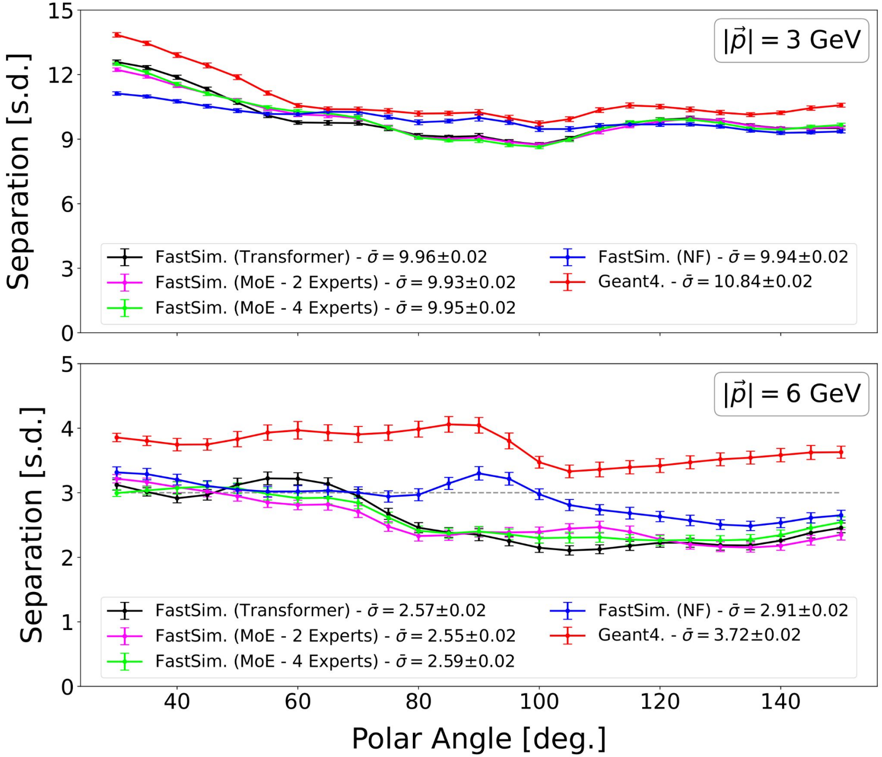
A Mixture of Experts - Conditional Computing

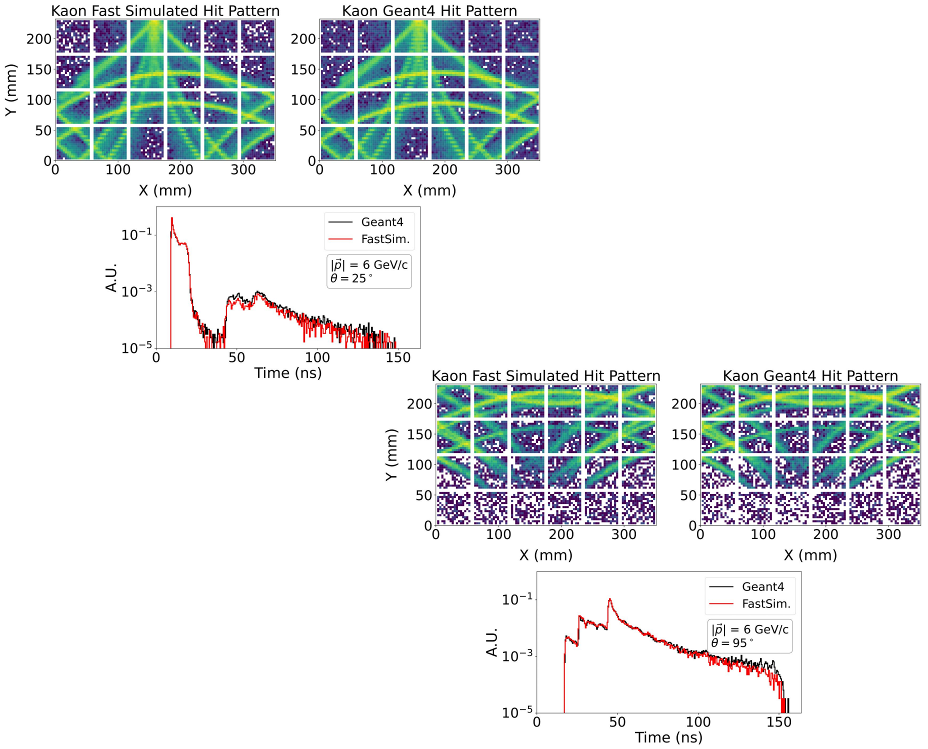


A Mixture of Experts - Conditional Computing

- The general idea
  - Turn of/on parts of the network given a specific type of particle we want to generate - **fixed routing**
  - Majority of parameters are shared - only experts are aligned with a specific class
  - Capture *general* relationship through attention blocks
  - Apply *fine grained* corrections through class conditional experts
- Extension in multiple ways
  - Multiple particles - π/K/e
  - Multiple experts per class (see [4] for more details)

A Mixture of Experts - Conditional Computing





Generation Quality does not degrade w.r.t. Independent models

WILLIAM & MARY CHARTERED 1693 | DATA SCIENCE

Key Takeaways

- **(Proto) Foundation Model** operating directly on low level detector signals
  - Generalizable to other detectors with discrete + continuous data streams
- Four core innovations
  - **Dual Vocabularies** for spatial and temporal data - fused through **Cross Attention**
  - **Scaleable**, **higher resolution tokenization** with joint vocab inflation - our resolutions correspond to 36M tokens in a joint vocabulary
  - **Continuous conditioning** using prepending kinematic embeddings
  - **Class conditional generation** through a Mixture of Experts
- Capable of multiple downstream tasks
  - Generation shown here.

Github

All code is open source and pre-trained models are provided.