Particle Flow Networks for Kinematic Reconstruction in Semi-Inclusive Deep-Inelastic Scattering at the EIC

Daniel Almonte[1], Anselm Vossen[2]

[1] Stony Brook University
[2] Duke University

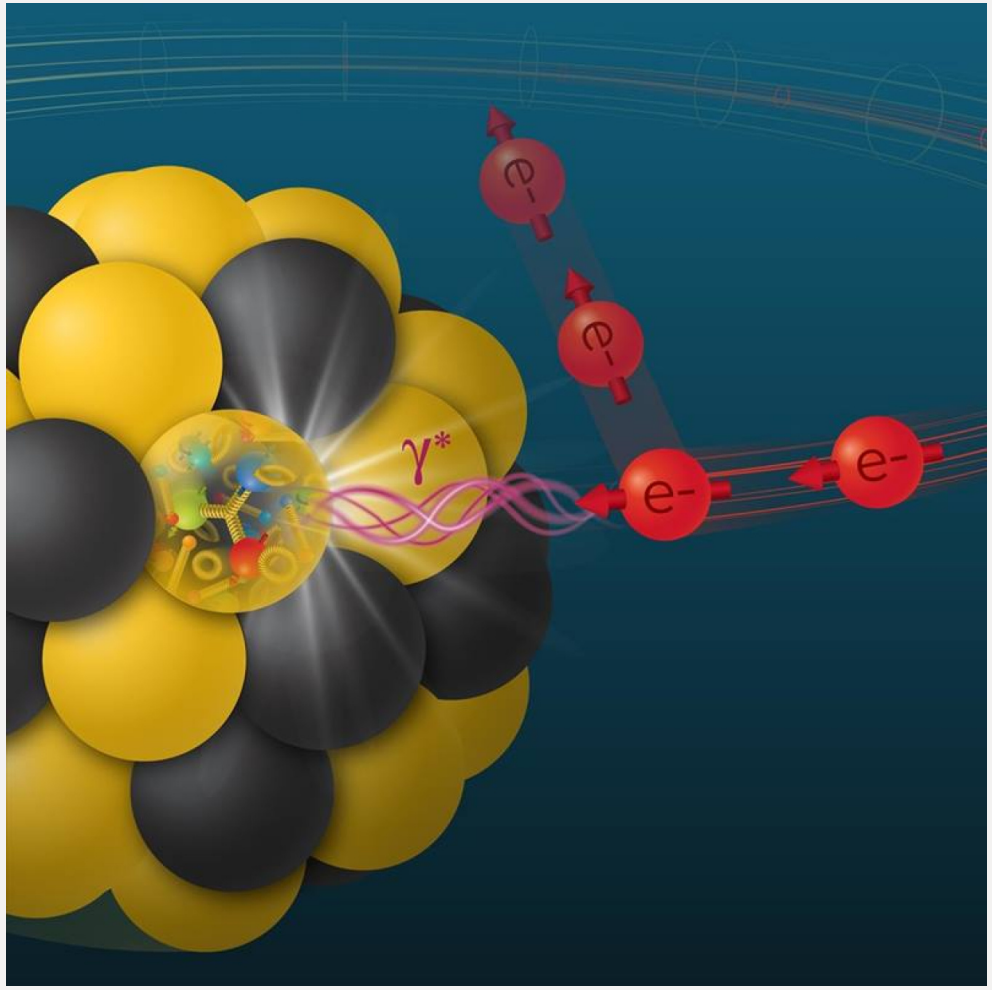
Duke

Agenda

1. **The Physics: SIDIS Kinematics & Reconstruction Results** (~20 min)
2. The Machine Learning: PFN Architecture & Training (~20 min)
3. Discussion & Questions (~20 min)

Duke

Deep Inelastic Scattering (DIS)



At Brookhaven National Laboratory, the **Electron-Ion Collider** will collide electrons with nuclei to study nucleon structure.

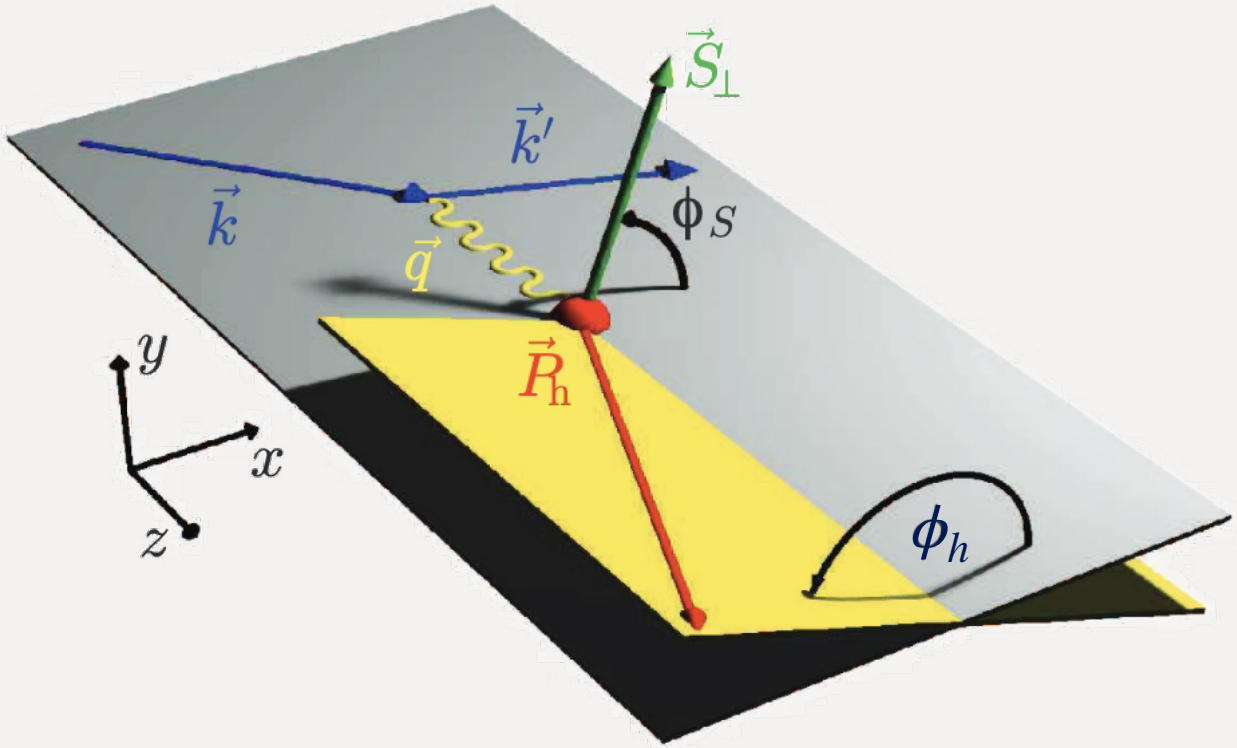
- Electrons are a very clean probe.
- High resolution allows us to see **inside the nucleon**

$$\text{resolution} \sim \frac{\hbar}{|\vec{q}|}$$

Duke

Semi-Inclusive Deep Inelastic Scattering (SIDIS)

$$e + N \rightarrow e' + h + X$$



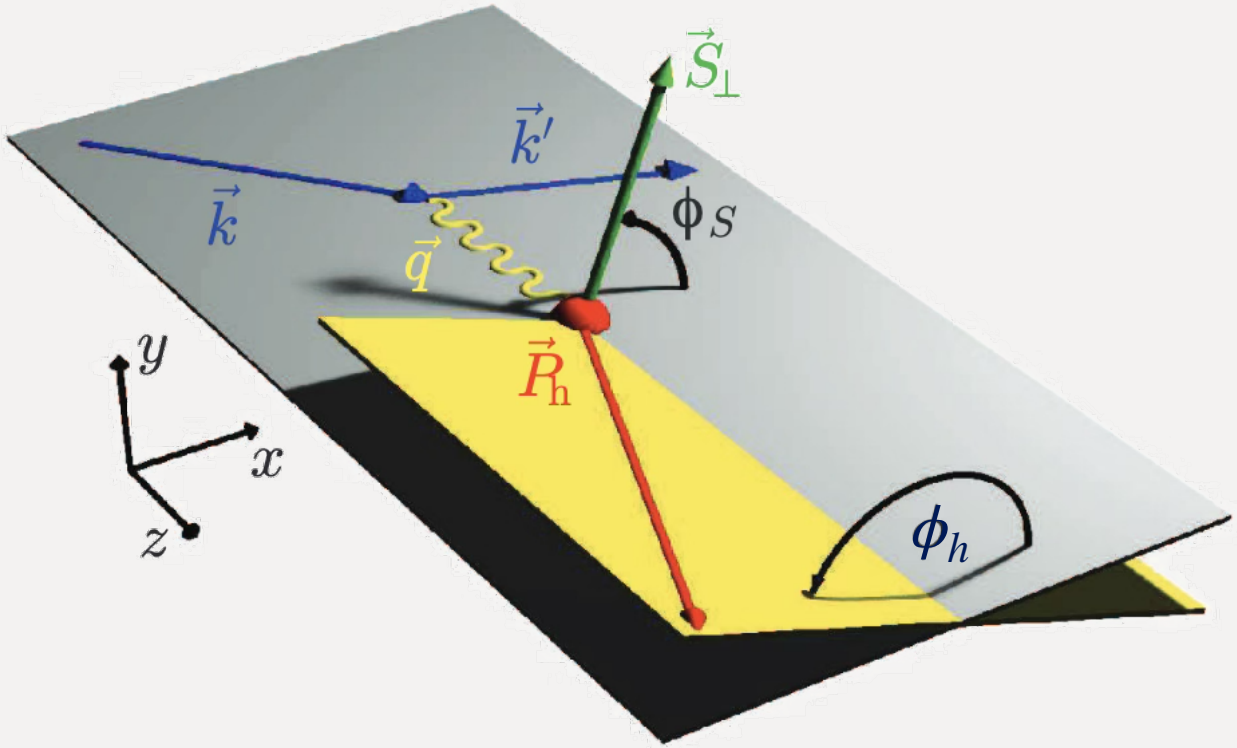
Differential Cross Section

$$\frac{d^5\sigma}{dx\ dQ^2\ dz\ dP_{hT}^2\ d\phi_h}$$

Duke

Semi-Inclusive Deep Inelastic Scattering (SIDIS)

$$e + N \rightarrow e' + h + X$$



Differential Cross Section

$$\frac{d^5\sigma}{dx\ dQ^2\ dz\ dP_{hT}^2\ d\phi_h}$$

DNNs have already been shown to improve the reconstruction of the inclusive variables …

Reconstructing the Kinematics of Deep Inelastic Scattering with Deep Learning

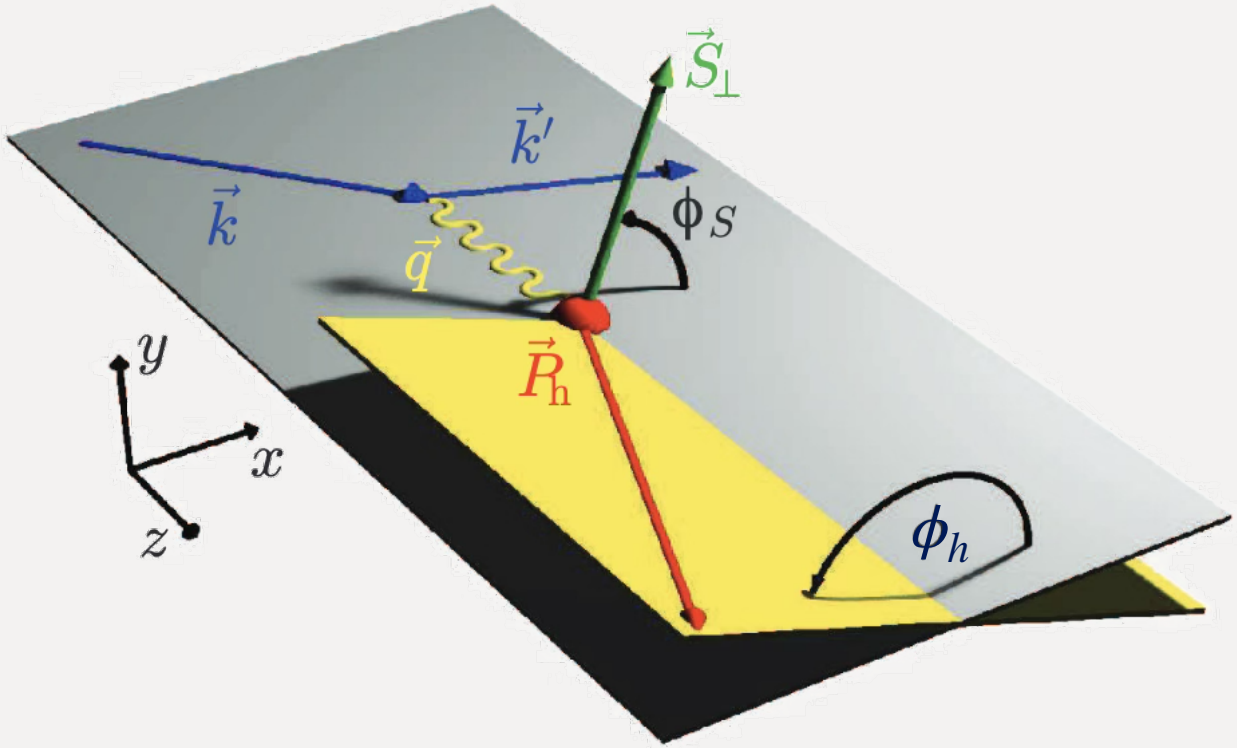
Miguel Arratia,[a,b] Daniel Britzger,[c] Owen Long,[*a] and Benjamin Nachman[d,e]

[a] *Department of Physics and Astronomy, University of California, Riverside, CA 92521, USA*
[b] *Thomas Jefferson National Accelerator Facility, Newport News, VA 23606, USA*
[c] *Max-Planck-Institut für Physik, Föhringer Ring 6, 80805 München, Germany*
[d] *Physics Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA*
[e] *Berkeley Institute for Data Science, University of California, Berkeley, CA 94720, USA*

Duke

Semi-Inclusive Deep Inelastic Scattering (SIDIS)

$$e + N \rightarrow e' + h + X$$



Differential Cross Section

$$\frac{d^5\sigma}{dx\ dQ^2\ dz\ dP_{hT}^2\ d\phi_h}$$

but ML has not yet been applied to the reconstruction of the semi-inclusive variables…

z : hadron momentum fraction

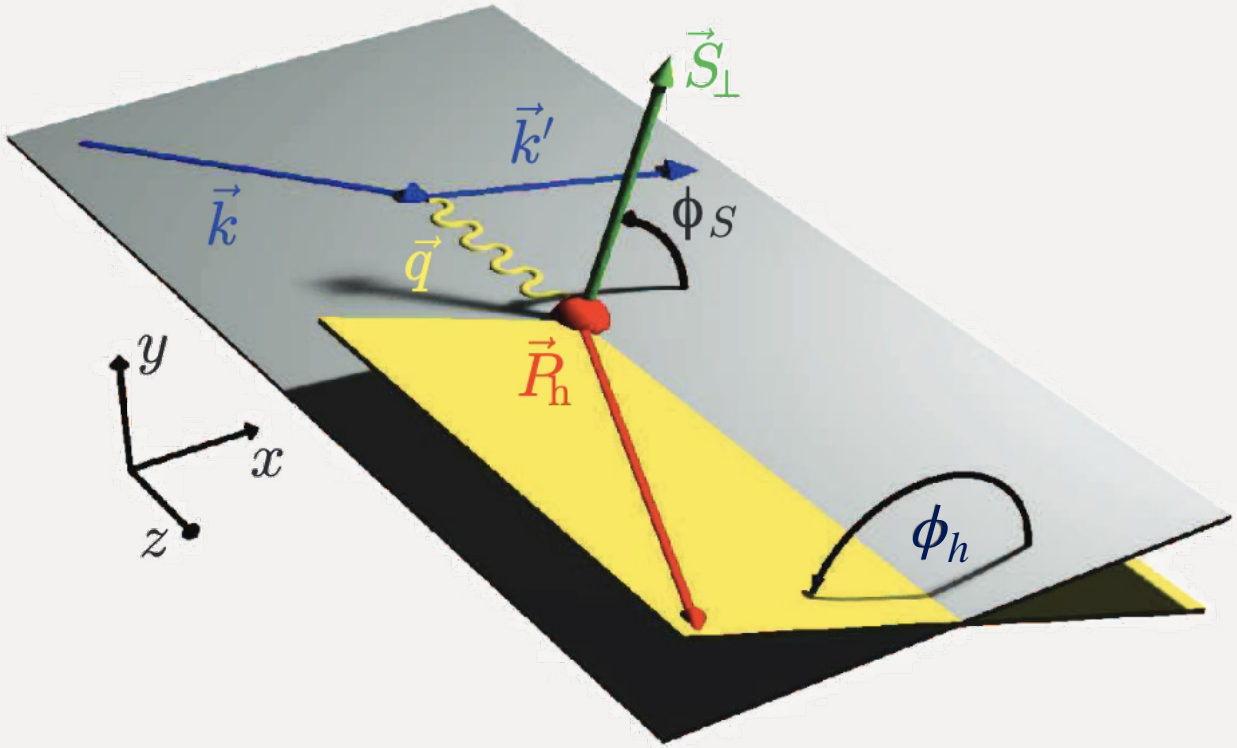
P_{hT} : hadron transverse momentum

ϕ_h : hadron plane angle

Duke

Kinematic Reconstruction

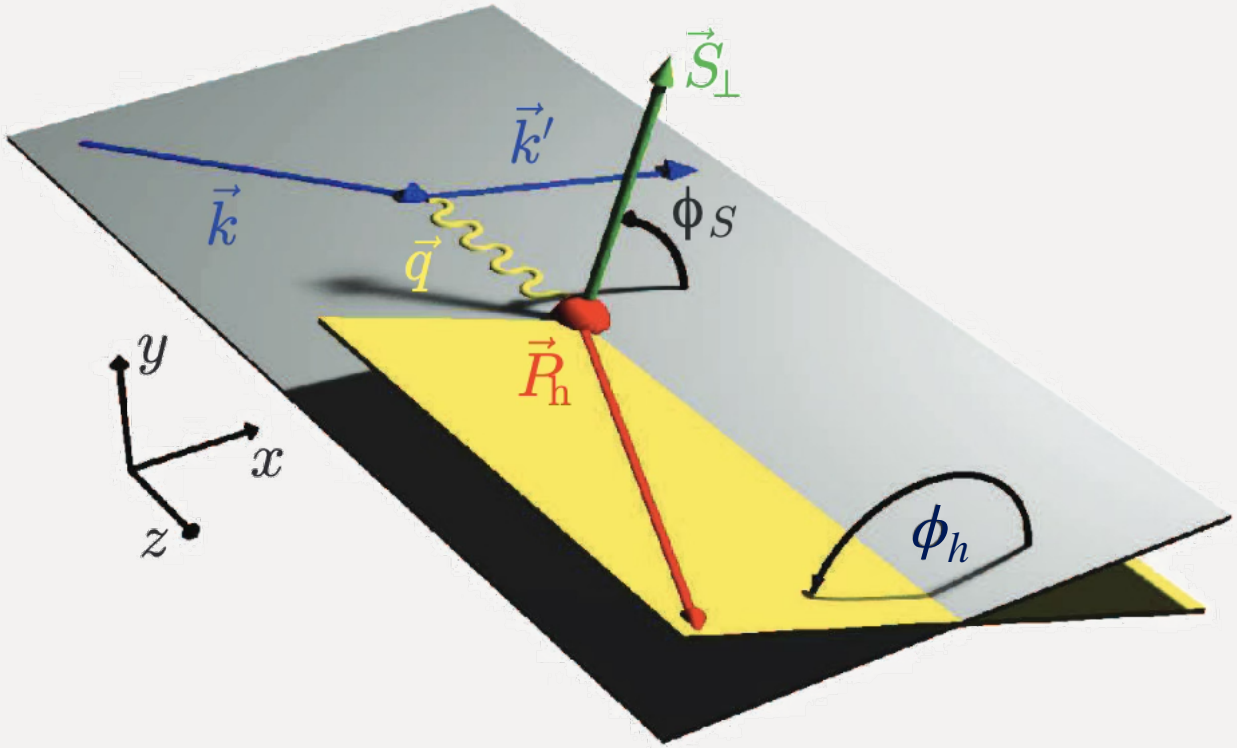
$$e + N \rightarrow e' + h + X$$



Duke

Kinematic Reconstruction

$$e + N \rightarrow e' + h + X$$



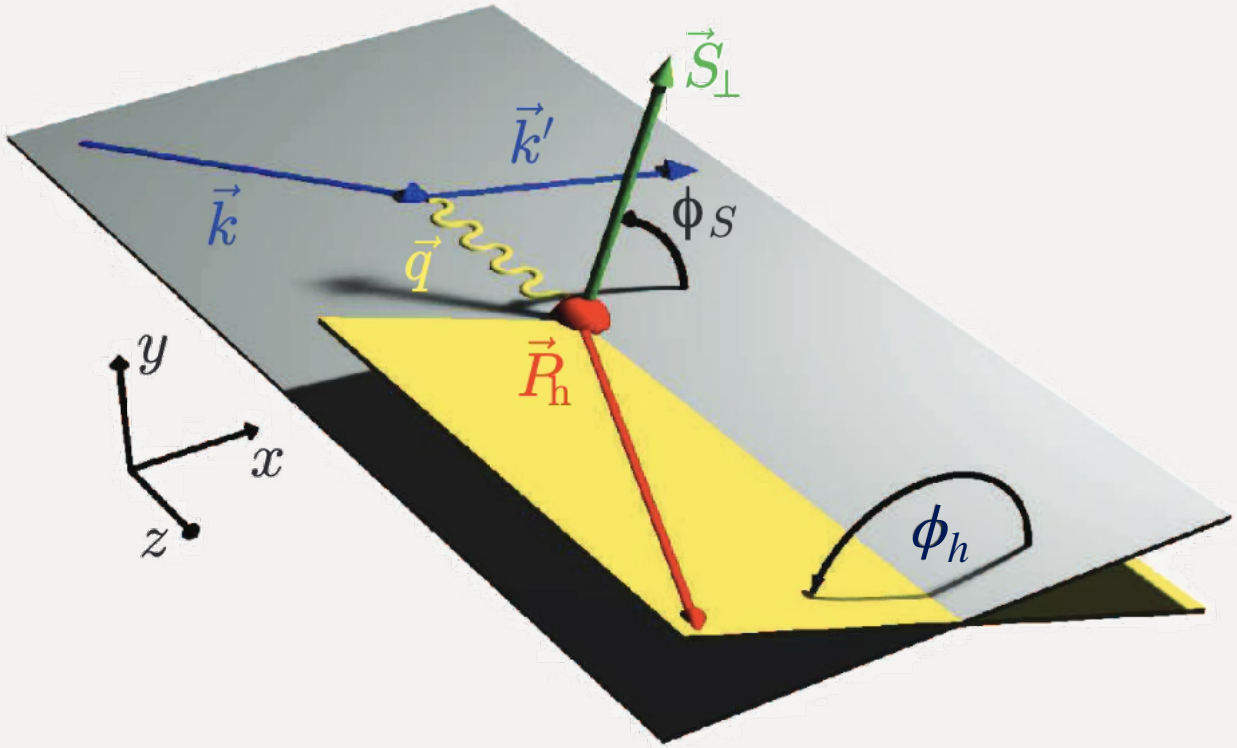
To calculate P_{hT} and ϕ_h we first need to **define our coordinate system** …

Our ability to reconstruct $\vec{q}$ will directly impact the accuracy of P_{hT} and ϕ_h

Duke

Kinematic Reconstruction

$$e + N \rightarrow e' + h + X$$



Methods for reconstructing $\vec{q}$

Electron Method
Only using scattered electron kinematics.

Jacquet-Blondel (JB) Method
Using the HFS kinematics.

Double Angle (DA) Method
Using both the scattered electron and HFS kinematics.

Machine Learning Method: Particle Flow Network

Duke

Results

Simulated Dataset Info:

- 25.10.4 ePIC Campaign
  - DJANGOH version 4.6.21-1.0
- Neutral Current
- 18 GeV e- beam
- 275 GeV proton beam
- $1\ GeV^2 \leq Q^2 \leq 10^4\ GeV^2$
- 60,000 events

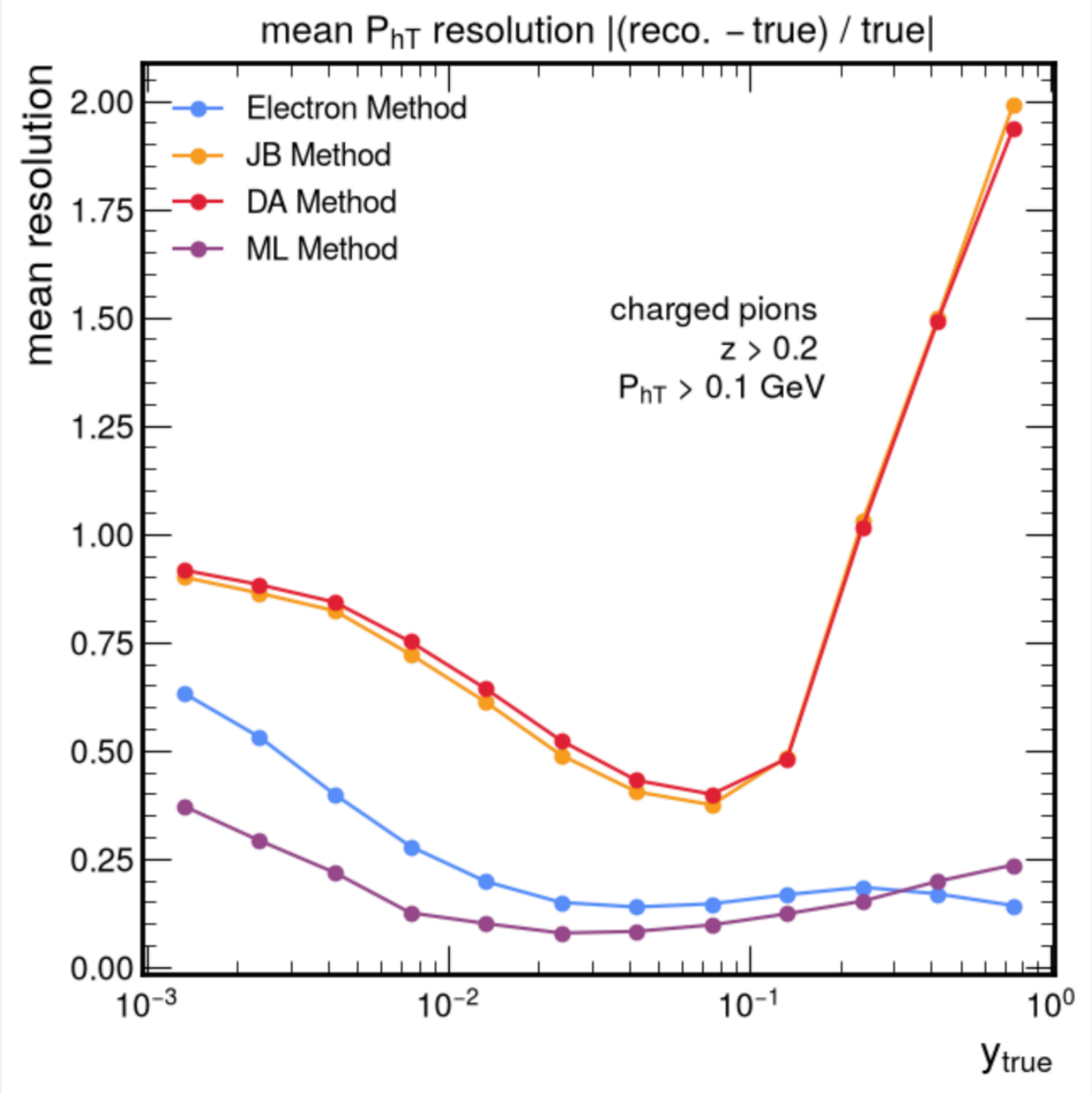
Duke

Results

Inelasticity

$$y = \frac{E_\gamma}{E_e}$$

(in the target rest frame)



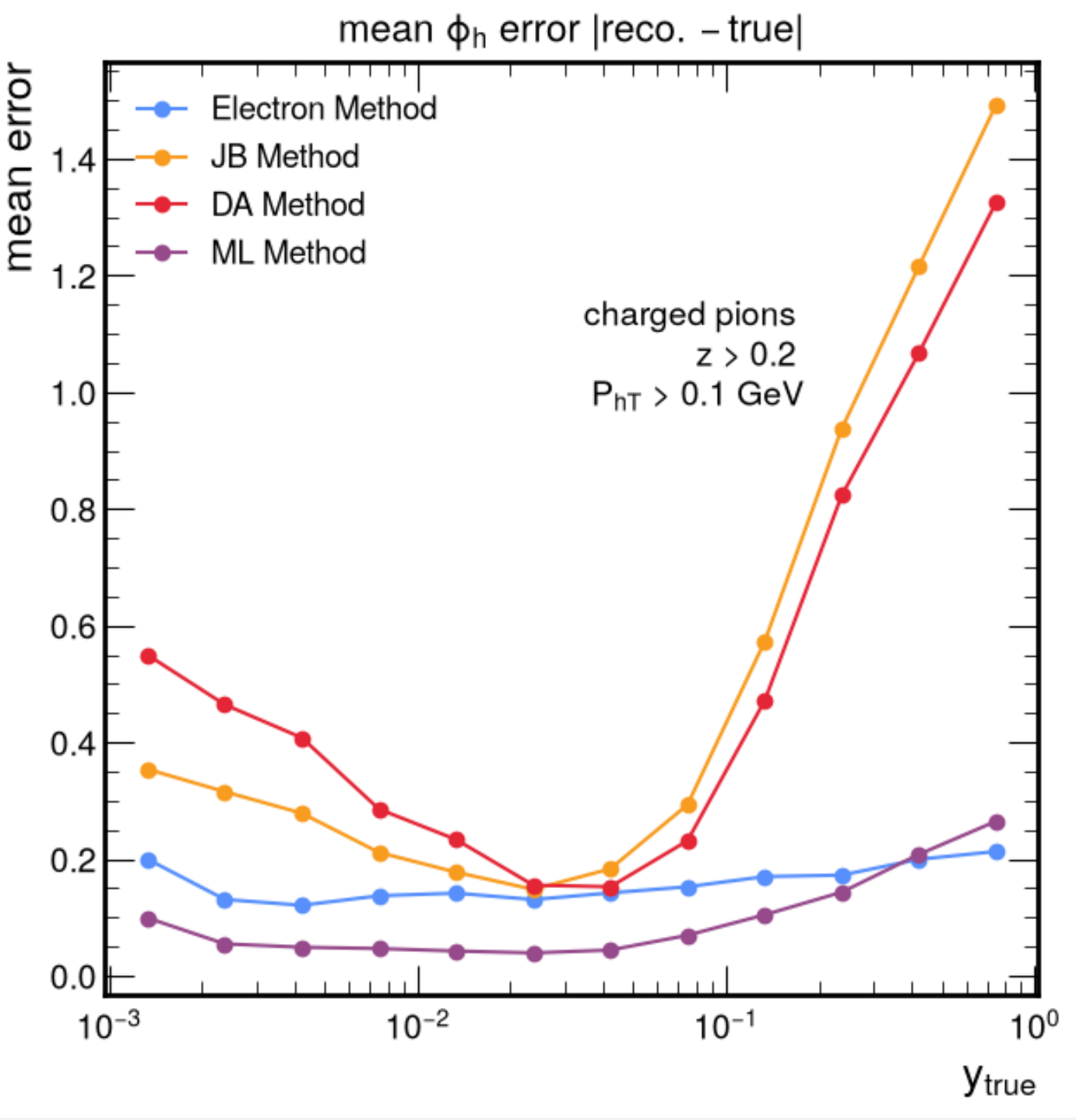
Duke

Results

Inelasticity

$$y = \frac{E_\gamma}{E_e}$$

(in the target rest frame)



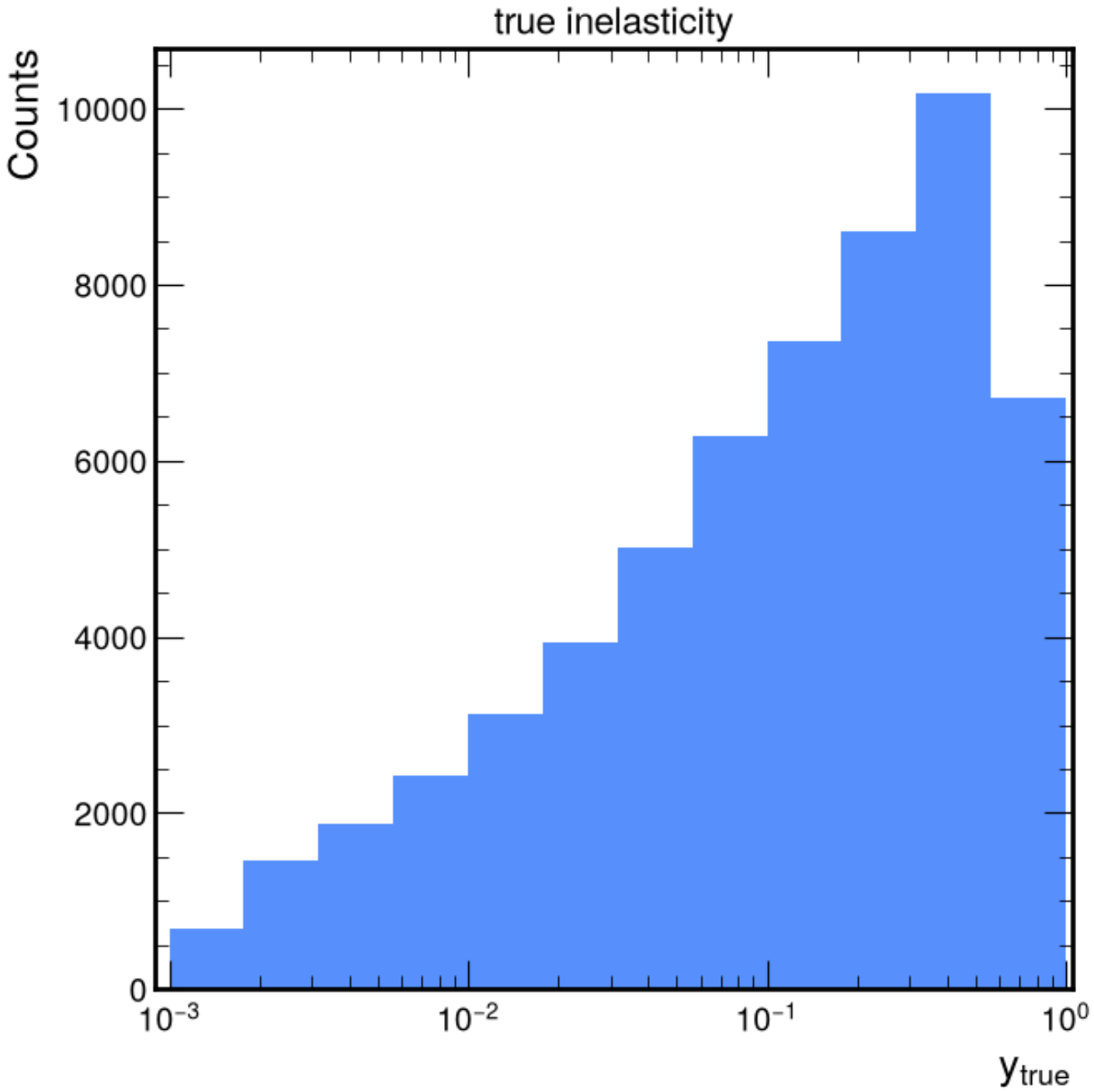
Duke

Results

Inelasticity

$$y = \frac{E_\gamma}{E_e}$$

(in the target rest frame)

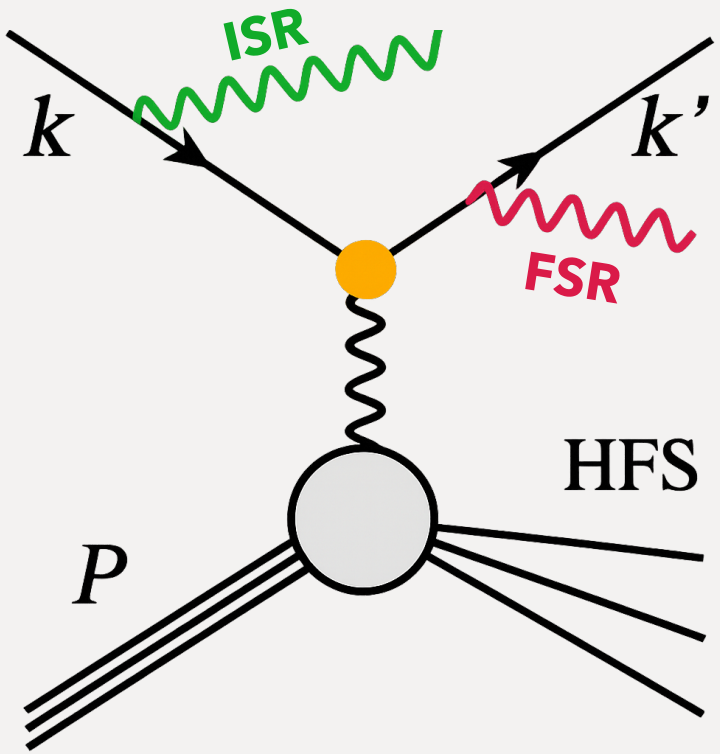


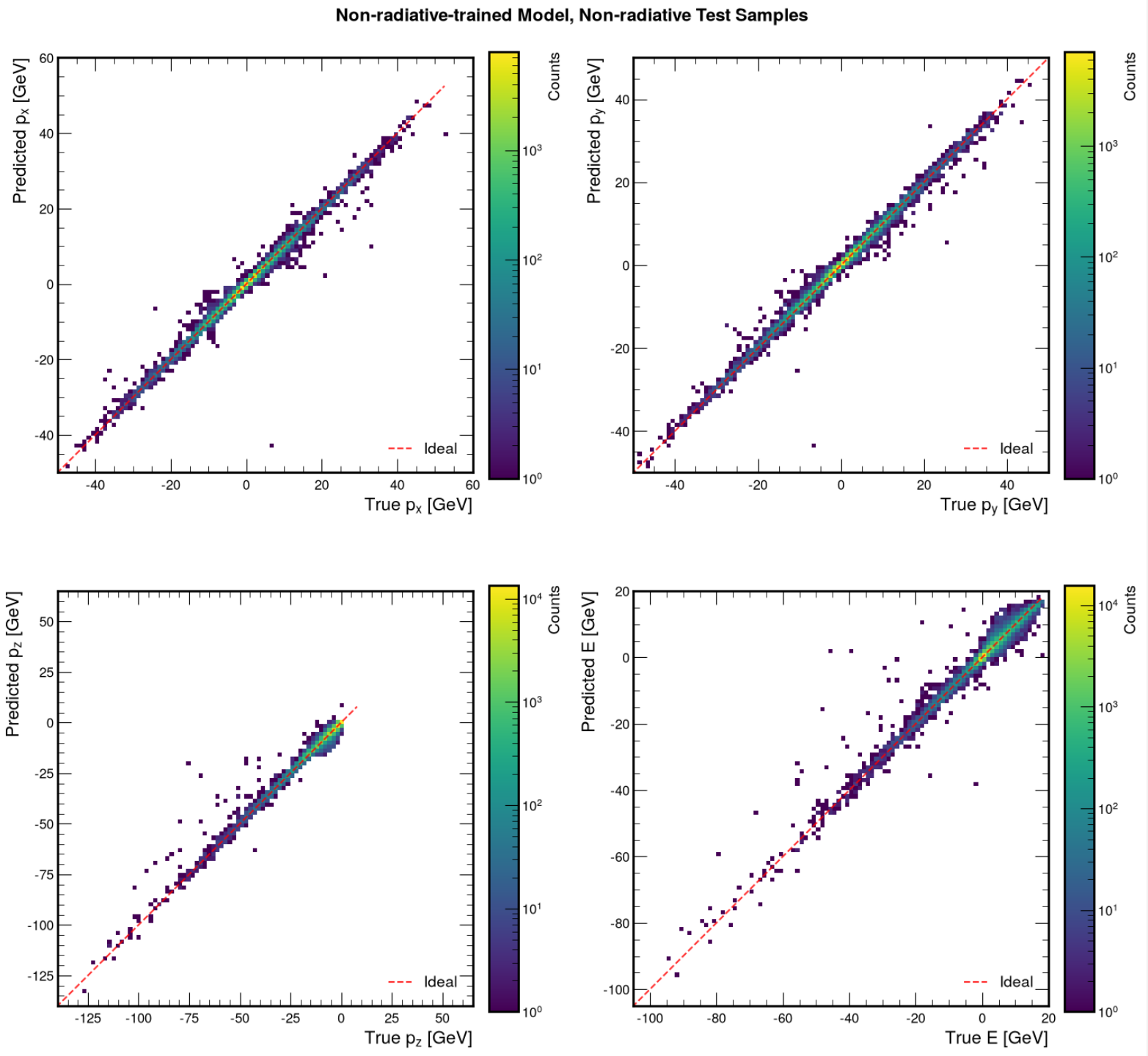
Duke

Results

How well does the PFN account for radiation?

- Initial State Radiation (ISR)
- Final State Radiation (FSR)

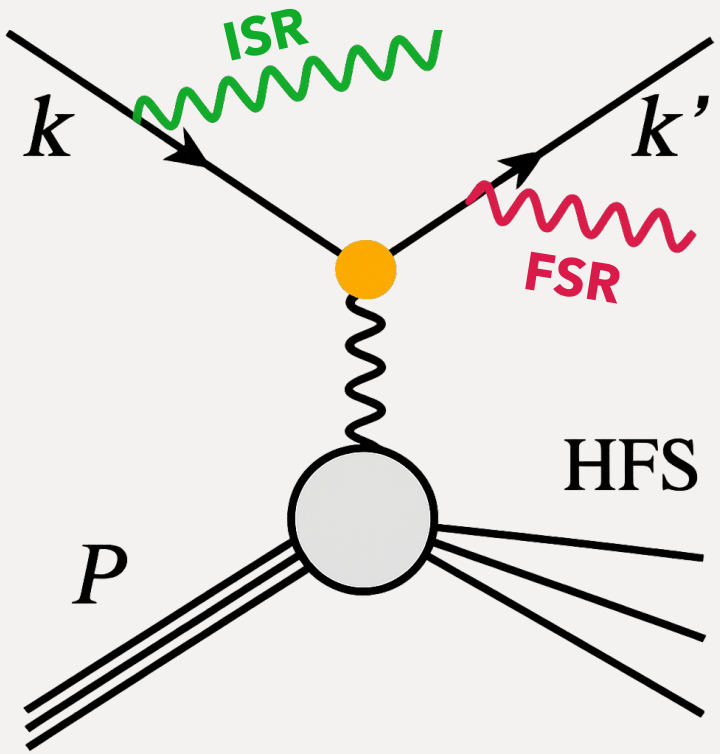


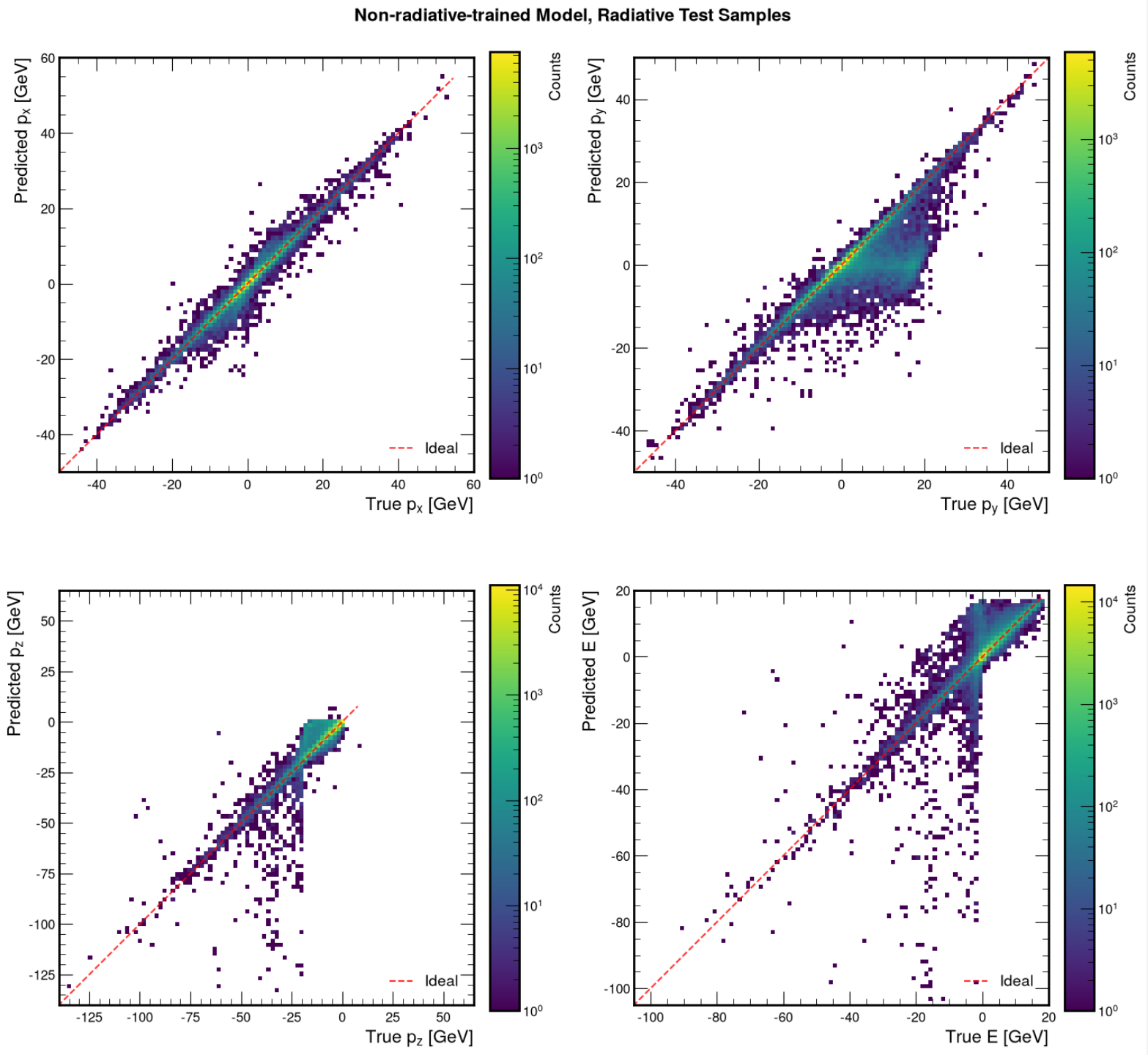


Results

How well does the PFN account for radiation?

- Initial State Radiation (ISR)
- Final State Radiation (FSR)

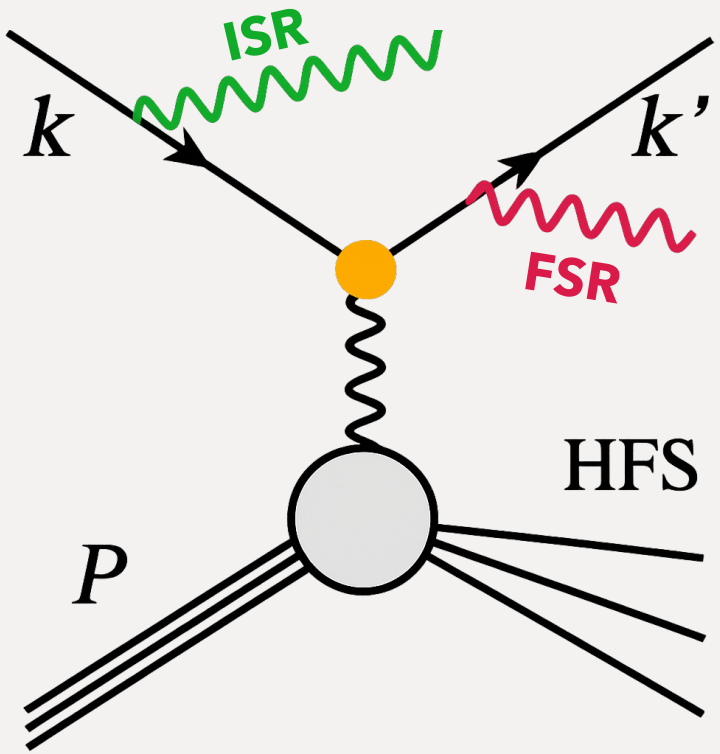


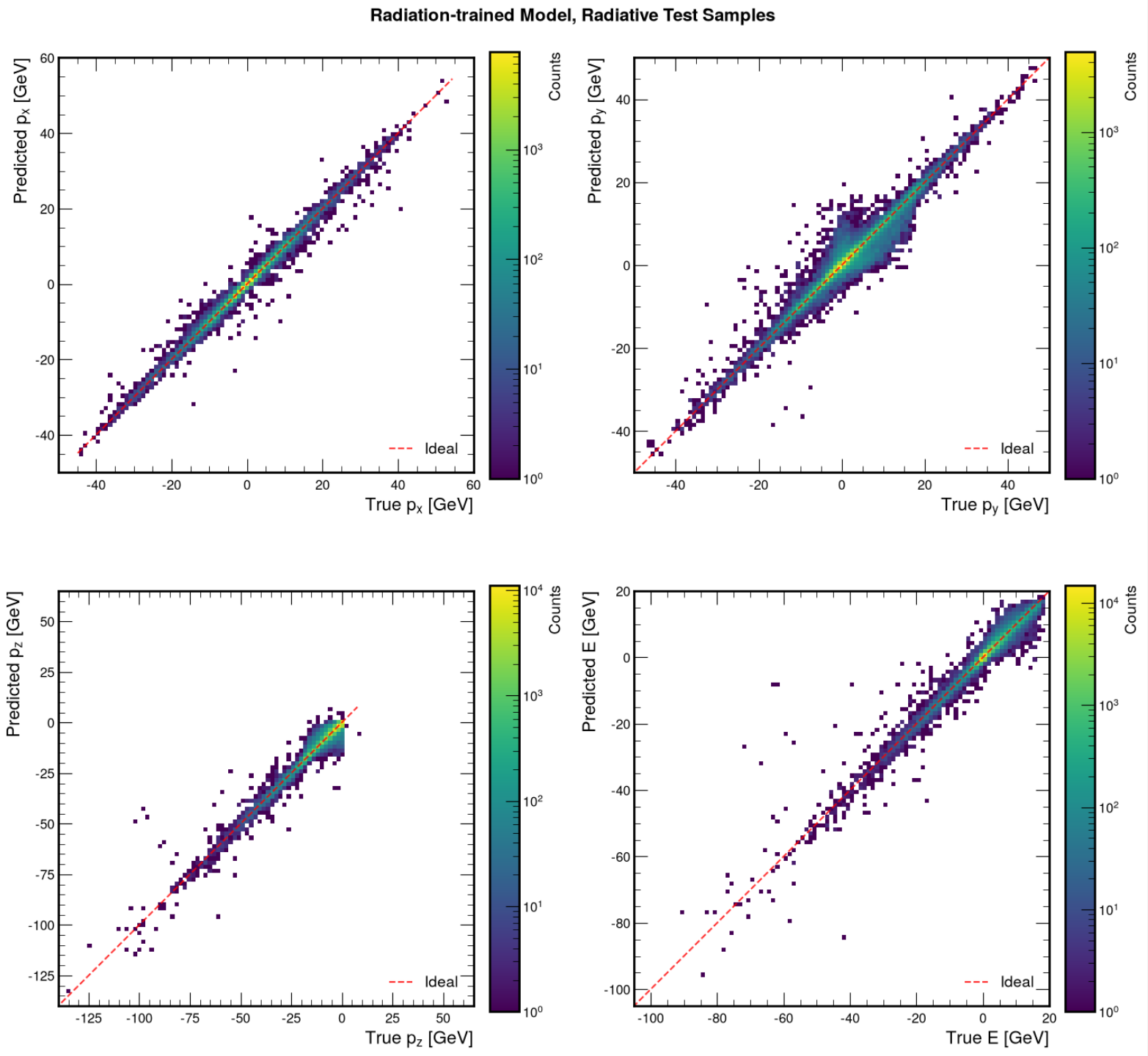


Results

How well does the PFN account for radiation?

- Initial State Radiation (ISR)
- Final State Radiation (FSR)

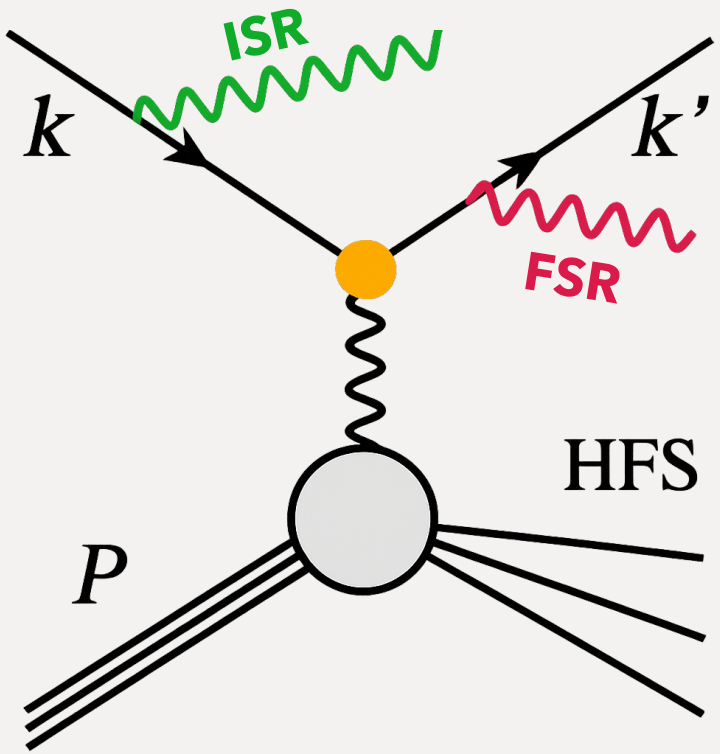


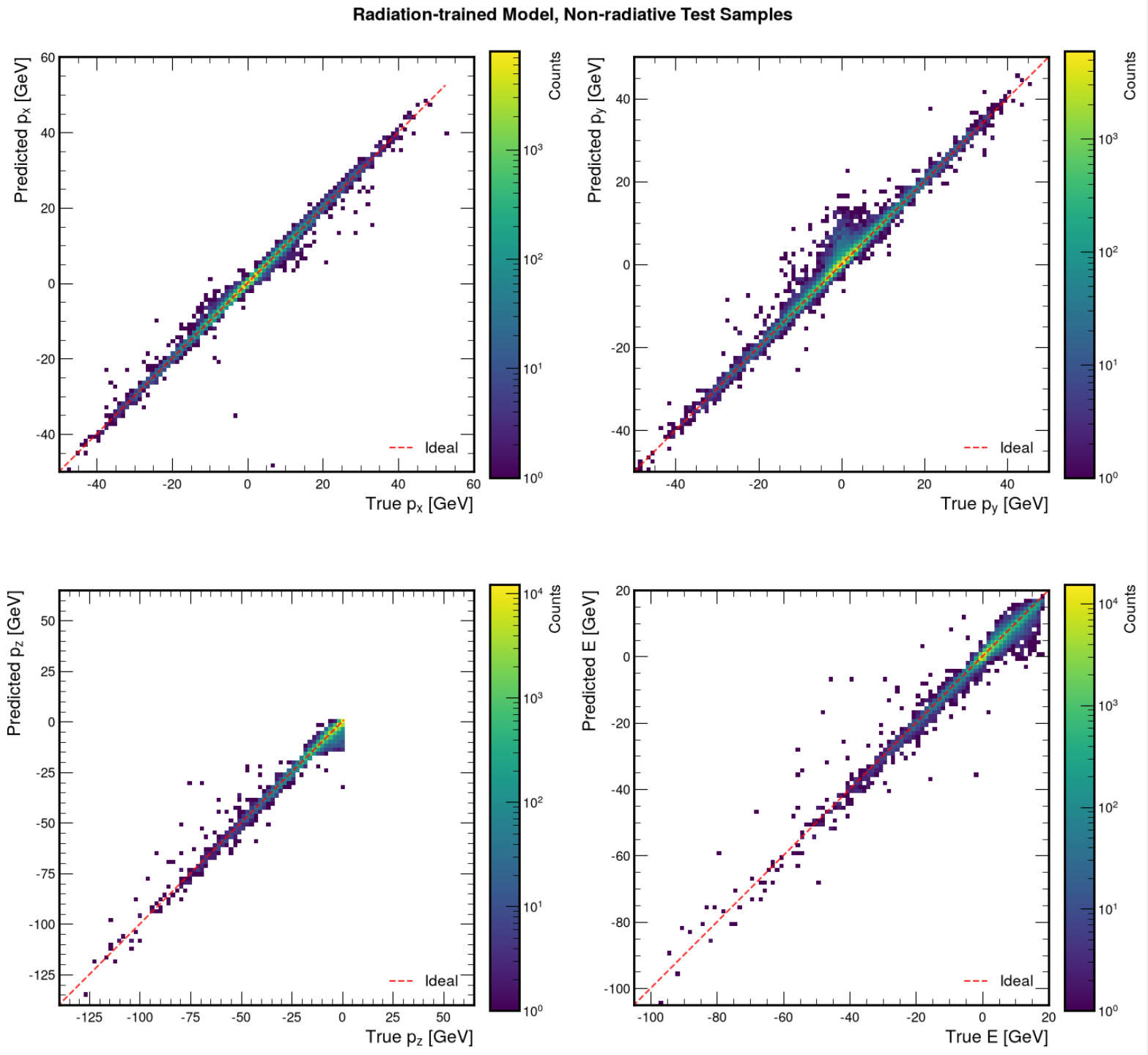


Results

How well does the PFN account for radiation?

- Initial State Radiation (ISR)
- Final State Radiation (FSR)





Agenda

1. The Physics: SIDIS Kinematics & Reconstruction Results (~20 min)
2. **The Machine Learning: PFN Architecture & Training** (~20 min)
3. Discussion & Questions (~20 min)

Duke

Deep Sets for Collider Events

Energy Flow Networks: Deep Sets for Particle Jets

Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler

Center for Theoretical Physics, Massachusetts Institute of Technology,
77 Massachusetts Avenue, Cambridge, MA 02139, U.S.A.

Department of Physics, Harvard University,
17 Oxford Street, Cambridge, MA 02138, U.S.A.

E-mail: pkomiske@mit.edu, metodiev@mit.edu, jthaler@mit.edu

ABSTRACT: A key question for machine learning approaches in particle physics is how to best represent and learn from collider events. As an event is intrinsically a variable-length unordered set of particles, we build upon recent machine learning efforts to learn directly from sets of features or "point clouds". Adapting and specializing the "Deep Sets" framework to particle physics, we introduce Energy Flow Networks, which respect infrared and collinear safety by construction. We also develop Particle Flow Networks, which allow for general energy dependence and the inclusion of additional particle-level information such as charge and flavor. These networks feature a per-particle internal (latent) representation, and summing over all particles yields an overall event-level latent representation. We show how this latent space decomposition unifies existing event representations based on detector images and radiation moments. To demonstrate the power and simplicity of this set-based approach, we apply these networks to the collider task of discriminating quark jets from gluon jets, finding similar or improved performance compared to existing methods. We also show how the learned event representation can be directly visualized, providing insight into the inner workings of the model. These architectures lend themselves to efficiently processing and analyzing events for a wide variety of tasks at the Large Hadron Collider. Implementations and examples of our architectures are available online in our EnergyFlow package.

Duke

Deep Sets Theorem

The Challenge: Each event has a variable number of particles, with no natural order.

What we want: A permutation-invariant model that can take a variable number of inputs.

Duke

Deep Sets Theorem

The Challenge: Each event has a variable number of particles, with no natural order.

What we want: A permutation-invariant model that can take a variable number of inputs.

Deep Sets Theorem

A function f is permutation-invariant if and only if there exist functions ϕ and F such that

$$f(\{x_1, \ldots, x_N\}) = F(\sum_{i=1}^{N} \phi(x_i))$$

Duke

Applications of Deep Sets Theorem

Energy Flow Network (EFN)

$$F\left(\sum_{i=1}^{N} z_i \, \phi(\hat{p}_i)\right)$$

- Input particle information is limited to particle direction ($\hat{p}_i$), and a weight z_i
- Preserves infrared and collinear (IRC) safety, allowing outputs to be compared with purtubative QCD predictions.

Particle Flow Network (EFN)

$$F\left(\sum_{i=1}^{N} \phi(z_i, \hat{p}_i, \ldots)\right)$$

- No restraint on input particle information
- Can learn from a broader selection of particle information
- Does not preserve IRC safety

Duke

Applications of Deep Sets Theorem

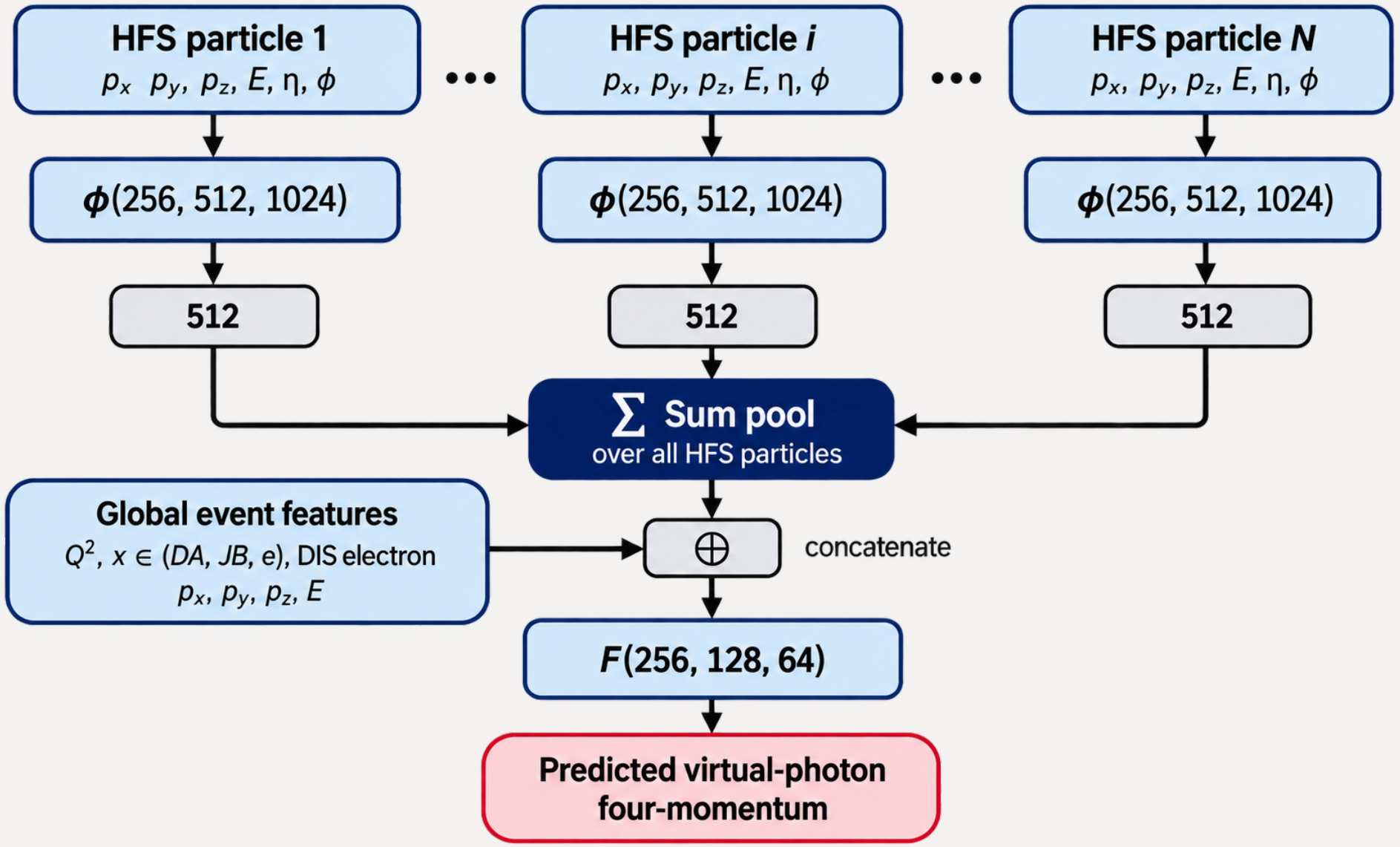
Particle Flow Network (EFN)

$$F(\sum_{i=1}^{N} \phi(z_i, \hat{p}_i, \dots))$$

- No restraint on input particle information
- Can learn from a broader selection of particle information
- Does not preserve IRC safety

Duke

Particle Flow Network



Duke

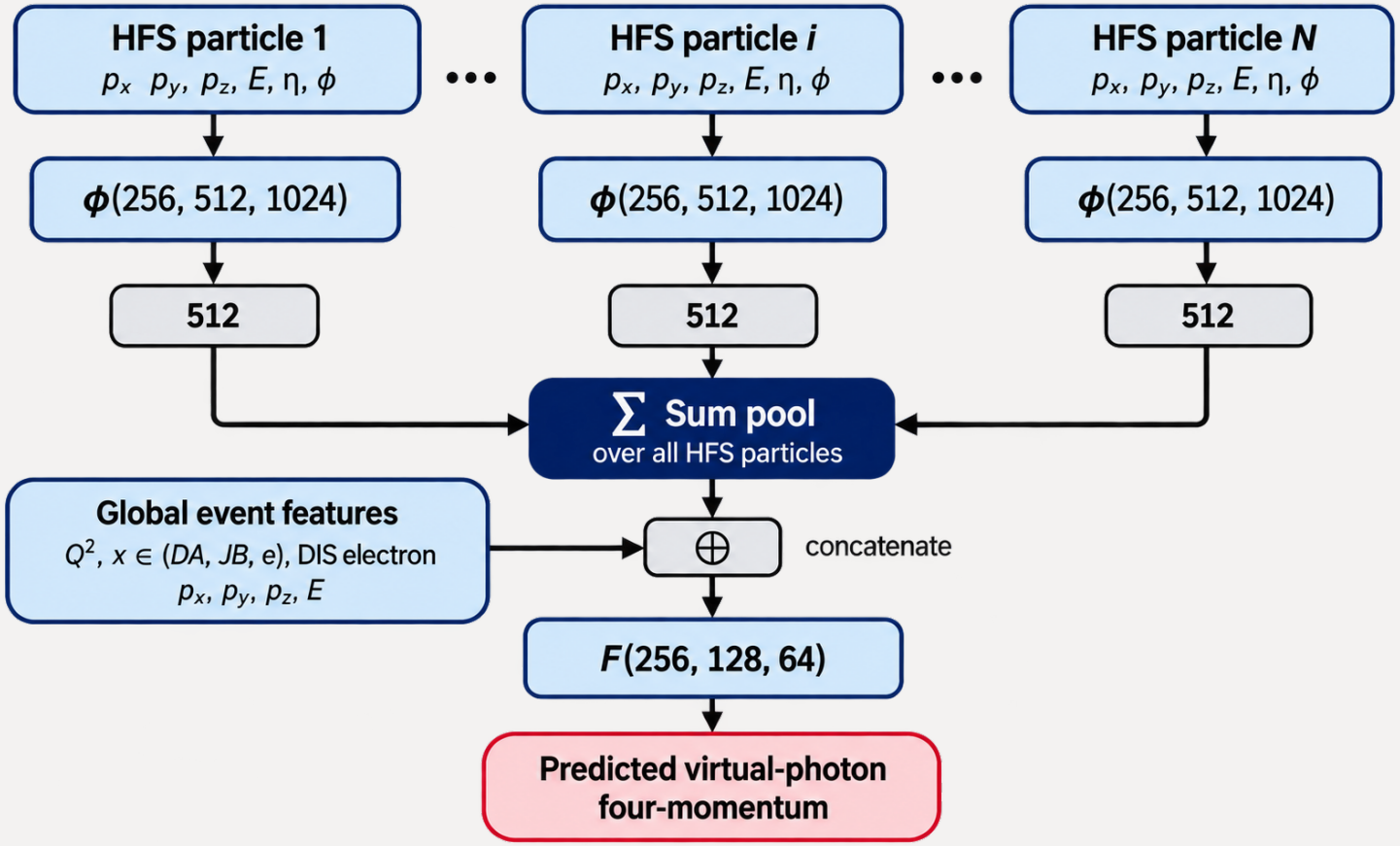
Machine Learning Method: Particle Flow Network

Designed and trained with Keras and TensorFlow

Training dataset: ~500,000 events
Validation dataset: ~60,000 events
Test dataset: ~60,000 events

All layers of ϕ and F use a ReLU activation function.

The final output layer uses a linear activation function.

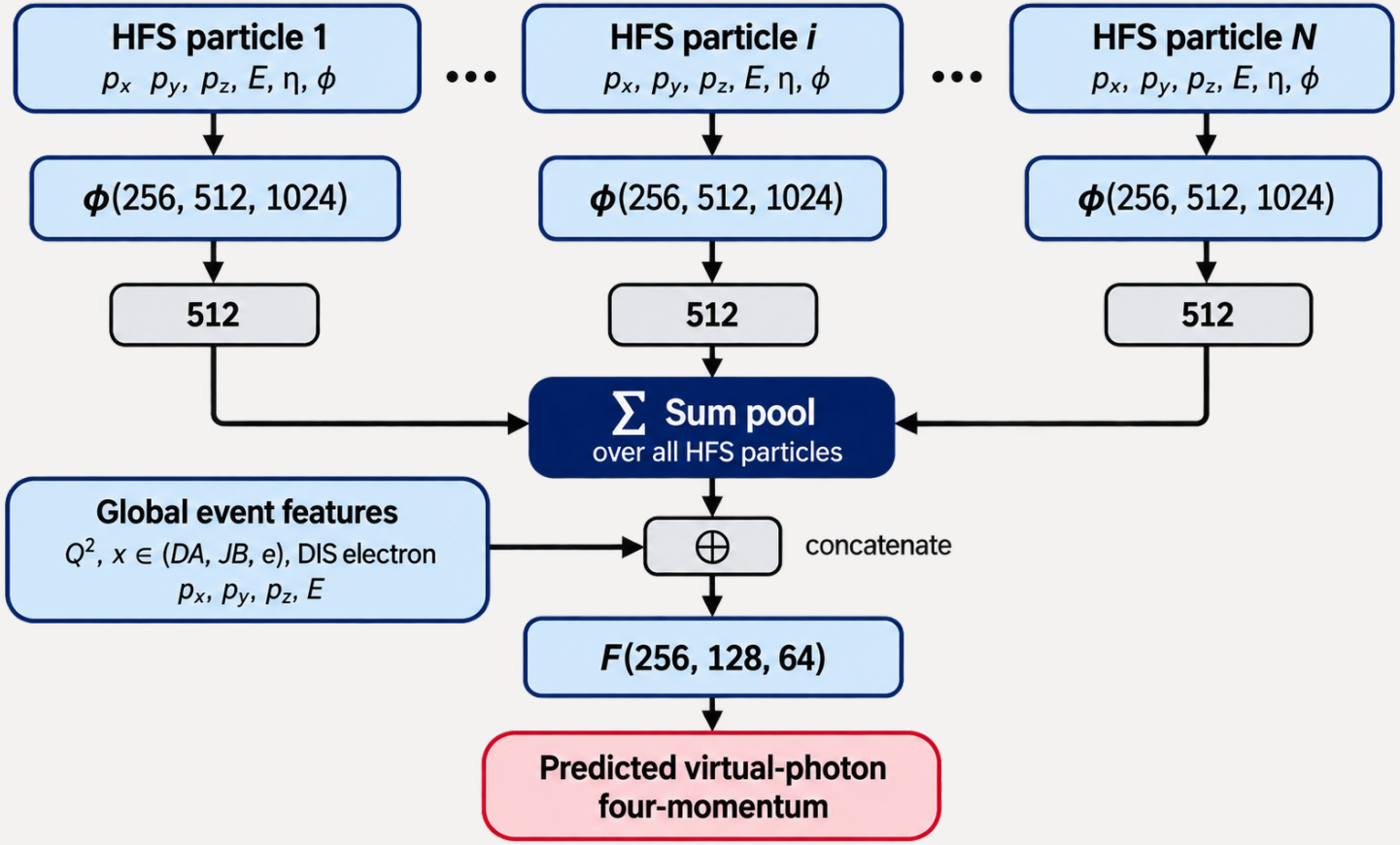


Machine Learning Method: Particle Flow Network

Mean Squared Error (MSE) Loss Function

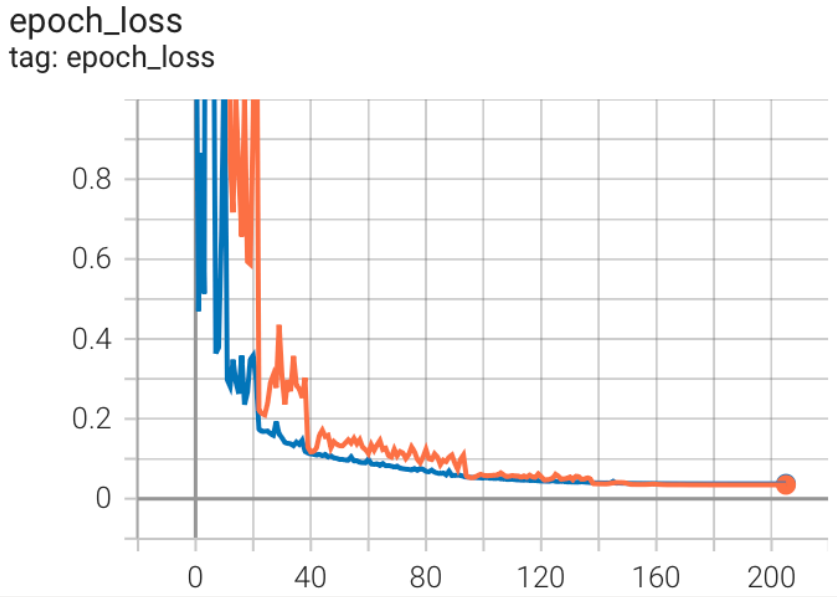
Batch size: 1000

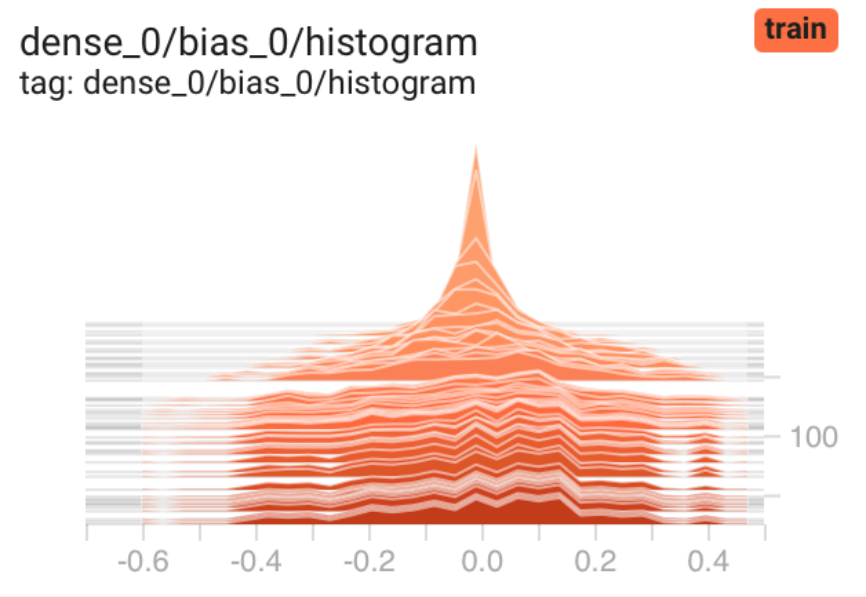
Initial learning rate is $3.0 \cdot 10^{-4}$. Learning rate is cut in half whenever validation loss has not improved after five epochs.

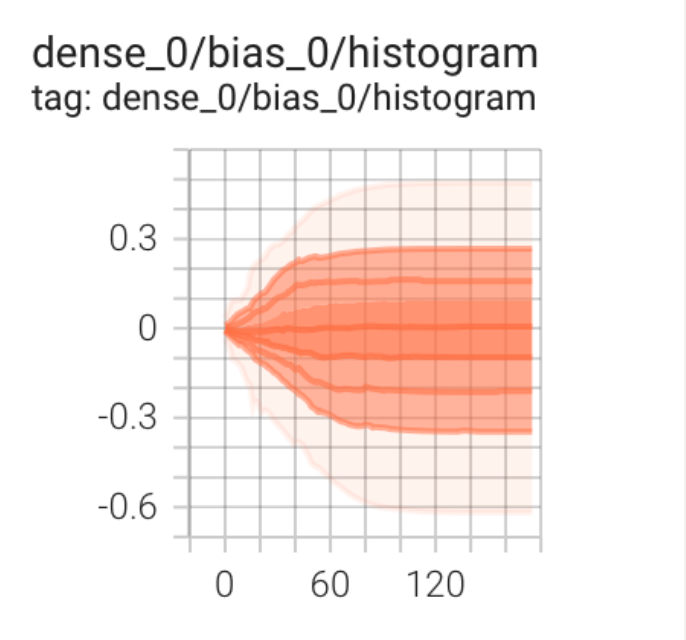


TensorBoard

A tool for visualizing metrics logged during model training.







Duke

Conclusions

1. The PFN outperforms the JB and DA methods across the full y range, and outperforms the electron method for $y < 0.4$
2. The PFN improves its reconstruction when trained on events with radiation, suggesting that it learns to account for radiative effects.

Future Work

- Presentation accepted to AI4EIC 2026. Conference proceedings will be published in JINST.

✳ Can we extract a mathematical representation of the PFN's learned latent representation?

✳ Can we further improve the accuracy of the reconstruction by altering our PFN design?

✳ Explore alternative architectures?

Duke

Agenda

1. The Physics: SIDIS Kinematics & Reconstruction Results (~15 min)

2. The Machine Learning: PFN Architecture & Training (~15 min)

3. **Discussion & Questions** (~20 min)

Duke

Backup slides

Duke

Kinematic Reconstruction Methods

Electron Method

$$q = k - k'$$

Jacquet-Blondel (JB) Method

$$\delta_h = \sum_i^{N_{HFS}} E_i - p_{z,i}$$

$$p_{t,h}^2 = \left(\sum_i^{N_{HFS}} p_{x,i} \right)^2 + \left(\sum_i^{N_{HFS}} p_{y,i} \right)^2$$

$$Q_{JB}^2 = \frac{p_{t,h}^2}{1 - y_{JB}}$$

$$y_{JB} = \frac{\delta_h}{2E_0}$$

$$q_x = \sum_i^{N_{HFS}} p_{x,i} \qquad q_y = \sum_i^{N_{HFS}} p_{y,i}$$

$$q_z, q_t \leftarrow \begin{cases} Q^2 = -\left(q_x^2 + q_y^2 + q_z^2 - q_t^2\right) \\ y = \dfrac{P_x q_x + P_y q_y + P_z q_z - P_t q_t}{P \cdot k} \end{cases}$$

Double Angle (DA) Method

$$\alpha_e = \tan \frac{\theta_e}{2}$$

$$\alpha_h = \frac{\delta_h}{p_{t,h}}$$

$$Q_{DA}^2 = \frac{4E_0^2}{\alpha_e(\alpha_e + \alpha_h)}$$

$$y_{DA} = \frac{\alpha_h}{\alpha_e + \alpha_h}$$

Kinematic Reconstruction Methods

Electron Method

$$q = k - k'$$

Jacquet-Blondel (JB) Method

$$\delta_h = \sum_i^{N_{HFS}} E_i - p_{z,i}$$

$$p_{t,h}^2 = \left(\sum_i^{N_{HFS}} p_{x,i}\right)^2 + \left(\sum_i^{N_{HFS}} p_{y,i}\right)^2$$

$$Q_{JB}^2 = \frac{p_{t,h}^2}{1 - y_{JB}}$$

$$y_{JB} = \frac{\delta_h}{2E_0}$$

$$q_x = \sum_i^{N_{HFS}} p_{x,i} \qquad q_y = \sum_i^{N_{HFS}} p_{y,i}$$

$$q_z, q_t \leftarrow \begin{cases} Q^2 = -(q_x^2 + q_y^2 + q_z^2 - q_t^2) \\ y = \dfrac{P_x q_x + P_y q_y + P_z q_z - P_t q_t}{P \cdot k} \end{cases}$$

Double Angle (DA) Method

$$\alpha_e = \tan\frac{\theta_e}{2}$$

$$\alpha_h = \frac{\delta_h}{p_{t,h}}$$

$$Q_{DA}^2 = \frac{4E_0^2}{\alpha_e(\alpha_e + \alpha_h)}$$

$$y_{DA} = \frac{\alpha_h}{\alpha_e + \alpha_h}$$

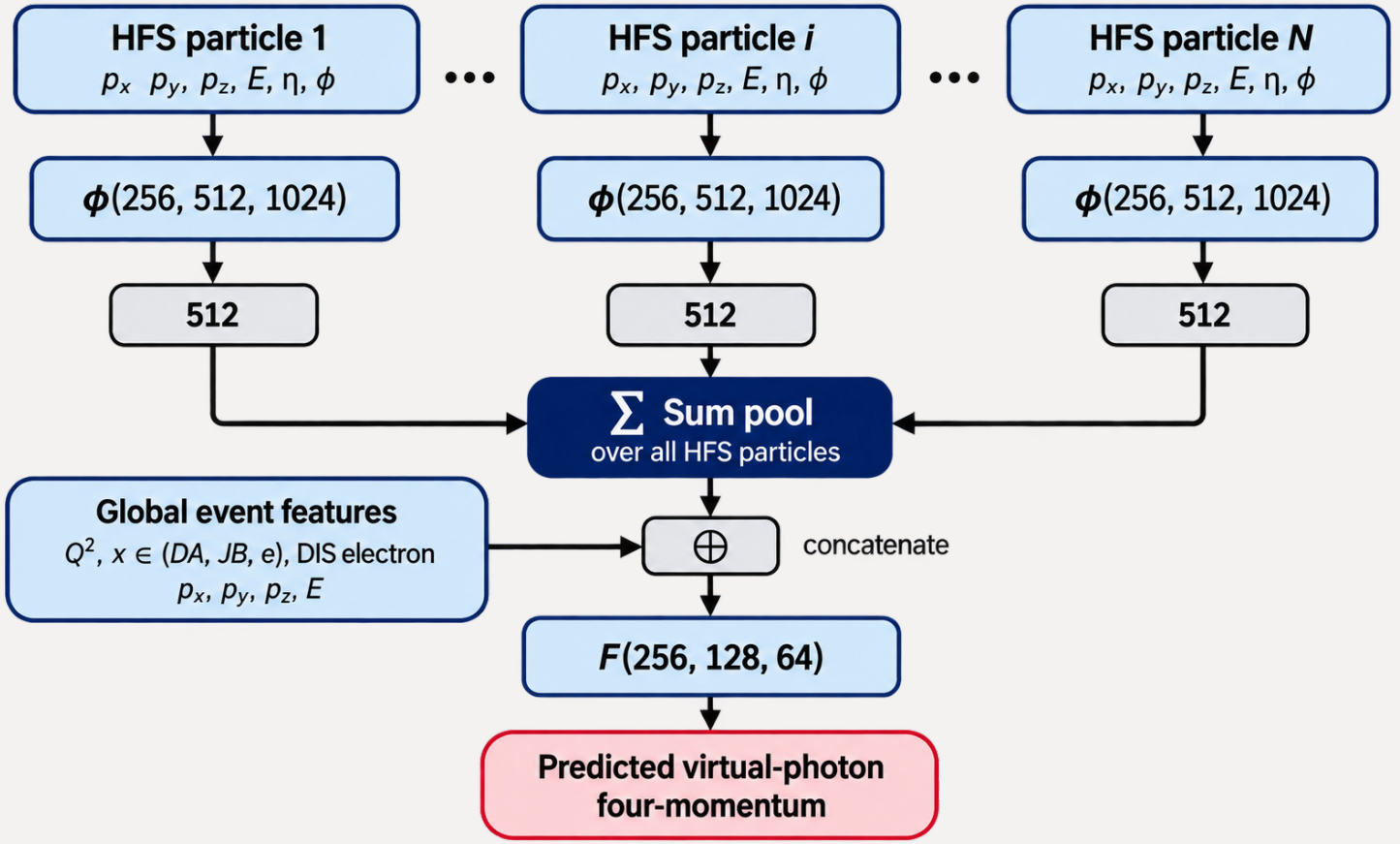
PFN Layer Dimensions

Model 1

$\phi(500, 500, 500)$, $F(200, 200, 200)$

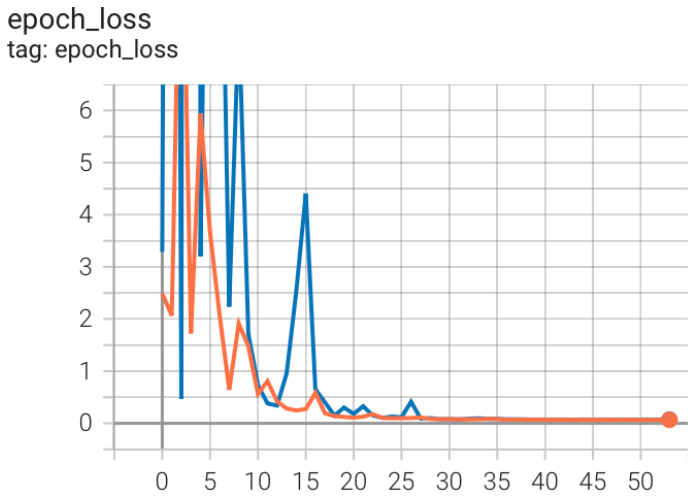
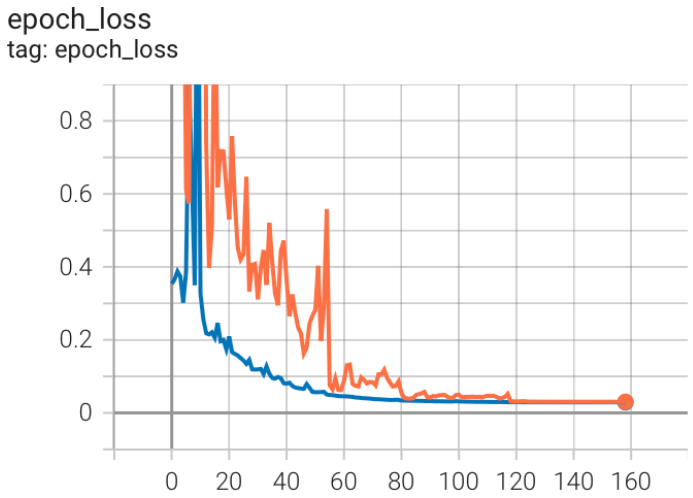
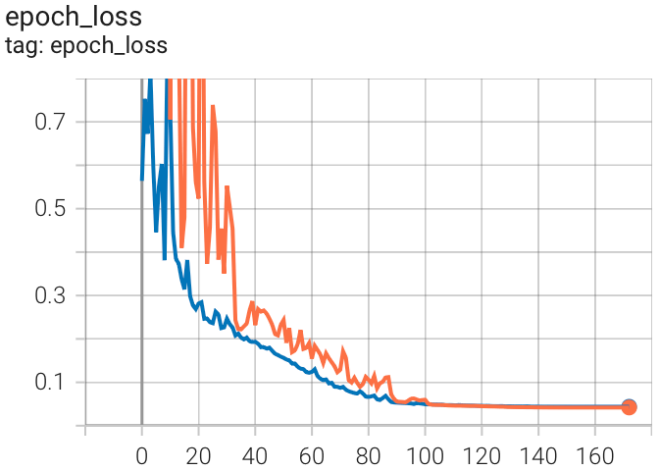
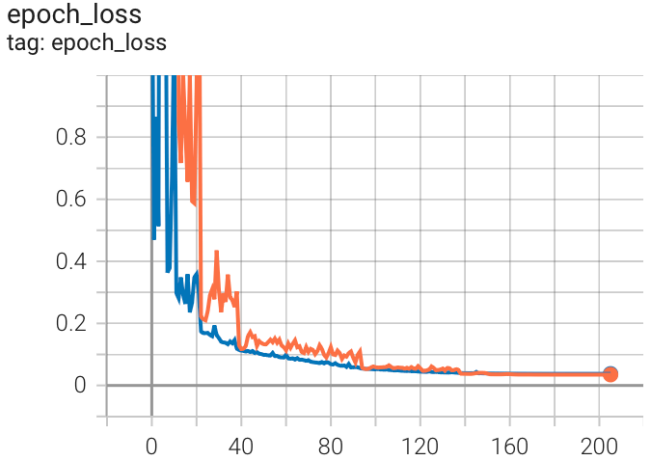
Model 2

$\phi(256, 512, 1024)$, $F(256, 128, 64)$



Duke

PFN Layer Dimensions

	Radiation-less Model	Radiation Model
Model 1		
Model 2		

Duke