

Generative AI for High Energy Nuclear Physics

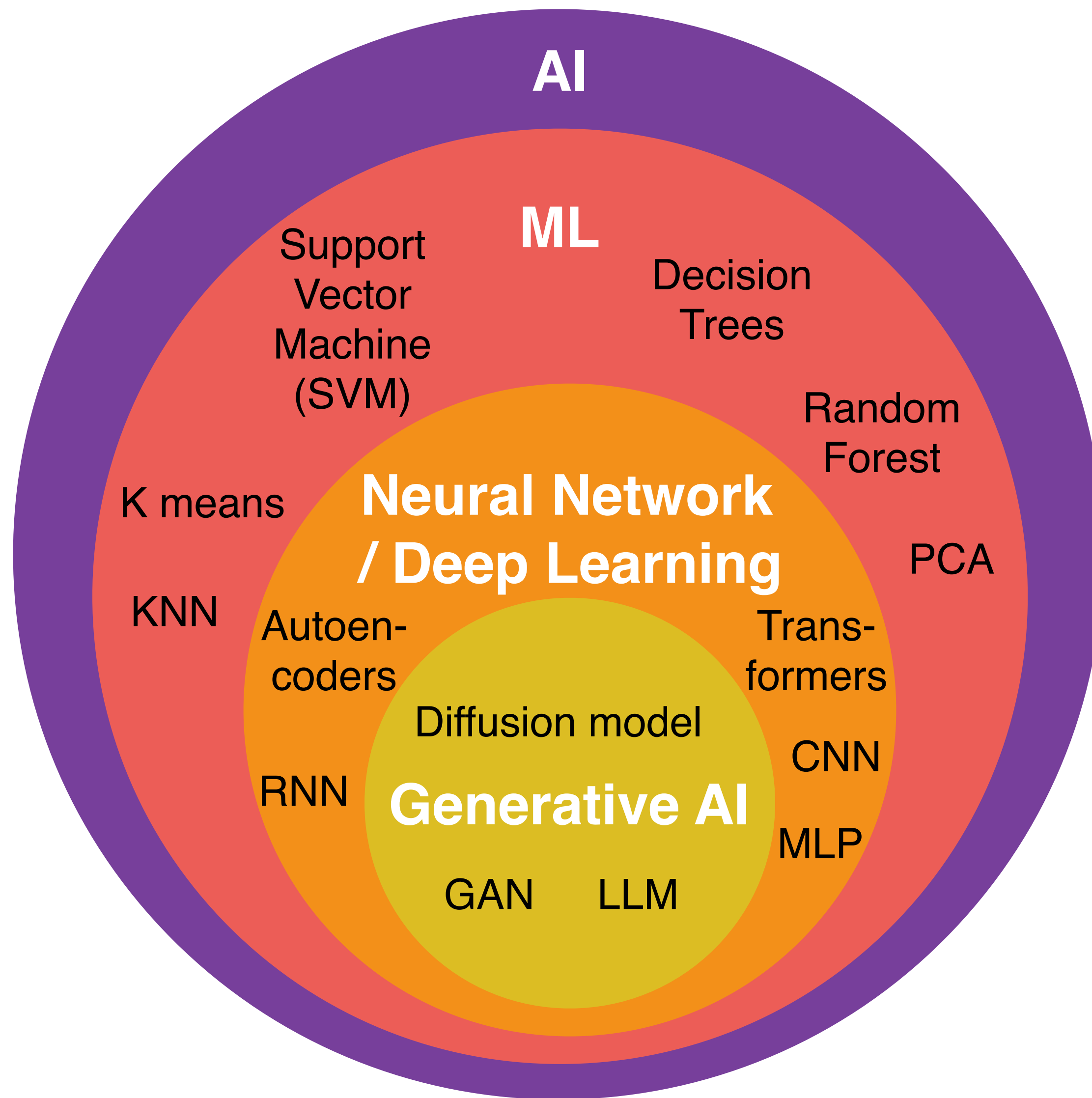
Yeonju Go

Brookhaven National Laboratory / Stony Brook University

*Lehigh University
Jul. 20-23, 2026*



Artificial Intelligence and Machine Learning



Artificial Intelligence (AI)

- **mimic** cognitive functions associated with **human intelligence**
e.g. see, understand, respond, analyze data, make recommendations, etc

Machine Learning (ML)

- **extract knowledge from data** and learn from it **autonomously**
- **no explicit programming**

Neural Network / Deep Learning

- inspired by human brain that uses “**neurons**”
- learn based on deep neural network

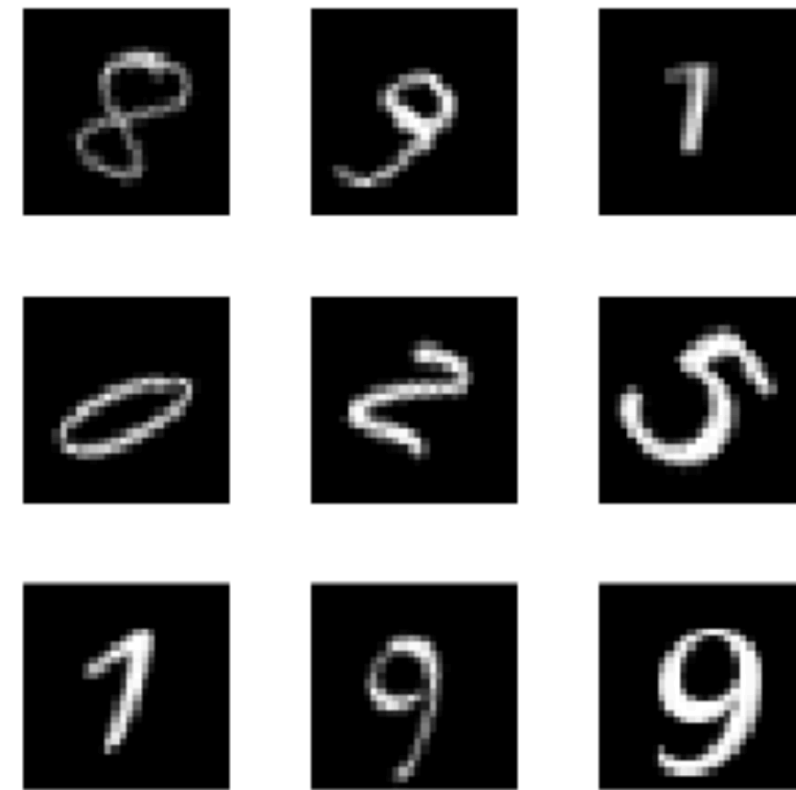
Generative AI

- **creates new content**, such as text, images, and audio, by learning patterns from existing data

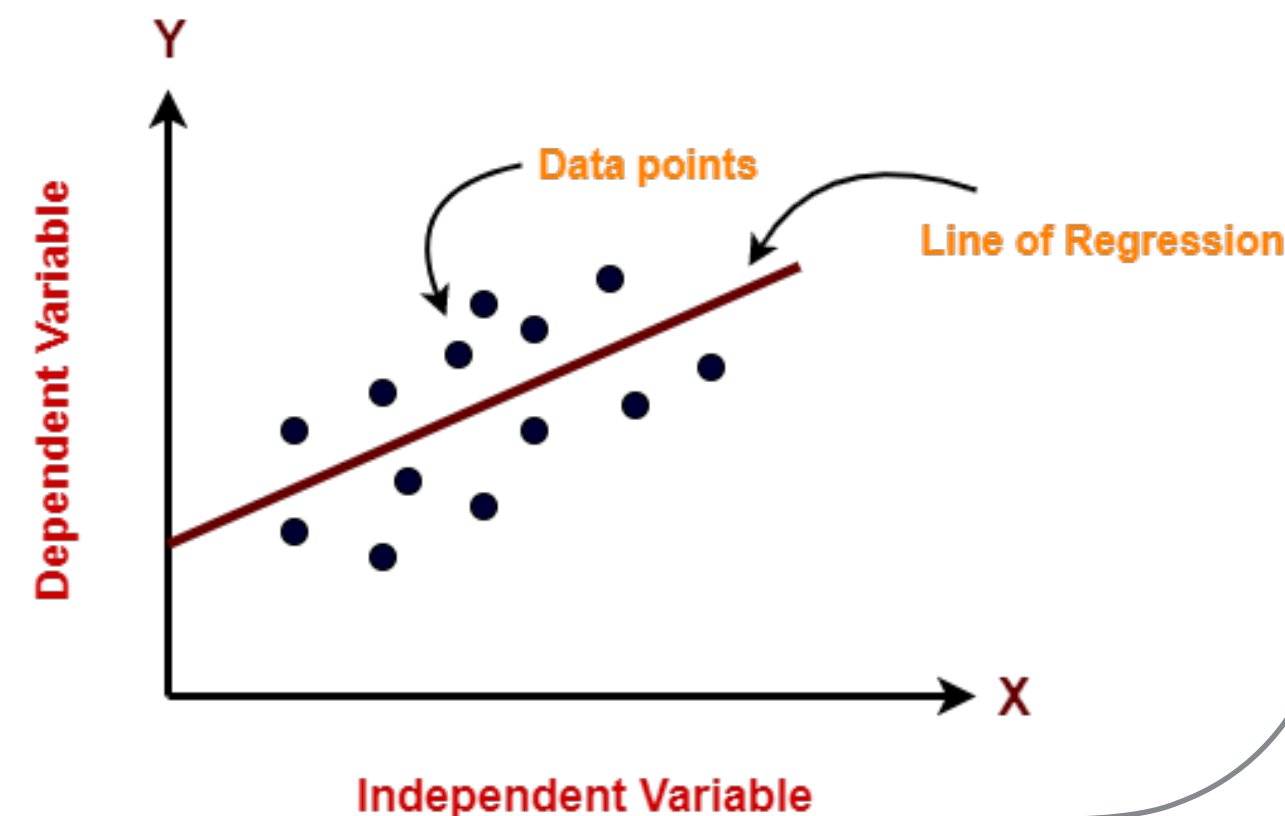
Types of Machine Learning

Supervised

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

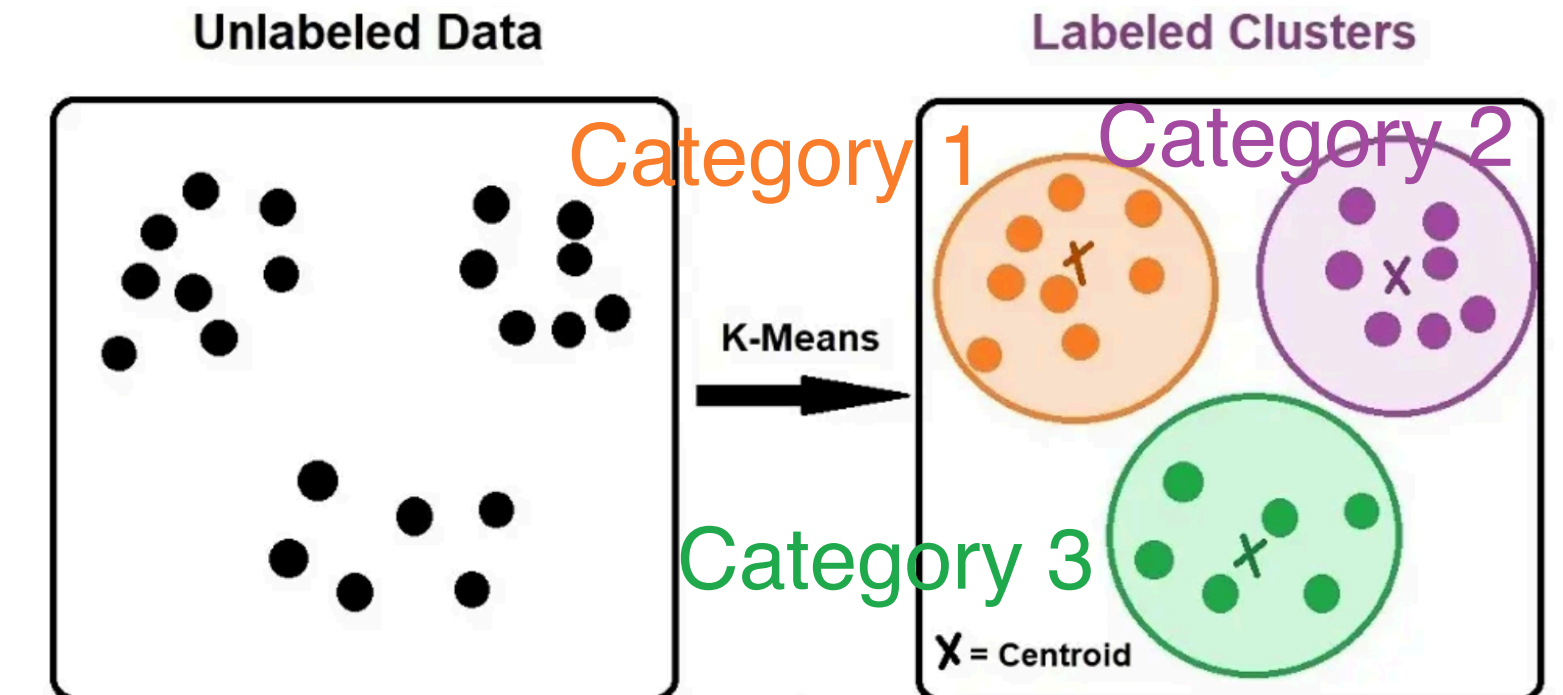


- Regression



Unsupervised

- Use **unlabelled** data
- Find **hidden patterns** w/o human intervention
- Clustering
- Association
- Dimension Reduction
- Anomaly detection



Various Neural Network Architecture

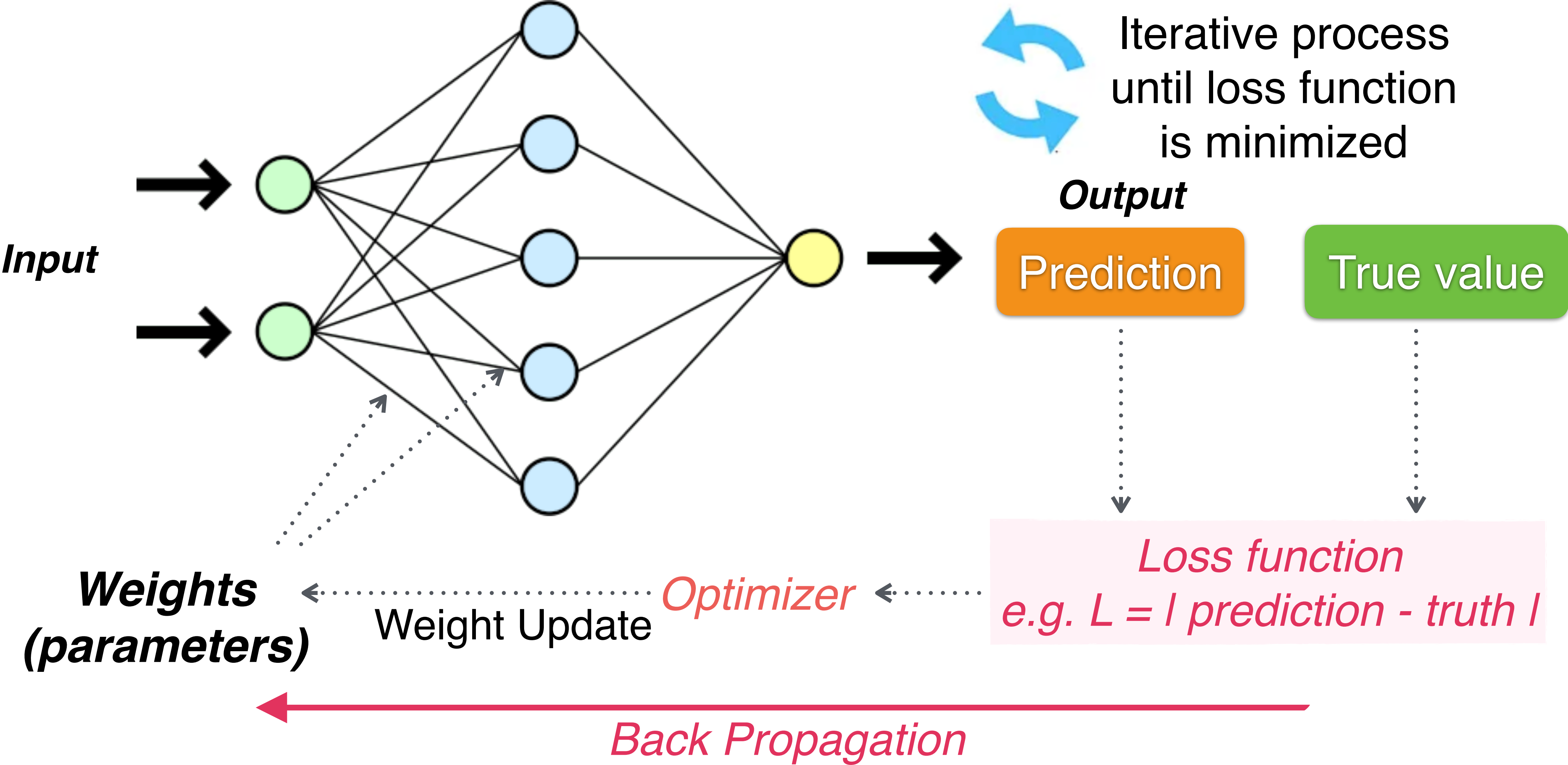
The different structure (architectures) based on how they organize, connect, and process inputs before or between MLPs

- **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
 - ➔ ...

Neural Network



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

What Controls the Performance/Fidelity of AI?

Defining the physics problem and ML task

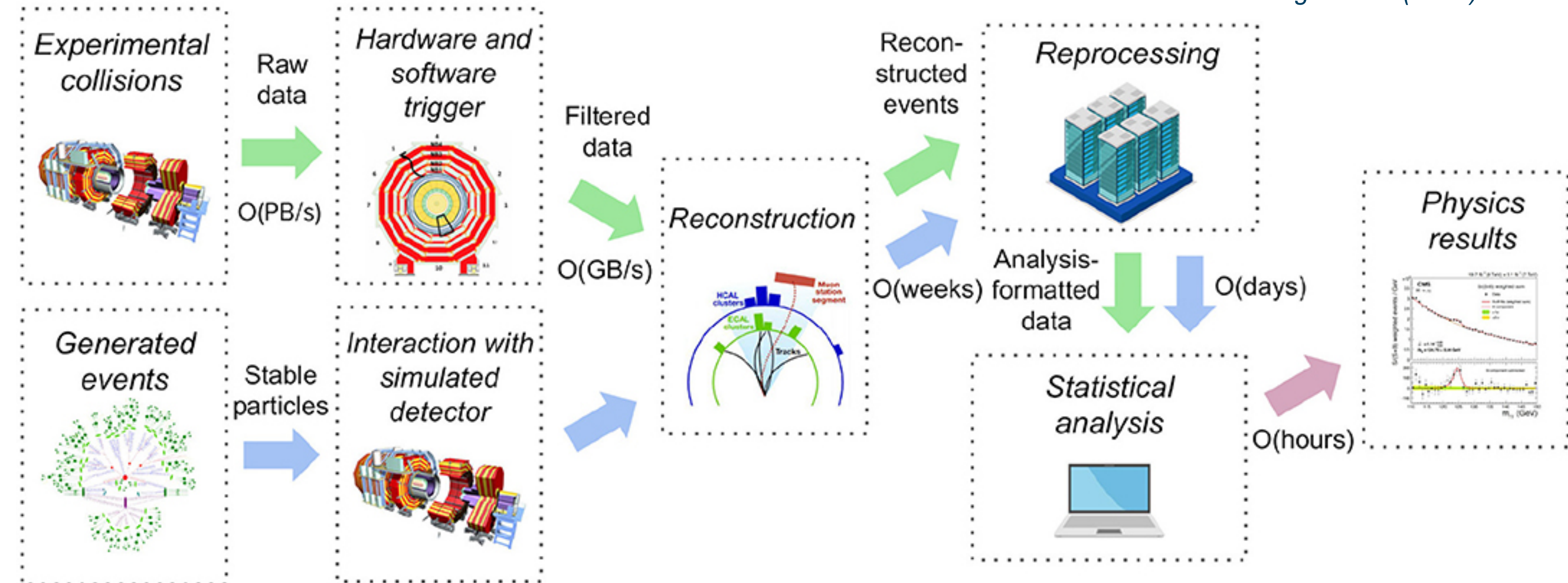
Choosing the model architecture

Curating and preprocessing the data

Defining loss, optimizer, and evaluation metrics

(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**

Fast Simulation:

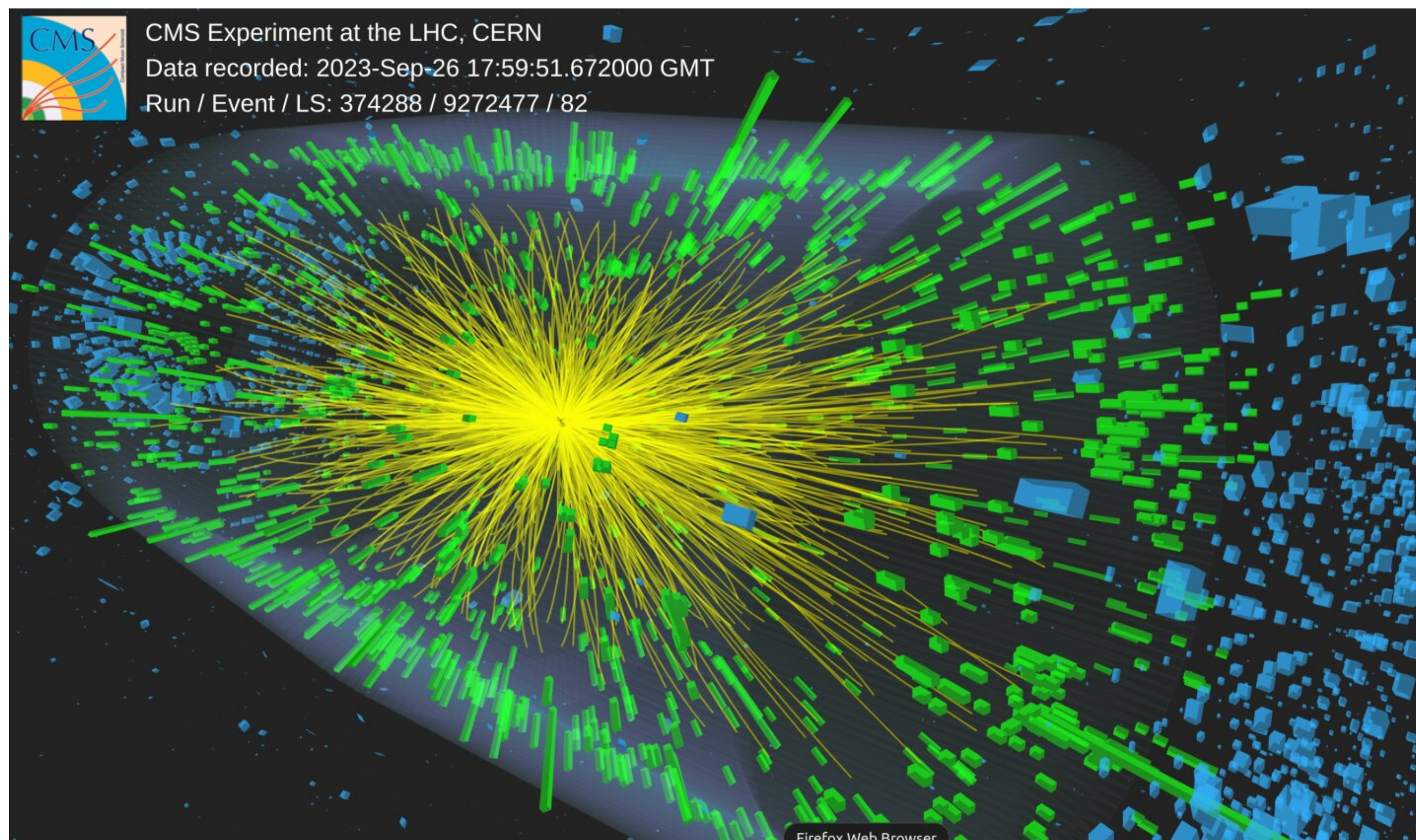
Diffusion Models

Generative Adversarial Network

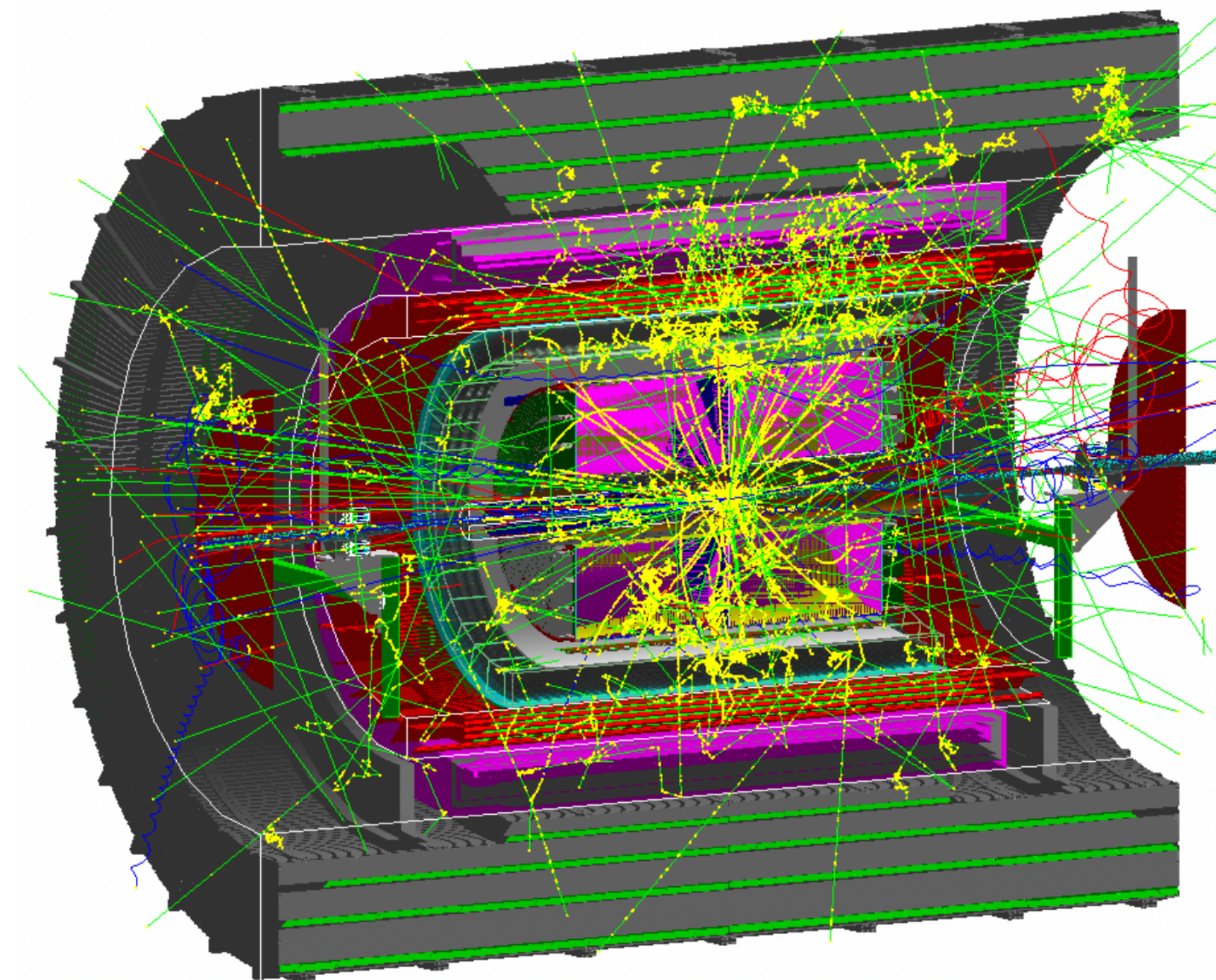
Heavy Ion Collision Simulation

- $O(1000)$ 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

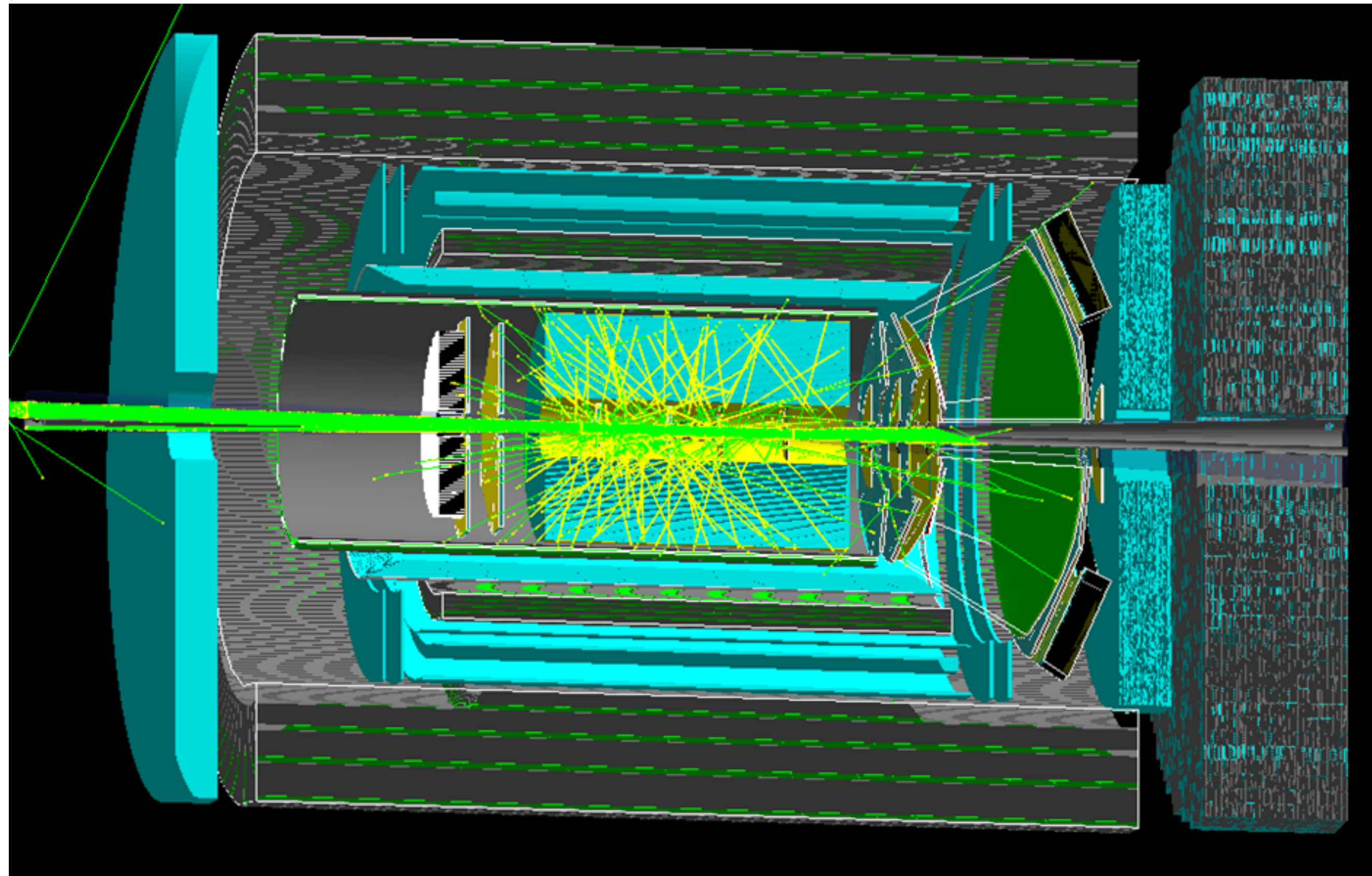


sPHENIX Geant4 detector simulation
for proton-proton collisions
(Au+Au central collisions have x1000 more particles)



EIC Collision Event Simulation

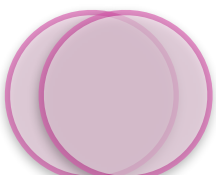
- **Electron-Ion Collider** will need a large amount of simulations of full detector with **both physics and machine background**

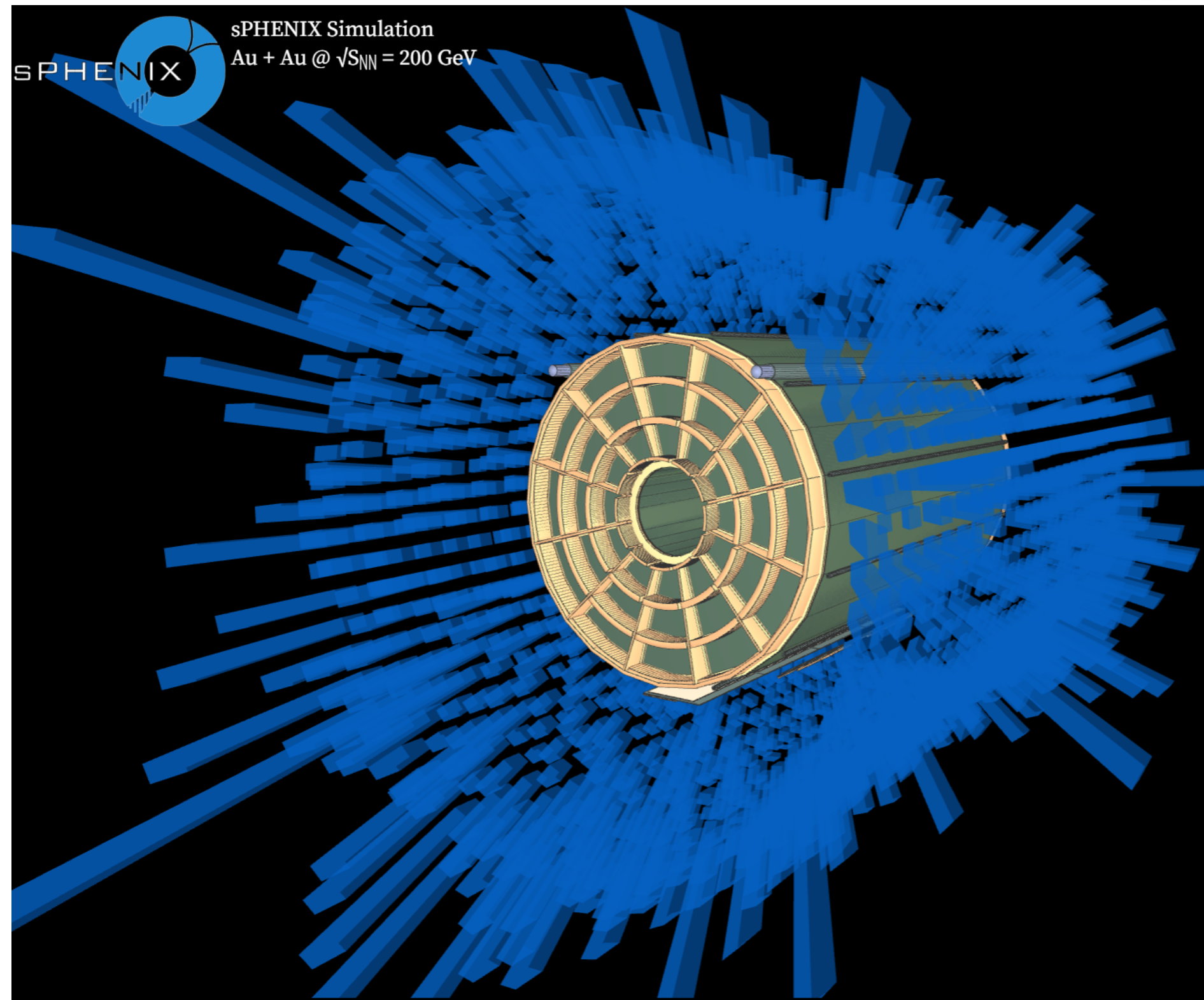


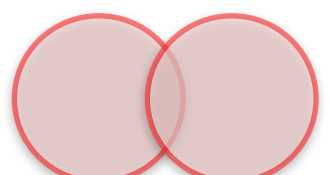
ML can speed up and produce large amount of the heavy ion event simulations!

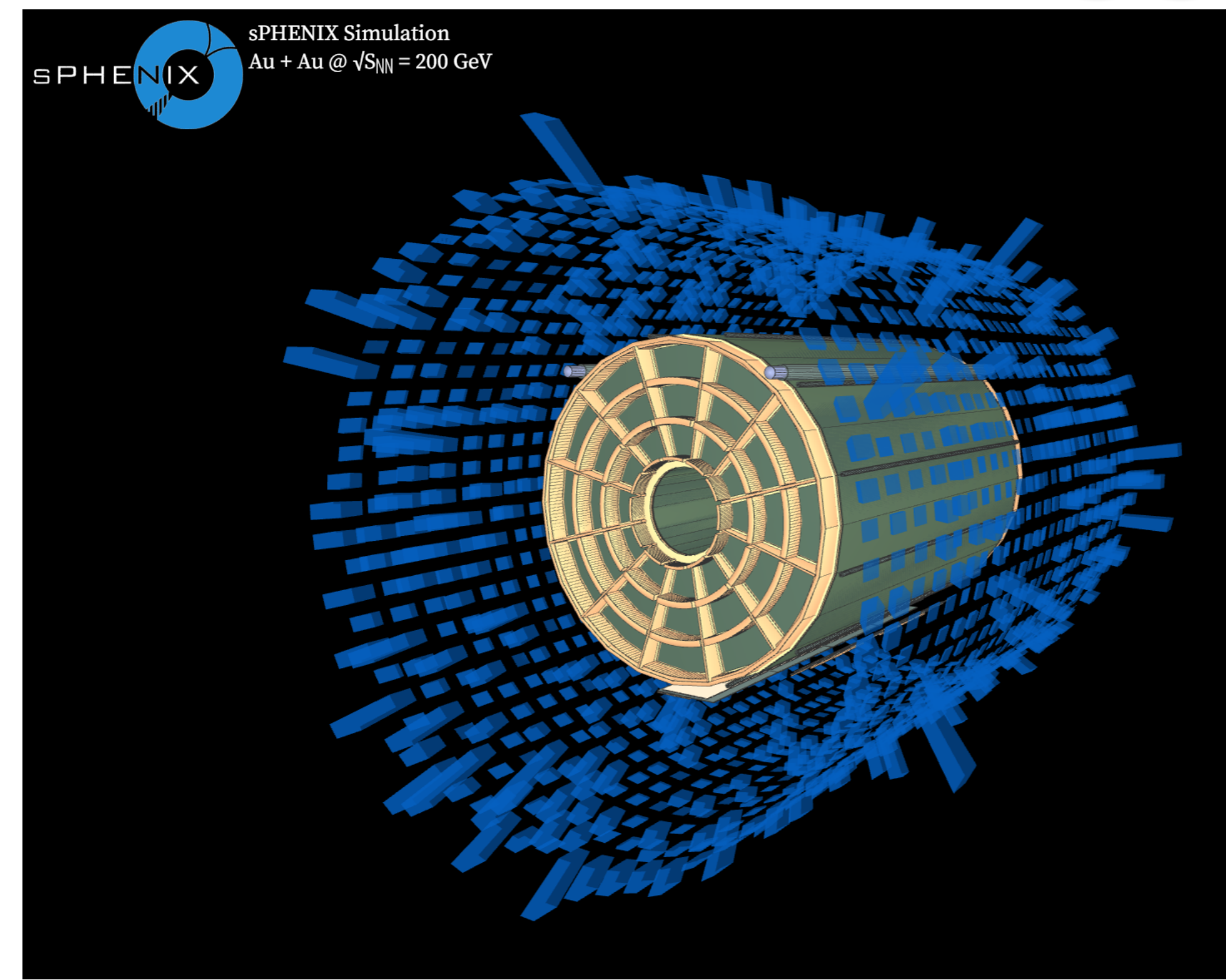
Heavy Ion Collision Event

- **HIJING** Monte Carlo event generator for Au+Au collisions at $\sqrt{s_{NN}}=200$ GeV
- **Geant4** full detector simulation with the sPHENIX geometry

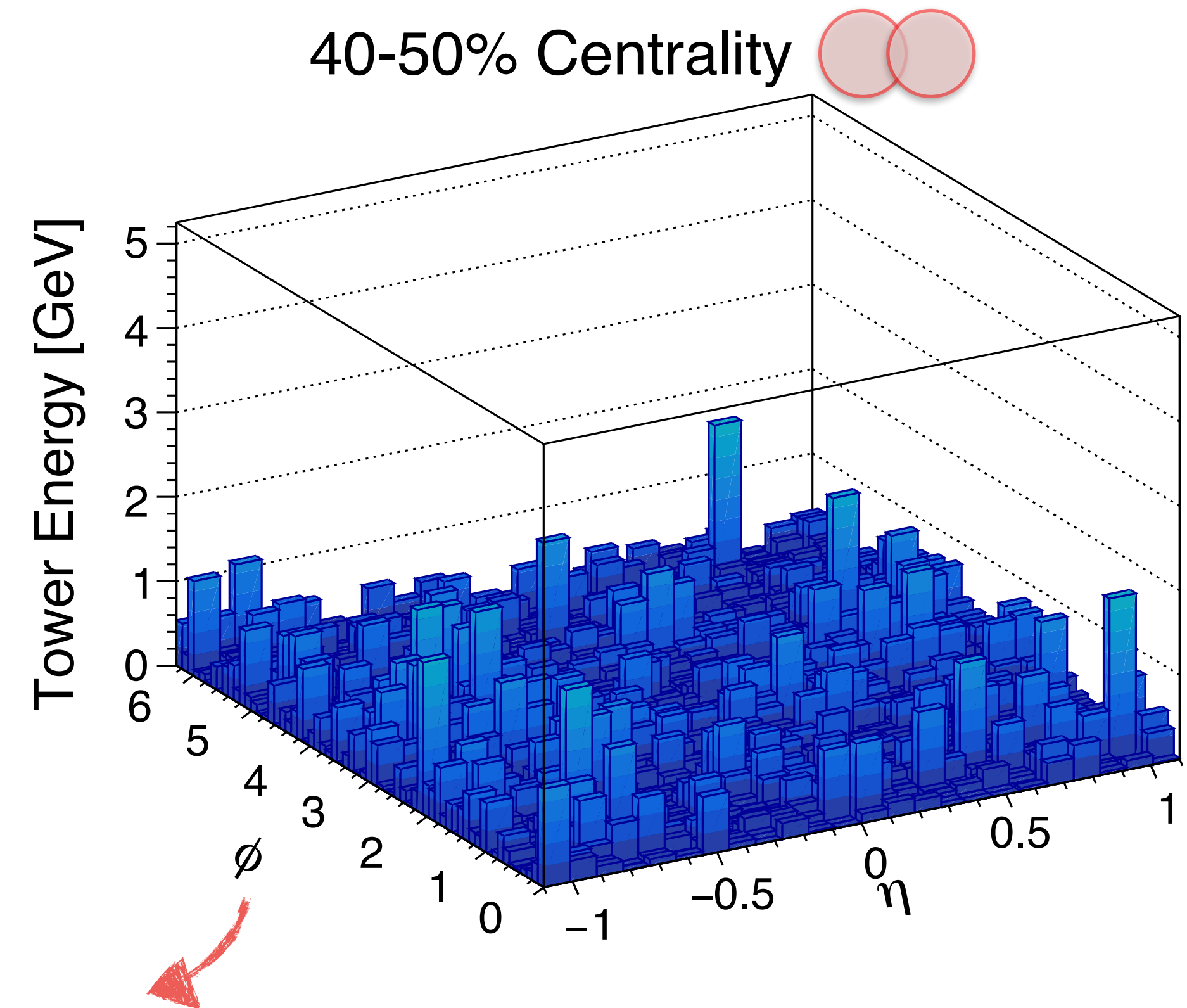
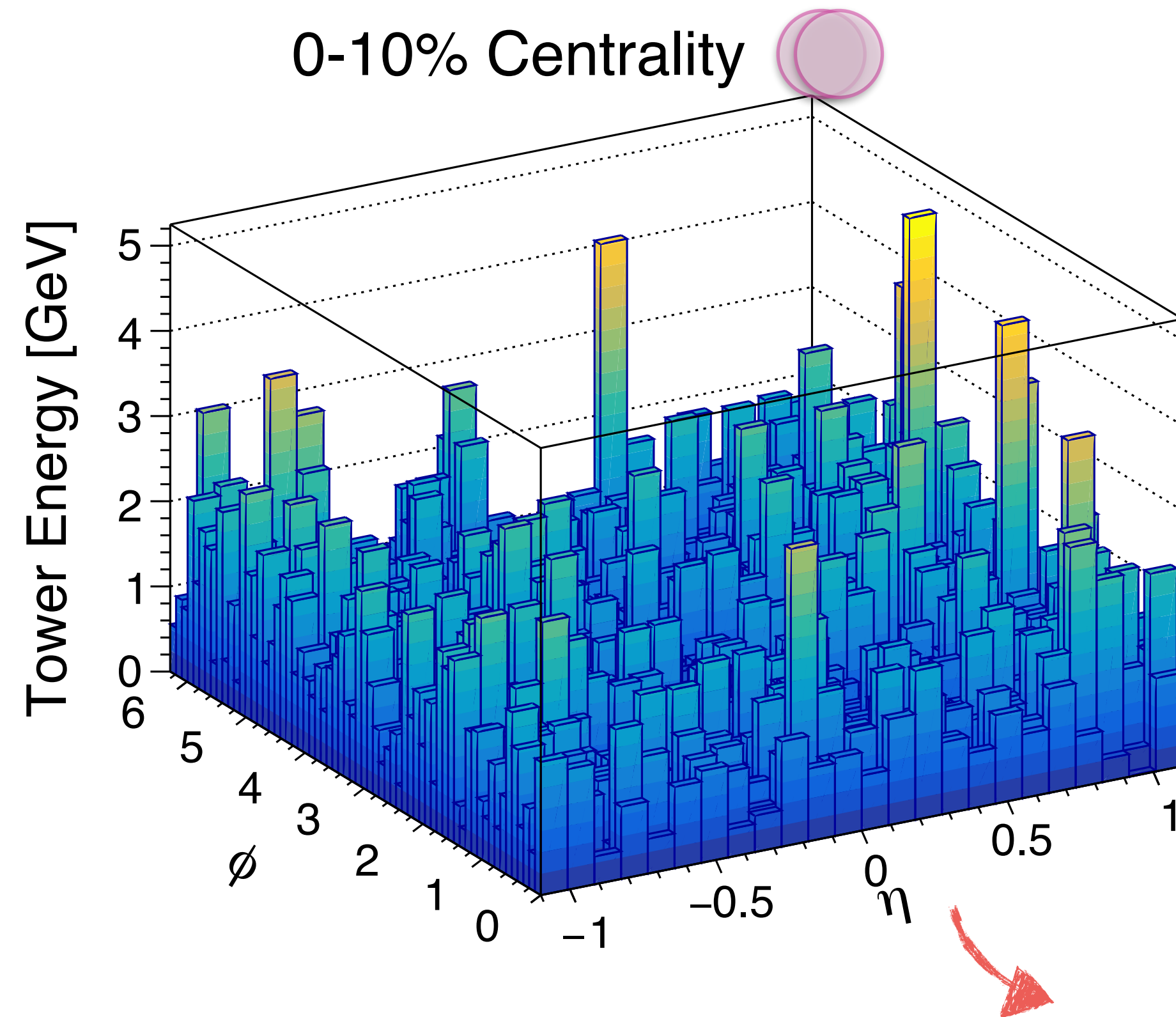
Head-on collision (0-10% Centrality) 



Side collision (40-50% Centrality) 

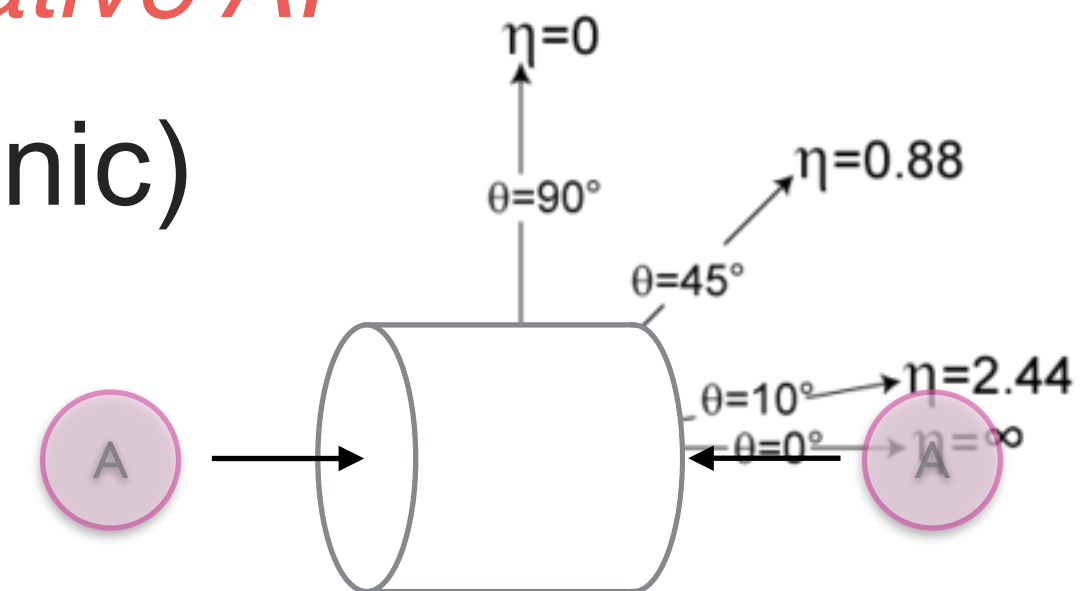


Heavy Ion Collision Calorimeter Energy Distribution



Let's generate these kind of event distribution (image) using generative AI

- Full calorimeter **towers** (Electromagnetic + Inner hadronic + Outer hadronic)
 - ➔ $-1.1 < \eta < 1.1$, $0 < \phi < 2\pi$
 - ➔ (24 x 64) bins in (η, ϕ)



Generative Adversarial Network (GAN)

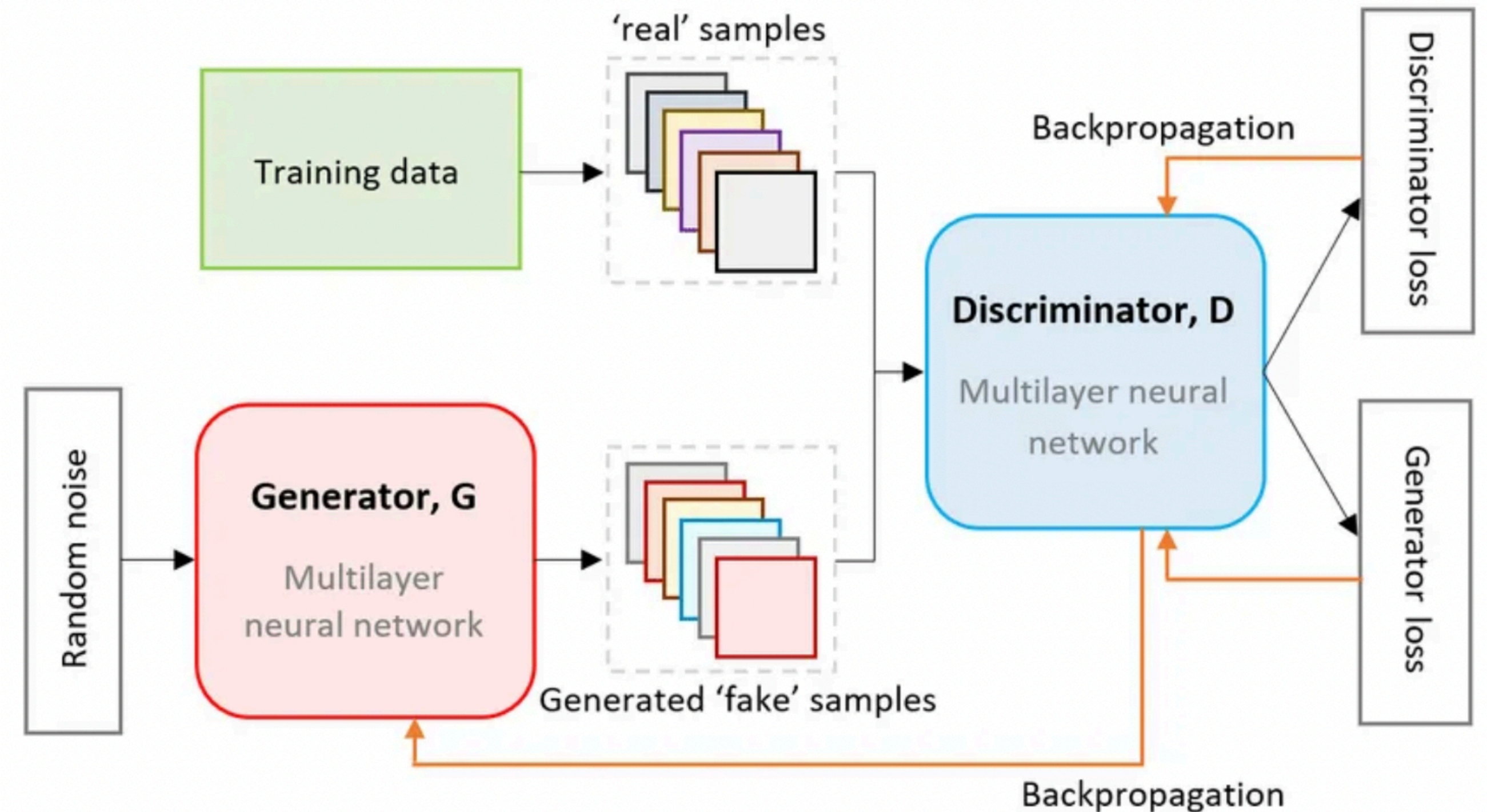
arXiv:1406.2661v1 [stat.ML] 10 Jun 2014

Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie*, Mehdi Mirza, Bing Xu, David Warde-Farley,
Sherjil Ozair[†], Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G . The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D , a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.



- 2 neural networks
 - ➡ Generator
 - ➡ Discriminator

Denoising Diffusion Probabilistic Model (DDPM)

arXiv:2006.11239v2 [cs.LG] 16 Dec 2020

Denoising Diffusion Probabilistic Models

Jonathan Ho

UC Berkeley

jonathanho@berkeley.edu

Ajay Jain

UC Berkeley

ajayj@berkeley.edu

Pieter Abbeel

UC Berkeley

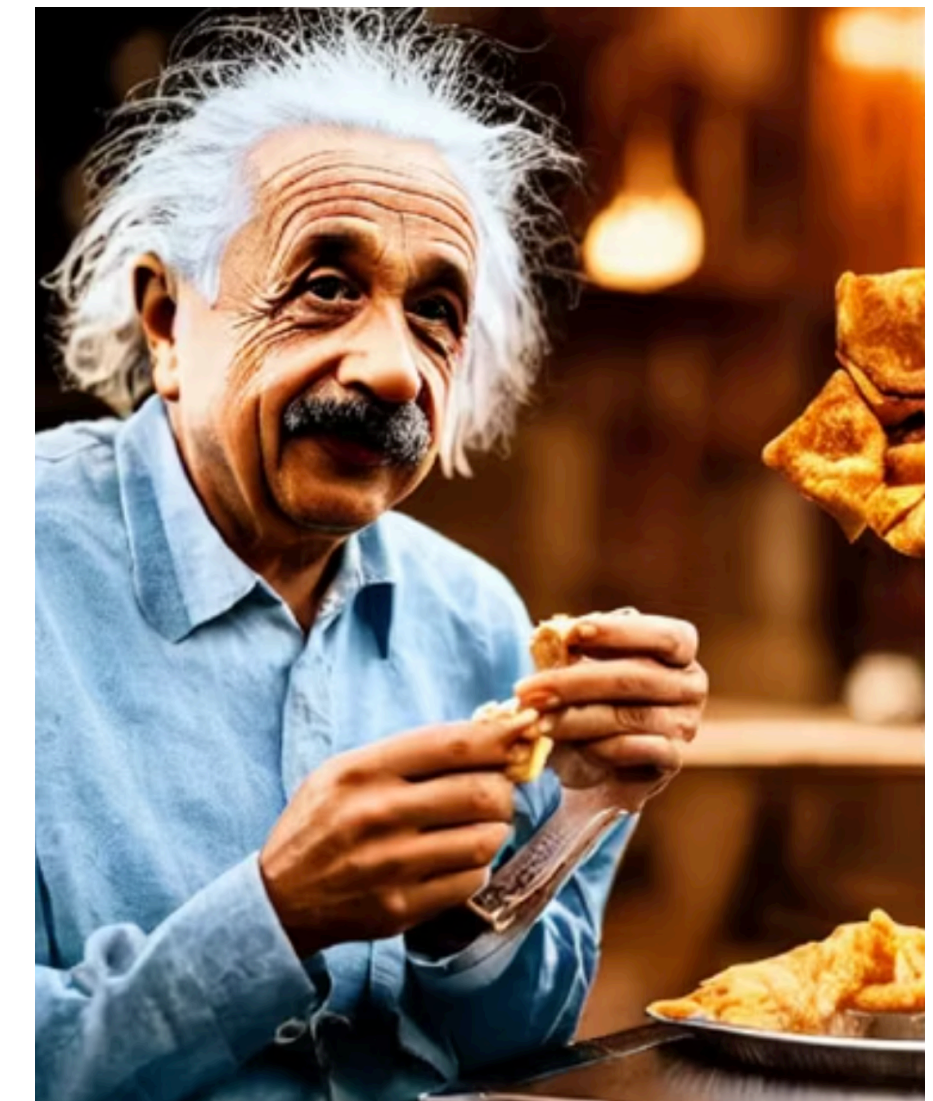
pabbeel@cs.berkeley.edu

Abstract

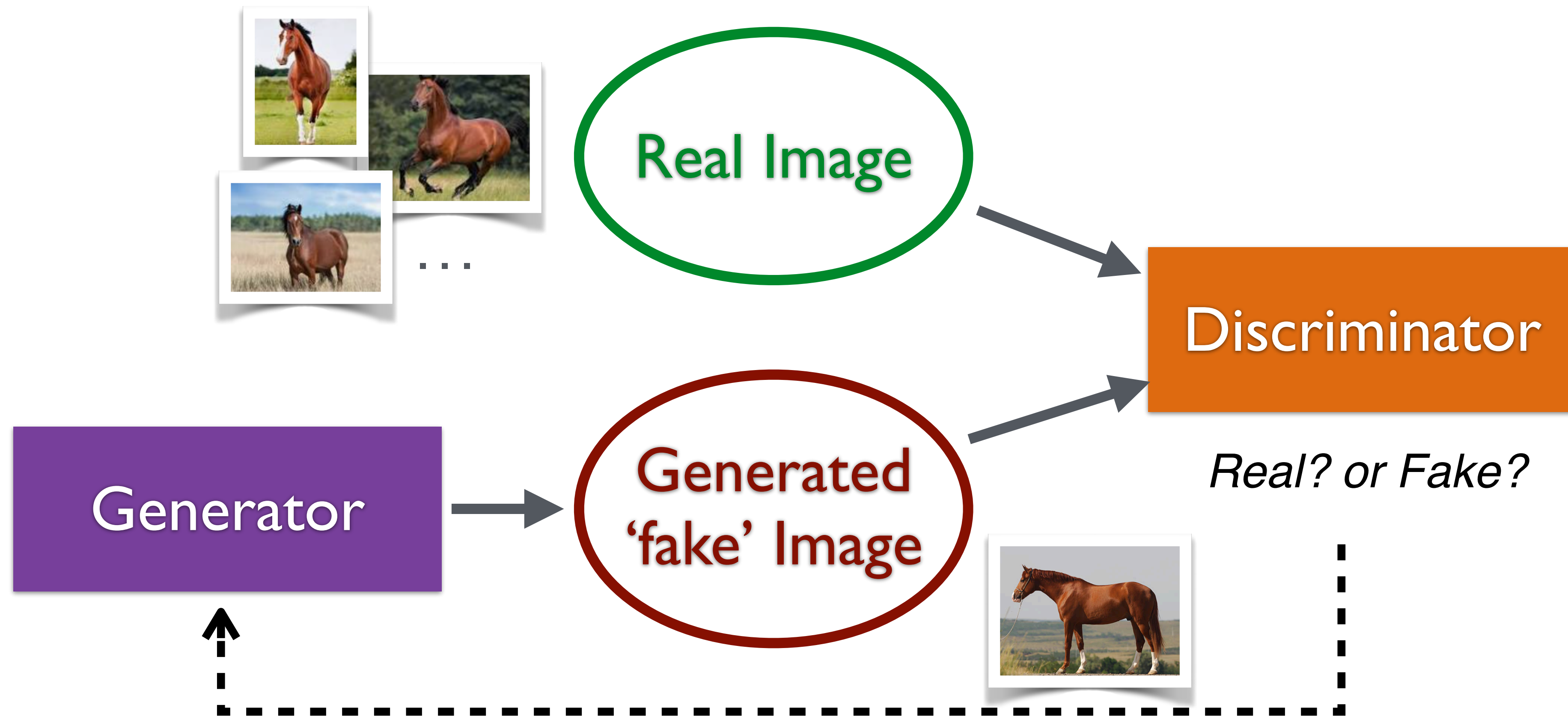
We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decomposition scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at <https://github.com/hojonathanho/diffusion>.

- **Diffusion Models**

- text-to-image, text-to-video generation (e.g. StableDiffusion, Midjourney, Dalle-3, Sora)
- very popular in industry



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

GAN Loss

- There are many ways to construct Loss for GAN, one way is

Discriminator Loss

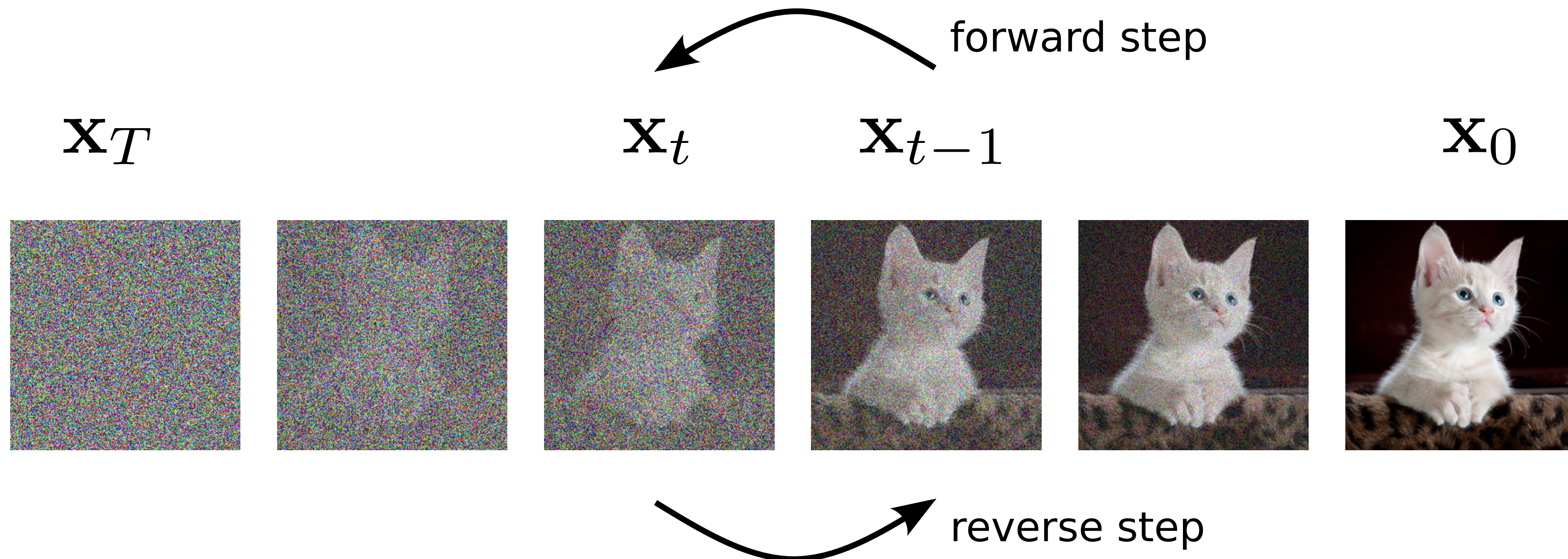
$$L_D = \langle \max(0, 1 - D(x_{\text{real}})) \rangle + \langle \max(0, 1 + D(x_{\text{fake}})) \rangle$$

Generator Loss

$$L_{G,\text{adv}} = - \langle D(x_{\text{fake}}) \rangle$$

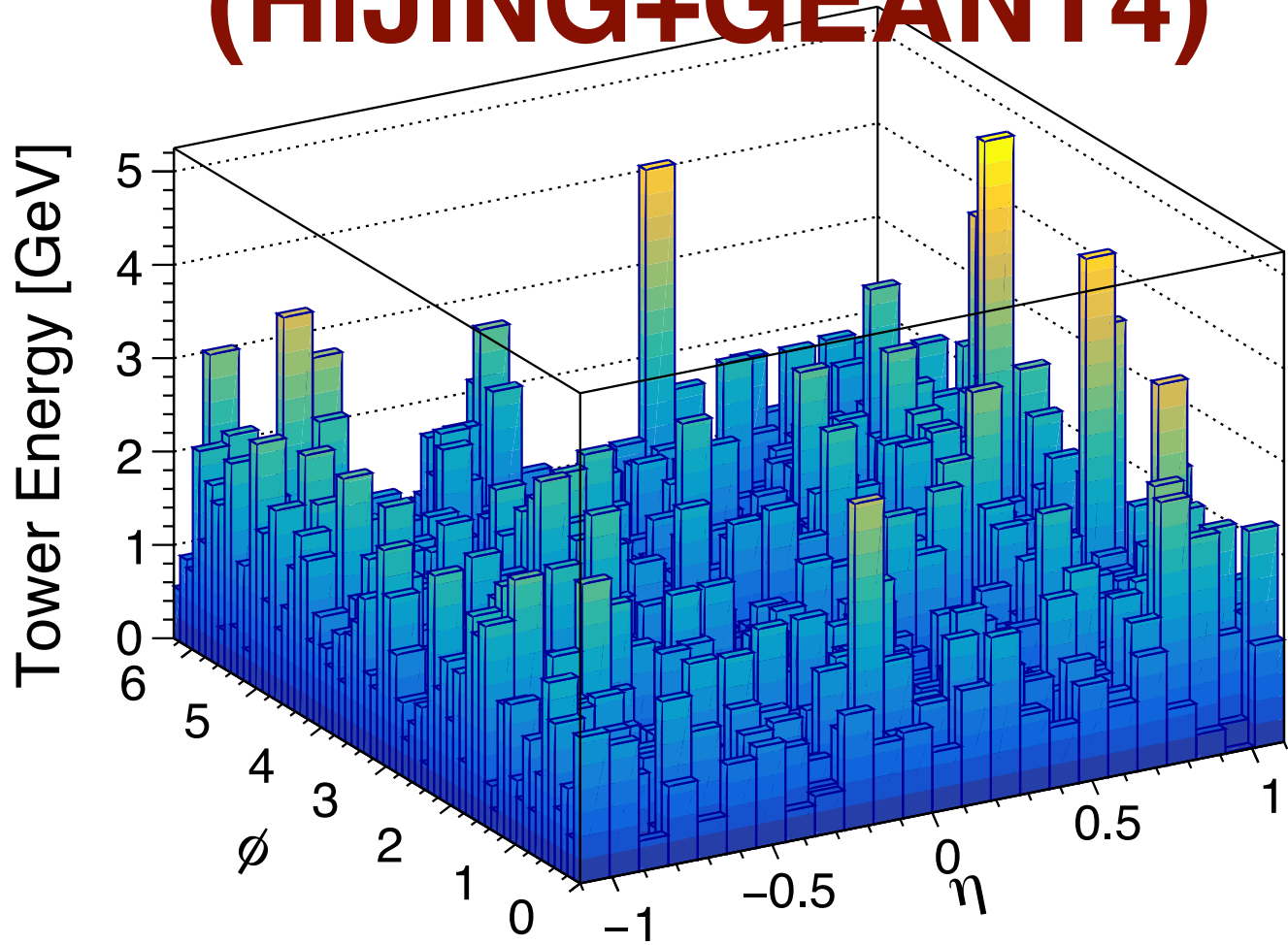
Denoising Diffusion Probabilistic Model (DDPM)

- **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

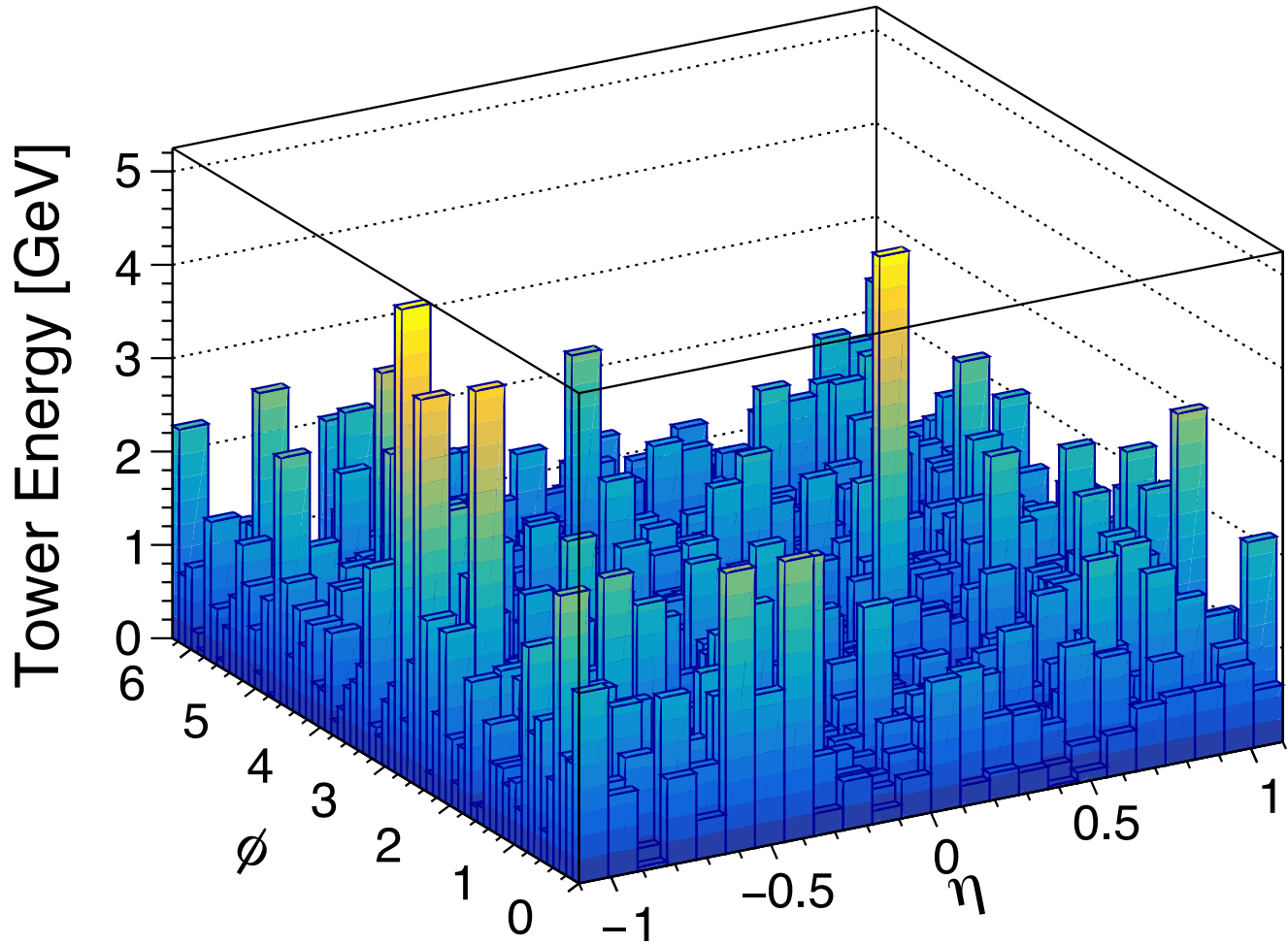


Display of AI-Generated Events

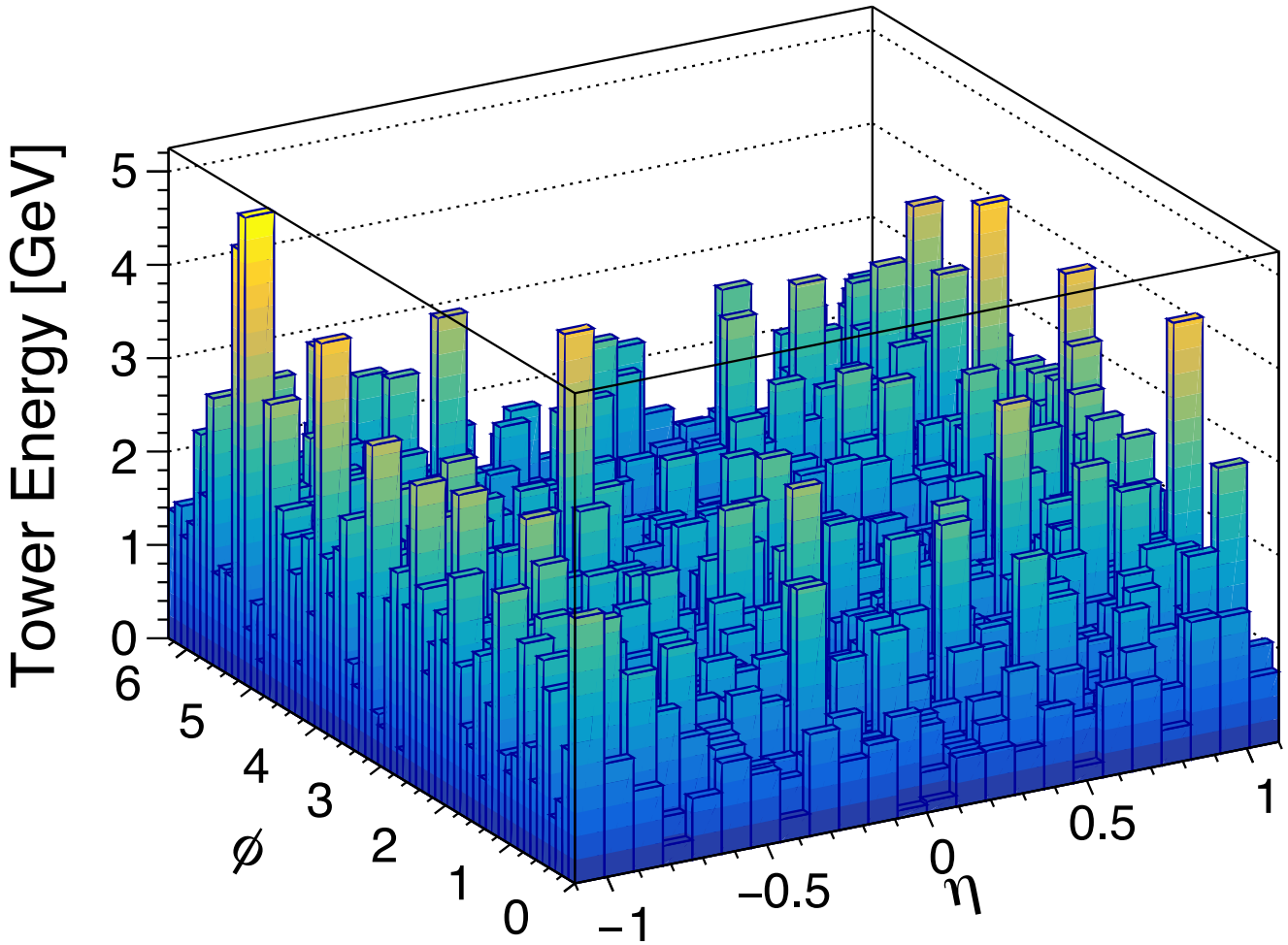
Training sample
(HIJING+GEANT4)



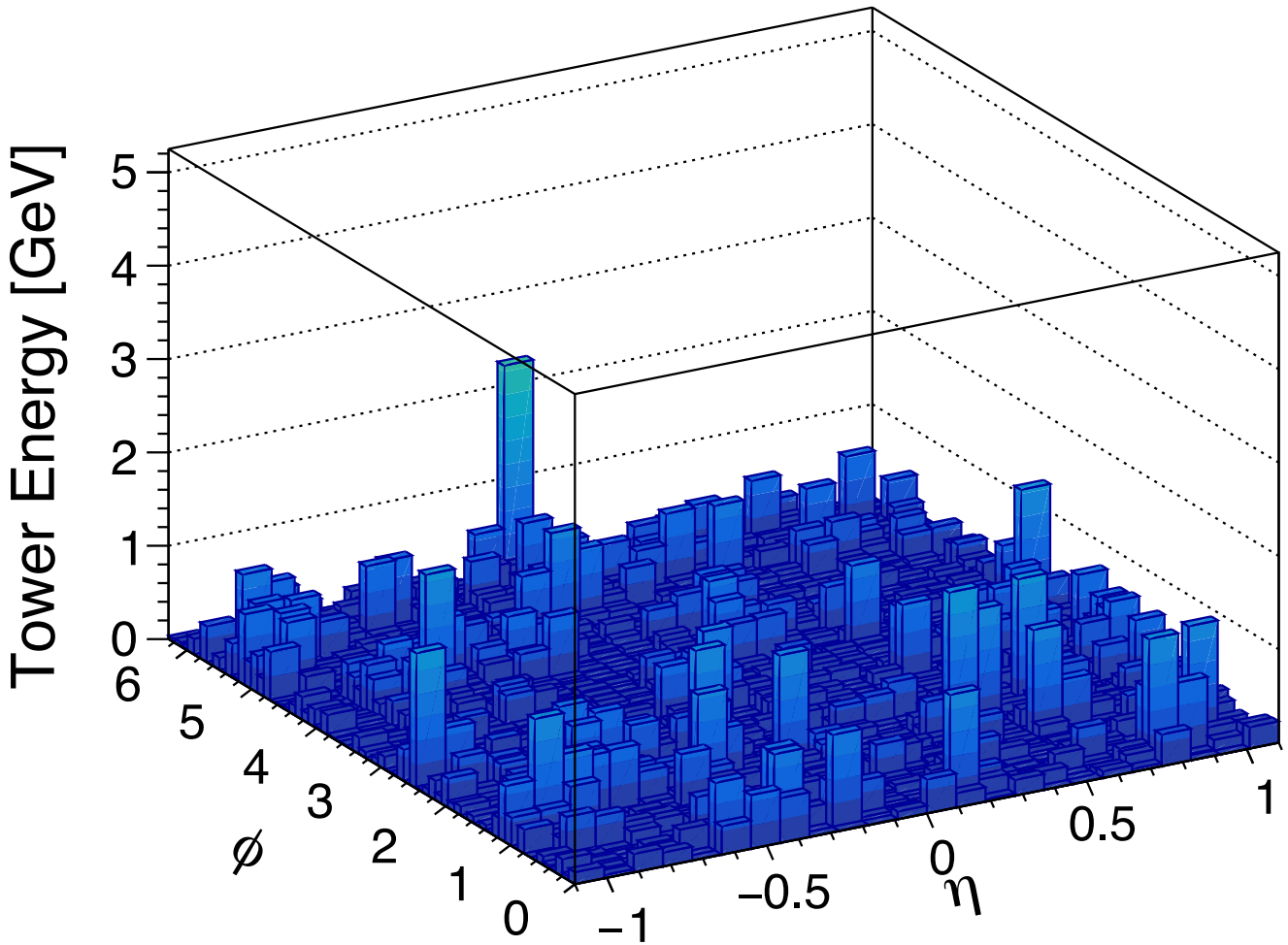
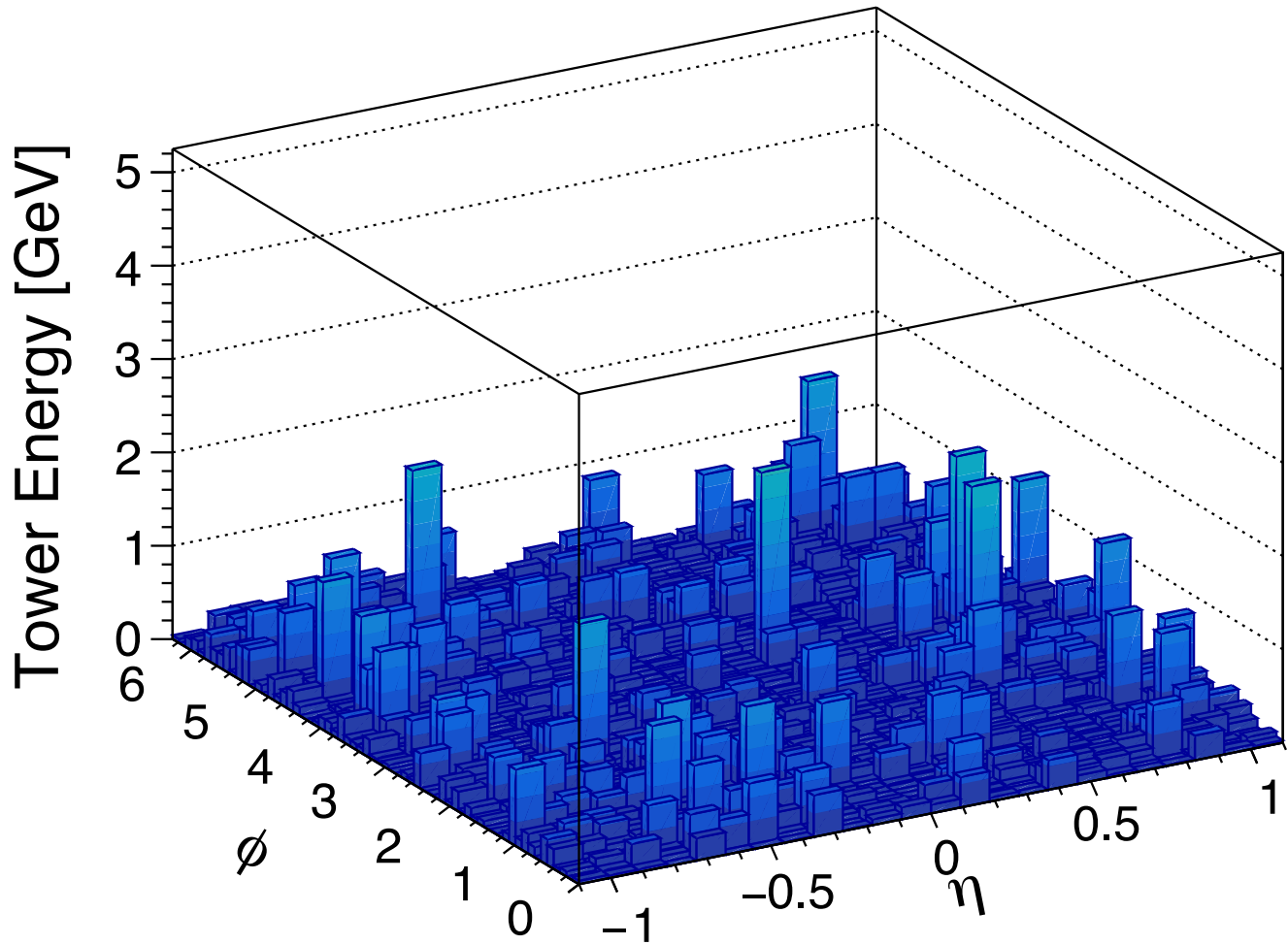
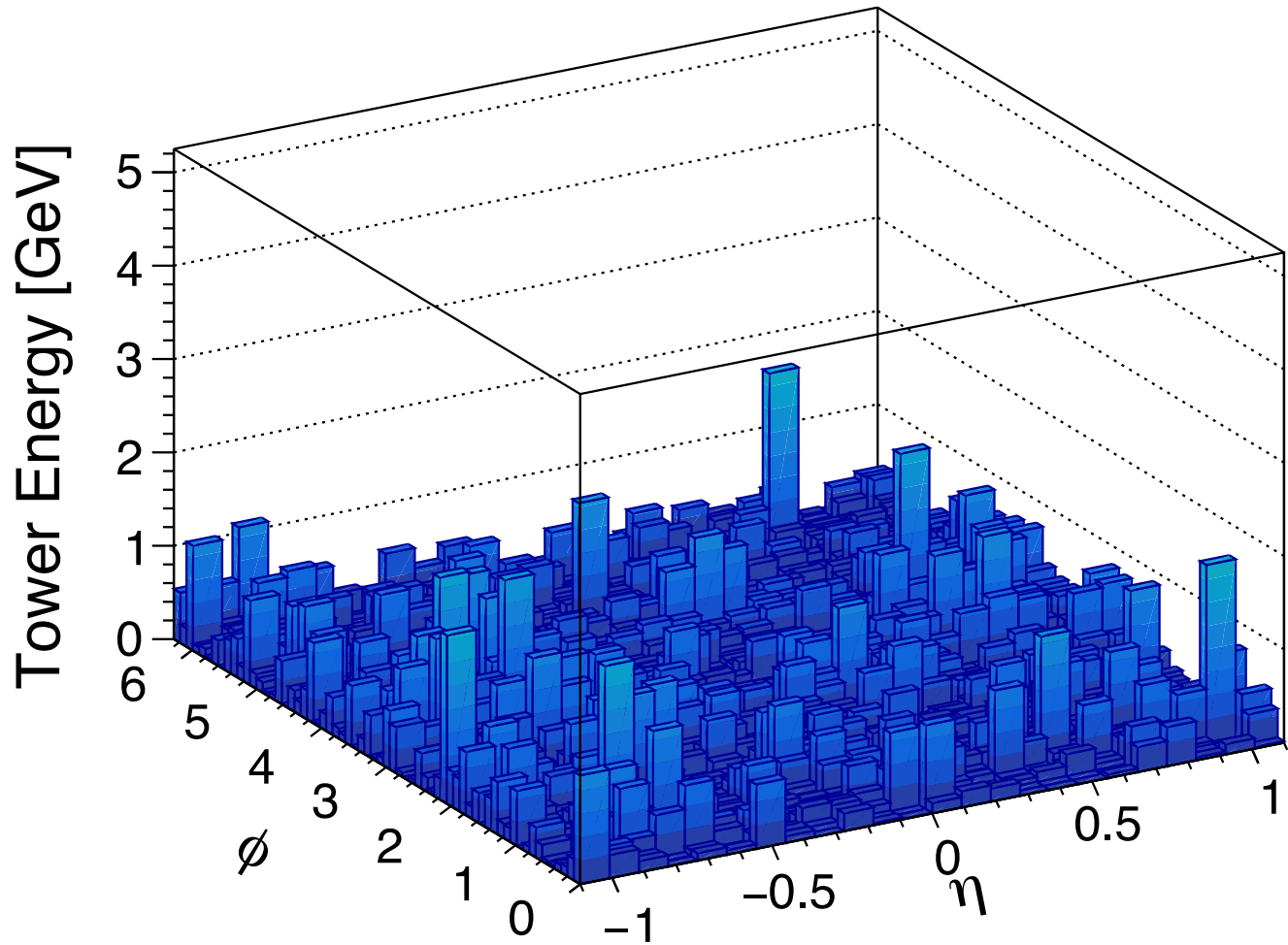
Generated (DDPM)



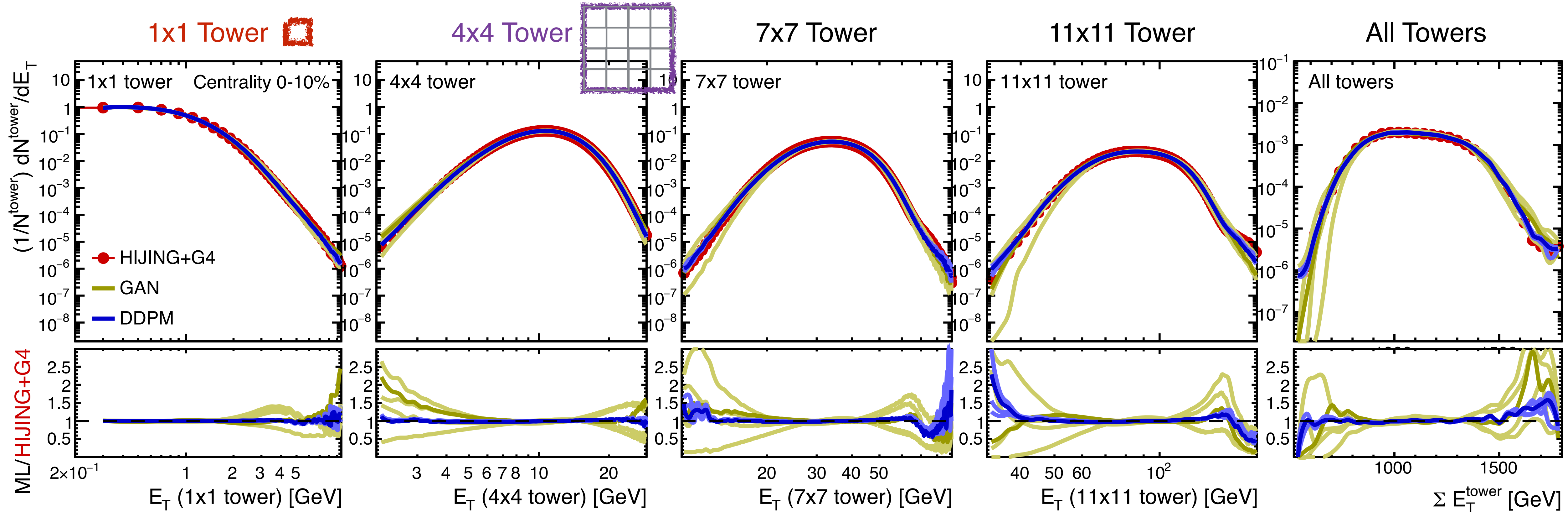
Generated (GAN)



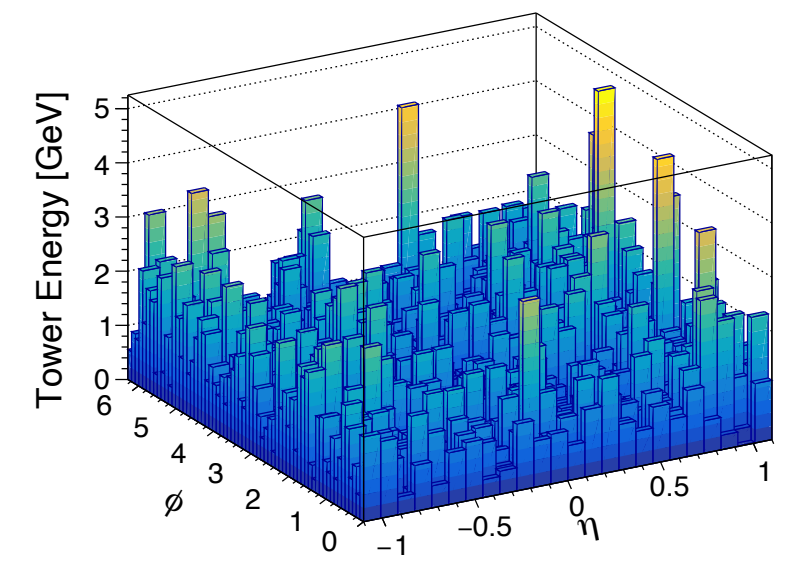
40-50%
Centrality



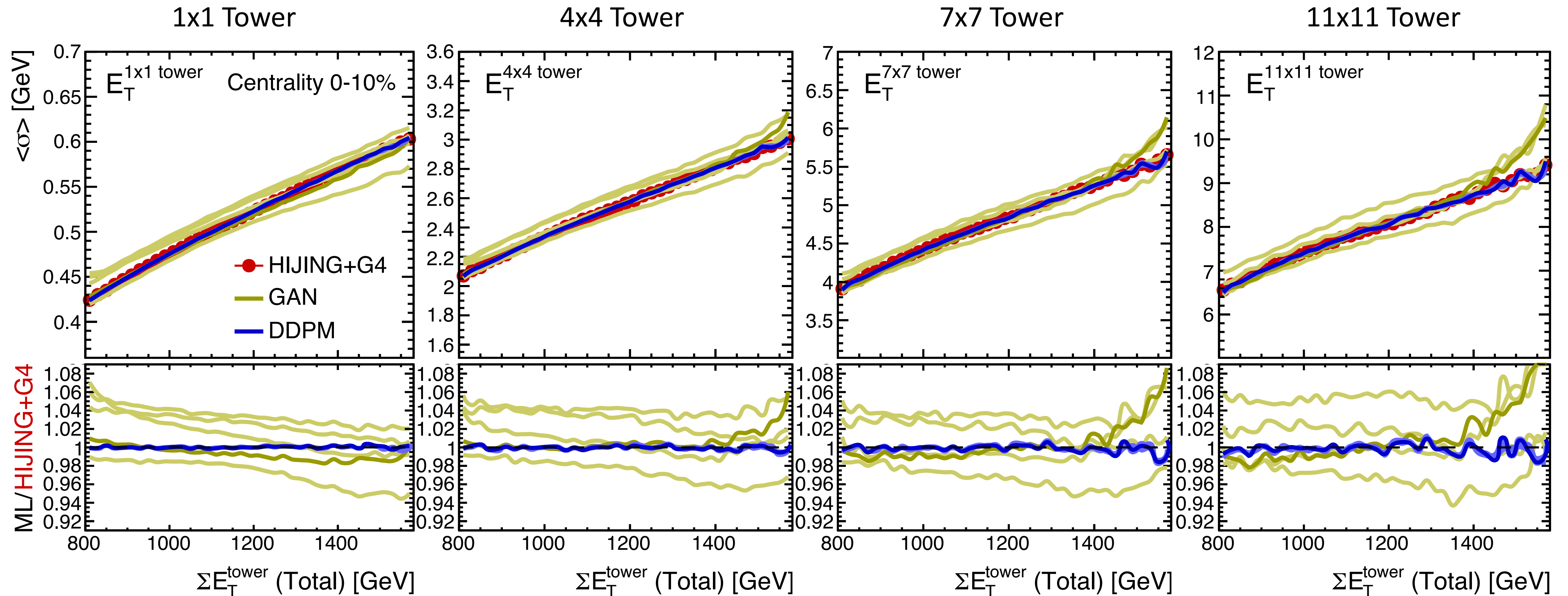
Performance: Transverse Energy (0-10%)



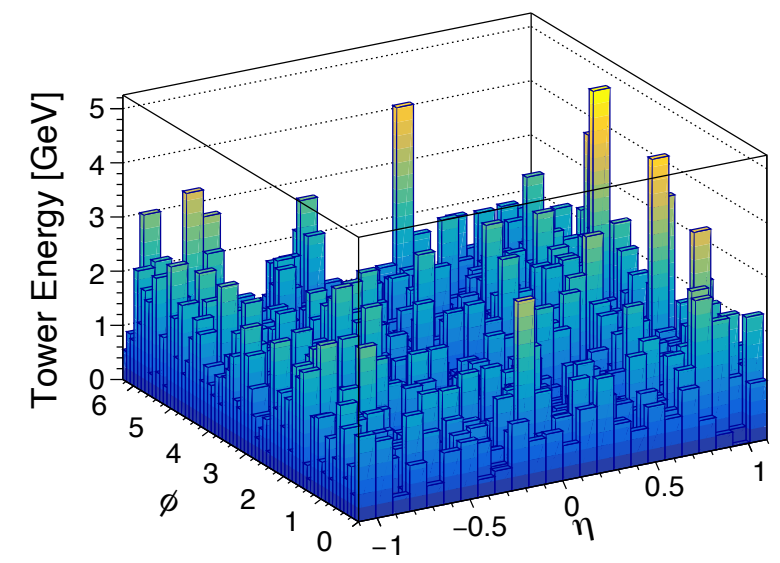
- Each model is retrained 5 times with different random seeds
- **HIJING+Geant4** used as training data (600k events) and testing data (100k events)
- Both **DDPM** and **GAN** reproduce the data distribution where the data are abundant
- **DDPM** outperforms **GAN** in overall distribution w/ great stability and accuracy



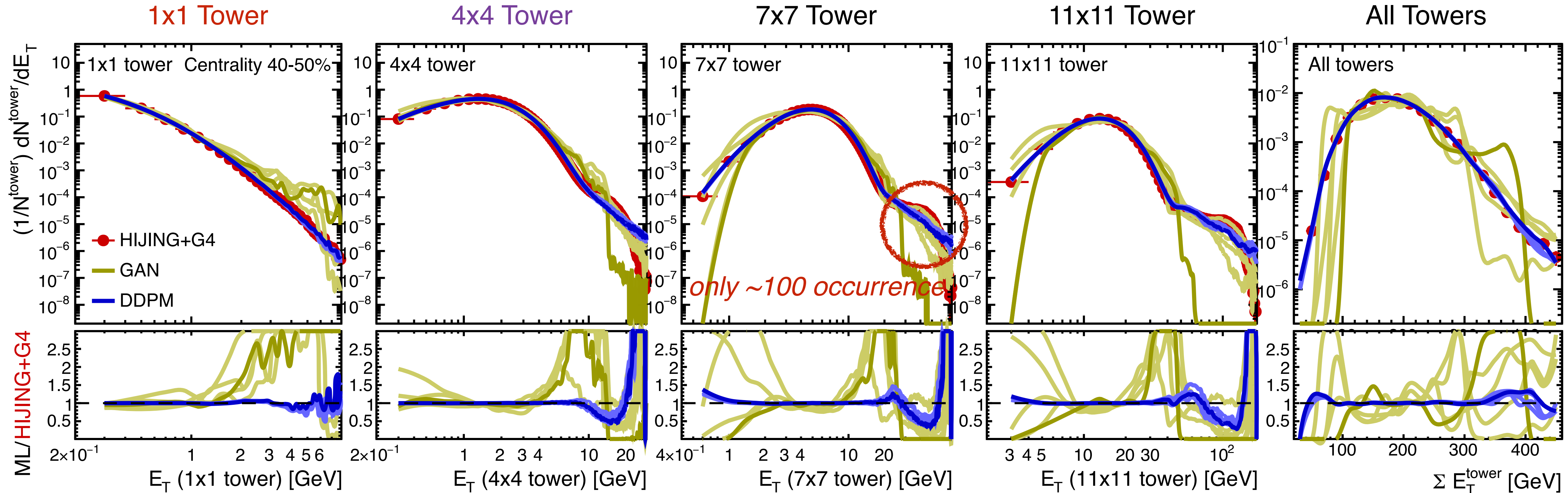
Performance: Transverse Energy Fluctuation (0-10%)



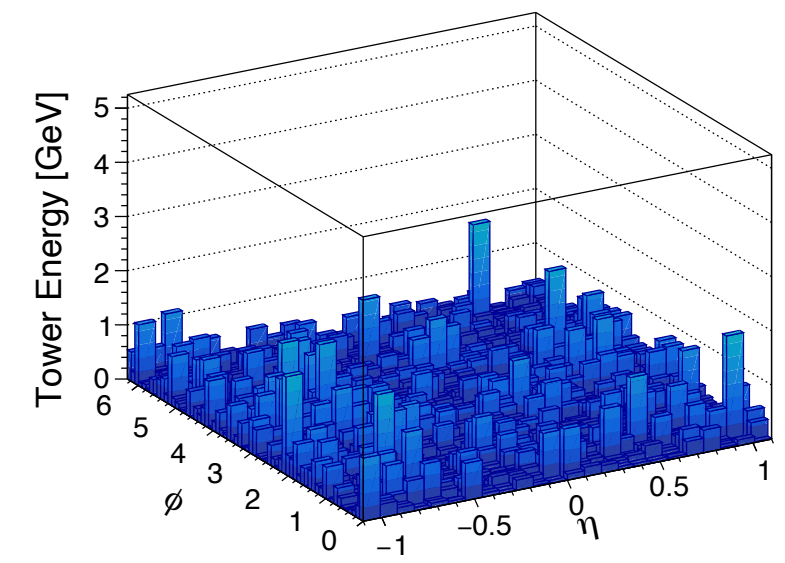
- **GAN** fails to describe fluctuation
- **DDPM** outperforms **GAN** w/ great stability, a few percent-level accuracy



Performance: Transverse Energy (40-50%)



- DDPM outperforms GAN
 - ➔ great stability, good agreement with HIJING+G4 at high probability region
- Non-gaussian rare tail at the high energy region → challenge to reproduce



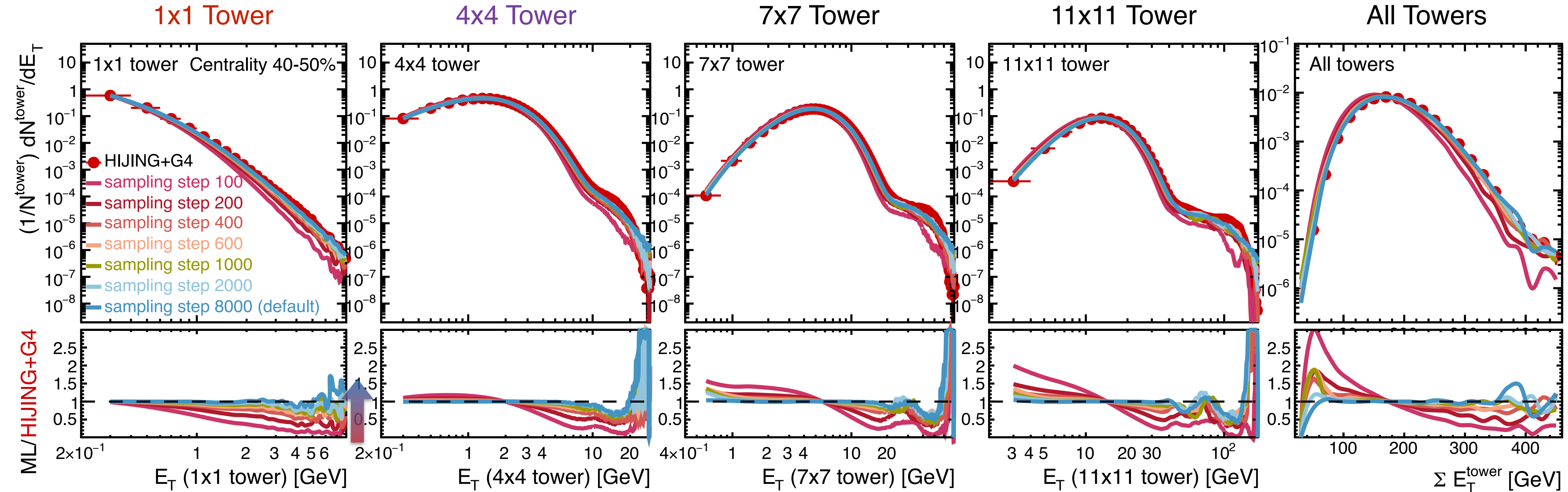
How long does it take to simulate a large sample?

	Generating time	Speedup	CPU/GPU
HIJING + GEANT4 (Conventional)	40 minutes / event	1	Single CPU
DDPM	1.34 s / event	~1,800X	NVIDIA RTX A6000
GAN	0.42 ms / event	~5,700,000X	NVIDIA RTX A6000

PRC 110 (2024) 034912

- **GAN** is faster, but the **DDPM** exhibits high fidelity in describing the truth ground (**HIJING+GEANT4**)
- **DDPM** provide a speedup of **O(100)**, considering a 32-core CPU equivalent to a GPU
- One can train the model using a relatively modest number (at the level of millions) and then accelerate the production of much larger samples (at the level of billions)

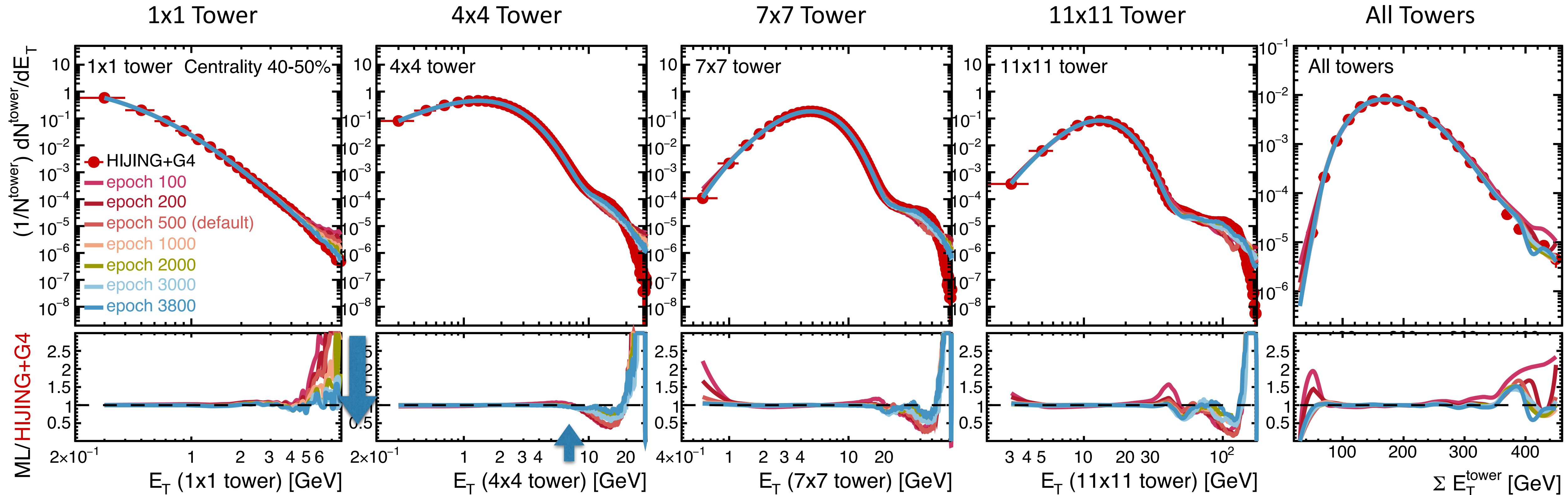
Trade-off between Generation time and Fidelity



PRC 110 (2024) 034912

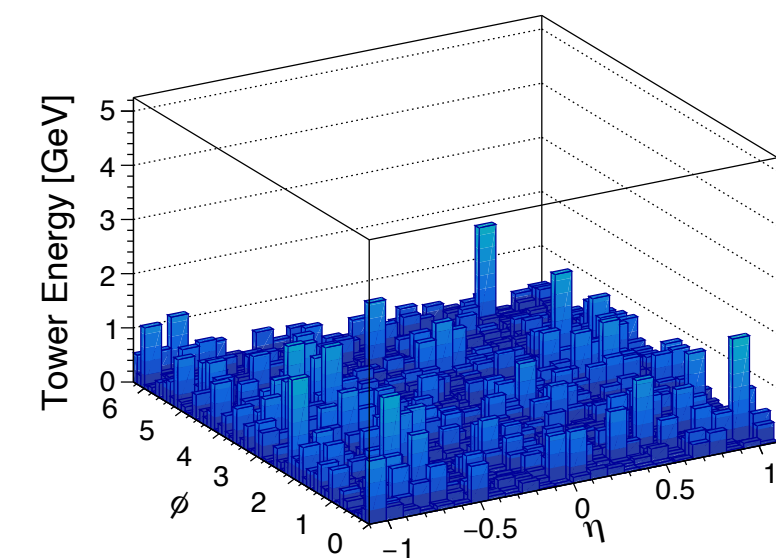
- **DDPM** models with the **higher de-noising steps** give **better performance!**
➡ but, the **higher the de-noising**, the *longer the generation time*

Trade-off between Training time and Fidelity



PRC 110 (2024) 034912

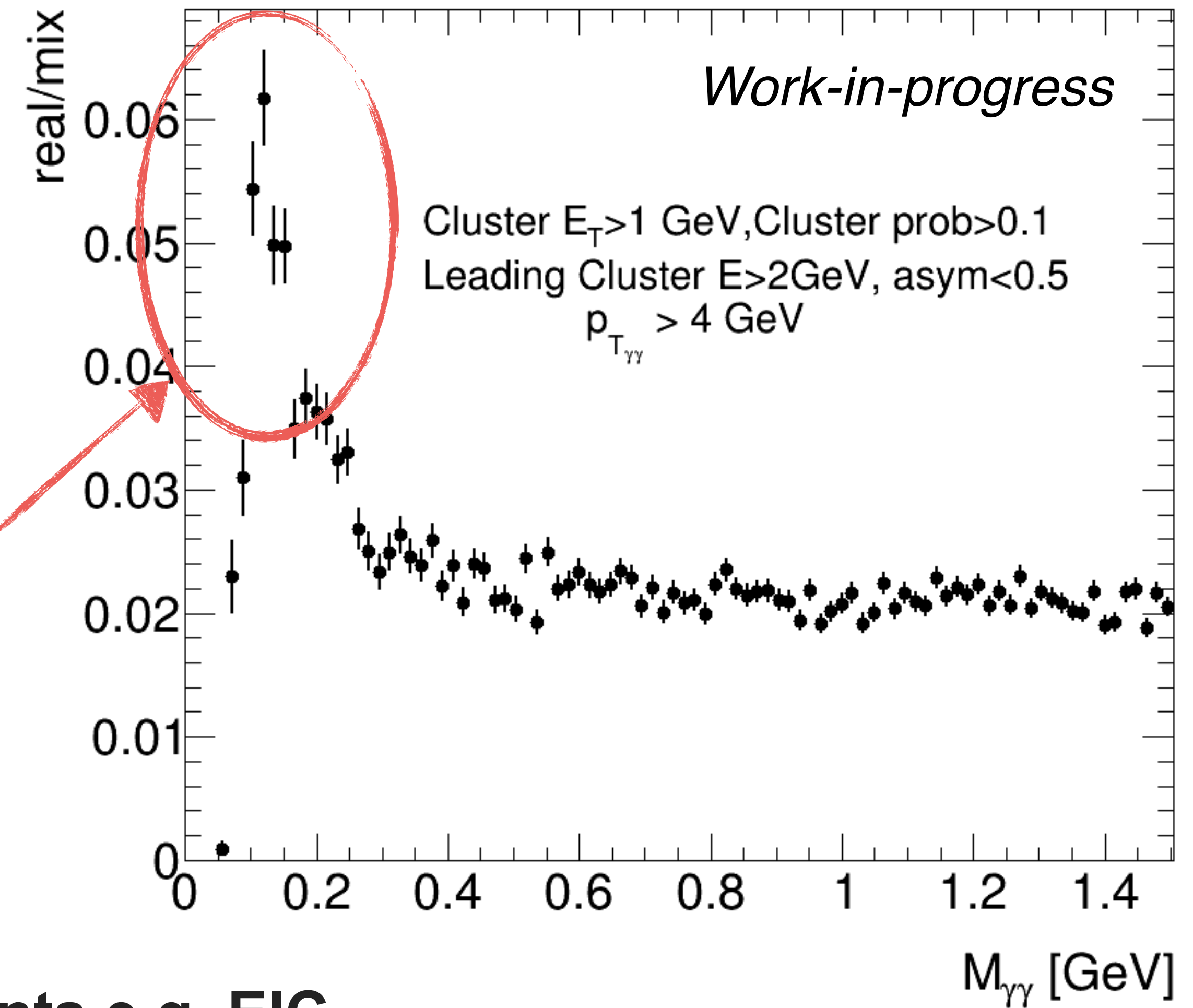
- epoch $\sim$ training duration
- **DDPM** models with the **higher epochs** give **better performance!**
 ➔ but, the **higher the epochs**, the *longer the training time*



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

- **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*

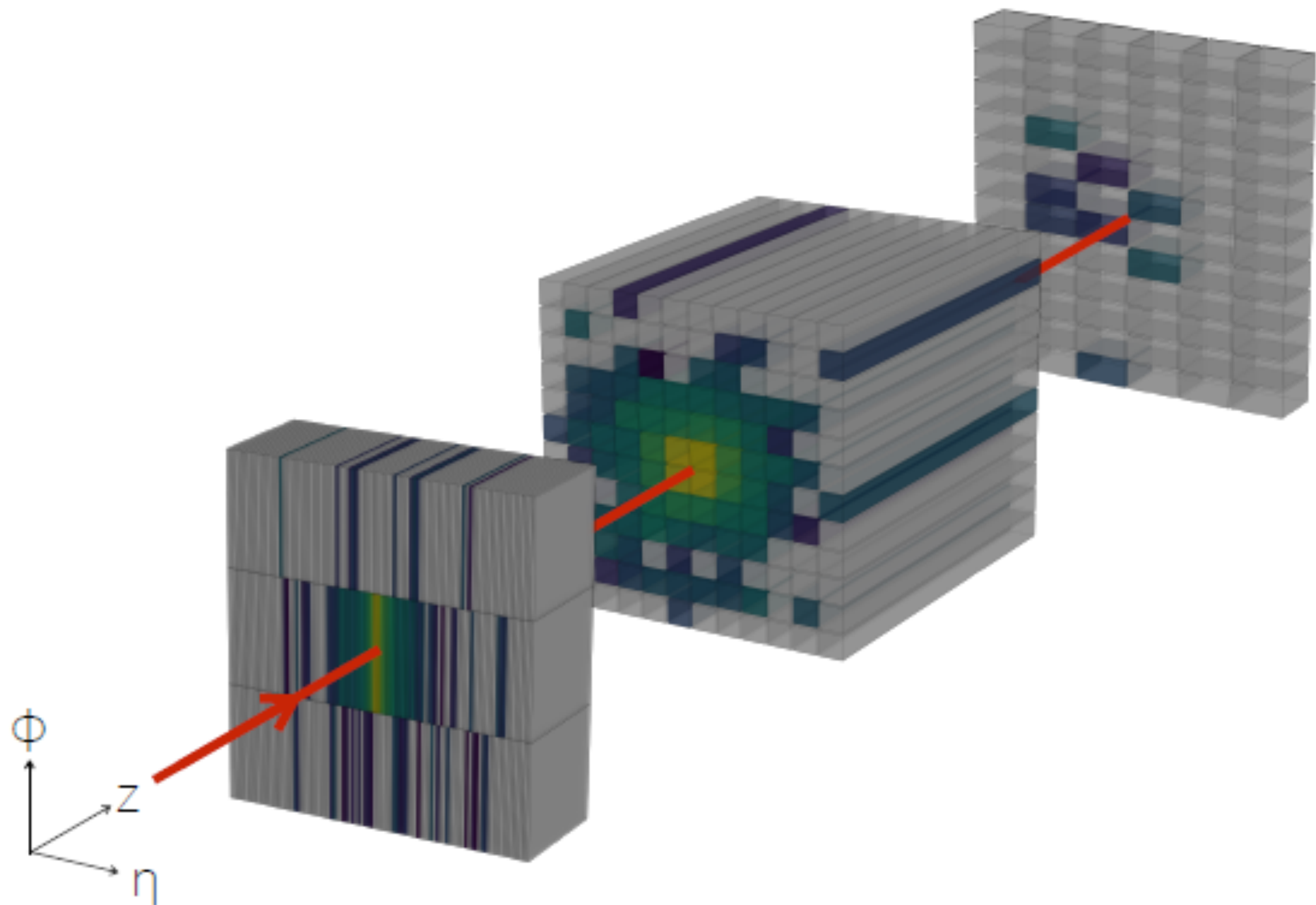


Table 1: Models submitted to the CaloChallenge.

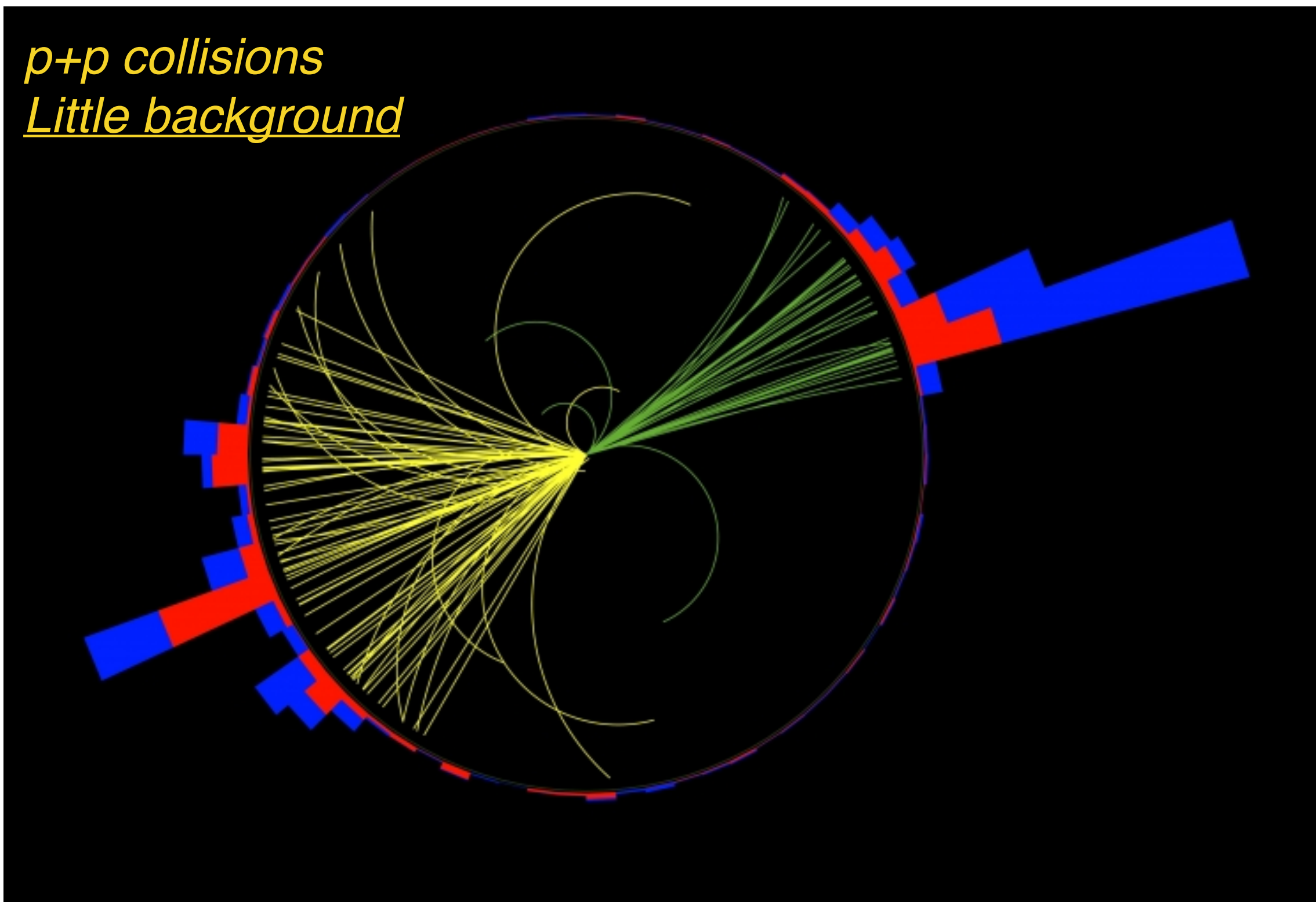
Approach	Model	Code	Dataset				Section
			$1 - \gamma$	$1 - \pi$	2	3	
GAN	CaloShowerGAN [21]	[22]	✓	✓			3.1
	MDMA [23, 24]	[25]			✓	✓	3.2
	BoloGAN [26]	[27]	✓	✓			3.3
	DeepTree [28, 29]	[30]			✓		3.4
NF	L2LFlows [31, 32]	[33]			✓	✓	4.1
	CaloFlow [34, 35]	[36, 37]	✓	✓	✓	✓	4.2
	CaloINN [38]	[39]	✓	✓	✓		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

Jet Background Subtraction:

Cycle-consistent Generative AI

Background in Jet Events in Heavy Ion Collisions

- **Signals (e.g. jet) lie on the significant underlying events (background) in heavy ion collisions**



sPHENIX

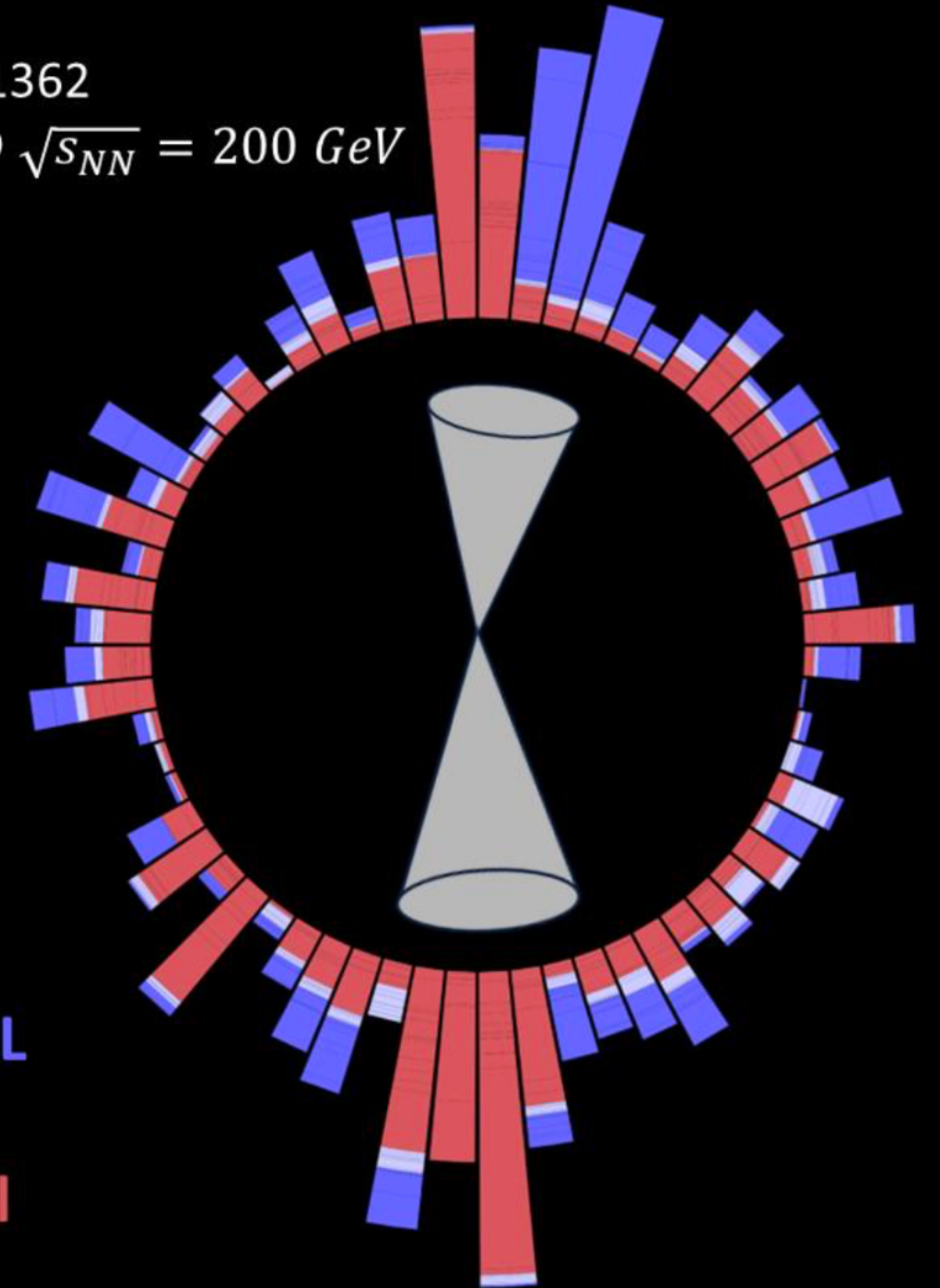
Run/Event: 21615 / 1362

Collisions: Au + Au @ $\sqrt{s_{NN}} = 200 \text{ GeV}$

Peripheral Collision

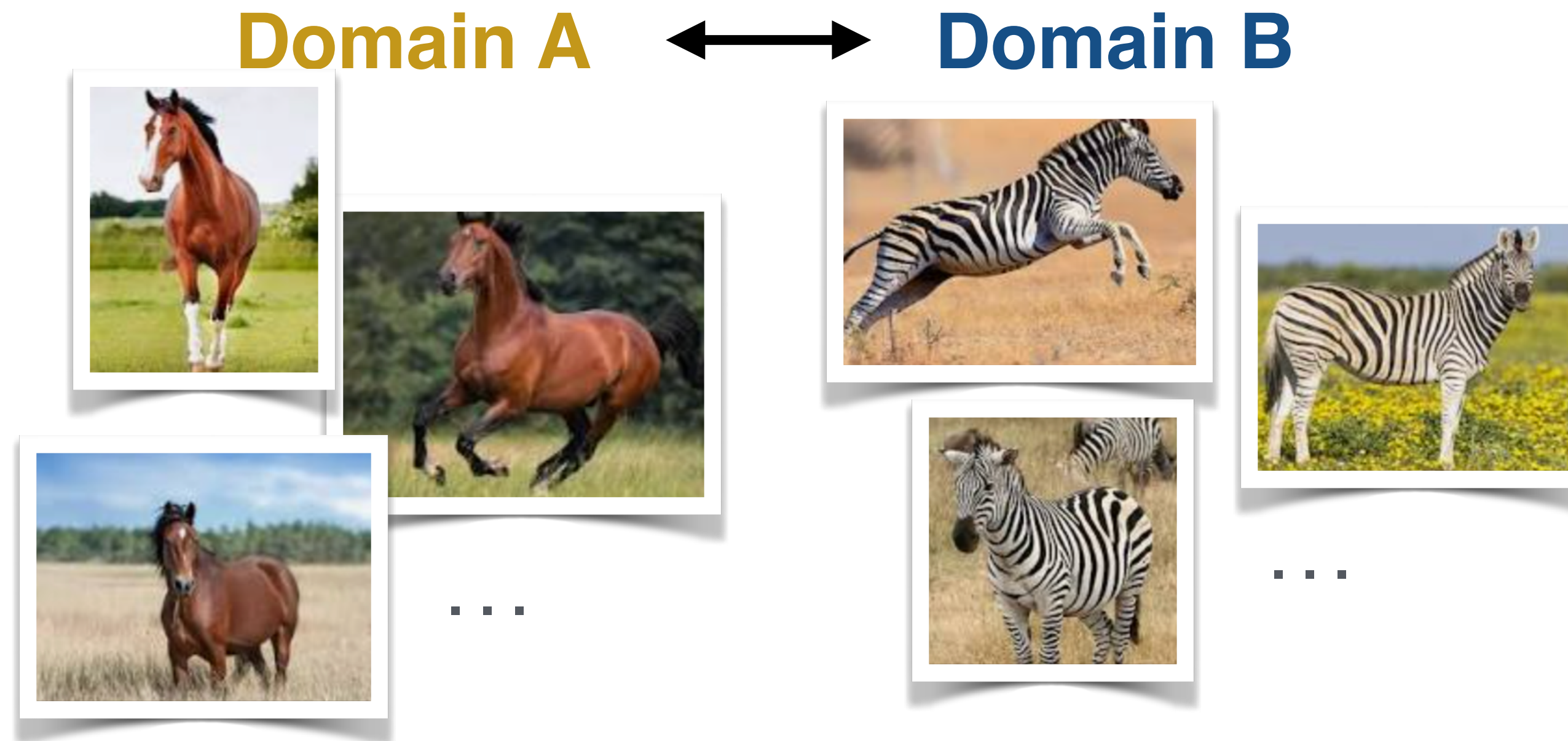
Au+Au collisions
Large background

OHCAL
IHCAL
EMCAL



Cycle-Consistent GAN

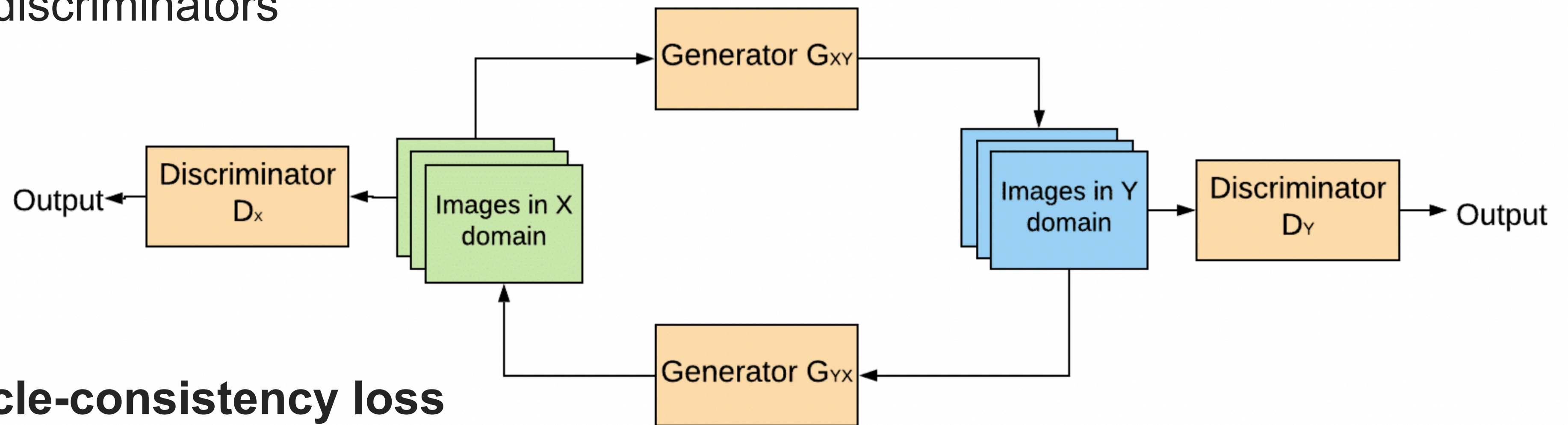
- **Unsupervised** learning, **unpaired** image-to-image translation



arXiv:1703.10593

Cycle-Consistent GAN

- **Unsupervised** learning, **unpaired** image-to-image translation
 - ➔ paired training data are usually harder to obtain
 - ➔ support tasks in real-world domain where only unpaired examples exist
- Two generators and two discriminators



- **adversarial loss** and **cycle-consistency loss**

Adversarial Loss

- $A \rightarrow B \sim B?$
- $B \rightarrow A \sim A?$

➔ *Encourages realism in generated images*

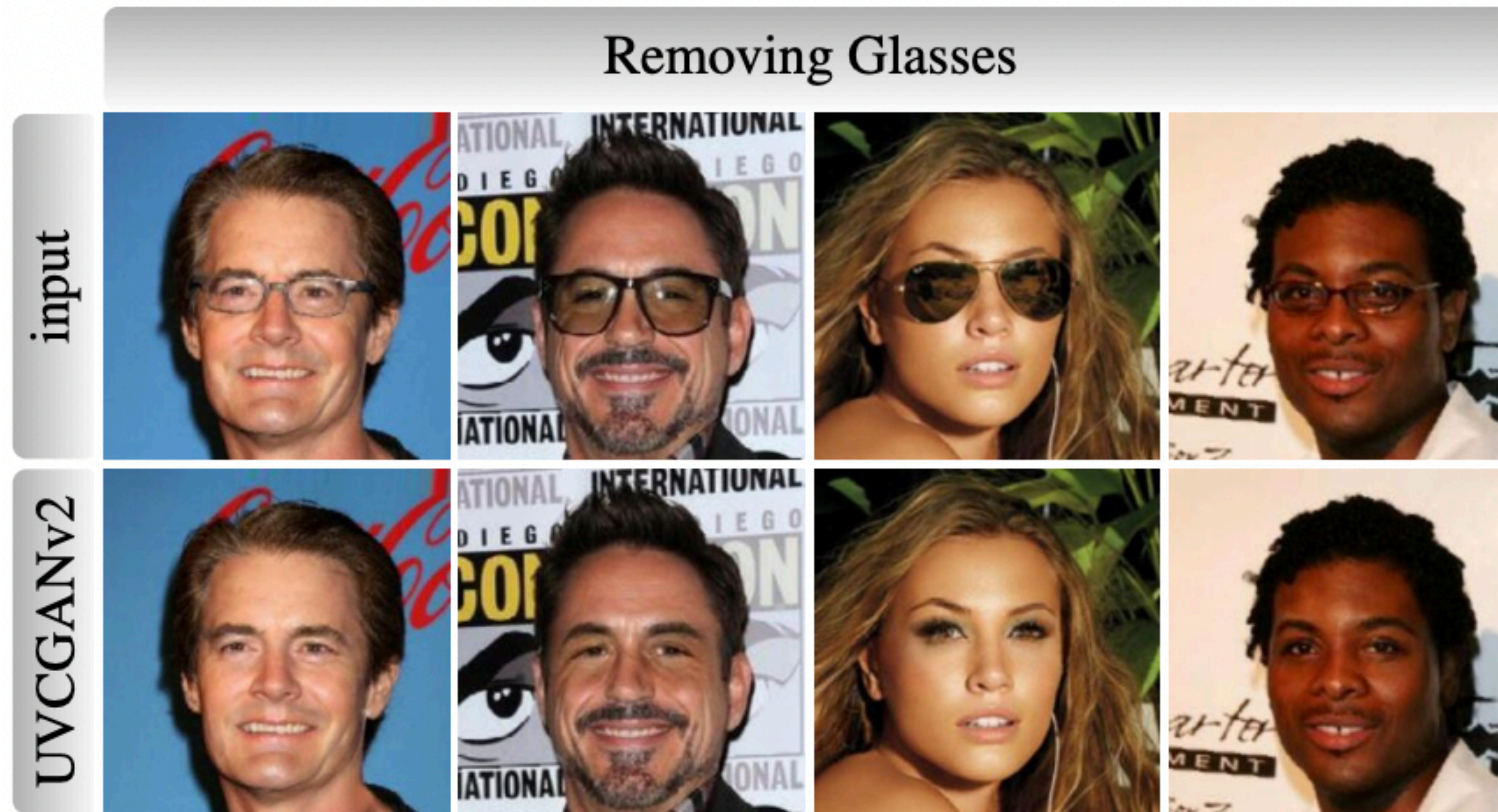
Cycle-consistency Loss

- $A \rightarrow B \rightarrow A \sim A?$
- $B \rightarrow A \rightarrow B \sim B?$

➔ *Encourages preservation of key content and structure*

UVCGAN

- **UVCGAN** (U-net Vision transformer Cycle-consistent Generative Adversarial Network)
 - ➔ **unpaired** image-to-image (I2I) translation; bridging gap between simulation and data reference

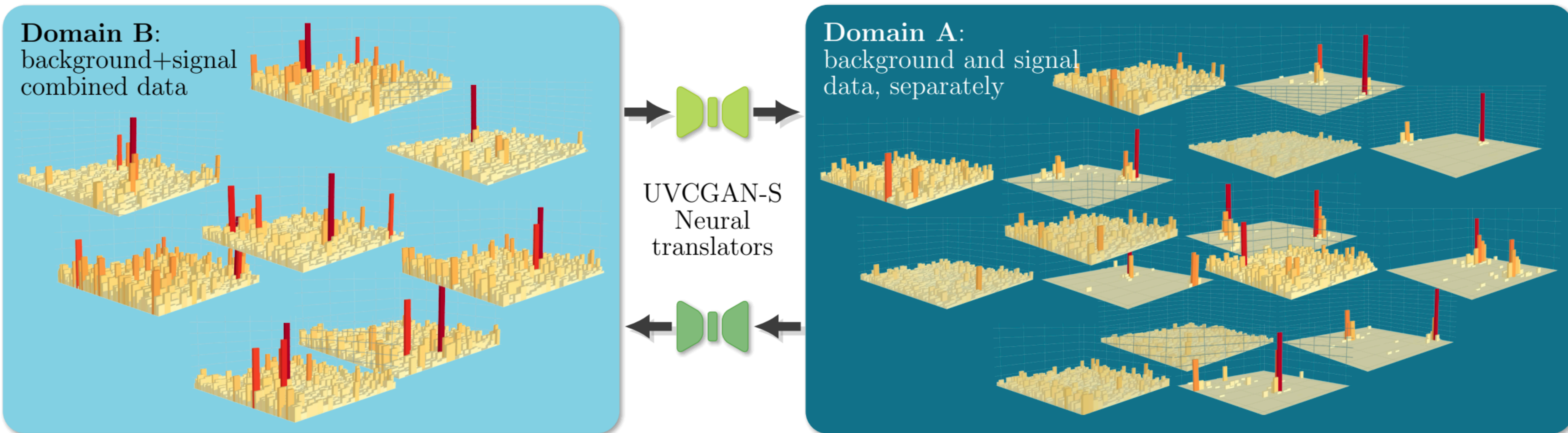


[arXiv:2303.16280](https://arxiv.org/abs/2303.16280)

UVCGAN-S: Unpaired Unsupervised Training

- As a proof-of-principle, UVCGAN-S model was trained on images of *EMCal* + *HCAL* combined calorimeter energies in (η, ϕ) with three samples
 - **PYTHIA**: Inclusive Jet $\hat{p}_T = 30$ GeV
 - **HIJING**: 0-10% Centrality HIJING
 - **PYTHIA+HIJING**: sum of calorimeter energies of PYTHIA and HIJING

[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)



Generative AI for Jet Background Subtraction

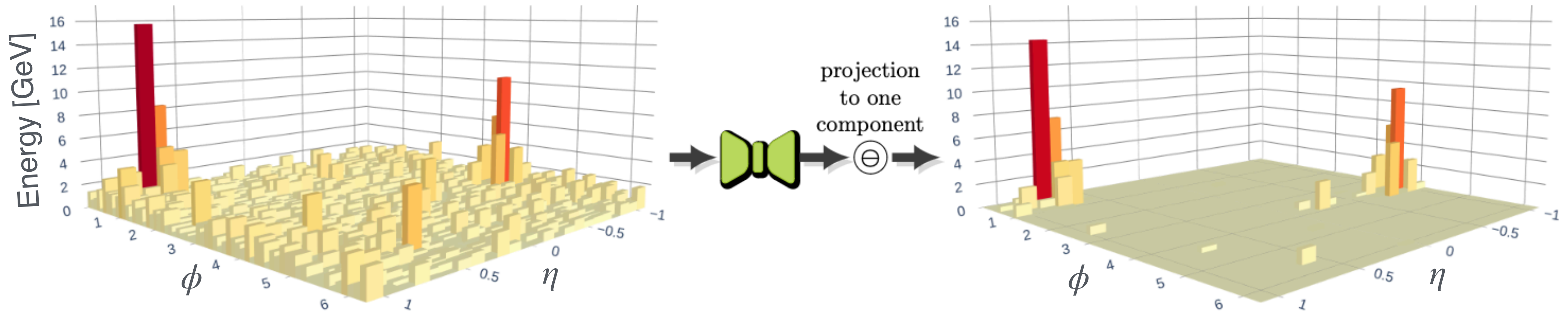
- Jet background subtraction as an “image” using cycle-consistent GAN

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

- **Unsupervised** generative AI

→ *preserves the higher dimensional information*

→ **can use experimental data (without label) for training**
: mitigate model-dependence

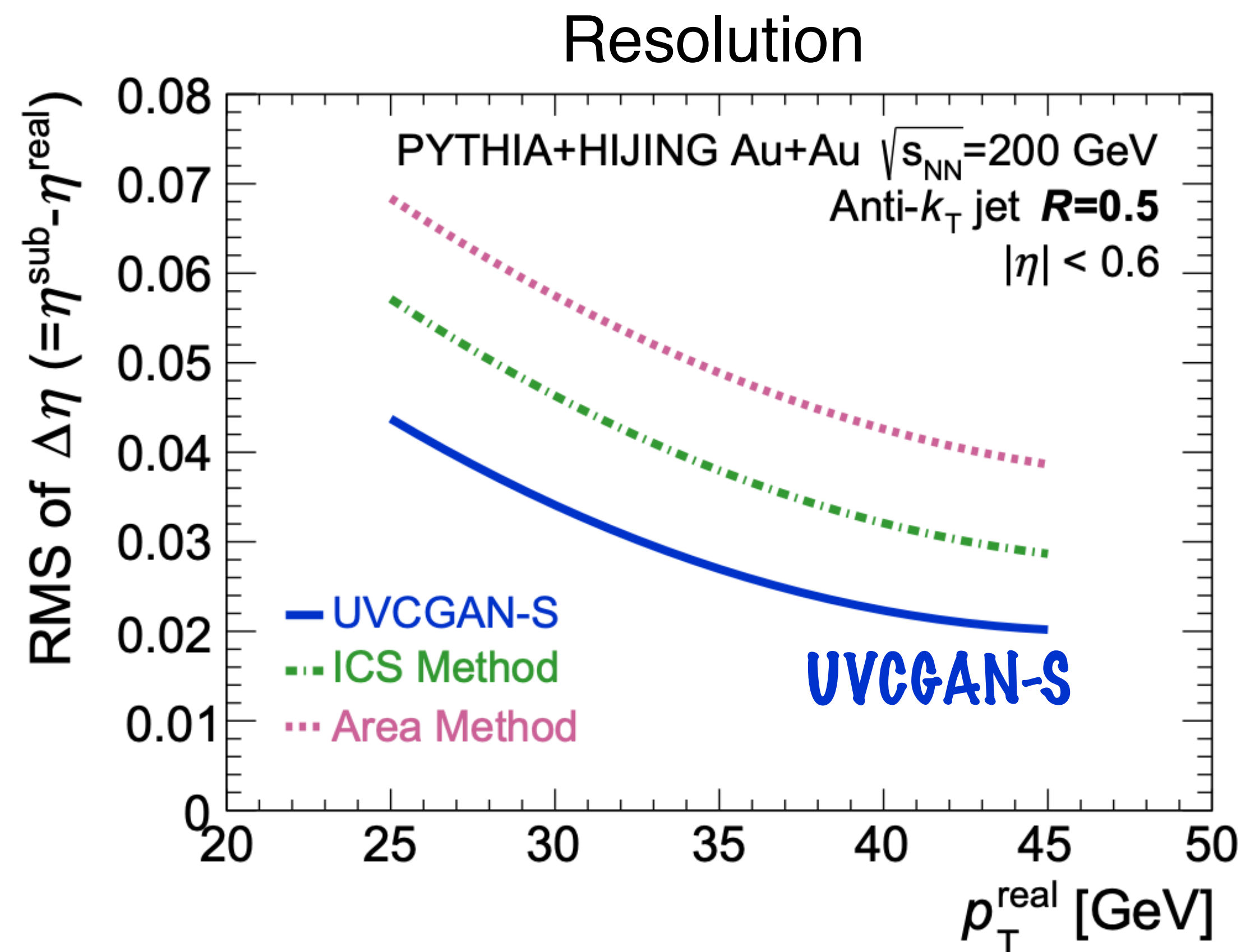
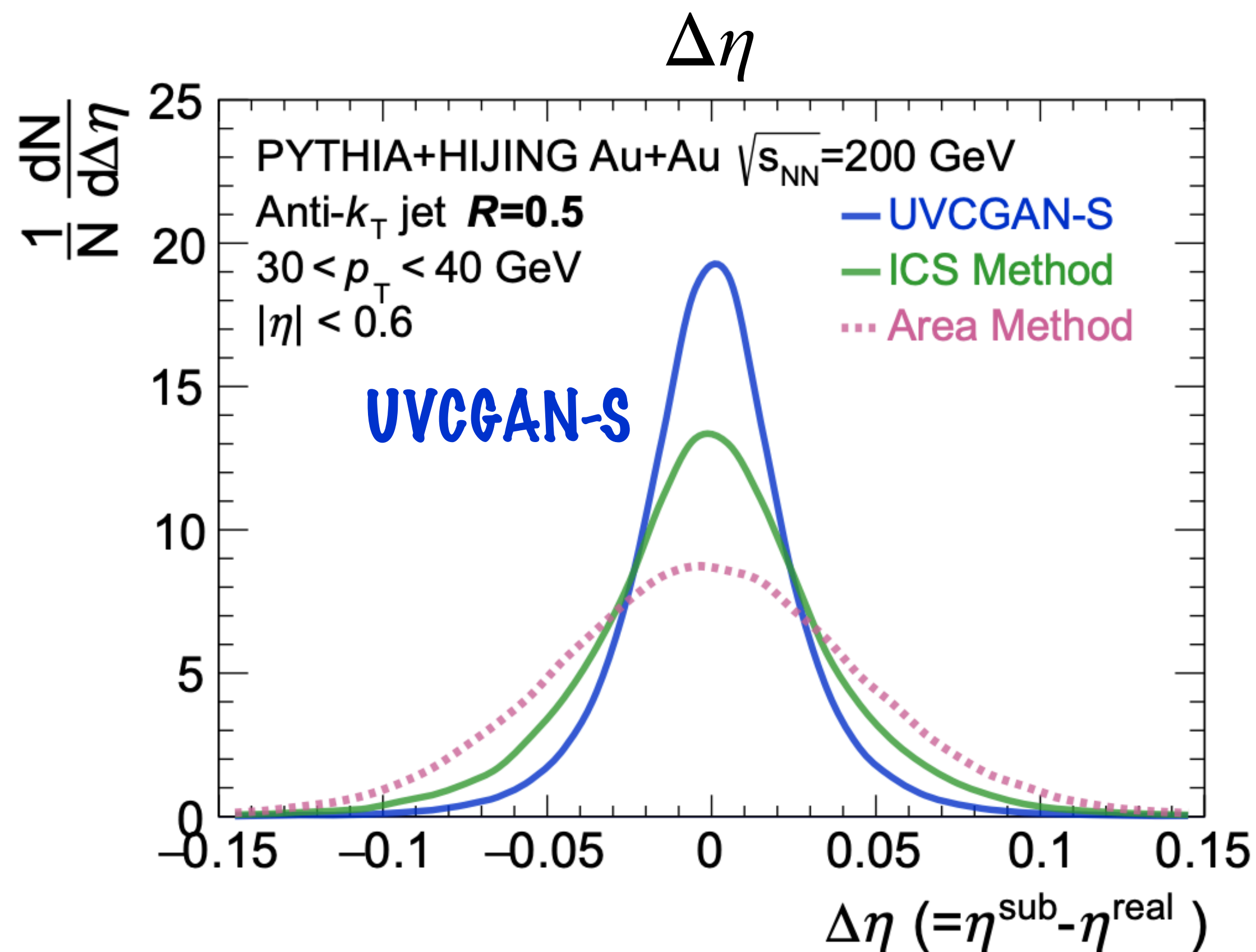


[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)

UVCGAN-S: Position Response

sub: bkg-subtracted
real: response in $p+p$

- Compared to **Iterative Constituent Subtraction (ICS)** and **Area-based Method**
- **UVCGAN-S** achieves up to 40% better position resolution than traditional methods

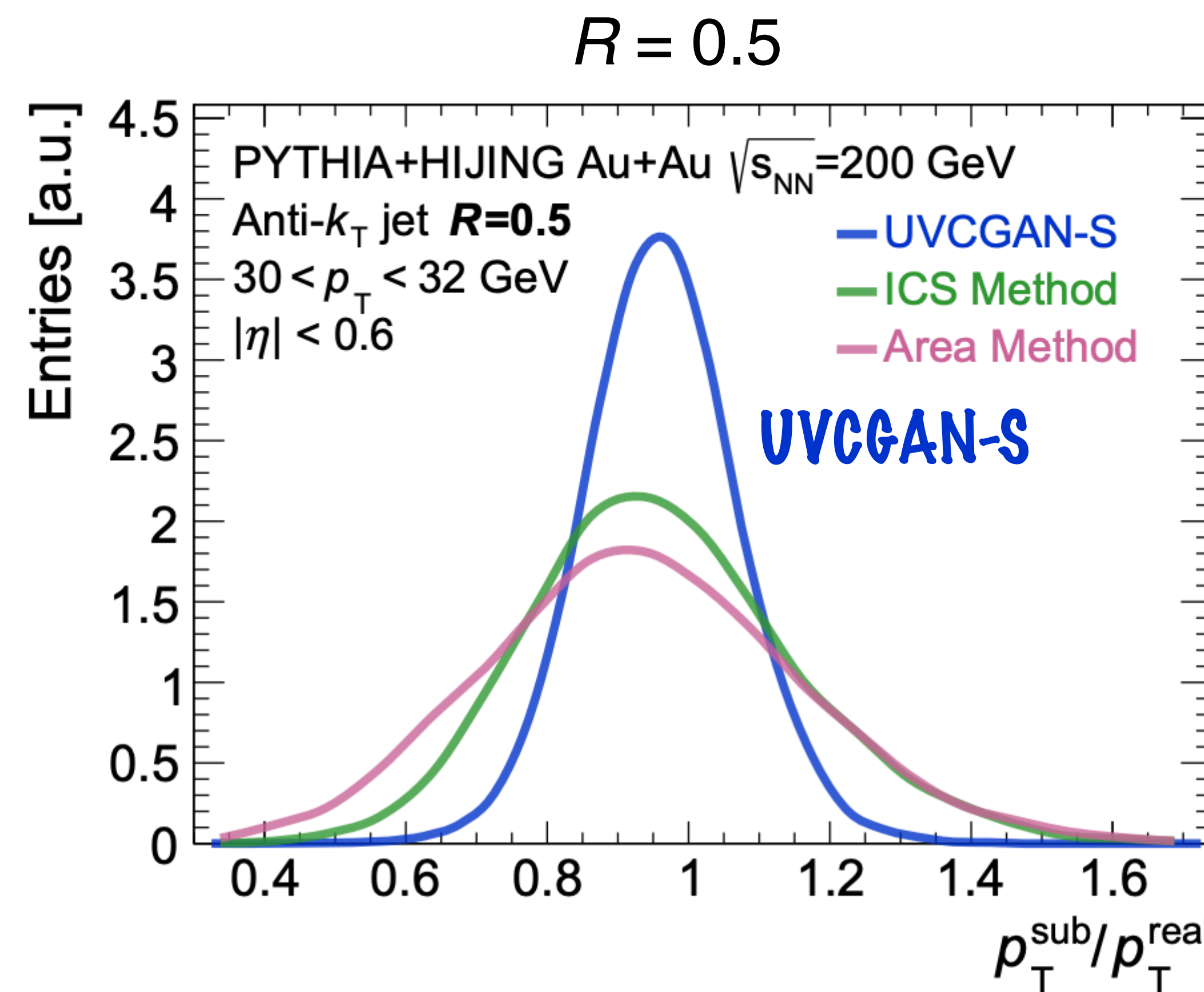
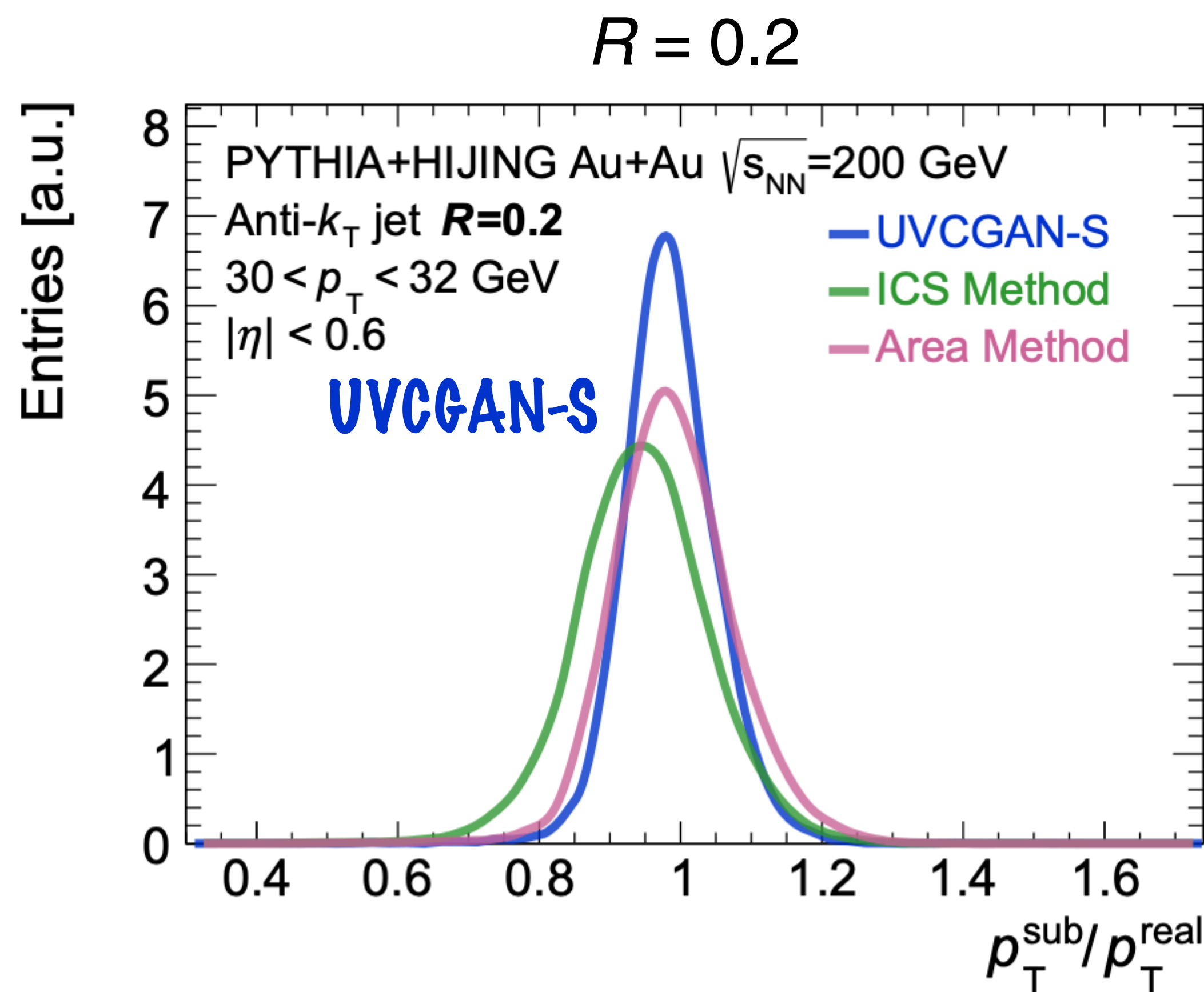


[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)

UVCGAN-S: Momentum Response

sub: bkg-subtracted
real: response in $p+p$

- Compared to **Iterative Constituent Subtraction (ICS)** and **Area-based Method**
- Dramatic improvement in resolution, especially at large R

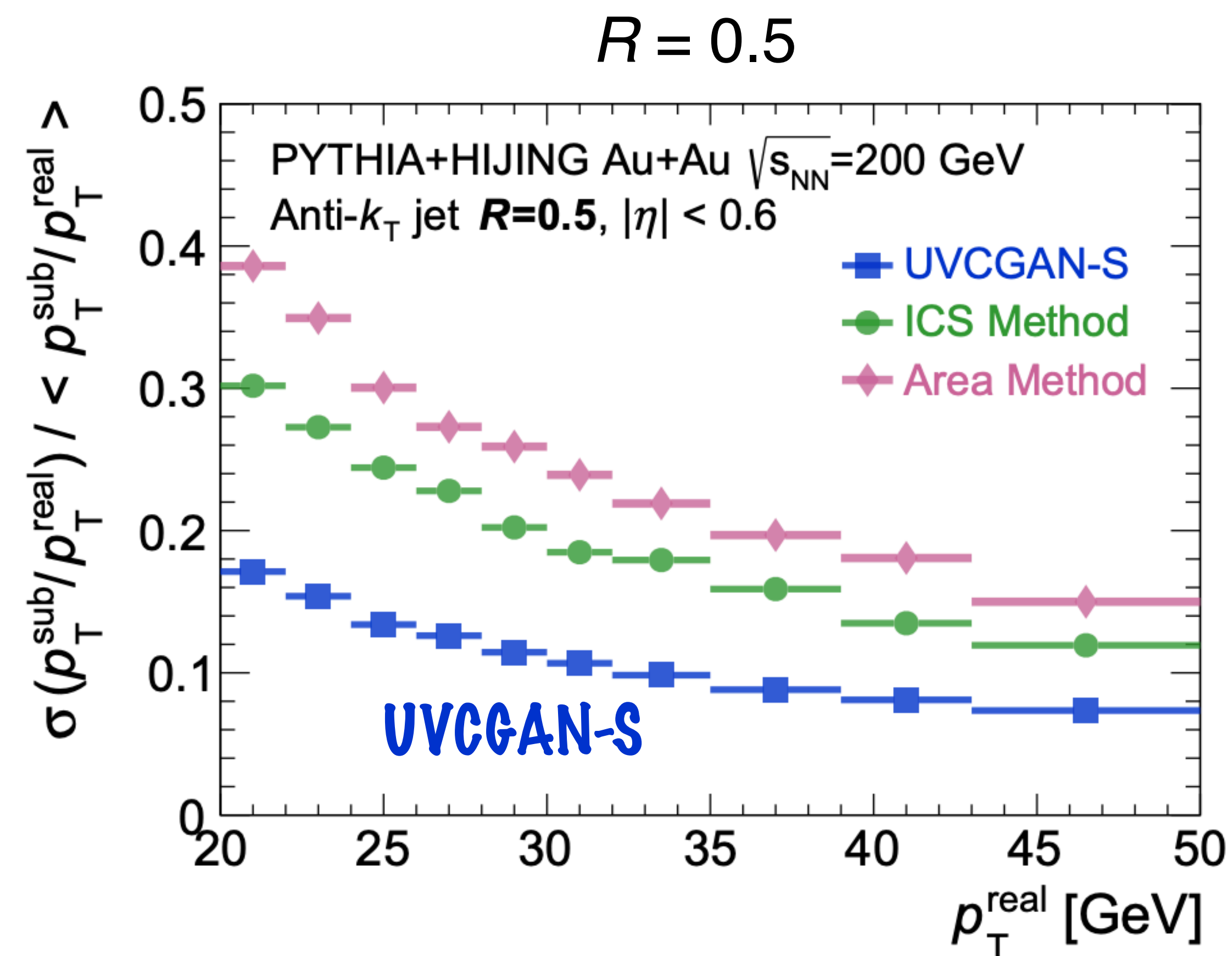
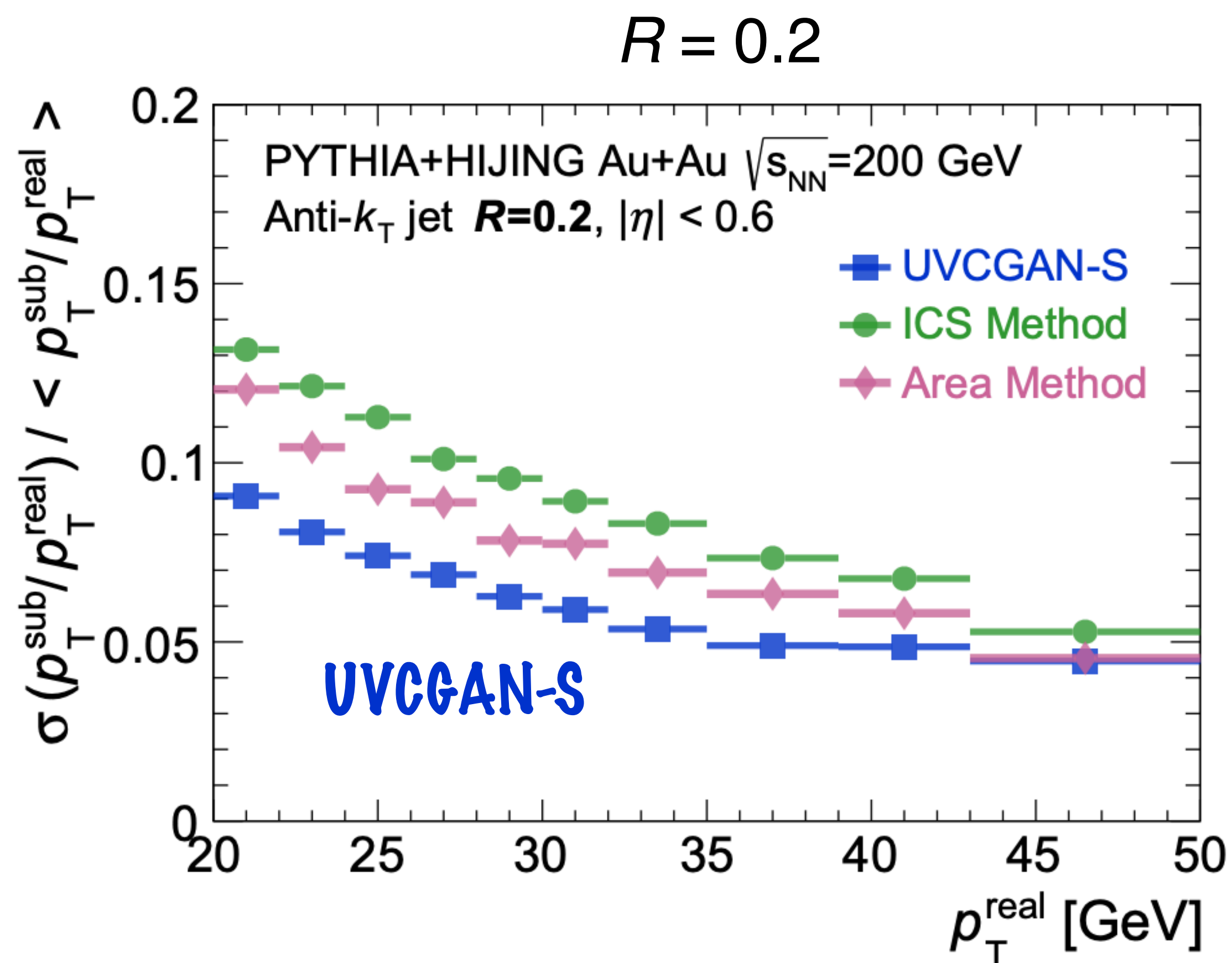


[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)

UVCGAN-S: Momentum Response

sub: bkg-subtracted
real: response in $p+p$

- Compared to **Iterative Constituent Subtraction (ICS)** and **Area-based Method**
- **Dramatic improvement in resolution, especially at large R**
 - ➔ 100% better momentum resolution for $R=0.5$

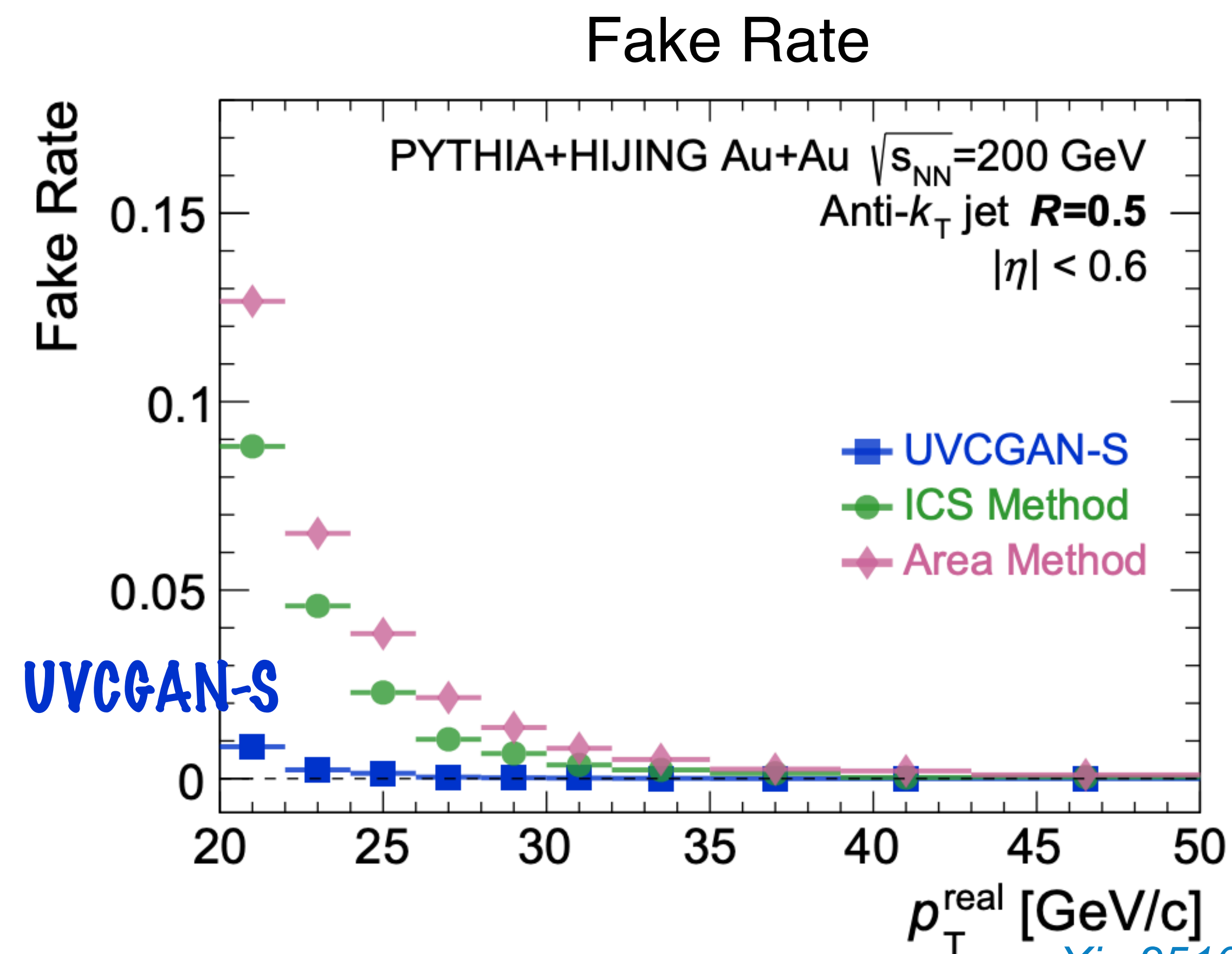
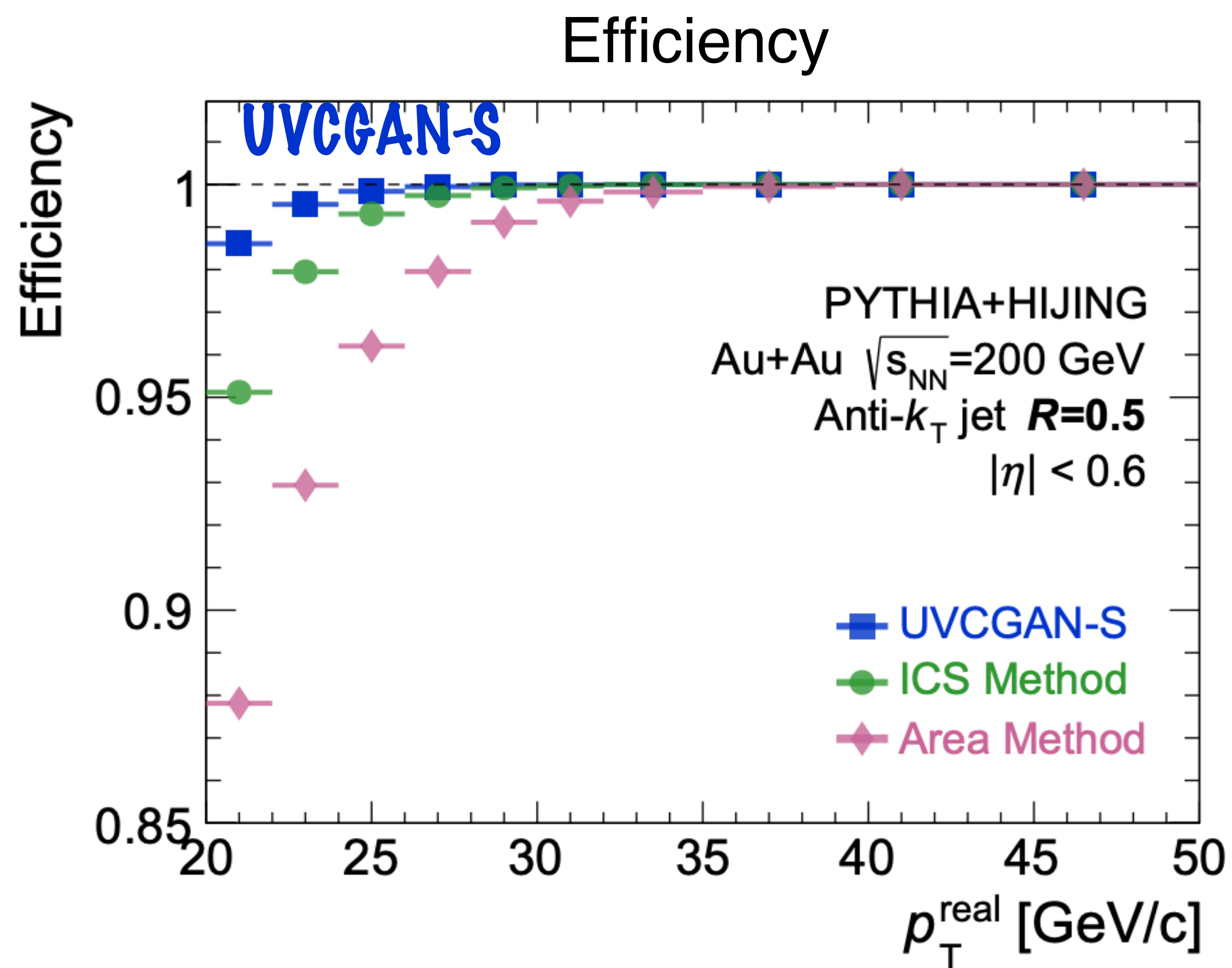


[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)

UVCGAN-S: Efficiency and Fake Rate

sub: bkg-subtracted
real: response in $p+p$

- Compared to **Iterative Constituent Subtraction (ICS)** and **Area-based Method**
- **Dramatic improvement in efficiency and fake rate, especially at large R**
➔ efficiency 88% vs. 99%, fake rate 13% vs. 1% for $R=0.5$

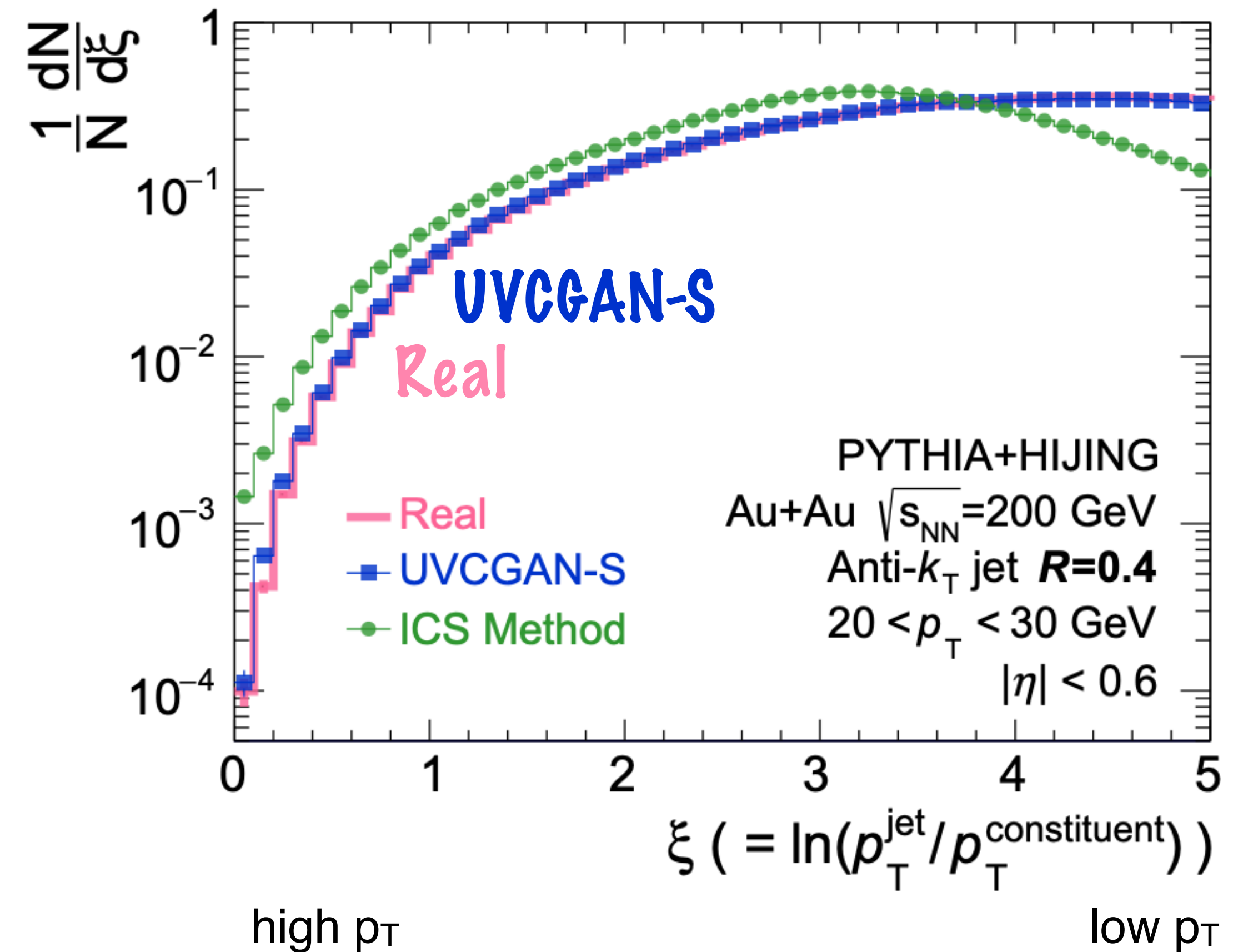
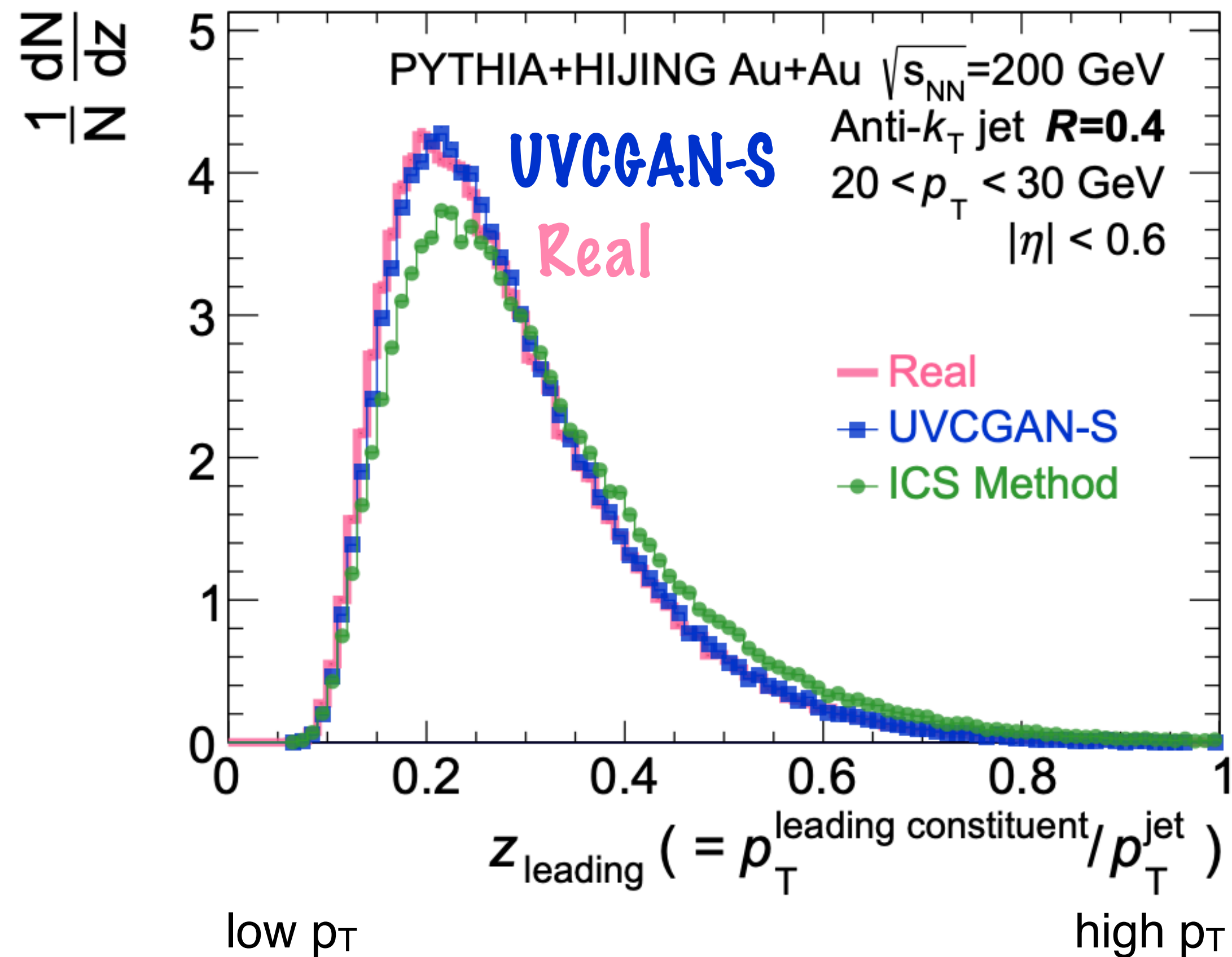


[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)

UVCGAN-S: Jet Substructure

sub: bkg-subtracted
real: response in p+p

- Compared to Iterative Constituent Subtraction (ICS)
- Superior jet substructure reproduction of UVCGAN-S



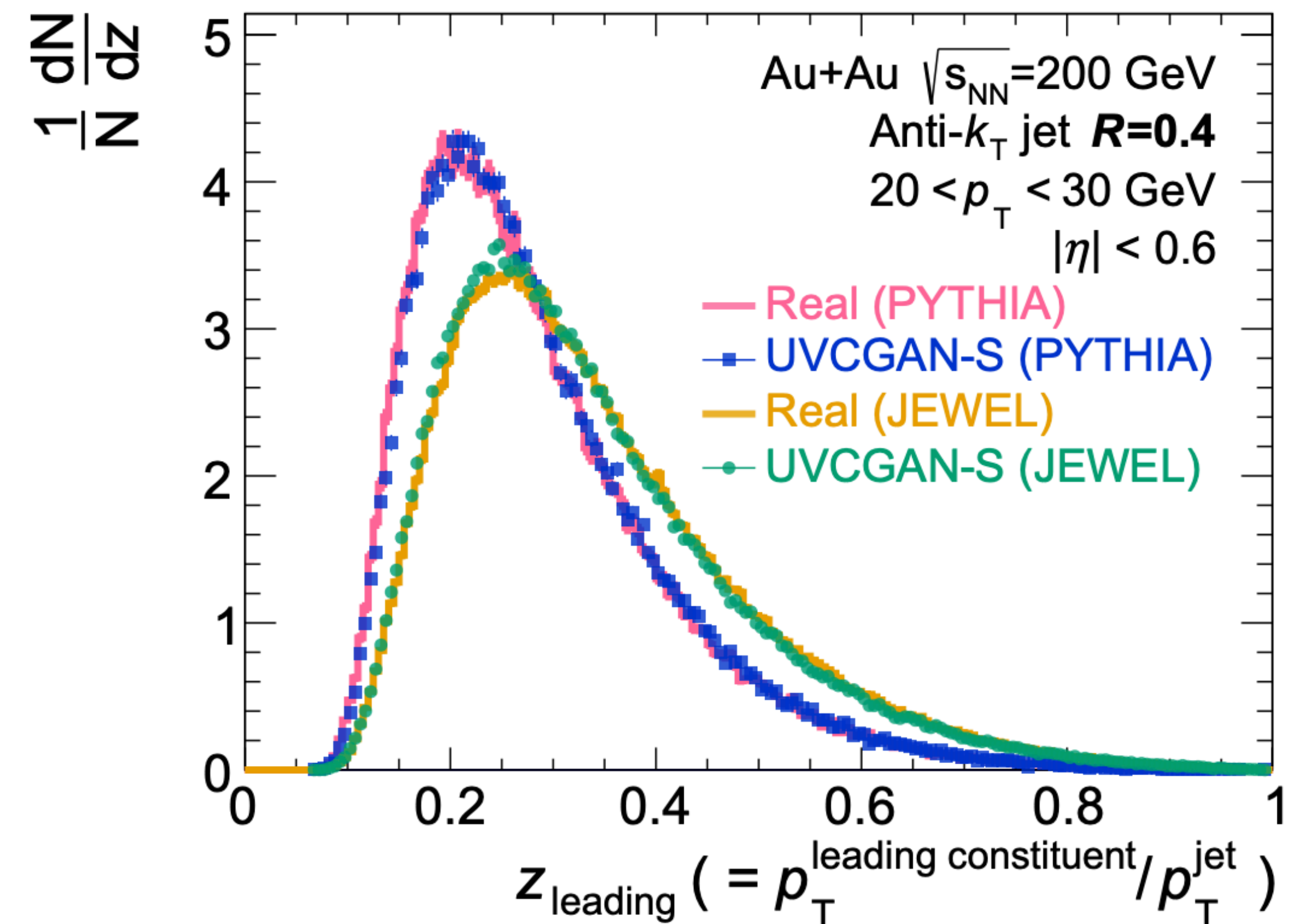
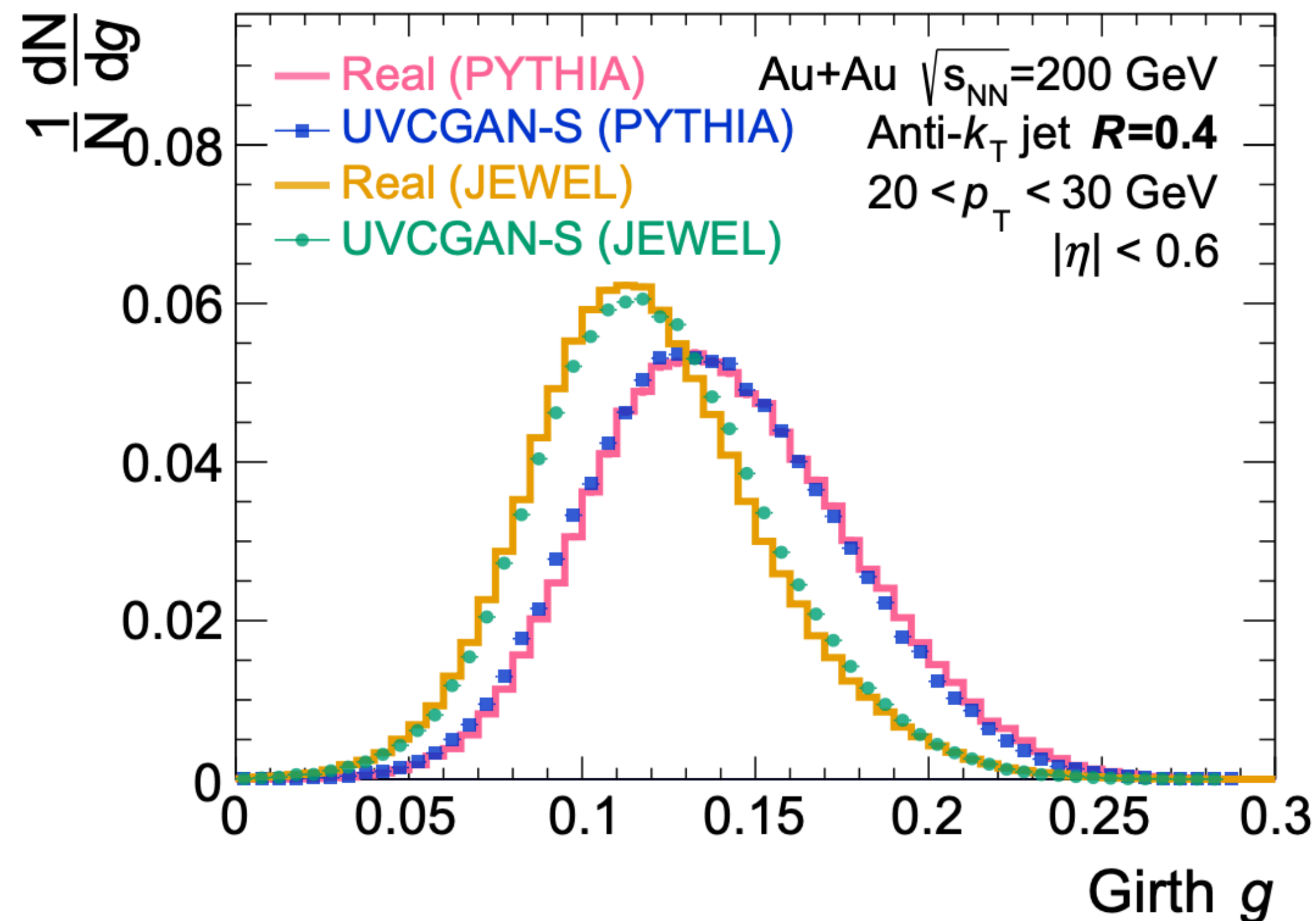
[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)

UVCGAN-S: Application on Quenched Jets

- UVCGAN-S trained on **PYTHIA** (unquenched jets) and tested on **JEWEL** (quenched jets)
 - ➔ significant difference in substructure distributions for quenched jets
 - ➔ **UVCGAN-S** successfully reproduce the **JEWEL**'s distribution

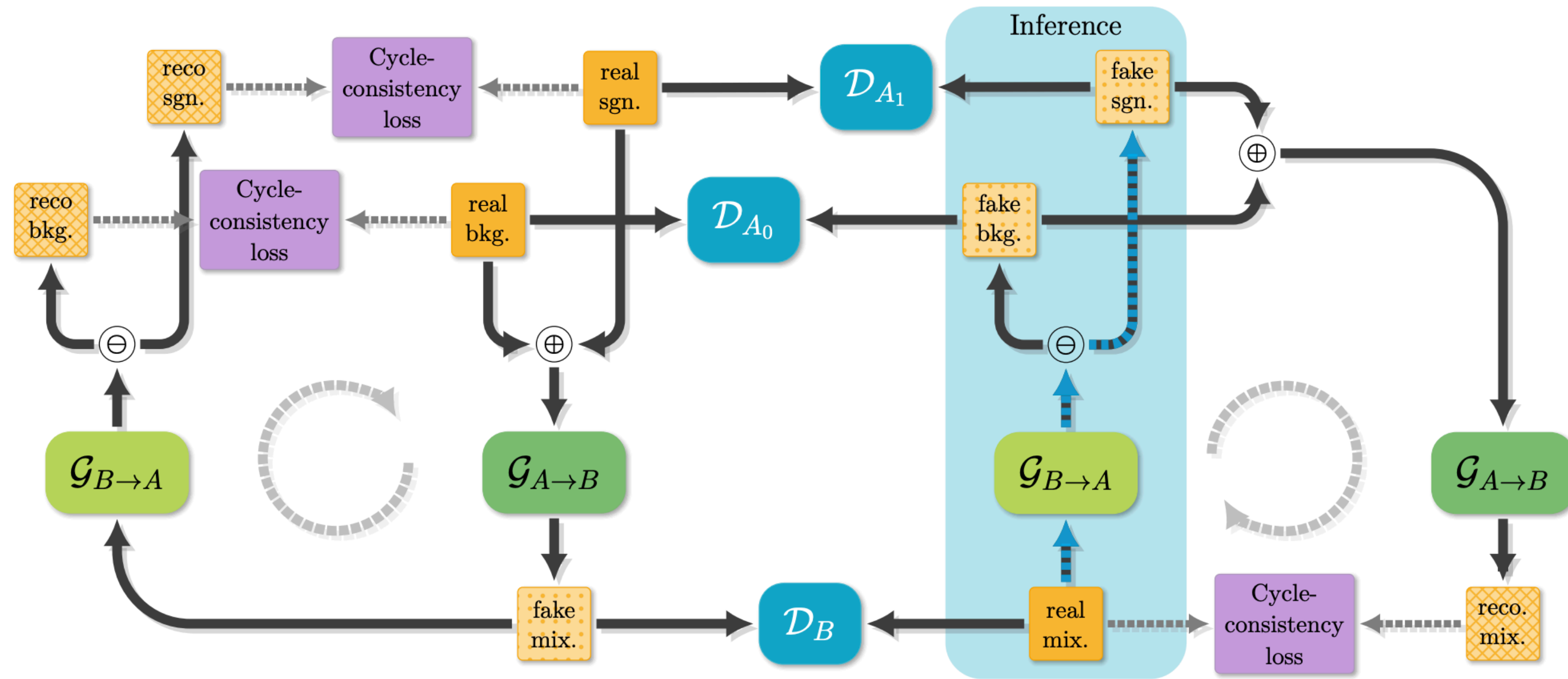
**Robust Generalizability of the
UVCGAN-S for out-of-distribution handling!**

[arXiv:2510.23717](https://arxiv.org/abs/2510.23717)

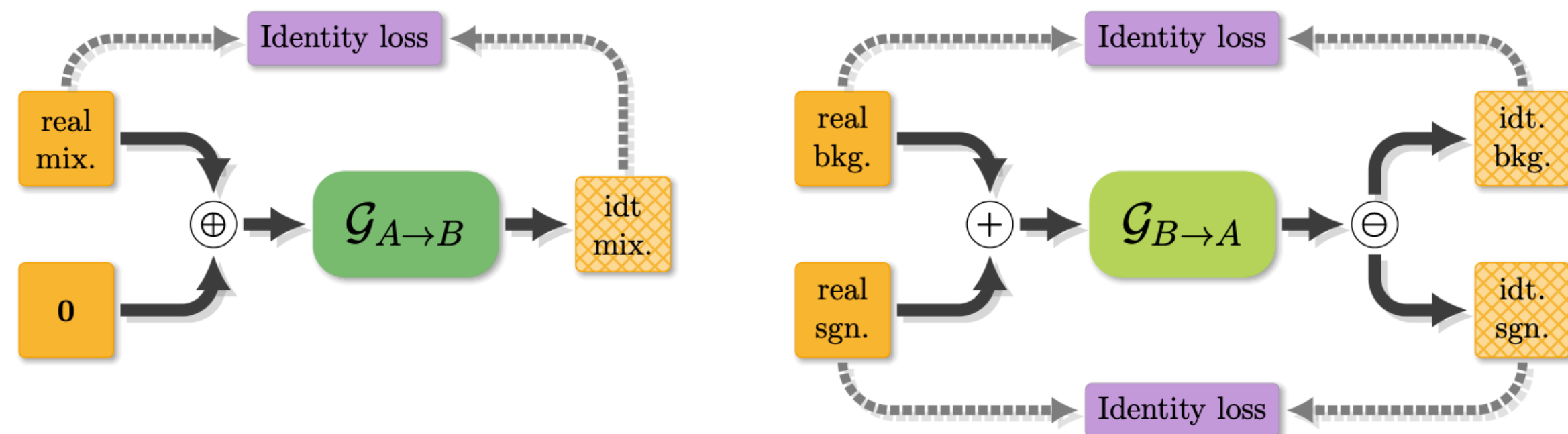


UVCGAN-S Architecture

(a) UVCGAN-S Architecture (identity losses shown separately below)



(b) Identity Losses



$\text{real bkg.} : a_0$ $\text{fake bkg.} : \Pi_{A_0}(\mathcal{G}_{B \rightarrow A}(b))$ $\text{reco bkg.} : \Pi_{A_0}(\mathcal{G}_{B \rightarrow A}(\mathcal{G}_{A \rightarrow B}(a_0, a_1)))$
 $\text{real sgn.} : a_1$ $\text{fake sgn.} : \Pi_{A_1}(\mathcal{G}_{B \rightarrow A}(b))$ $\text{reco sgn.} : \Pi_{A_1}(\mathcal{G}_{B \rightarrow A}(\mathcal{G}_{A \rightarrow B}(a_0, a_1)))$
 $\text{real mix.} : b$ $\text{fake mix.} : \mathcal{G}_{A \rightarrow B}(a_0, a_1)$ $\text{reco mix.} : \mathcal{G}_{A \rightarrow B}(\Pi_{A_0}(\mathcal{G}_{B \rightarrow A}(b)), \Pi_{A_1}(\mathcal{G}_{B \rightarrow A}(b)))$
 $\text{idt bkg.} : \Pi_{A_0}(\mathcal{G}_{B \rightarrow A}(a_0 + a_1))$ $\oplus$ concatenate along the channel dimension
 $\text{idt sgn.} : \Pi_{A_1}(\mathcal{G}_{B \rightarrow A}(a_0 + a_1))$ $\ominus$ split along the channel dimension
 $\text{idt mix.} : \mathcal{G}_{A \rightarrow B}(b, 0)$ $+$ addition

training data flow
 inference data flow
 loss data flow