DiT-based fast simulation for the CEPC long-bar crystal electromagnetic calorimeter

Zhihao Li[1,2], Tao Lin[2], Simon C Blyth[2], Weidong Li[1,2]

[1] University of Chinese Academy of Sciences, China

[2] Institute of High Energy Physics, Chinese Academy of Sciences, China

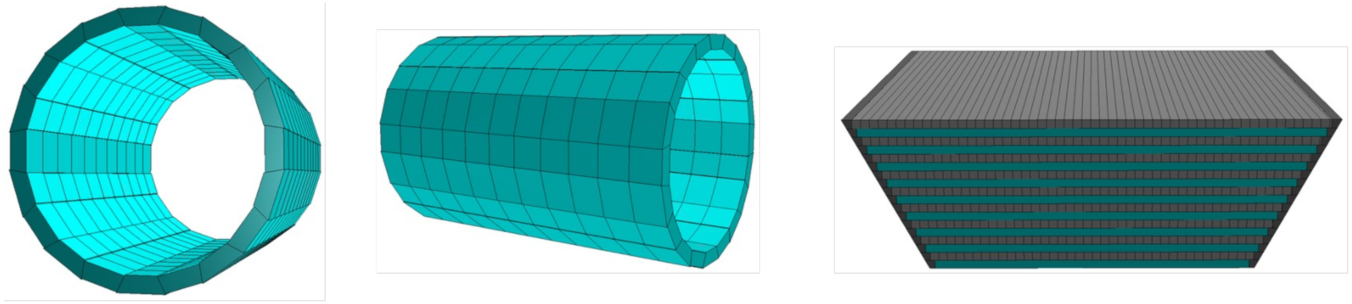
Outline

- Motivation
- DiT-based fast simulation method
- Integration of fast simulation
- Summary

CEPC ECAL

- **ECAL:** The long-bar crystal ECAL forms an interleaved three-dimensional mesh that delivers about 3% energy resolution with fine imaging.

$$\frac{\sigma_E}{E} \approx \frac{3\%}{\sqrt{E/GeV}}$$



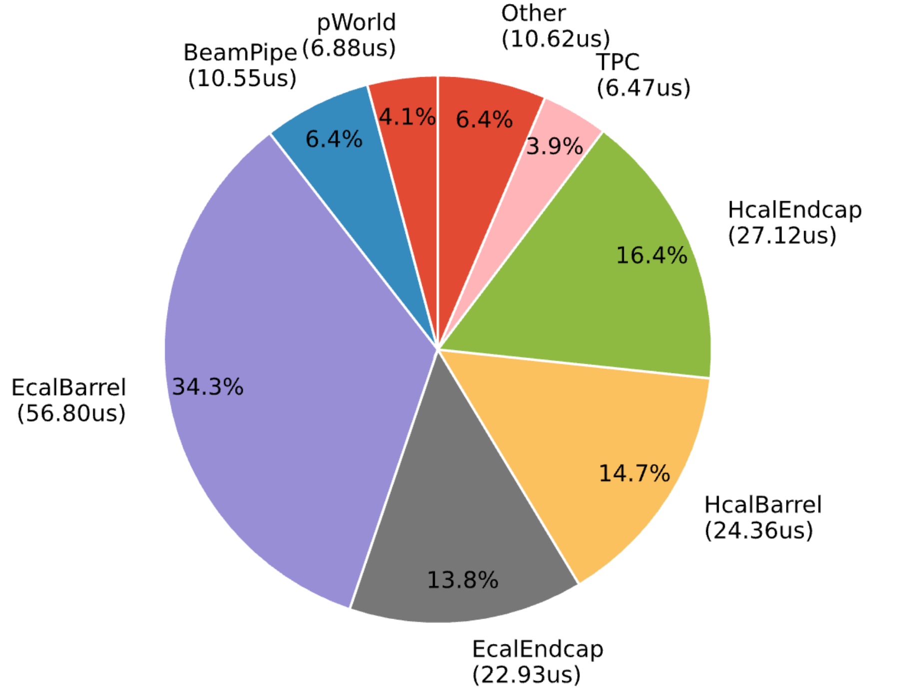
CEPCSW Computing Bottlenecks

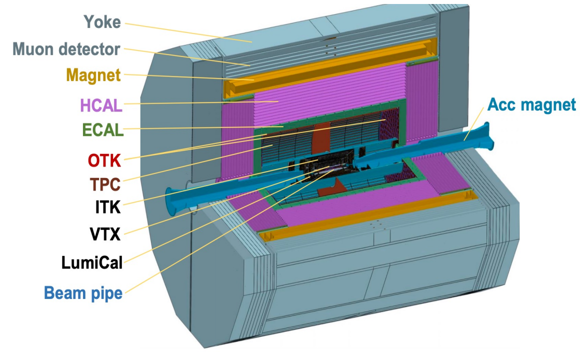
- **Classical Computing Resource Bottleneck:** The CEPC full-simulation program is expected to produce about 10^{11} events per year, yielding $284\ PB$ of full-simulation data and consuming roughly 1570 kHS23-yr of CPU computing time annually.

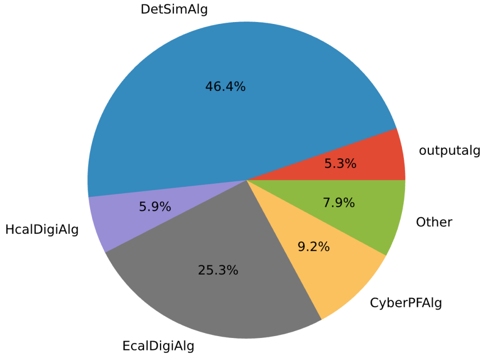
Run stage	MC events	CPU time (kHS23-yr)	MC data (PB)
Higgs	2.0×10^{9}	105	4.00
Low Z	1.4×10^{11}	1465	280.00

CEPCSW Computing Bottlenecks

- **Geant4 and ECAL Simulation Bottlenecks**: For $e^+e^- \rightarrow q\bar{q}$ at 240 GeV the GEANT4 step dominates 46.4% of the wall time; ECAL barrel and endcaps alone consume 34.4% and 13.8% of the total simulation budget, respectively.

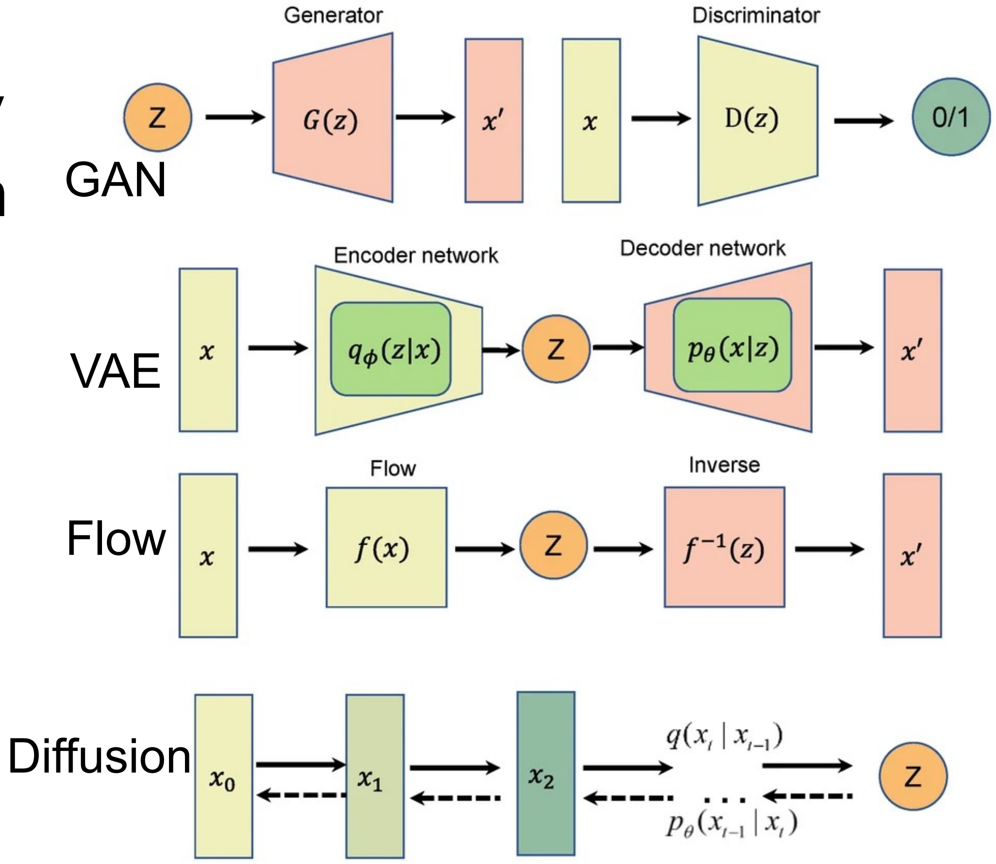






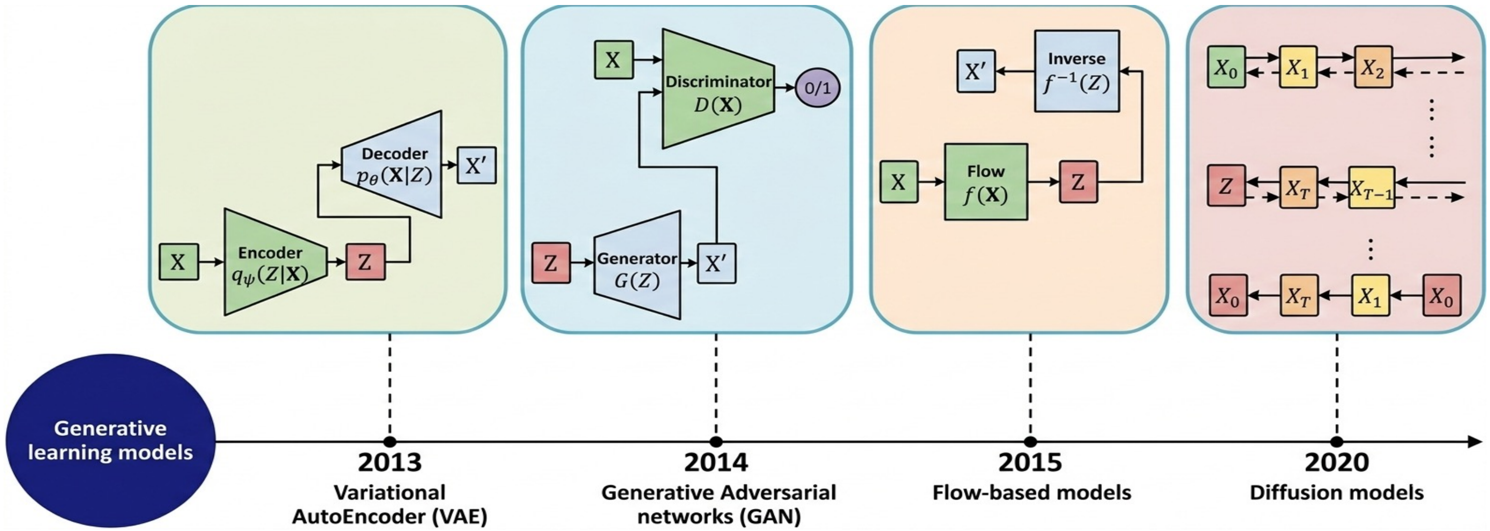
Generative Models

- **Generative models** aim to learn the true data distribution $p_{\text{data}}(x)$ by approximating it with a parameterized model $p_\theta(x)$such that
$$p_\theta(x) \approx p_{\text{data}}(x)$$
- New samples are generated by sampling
$$x \sim p_\theta(x).$$



Generative Models

Year	Model	Strength	Weakness
2013	VAE	Stable	Blurry samples
2014	GAN	Sharp outputs	Unstable training
2015	Flow	Exact likelihood	Expensive
2017	Transformer	Strong long-range modeling	Slow autoregressive sampling
2020	Diffusion	Best overall quality	Slow multi-step sampling



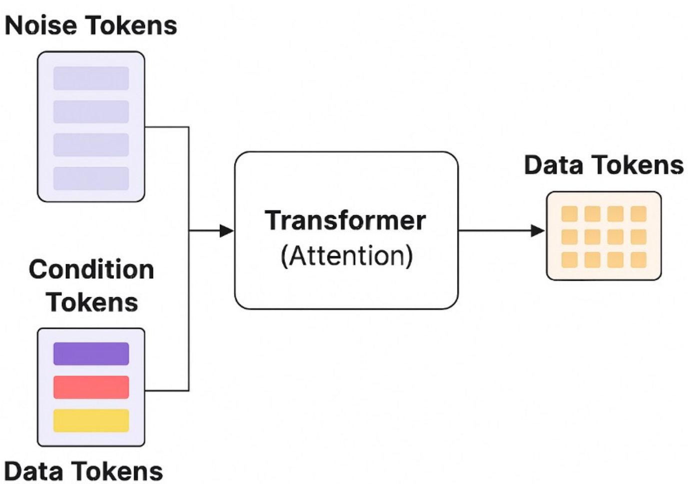
Generative Models : Timeline

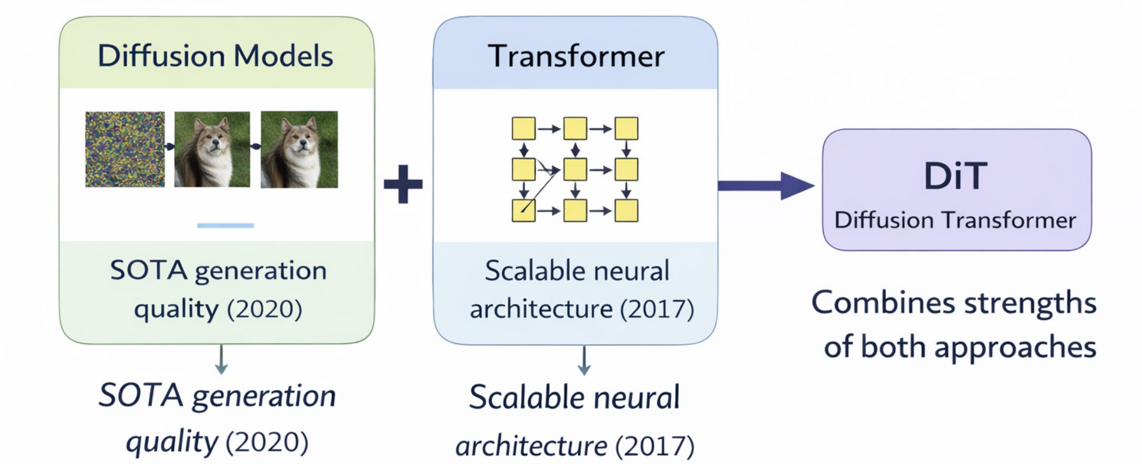
Year	Model Category	Representative Work	Key Contribution
2017	GAN	FastCaloGAN	First use of GANs for calorimeter fast simulation with very large speedup vs Geant4.
2021	Normalizing Flows	CaloFlow	Introduced normalizing flows for calorimeter showers with high fidelity & stable training.
2022	Normalizing Flows	CaloFlow II	Enhanced flows (v2) with probability distillation; validated on community benchmark.
2023	Diffusion Models	CaloClouds / CaloDiffusion	Applied diffusion & point-cloud methods; geometry-independent generation.
2025	Generalizable Diffusion	**CaloDiT-2**	Transformer-based diffusion with pretraining across detectors; rapid adaptation with reduced data/time cost.

- Higher **accuracy**
- More **stable** generation
- Better **generalization**

DiT I: Overview

- **DiT (Diffusion Transformer)** is a generative architecture for conditional diffusion which is particularly suitable for high-dimensional, complex, and conditional generation tasks.
  - **Noise Tokens:** Randomly initialized noise inputs representing the starting noisy state.
  - **Condition Tokens:** Tokens encoding conditioning information such as particle type, total energy, angles, etc.
  - **Transformer Attention Block**: The core of DiT, using self-attention and cross-attention mechanisms to enable interaction between different tokens.
  - **Data Tokens**: Representations of the crystal array or energy deposits, from which the model produces denoised outputs that satisfy the conditioning.





DiT II: Diffusion Process

Forward diffusion process

- Starting from original data x_0, Gaussian noise is gradually added in steps to produce a sequence

$$x_0 \to x_1 \to \cdots \to x_T$$

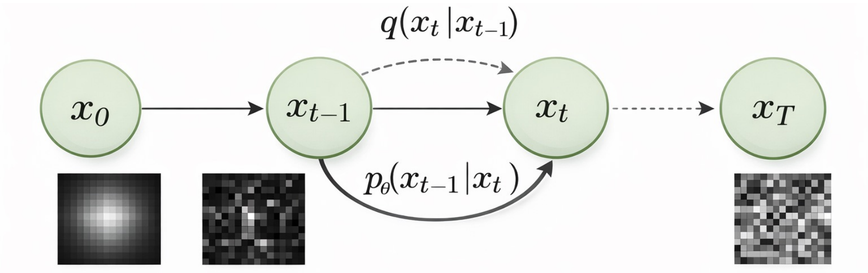
where x_T approaches an isotropic Gaussian distribution as T.

Reverse diffusion process

- The model learns a parameterized denoising distribution

$$p_\theta(\, x_{t-1} \mid x_t \,)$$

which predicts how to iteratively remove noise from x_t to recover x_{t-1} effectively reversing the forward process.



DiT III: Dataset

Data Preprocess

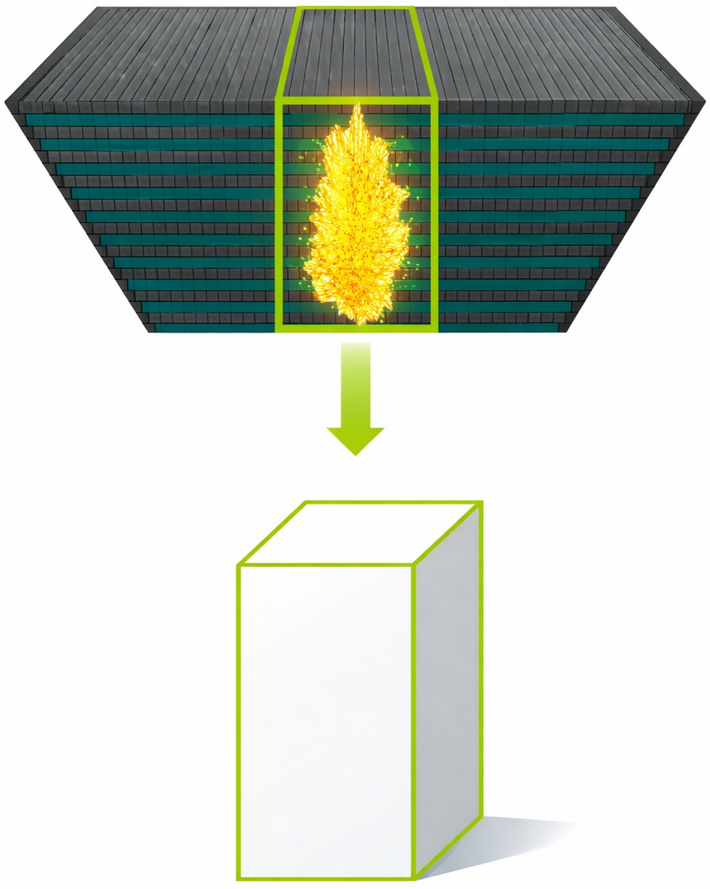
- Geant4 photon simulations based on CEPC Ref-TDR geometry.
- Segmented with DD4hep into a 15×15×18 voxel grid (15 mm cube size).
- Log-transformed and normalized voxel energies.

$$\hat{x}_i = \frac{\log(x_i + \epsilon) - \mu}{\sigma} \quad \text{where}$$

$$\mu = \mathbb{E}[\log(x_i + \epsilon)]$$

$$\sigma = \sqrt{\mathbb{E}[(\log(x_i + \epsilon) - \mu)^2]}, \quad \text{and}$$

$$\epsilon = 10^{-7}\ GeV$$

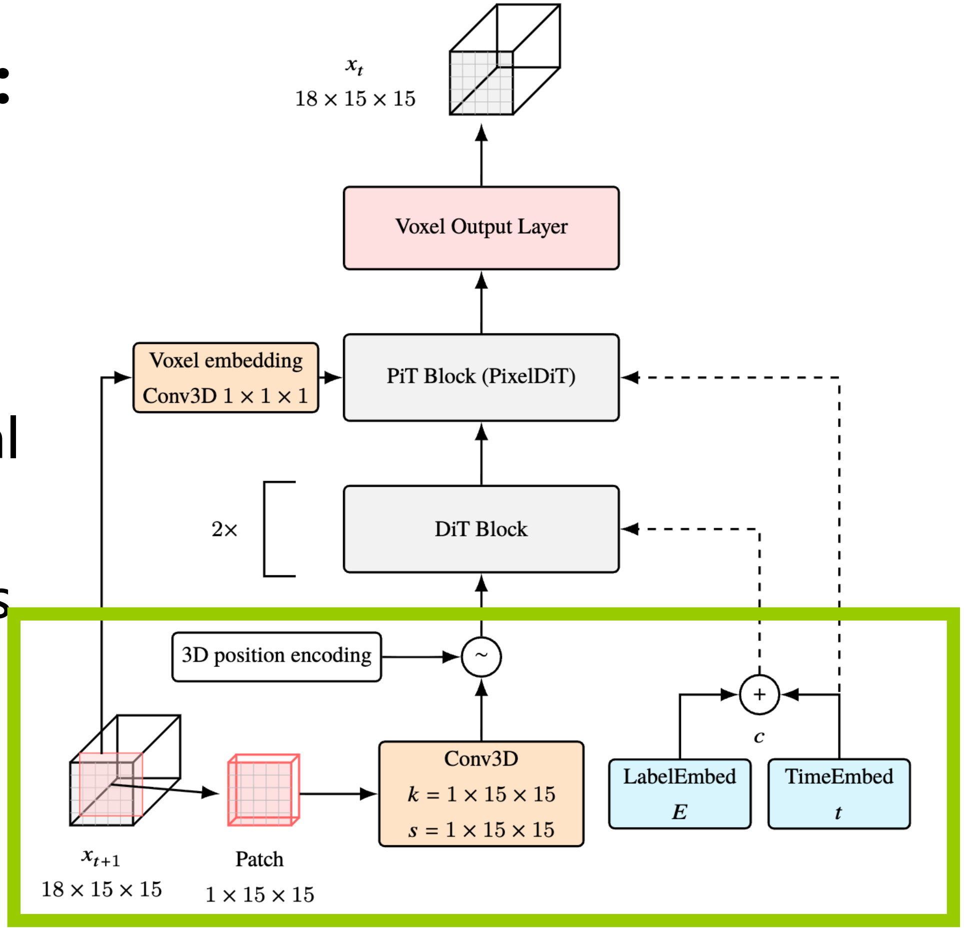


Area of interest selected as dataset

DiT IV: Model Architecture

VoDiT4CAL Architecture:

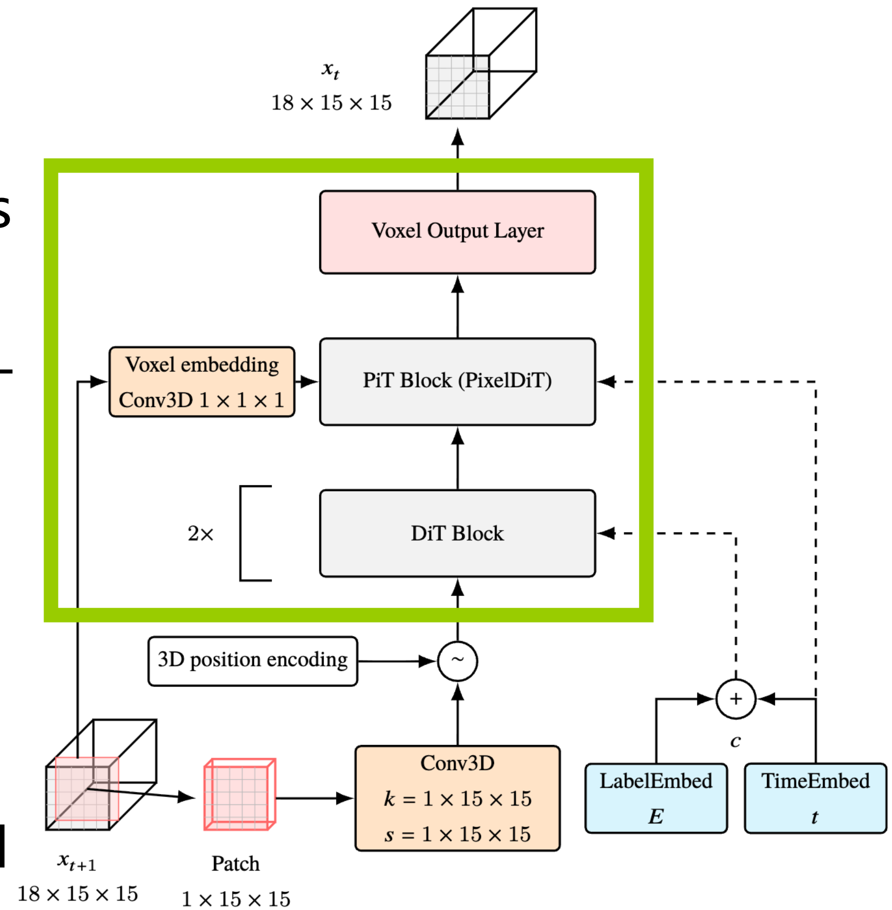
- **Patch Embedding**: Use 3D convolutions to split the ECAL volume into small patches, mapping each patch to a high-dimensional latent token.
- **Conditioning**: Each token is augmented with relative positional encoding, noise timestep t encoding, total energy E particle type, etc.



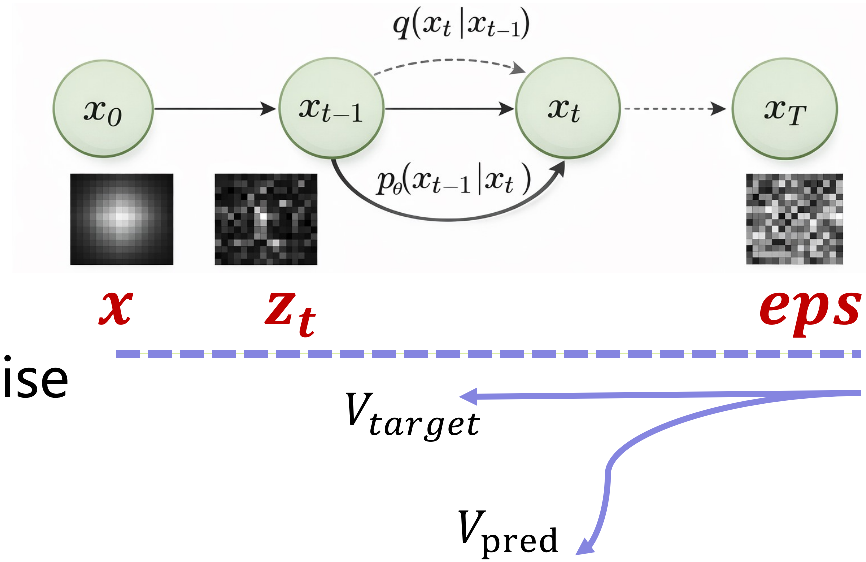
DiT IV: Model Architecture

VoDiT4CAL Architecture:

- **DiT Blocks:** Transformer self-attention blocks exchange information across tokens globally.
- **PiT Blocks**: Transformer self-attention blocks exchange information within each token across voxels.
- **Voxel Output**: The final token representations are mapped back to energy space to produce voxel-level predictions.



DiT V: Training



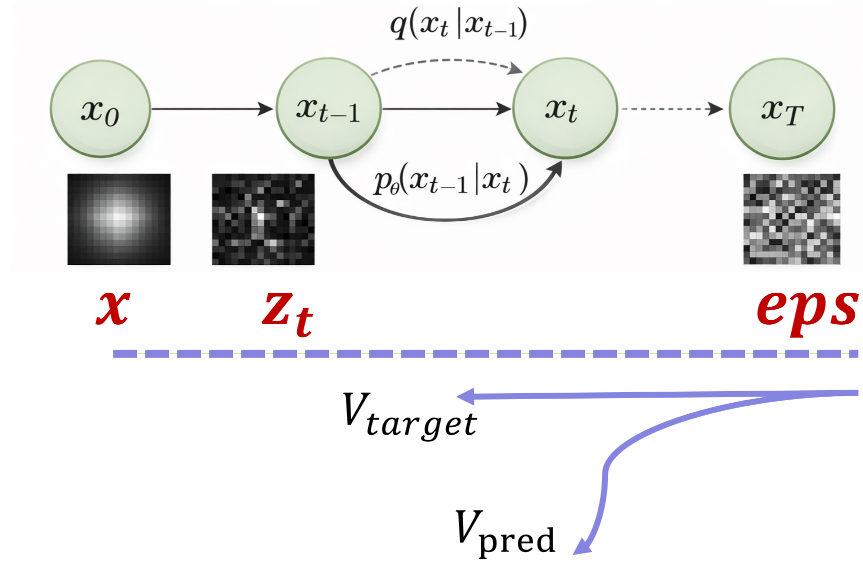
Training Steps:

- Sample a random timestep $t \sim \mathcal{U}(0,1)$.
- Construct a noisy state z_t by mixing data x with Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$.
$$z_t = (1-t)\epsilon + tx$$
- Use v-parameterization with target
$$v_{\text{target}} = \frac{x - z_t}{1-t}$$
$$v_{\text{pred}} = f_\theta(z_t, t, \text{cond}),$$
$$\text{cond} = E,\ \text{particle type}, \ldots$$
- Optimize with MSE loss:
$$\mathcal{L} = \text{MSE}\left(v_{\text{pred}}, v_{\text{target}}\right)$$

```
t = sample_t()
eps = randn_like(x) * noise_scale
z_t = (1 - t) * eps + t * x

v_target = (x - z_t) / (1 - t)
v_pred = model(z_t, t, E_cond)
loss = mse(v_pred, v_target)
```

DiT V: Training

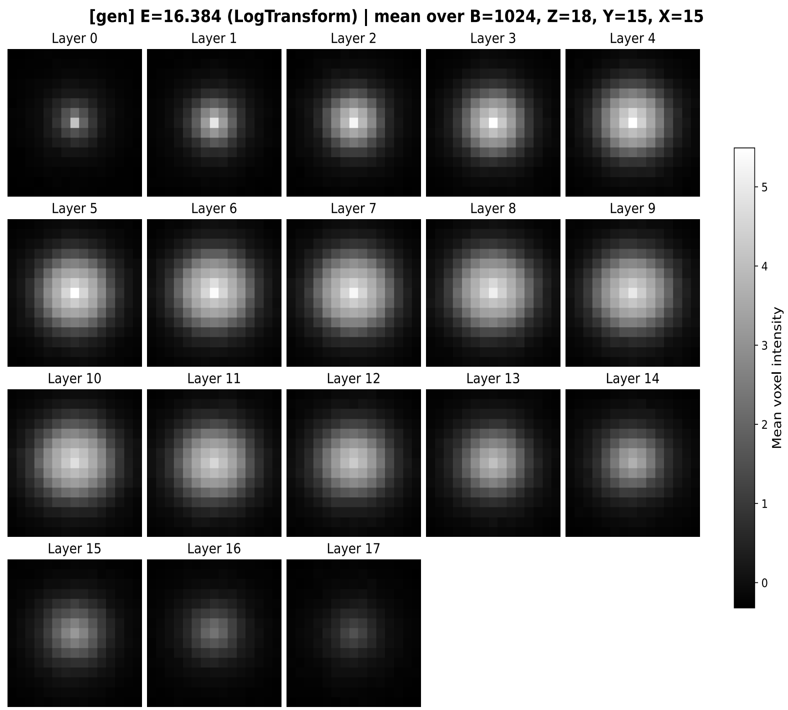
- **Training:**
  - **PyTorch** provides tensor computation and automatic differentiation, while **PyTorch Lightning** offers a high-level training framework.
  - **Training Time:** The total effective training time is **3h23m19s** on a single RTX 5090 GPU.



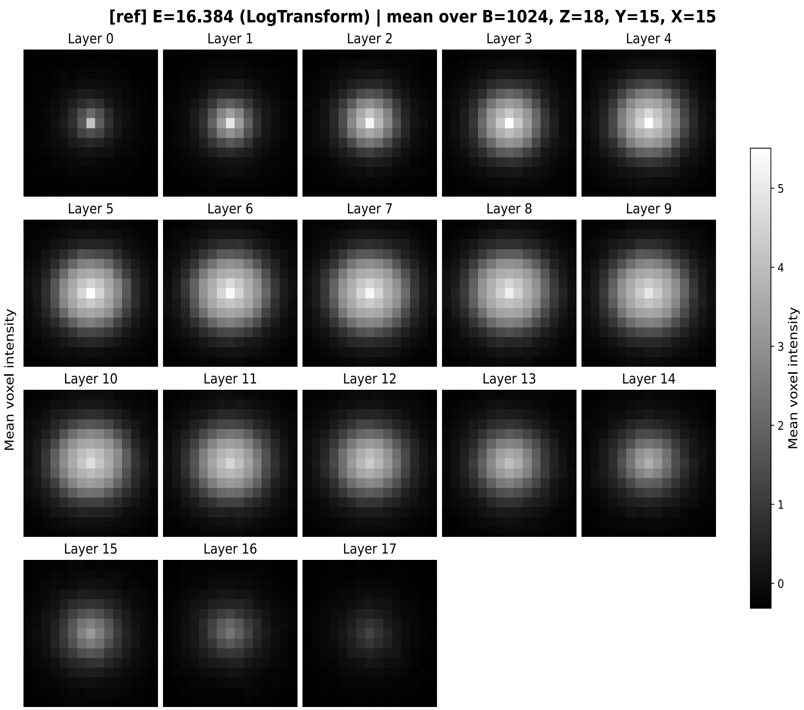

```
t = sample_t()
eps = randn_like(x) * noise_scale
z_t = (1 - t) * eps + t * x

v_target = (x - z_t) / (1 - t)
v_pred = model(z_t, t, E_cond)
loss = mse(v_pred, v_target)
```

Result I: Generation Quality

- Generate photon shower at **16 GeV** for example



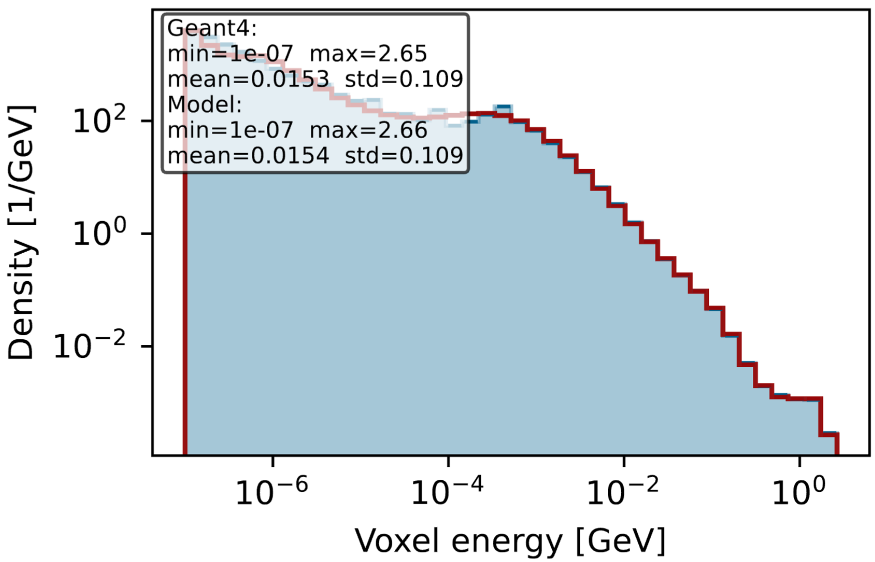
GEANT4

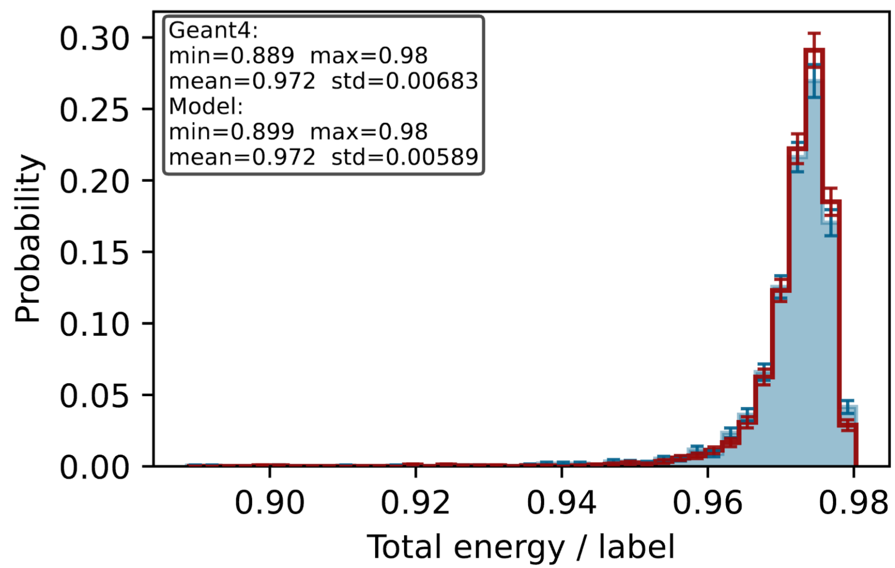


MODEL

Result I: Generation Quality

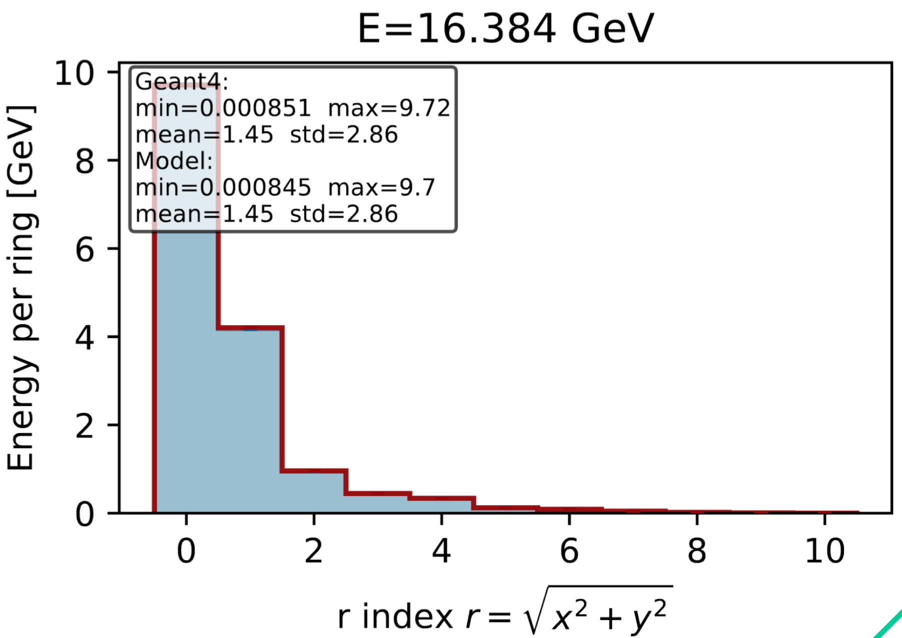
❖ Overview

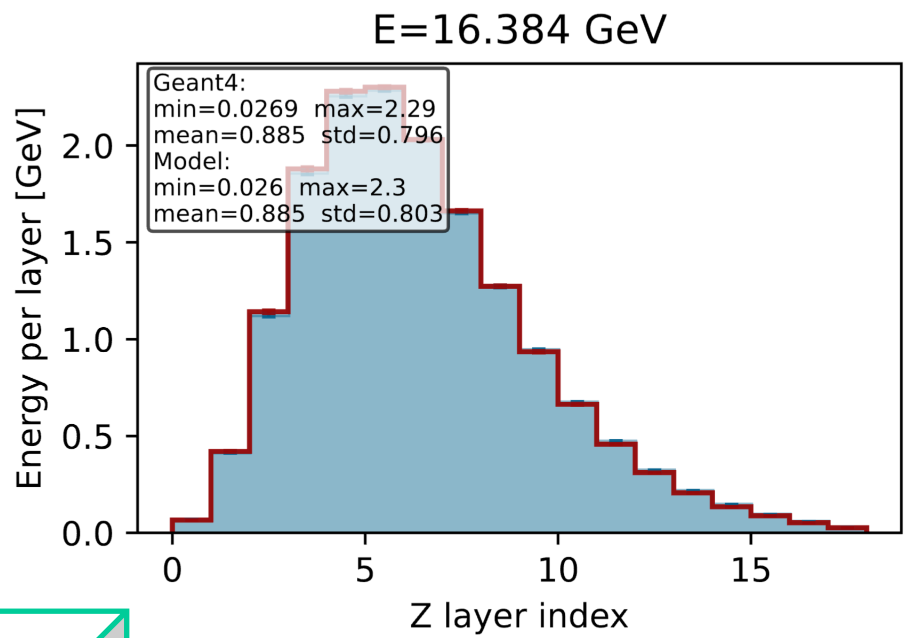


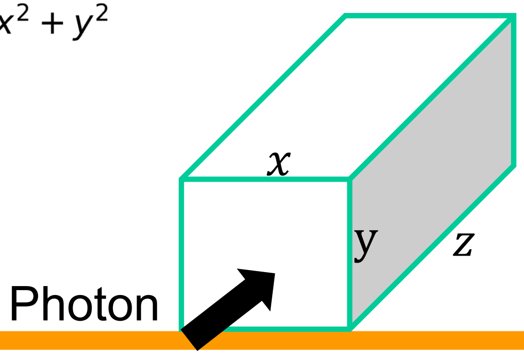


Result I: Generation Quality

- Geometry

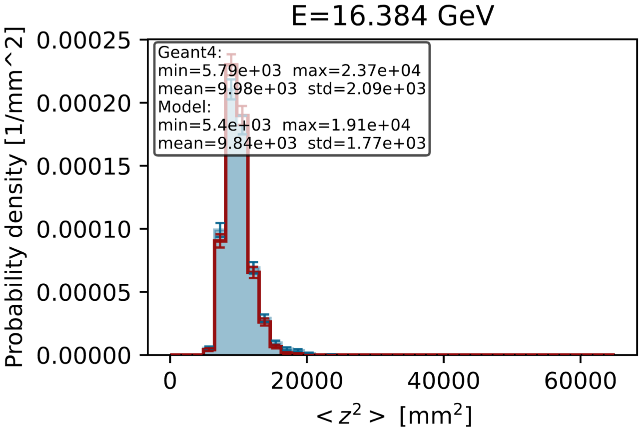


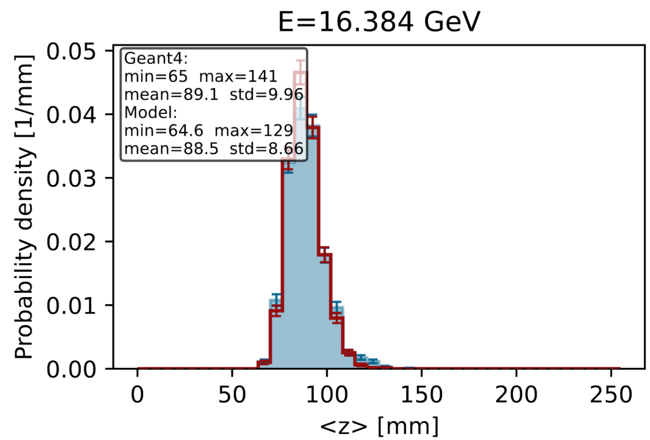




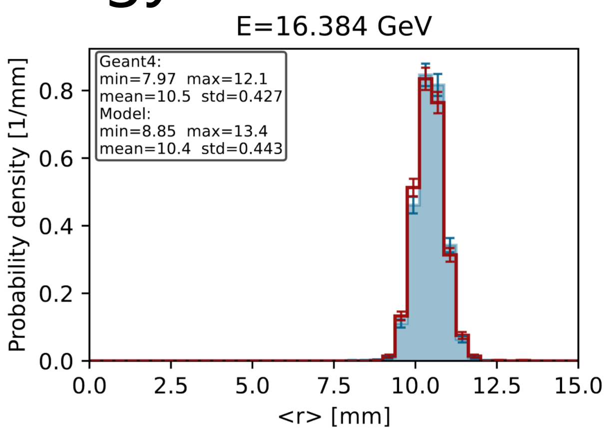
Result I: Generation Quality

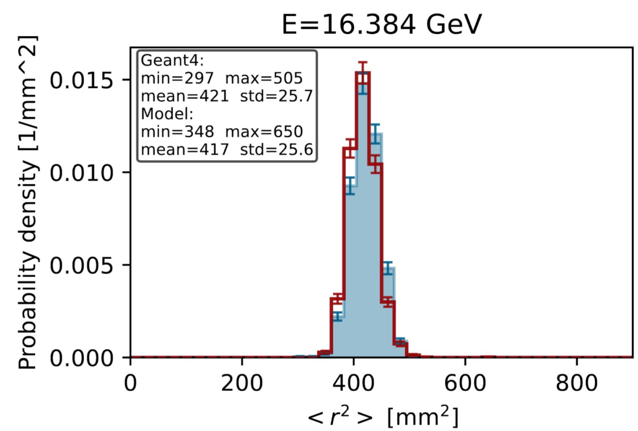
- Z energy center





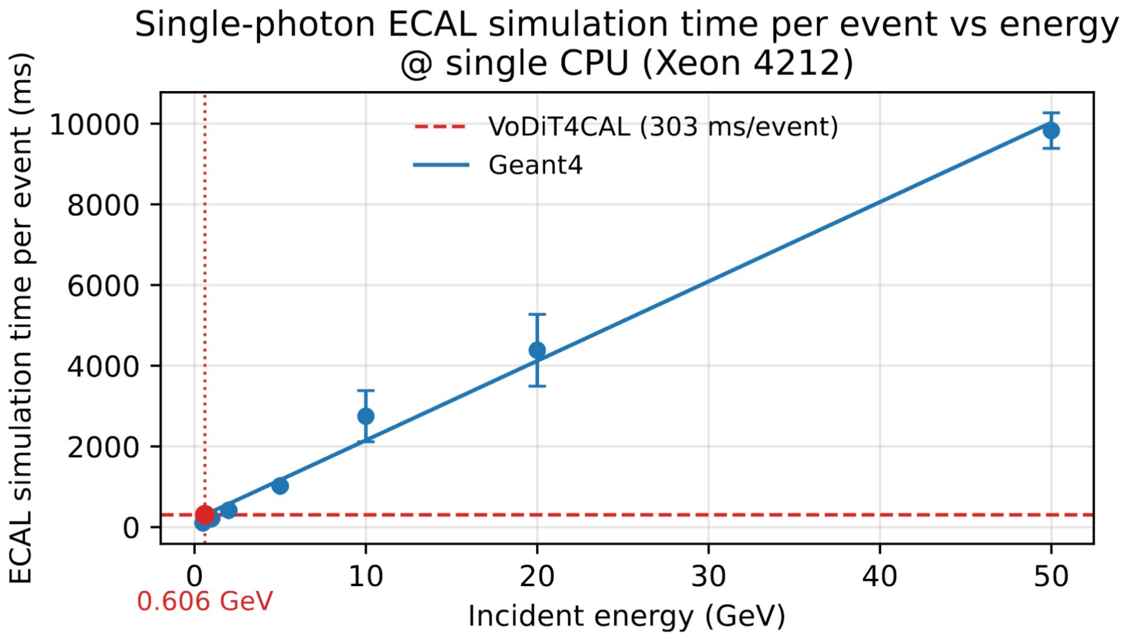
- R energy center





Result II: Generation Speed

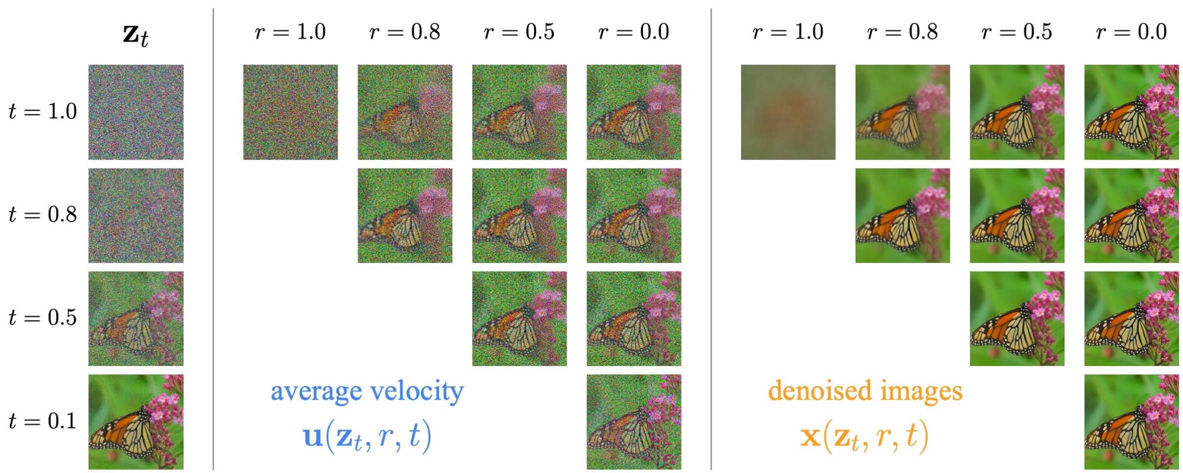
- Benchmark:
  - GPU (Nvidia RTX 8000): **3.19 ms /event**
  - CPU (Xeon 4214, single thread): **303.45 ms /event**
  - Geant4: generation time $T \sim E$ with particle energy E

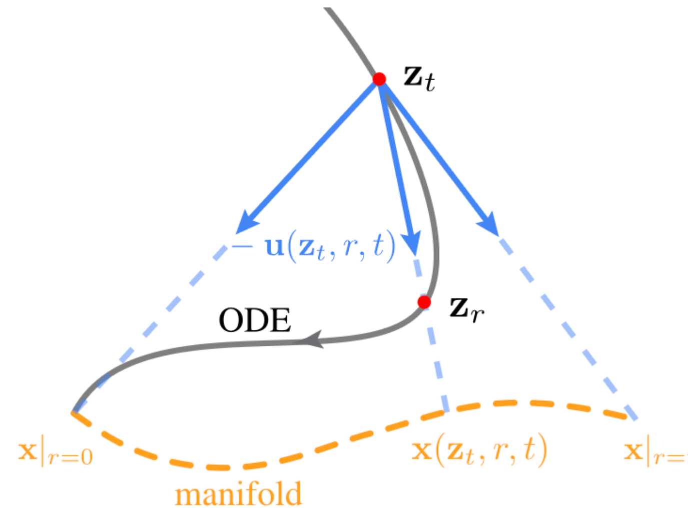


WIP I: Distillation-based Acceleration

- **Pixel Mean Flow:** A **one-step**, latent-free generative model that maps noise directly to output.
  - Only one-step generation
  - Quality decreases
  - 10x speedup

$$\mathbf{x}(\mathbf{z}_t, r, t) \triangleq \mathbf{z}_t - t \cdot \mathbf{u}(\mathbf{z}_t, r, t).$$

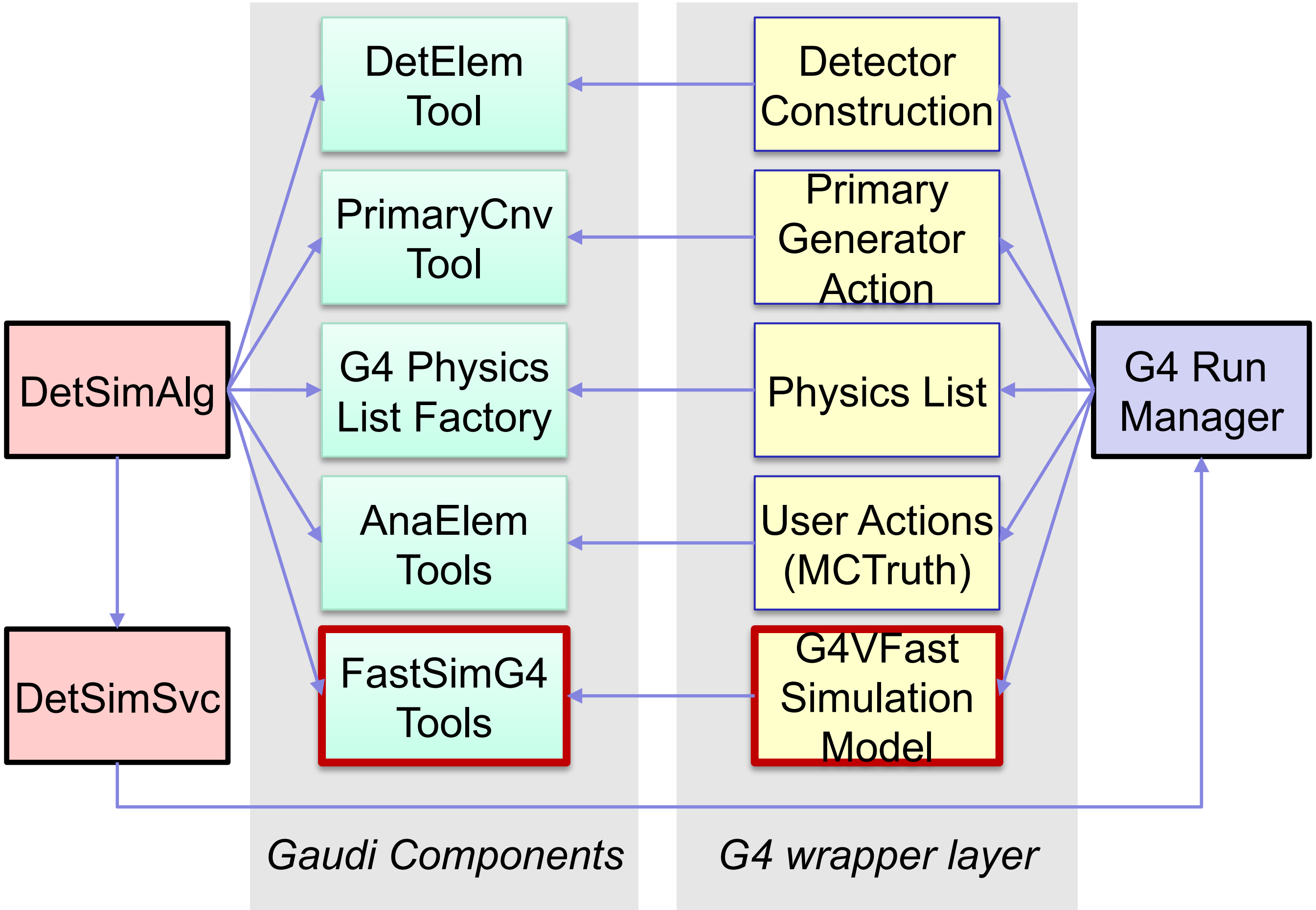




WIP II: Future Plan

- **One-Step Distillation**:
  - Model currently uses **multi-step diffusion (16 steps)**.
  - Enable **single-step generation** with Pixel Mean Flow (pMF).
- **Deployment & Integration:**
  - Develop interfaces for **deployment within Geant4 workflows**.
- **Multi-Particle & Incident States:**
  - Handle **different particle types and incoming states** (e.g., electrons, photons, hadrons) with a unified model.
- **Cross-Detector Generalization:**
  - Strengthen **generalization across different detector** designs and experiments.

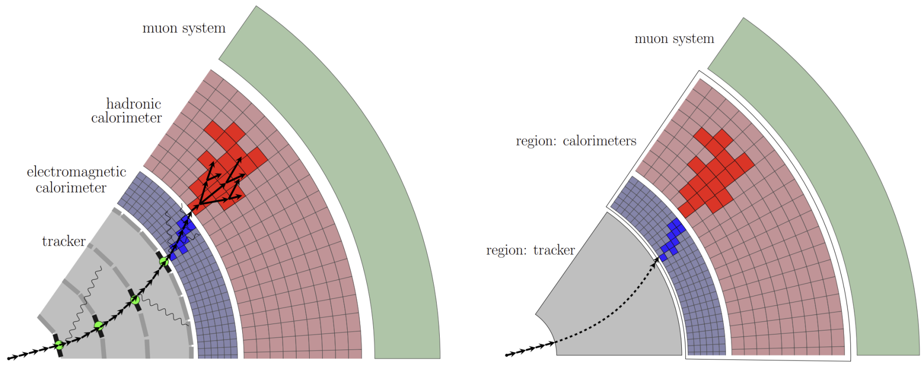
Integration of fast simulation (I)

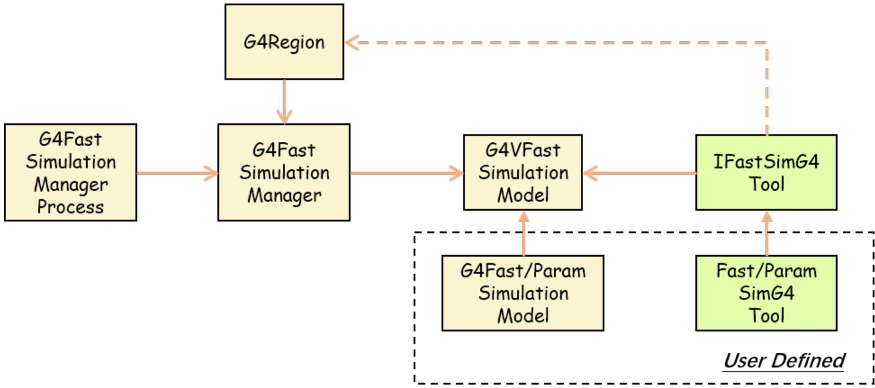


Within CEPCSW, a lightweight simulation framework has been developed to enable seamless integration of Geant4 into Gaudi framework

Integration of fast simulation (II)

- Region based fast simulation is adopted
  - When a particle enter a region, fast simulation will be triggered by Geant4.
  - Machine learning inference could be integrated via ONNX.





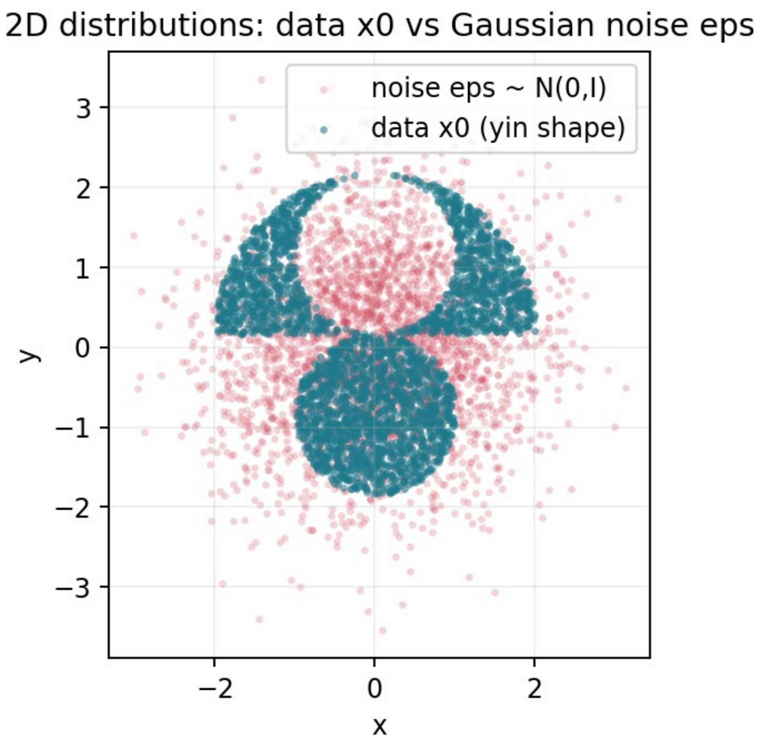
A Zaborowska 2017 J. Phys.: Conf. Ser. 898 042053

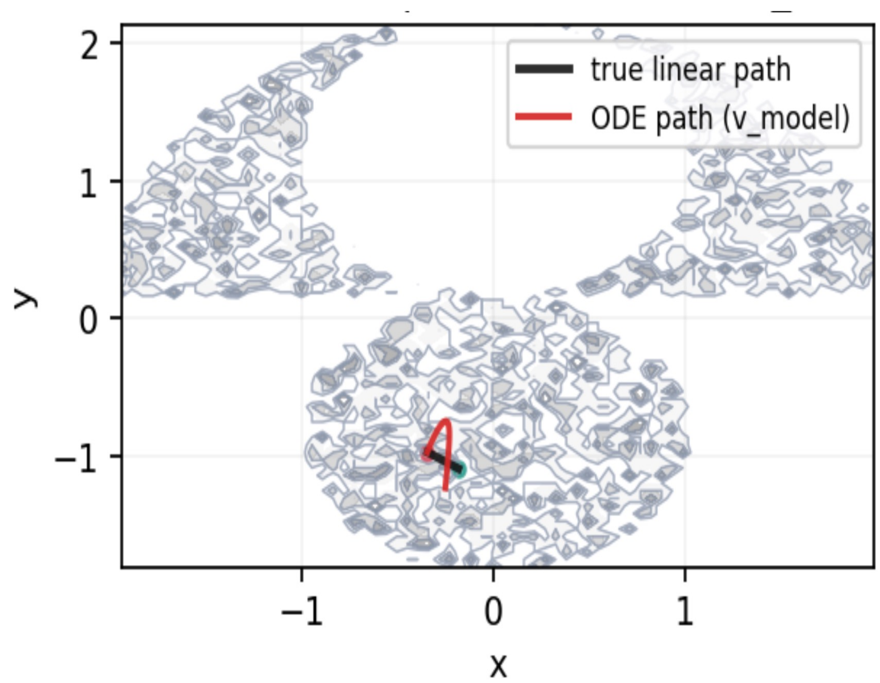
Summary

- VoDiT4CAL achieves high-precision fast simulation of the CEPC crystal ECAL at **~100ms** level.
  - While preserving the quality of showers and key physical observables, it delivers approximately a **100× speedup** for 50 GeV events on a single CPU core.
  - This provides a practical solution to **mitigate computing bottlenecks** and paves the way for larger-scale physics studies.
  - Will apply distillation techniques to **reduce inference time** while maintaining generation quality
  - Will extend to **multi-particle** and **more complex incident** conditions
  - Will ultimately **integrate it into CEPCSW framework** to effectively replace the Geant4 ECAL simulation module

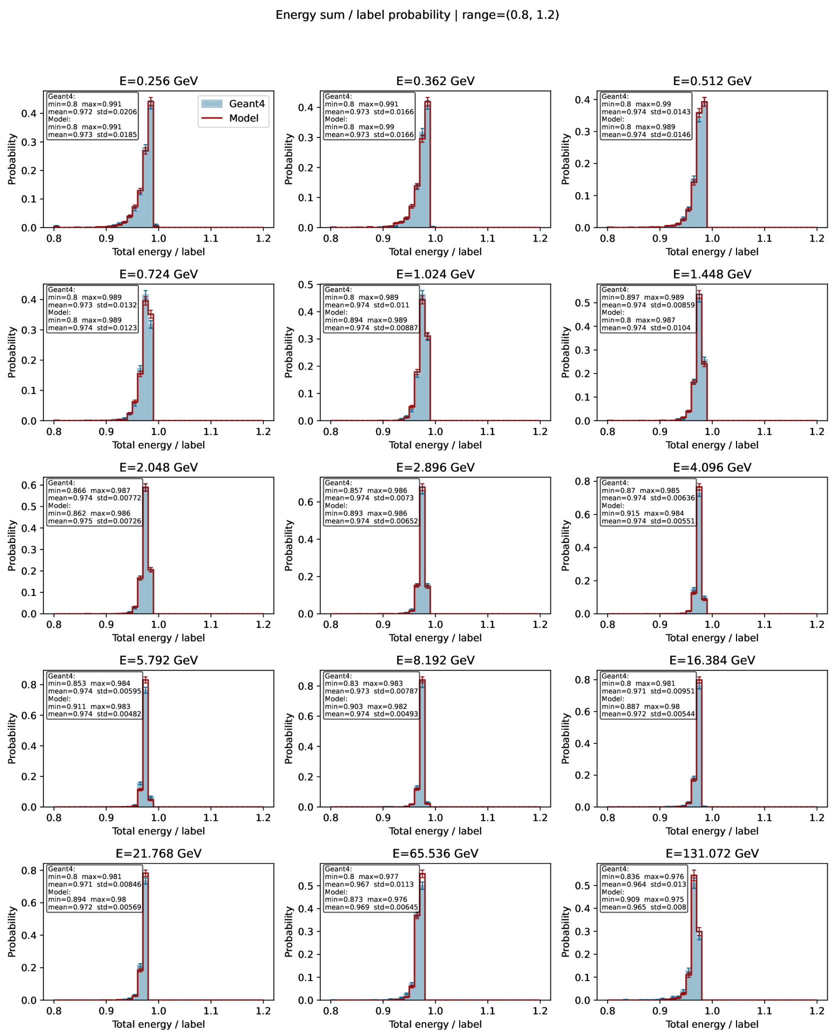
Thank you!

TOY Data Demo

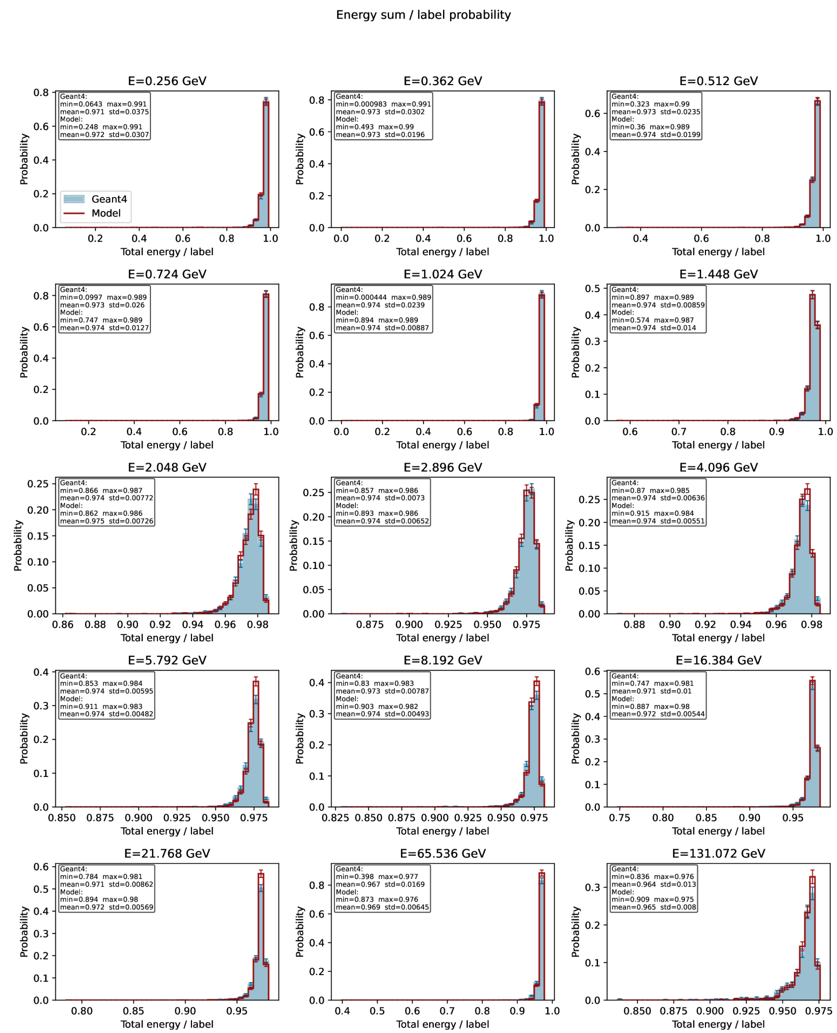




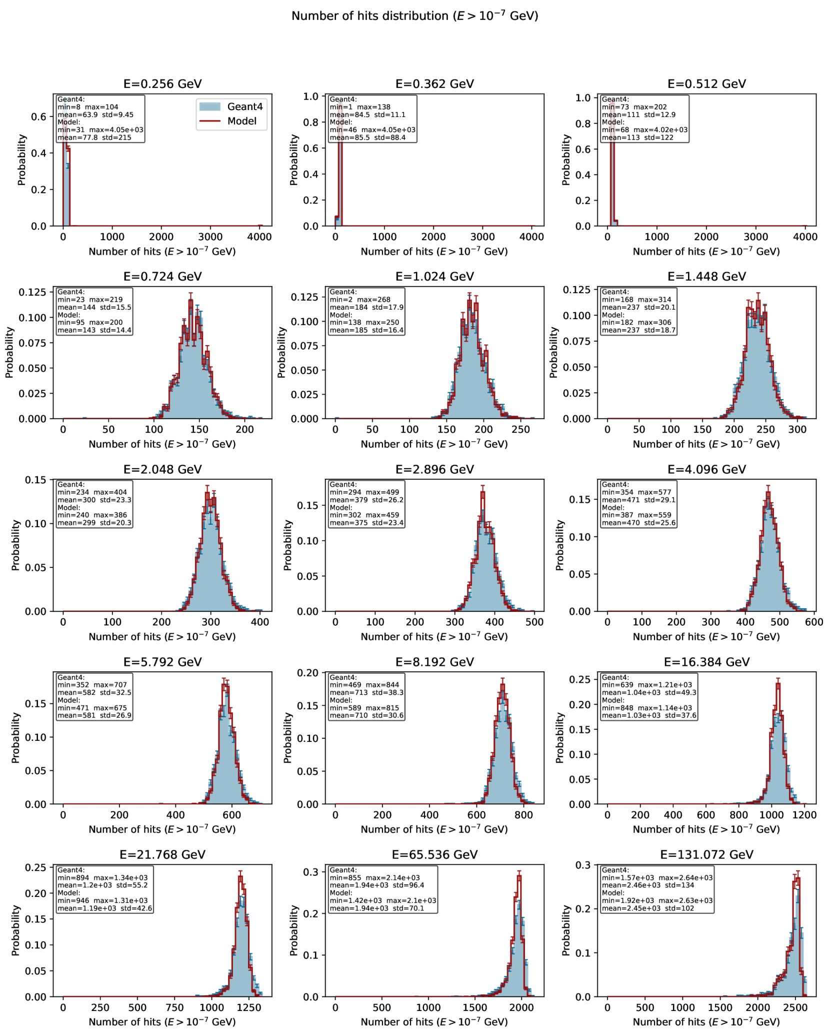
SUM

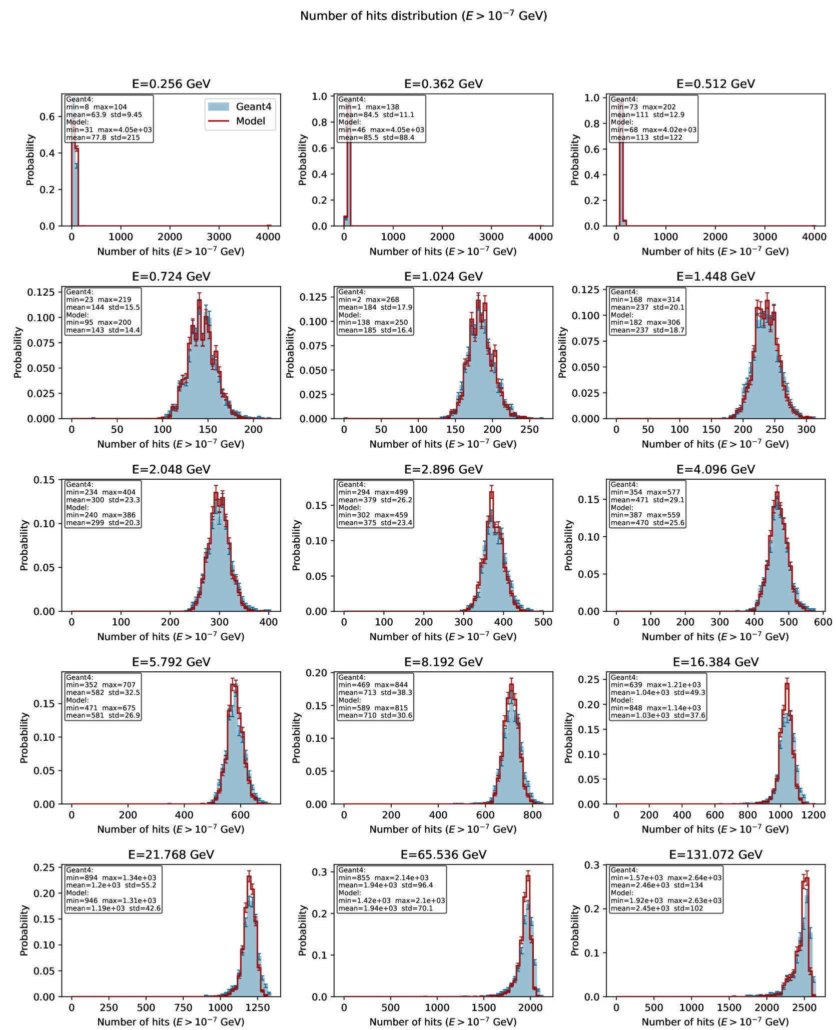


Energy sum / label probability | range=(0.8, 1.2)
E=0.256 GeV
E=0.362 GeV
E=0.512 GeV
E=0.724 GeV
E=1.024 GeV
E=1.448 GeV
E=2.048 GeV
E=2.896 GeV
E=4.096 GeV
E=5.792 GeV
E=8.192 GeV
E=16.384 GeV
E=21.768 GeV
E=65.536 GeV
E=131.072 GeV
Geant4
Model
Probability
Total energy / label

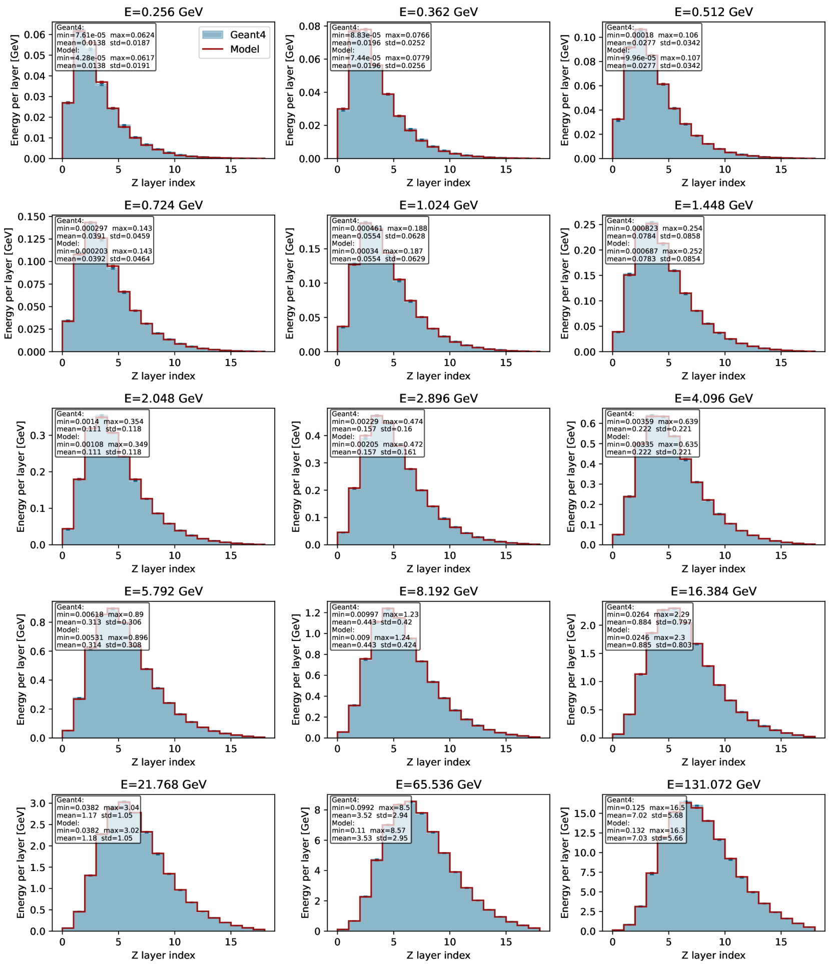


Energy sum / label probability
E=0.256 GeV
E=0.362 GeV
E=0.512 GeV
E=0.724 GeV
E=1.024 GeV
E=1.448 GeV
E=2.048 GeV
E=2.896 GeV
E=4.096 GeV
E=5.792 GeV
E=8.192 GeV
E=16.384 GeV
E=21.768 GeV
E=65.536 GeV
E=131.072 GeV
Geant4
Model
Probability
Total energy / label

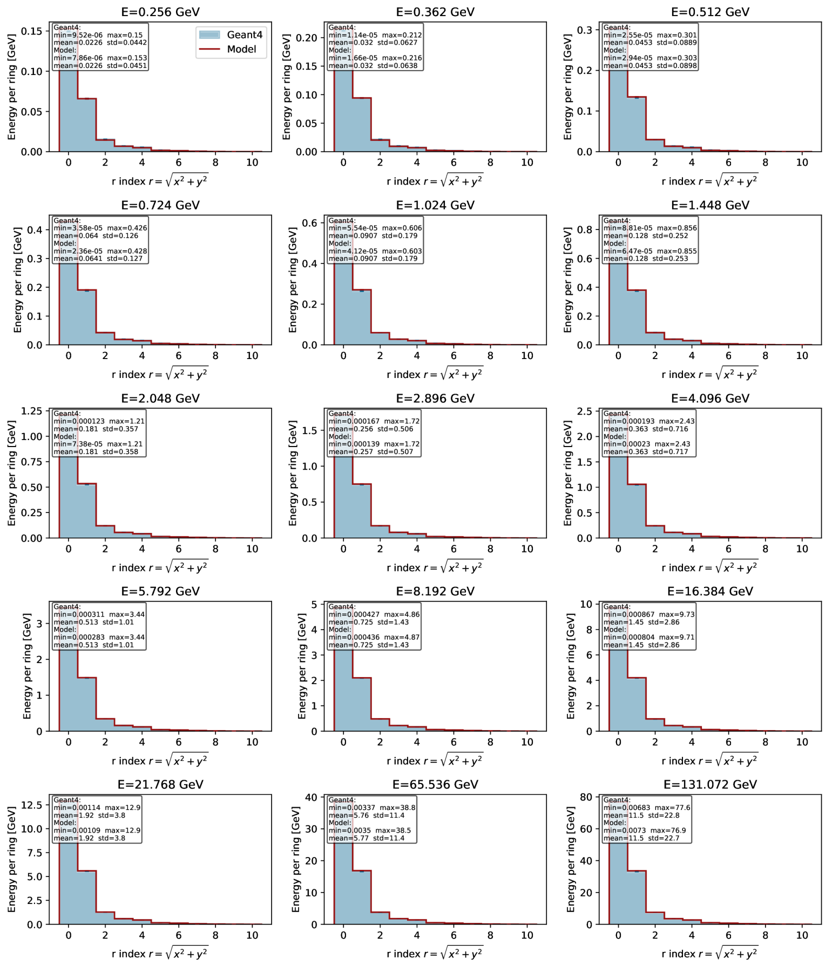

Number of hits distribution ($E > 10^{-7}$ GeV)
E=0.256 GeV
E=0.362 GeV
E=0.512 GeV
E=0.724 GeV
E=1.024 GeV
E=1.448 GeV
E=2.048 GeV
E=2.896 GeV
E=4.096 GeV
E=5.792 GeV
E=8.192 GeV
E=16.384 GeV
E=21.768 GeV
E=65.536 GeV
E=131.072 GeV
Geant4
Model
Probability
Number of hits ($E > 10^{-7}$ GeV)


Number of hits distribution ($E > 10^{-7}$ GeV)
E=0.256 GeV
E=0.362 GeV
E=0.512 GeV
E=0.724 GeV
E=1.024 GeV
E=1.448 GeV
E=2.048 GeV
E=2.896 GeV
E=4.096 GeV
E=5.792 GeV
E=8.192 GeV
E=16.384 GeV
E=21.768 GeV
E=65.536 GeV
E=131.072 GeV
Geant4
Model
Probability
Number of hits ($E > 10^{-7}$ GeV)

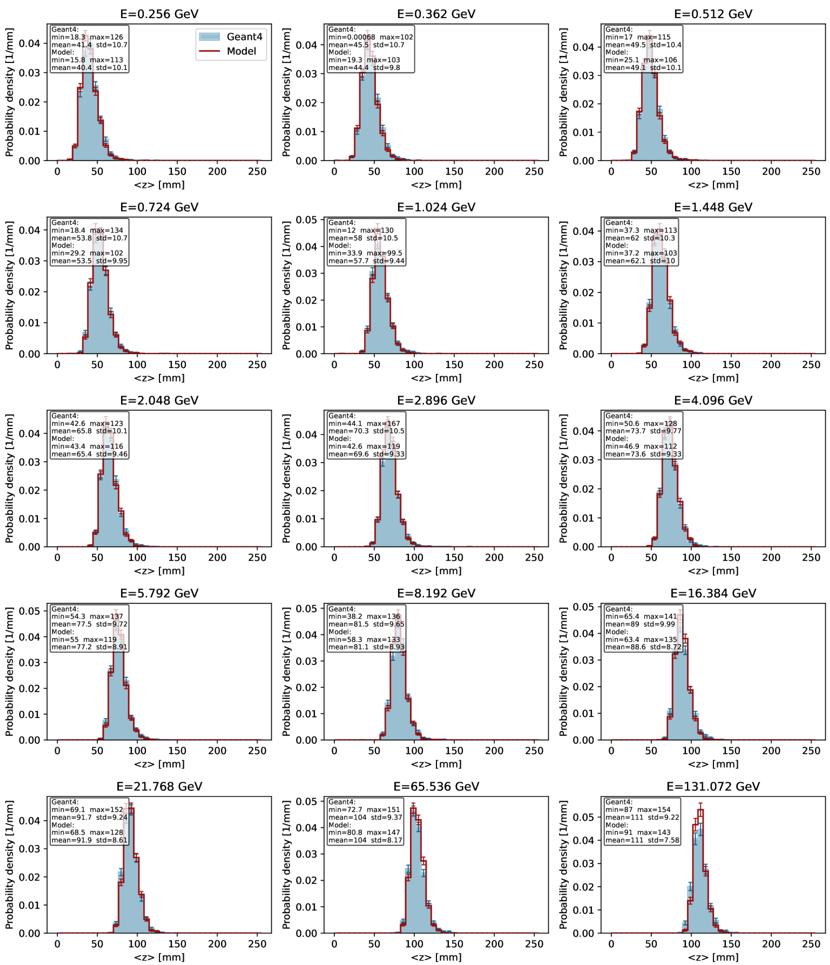
Mean energy per Z layer [GeV]



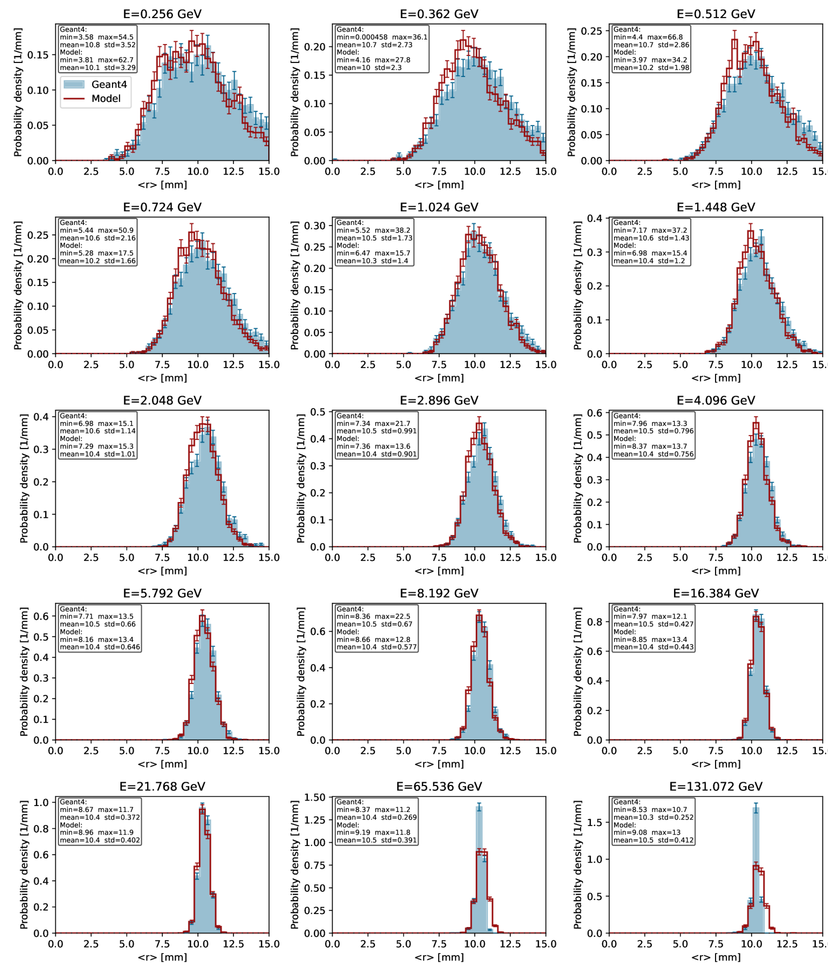
Mean energy vs radial index [GeV]



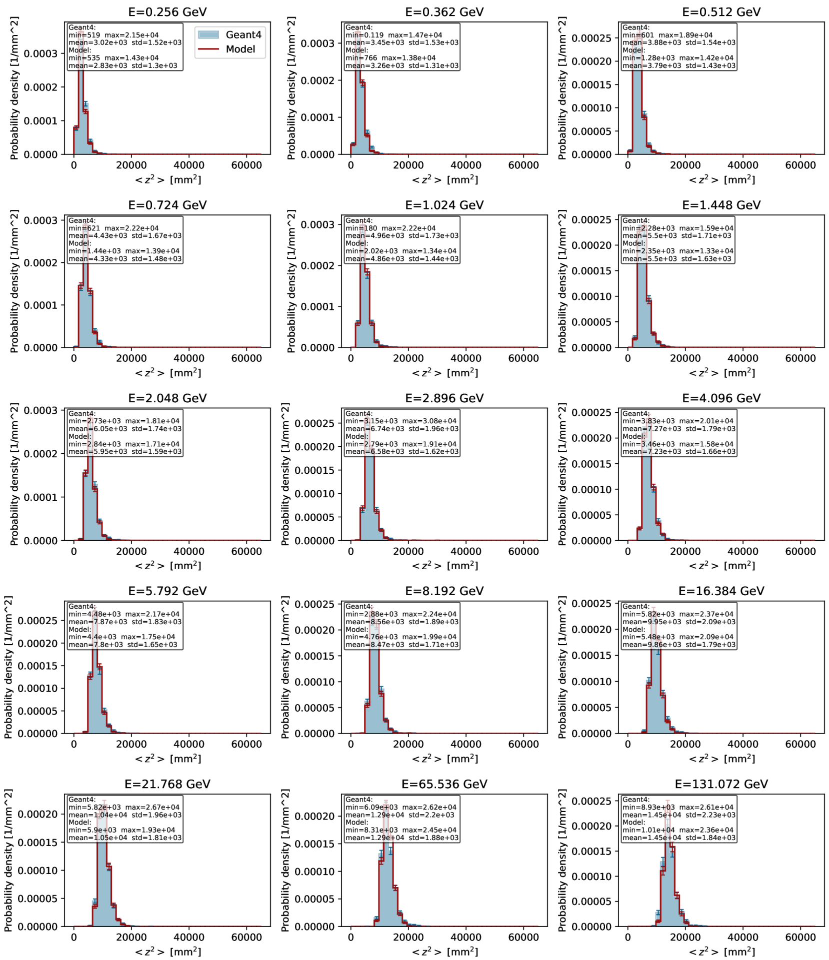
Energy-weighted Z barycenter [mm]



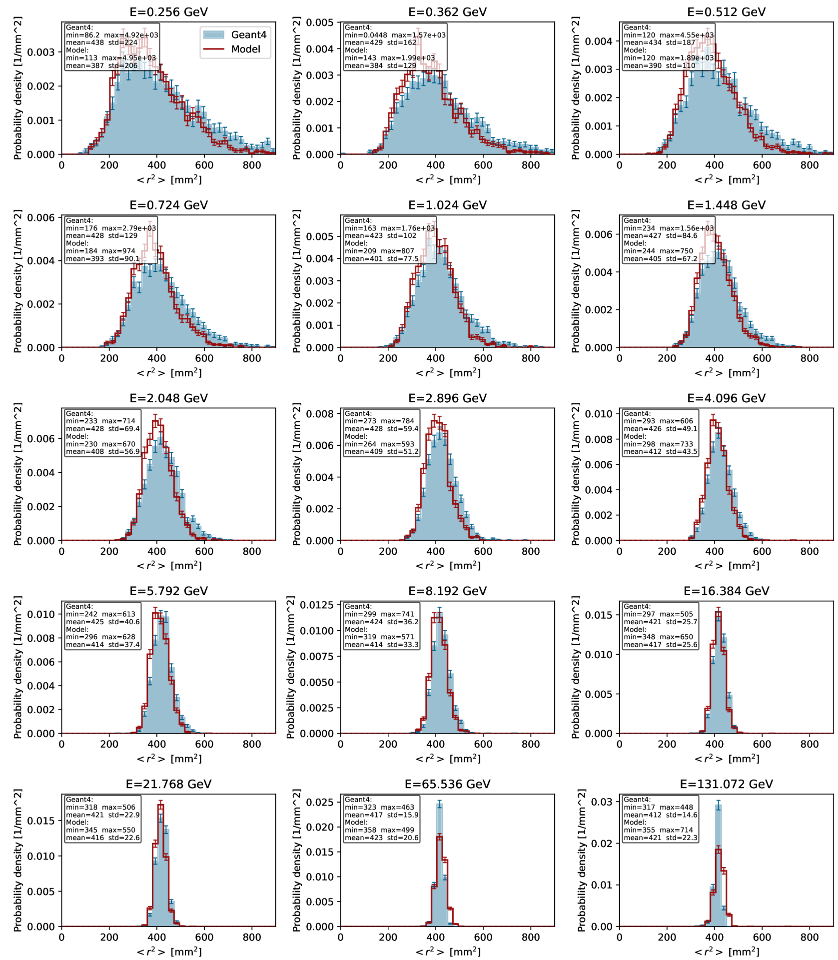
Energy-weighted radial barycenter [mm]



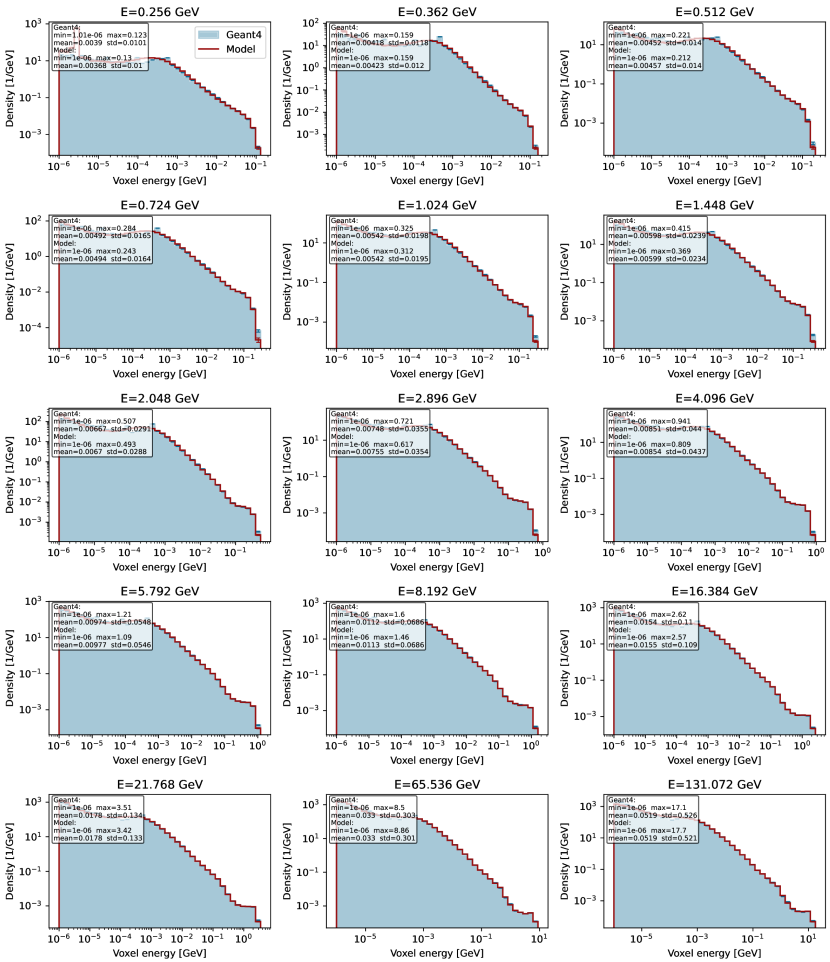
Energy-weighted second moment along Z [mm^2]



Energy-weighted second moment in XY [mm^2]



Voxel energy density | combined range | log y | log x



Voxel energy density (ALRTransform) | combined range | log y