BERKELEY LAB

NERSC

Diffusion Generative Models for EIC Simulations

vmikuni@lbl.gov

vinicius-mikuni

Vinicius M. Mikuni

Generative Models

1 2 3 4 5
6 7 8 9 10

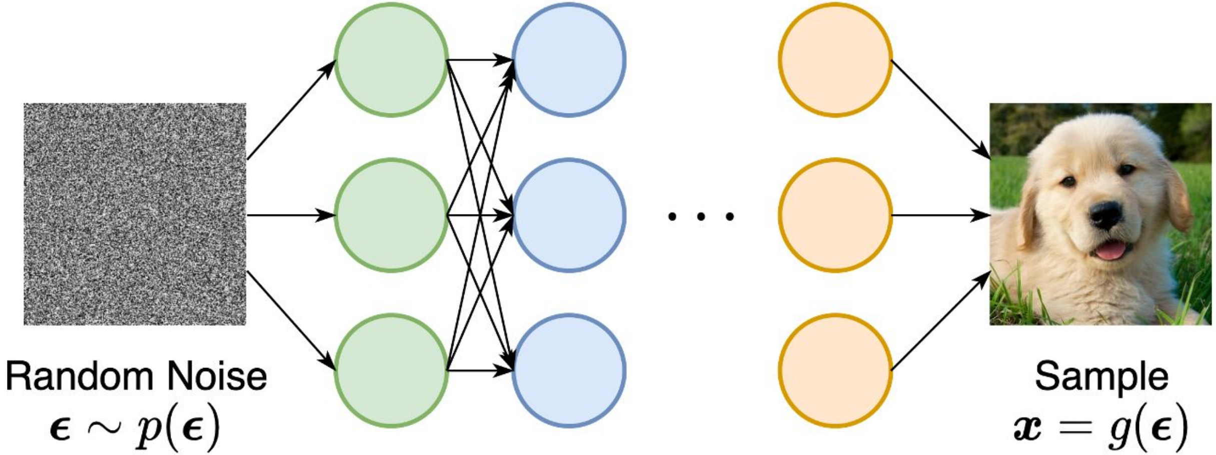
- Which of these people you think are **AI generated?**

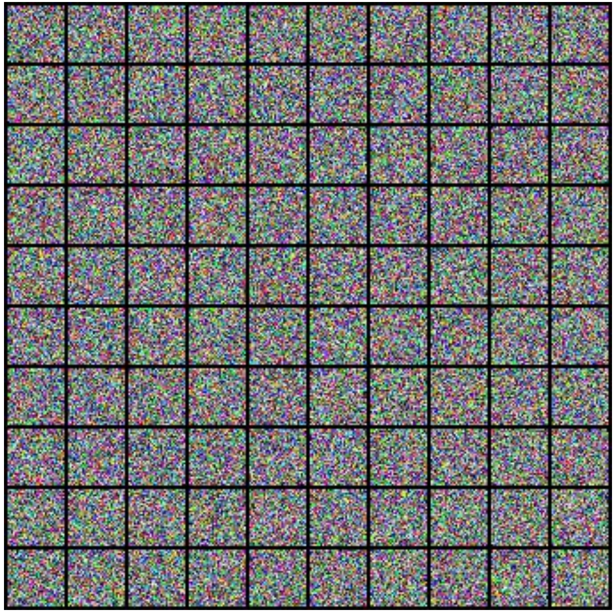
Generative Models

1 2 3 4 5
6 7 8 9 10

- Which of these people you think are **AI generated?**
- **Answer: All of them** https://generated.photos/faces

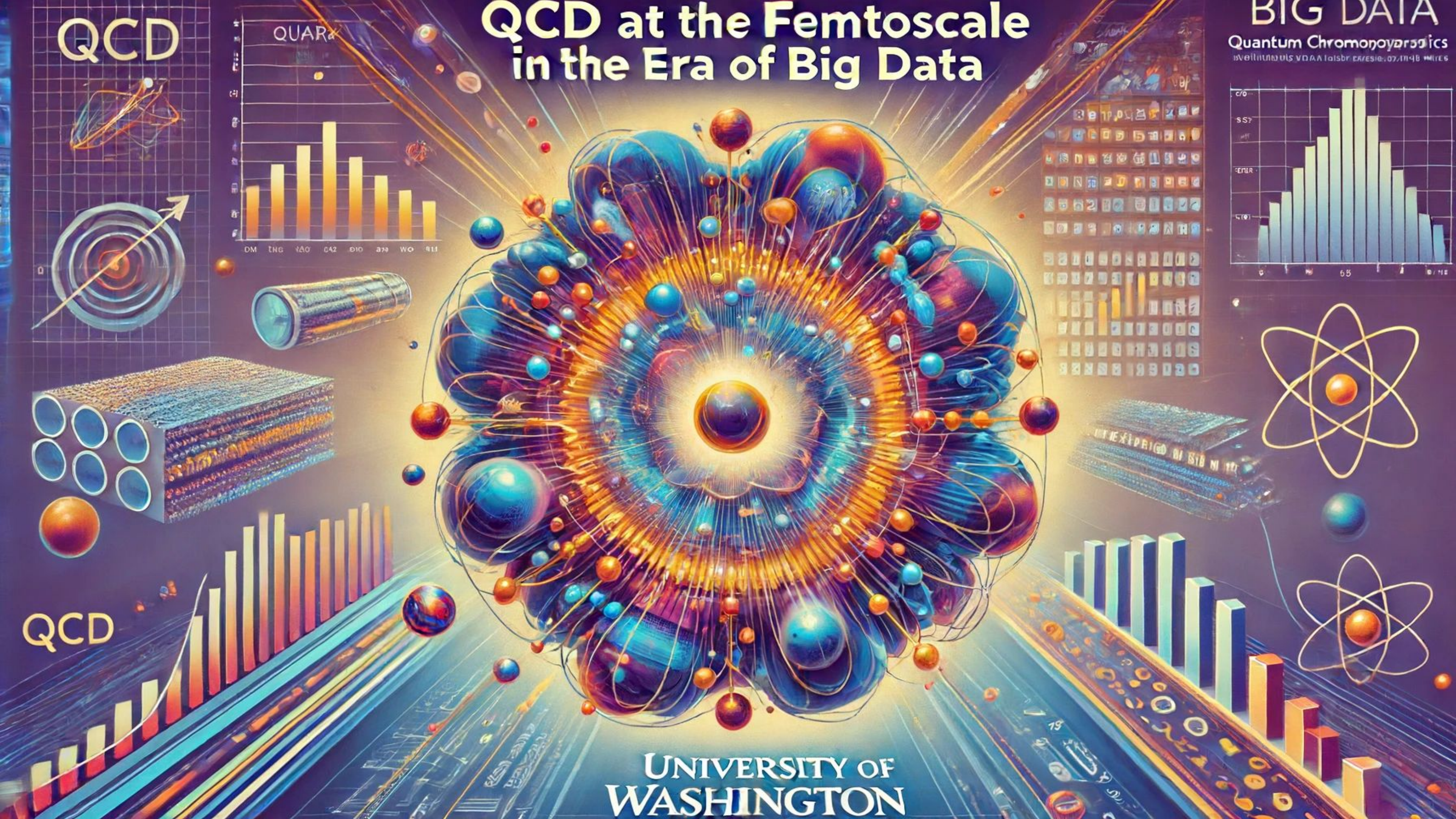
Generative models





Source: https://yang-song.net/blog/2021/score/

Generative models are a class of algorithms trained to transform easy-to-sample noise into data


QCD at the Femtoscale
in the Era of Big Data
QCD
BIG DATA
QCD
UNIVERSITY OF
WASHINGTON

Diffusion models

A Mechanical Model of Brownian Motion

Communications in Mathematical Physics
© Springer-Verlag 1981

D. Dürr*, S. Goldstein**, and J. L. Lebowitz***

Department of Mathematics, Rutgers University, New Brunswick, NJ 08903, USA

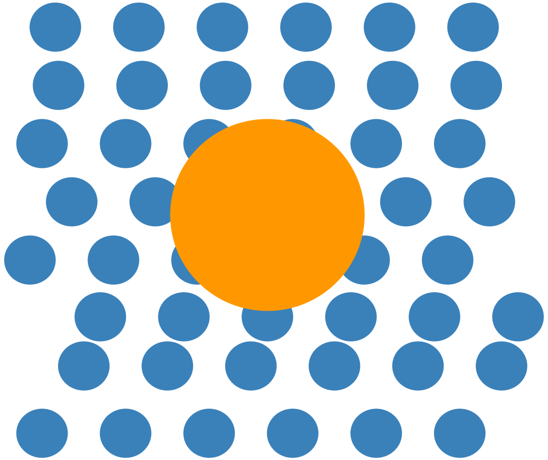
Abstract. We consider a dynamical system consisting of one large massive particle and an infinite number of light point particles. We prove that the motion of the massive particle is, in a suitable limit, described by the Ornstein-Uhlenbeck process. This extends to three dimensions previous results by Holley in one dimension.

The ultimate mathematical idealization of this phenomenon is the Ornstein-Uhlenbeck process for the position and velocity of the Brownian particle $(\underline{X}_t, \underline{V}_t)$, described by the stochastic differential equations

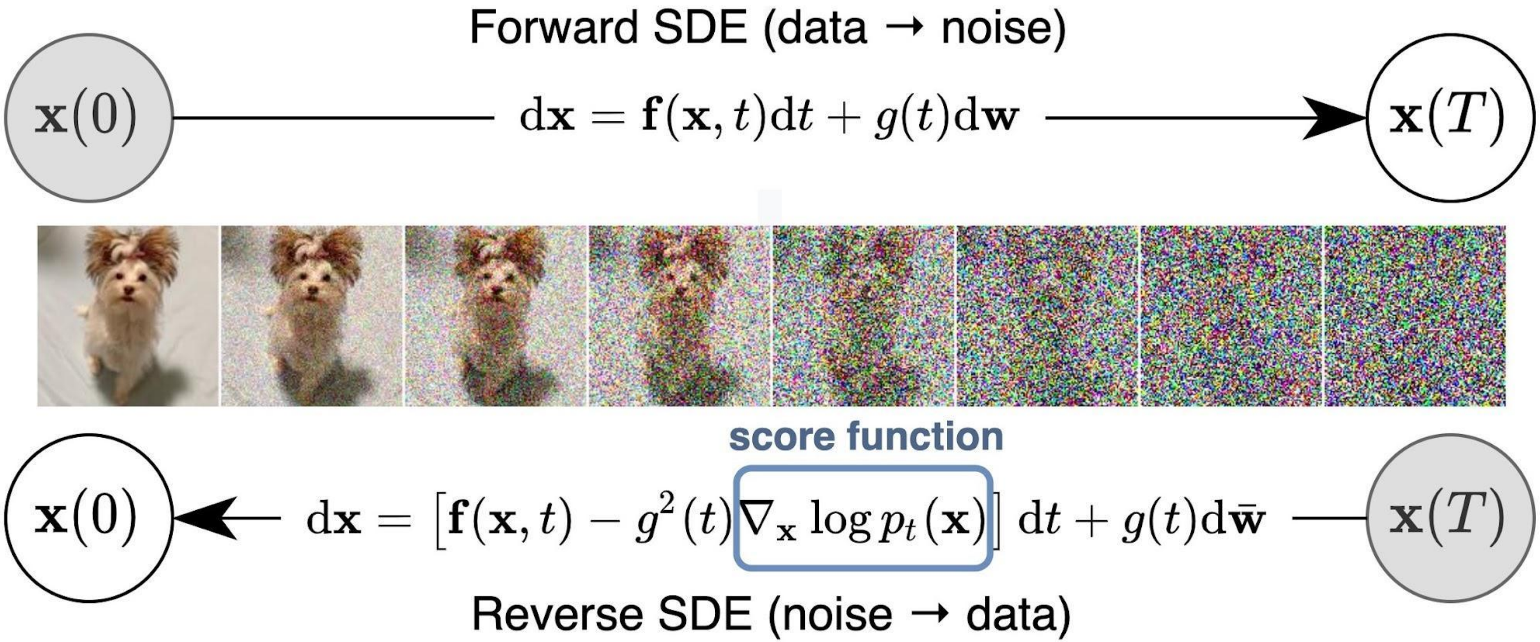
$$d\underline{X}_t = \underline{V}_t dt\,, \qquad (0.1)$$

$$d\underline{V}_t = -a\underline{V}_t dt + \sqrt{D}\, d\underline{W}_t\,, \quad a \geqq 0\,, \quad D \geqq 0\,, \quad \underline{W}_t = \text{Wiener process}. \qquad (0.2)$$

The position process $\underline{X}_t$ converges in an appropriate limit (e.g. $a \to \infty$, $a^2/D = \text{const}$) to a Wiener process.



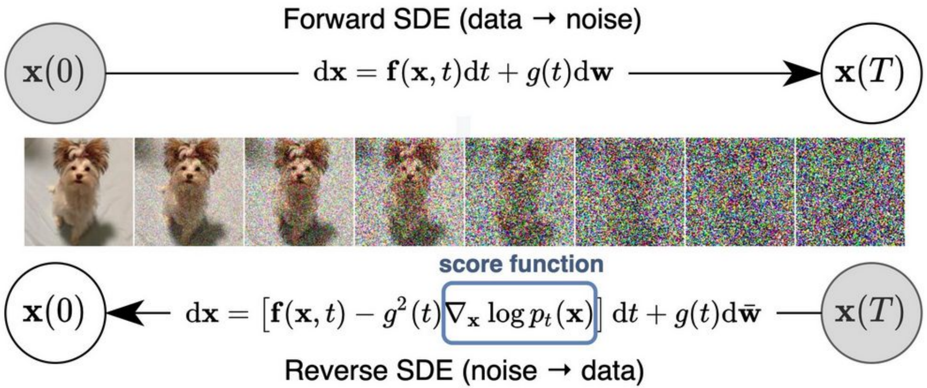
Diffusion Generative Models



Source: https://yang-song.net/blog/2021/score/

Generation

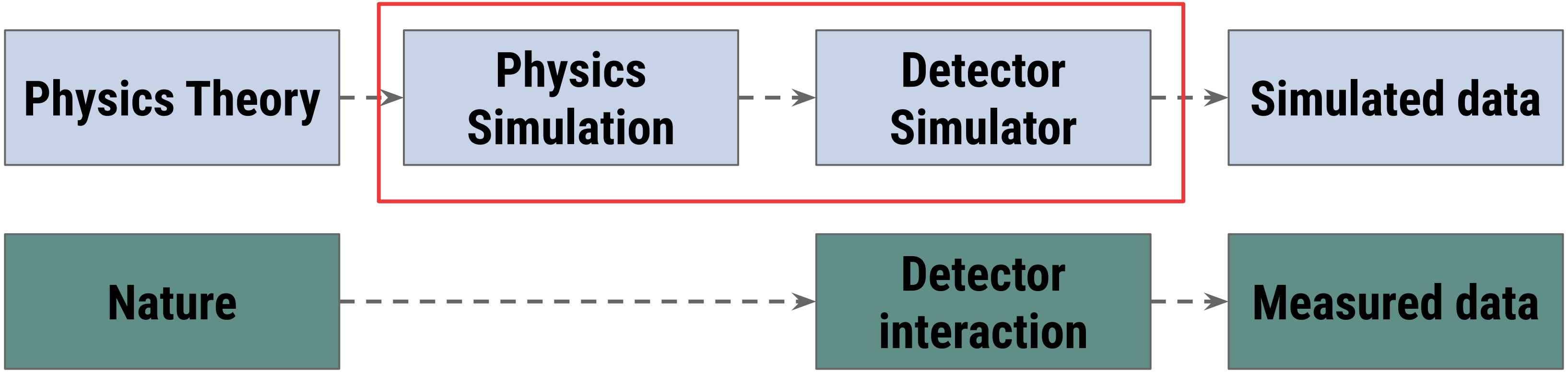
- Langevin dynamics is used to draw samples from **p(x)** using only the **score function**
- High fidelity samples require small time steps,



$$\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\, \mathbf{z}_i, \quad i = 0, 1, \cdots, K,$$

Fast Detector Simulation

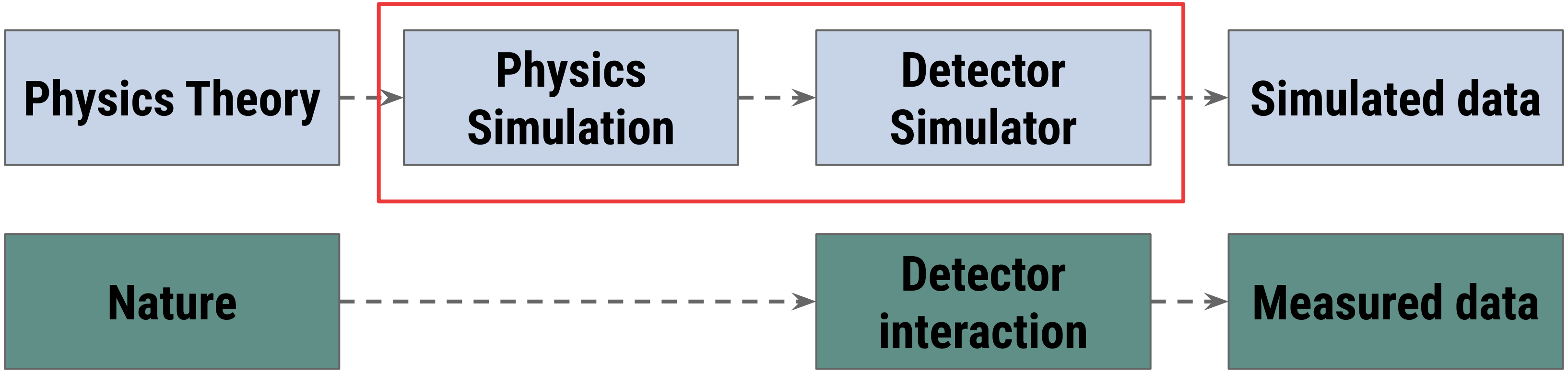
Introduction



We can only compare our physics predictions with experiments through the use of **simulations**.

Full simulation chain can be computationally expensive

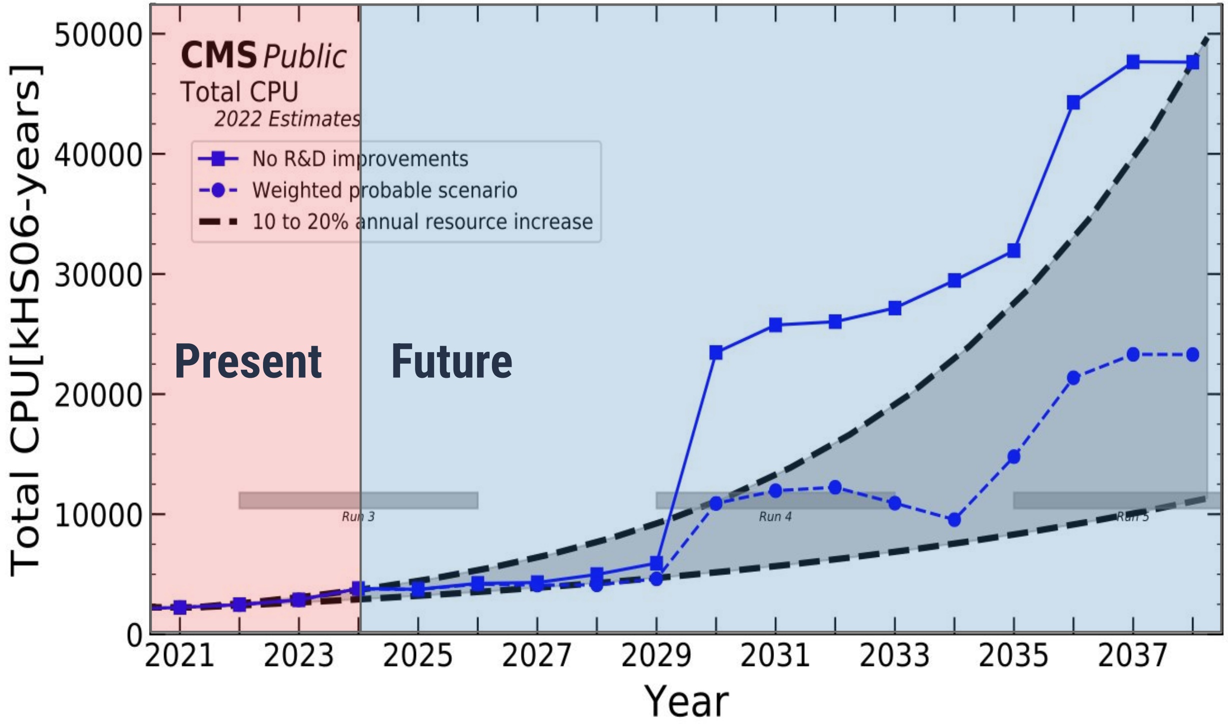
Introduction



Alternatively, **fast surrogate** models can be used to reduce the simulation time while maintaining similar level of fidelity

The Challenge

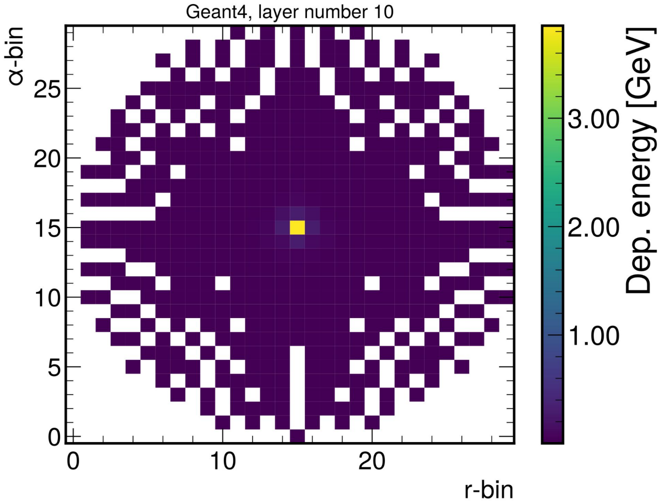
BERKELEY LAB



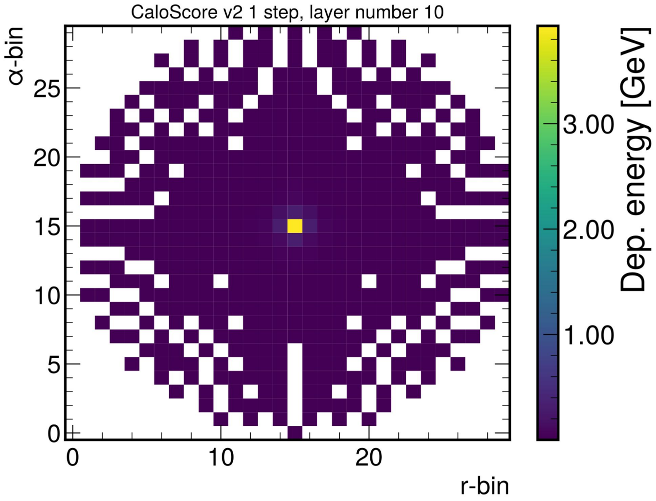
Future upgrades of the LHC experiment will aim to increase the likelihood of collisions happening, **exceeding the current computing budget**

Source: CMS-NOTE-2022-008

Diffusion Generative Models for Detector Simulation



Physics Simulation



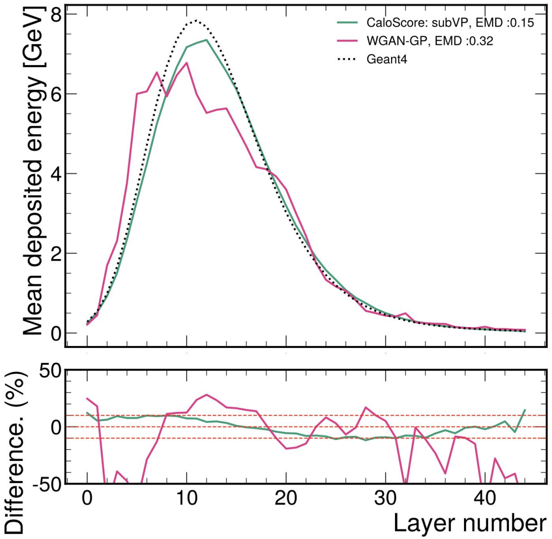
Generated by **CaloScore**

First Diffusion model in High Energy Physics named **CaloScore**.
Up to 50k Detector Components simulated

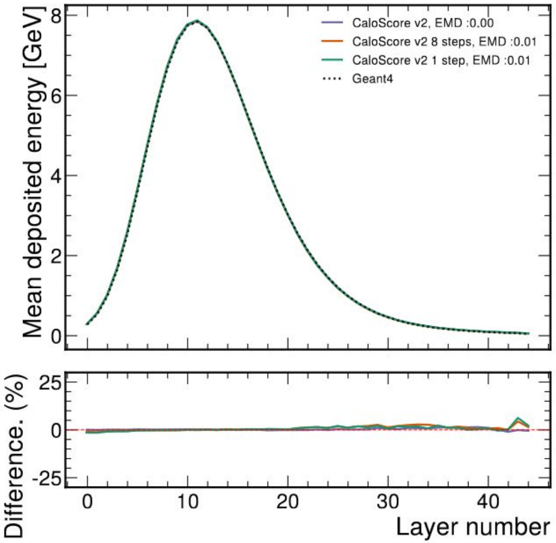
- **V. Mikuni** and B. Nachman Phys. Rev. D **106**, 092009
- **V. Mikuni** and B. Nachman 2024 *JINST* **19** P02001

Diffusion Generative Models for Detector Simulation

BERKELEY LAB



Energy deposition inferred from sum of pixels



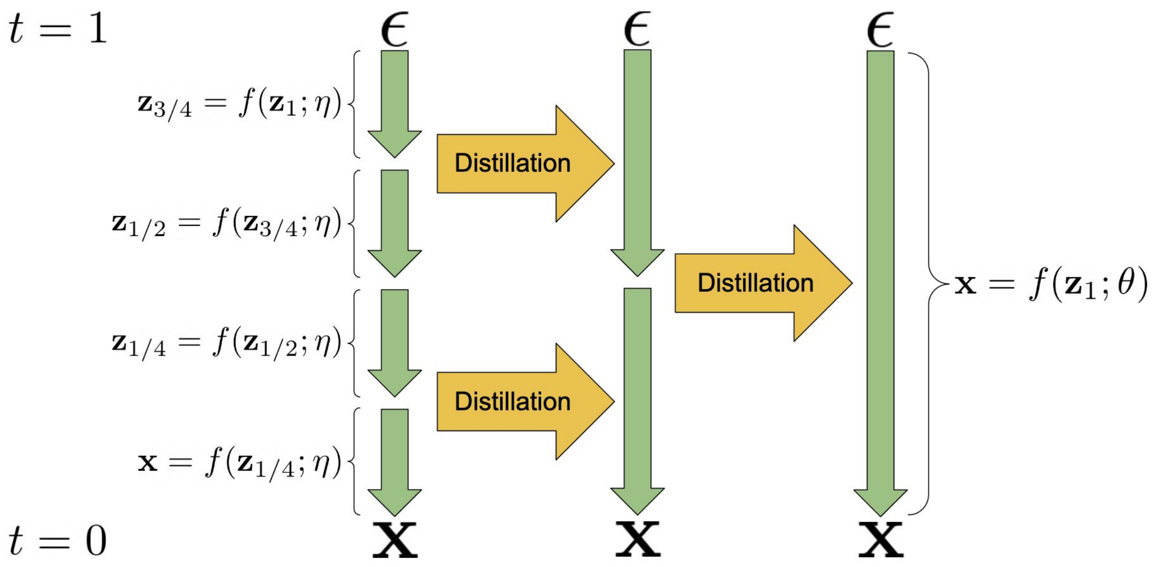
Additional model trained to learn the energy sum

Improve energy conservation by training **2 conditional diffusion models**: One on normalized pixel responses and one to determine the total energy deposition

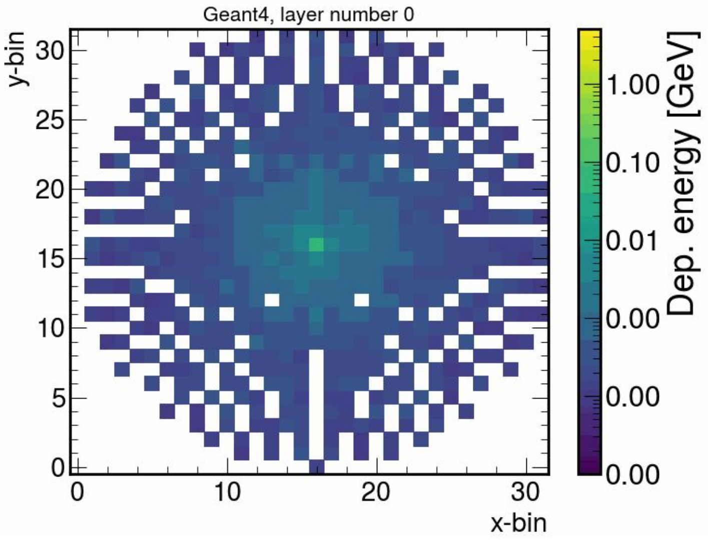
$$\nabla \log p(x_{\text{norm}}, E) = \nabla \log p(x_{\text{norm}}|E) + \nabla \log p(E)$$

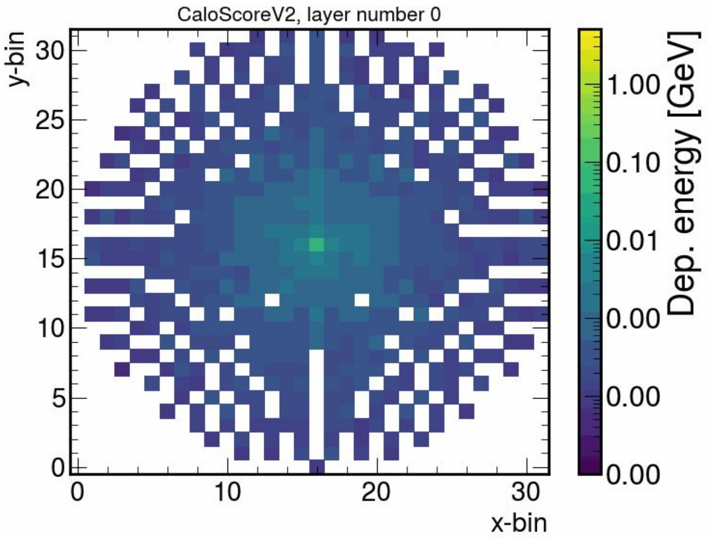
CaloScore v2

- Progressive distillation is used to iteratively **reduce the number of time steps** used during generation
- Train a follow up model that learns how to predict **2 steps at a time**
- Repeat **multiple times**



Diffusion Generative Models for Detector Simulation





10^5-10^6 times faster than full physics simulation!

Physics Simulator

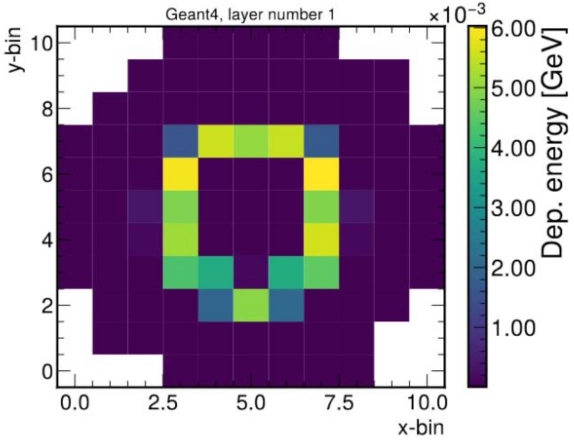
CaloScore

Calorimeter Simulation

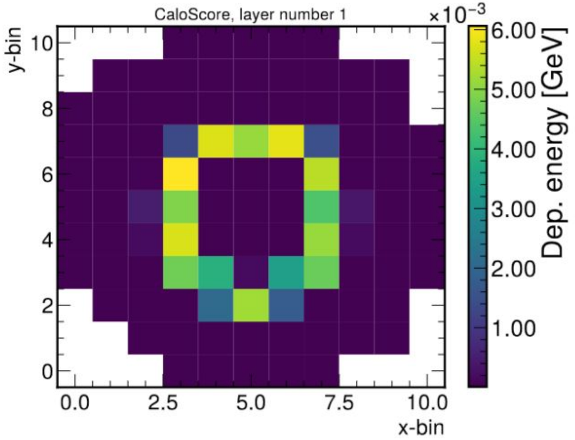
F. T. Acosta, **V. Mikuni**, *et al* 2024 *JINST* **19** P05003

- **Calorimeter** design based on the forward hadronic calorimeter of the ePIC detector
- Original dataset consisting of **55** layers with **55x55** cells
- Cells are merged to voxels: **11x11x11 voxels**

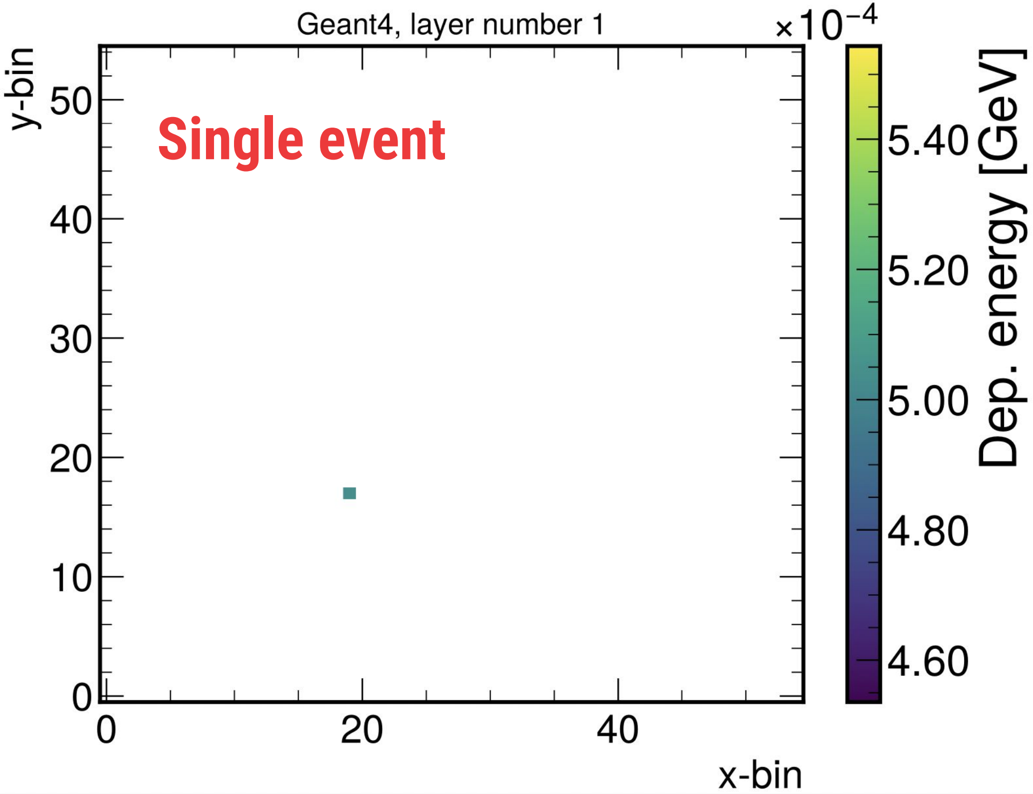
Full Simulation



Diffusion



Calorimeter Simulation



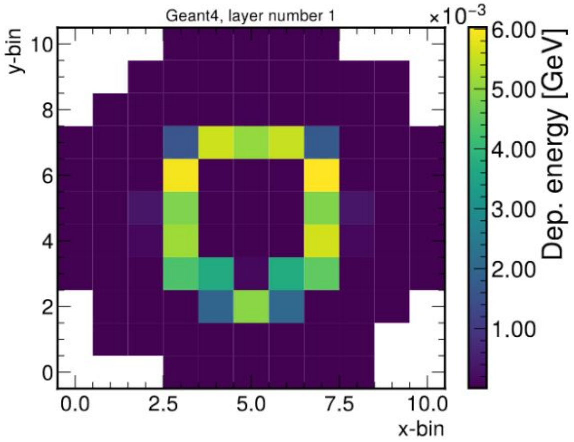
- Representing the full detector granularity is **expensive**
- However, most showers are localized and have low occupancy

Idea: Model only the cells with energy depositions: point clouds

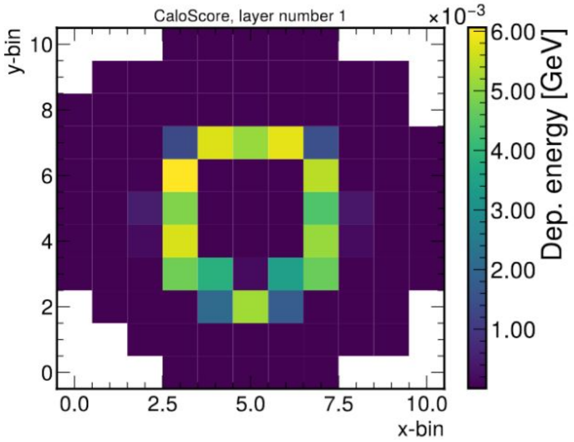
Calorimeter Simulation

F. T. Acosta, **V. Mikuni**, *et al* 2024 *JINST* **19** P05003

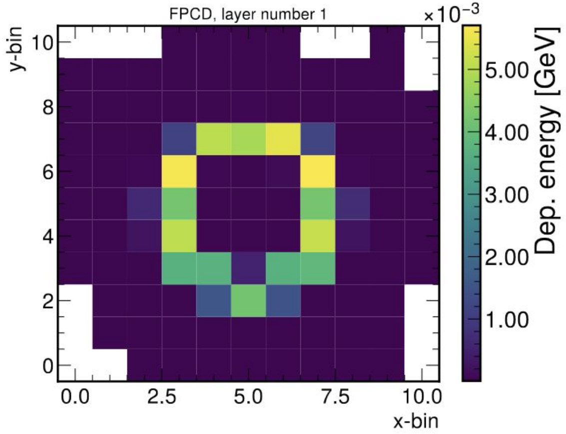
Full Simulation



Diffusion images



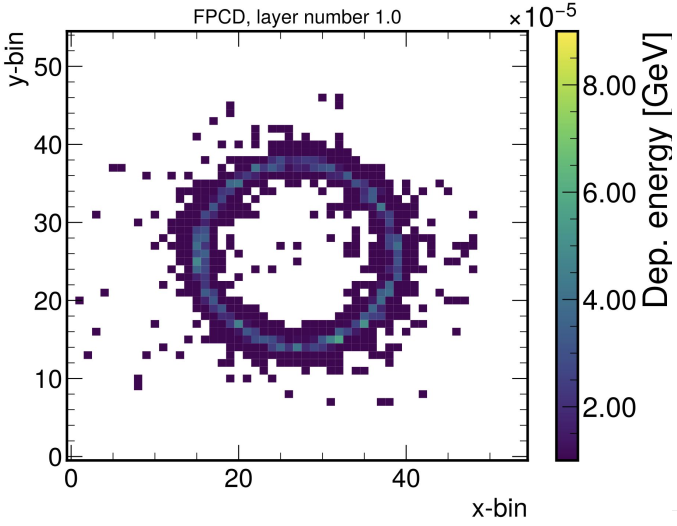
Diffusion point cloud

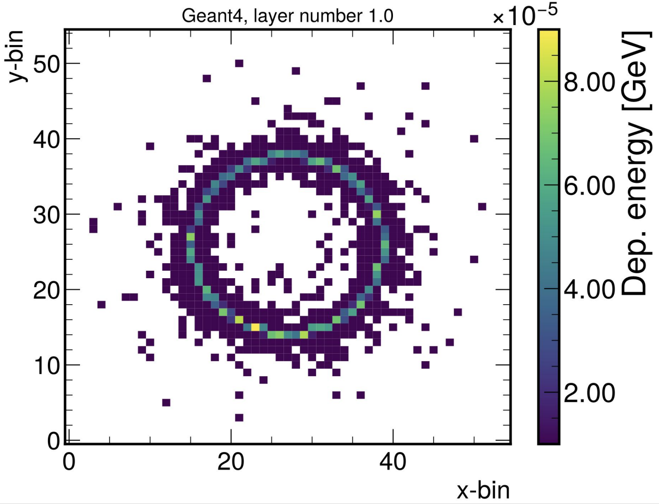


Calorimeter Simulation

F. T. Acosta, **V. Mikuni**, *et al* 2024 *JINST* **19** P05003

- Point cloud is trained on the full granularity

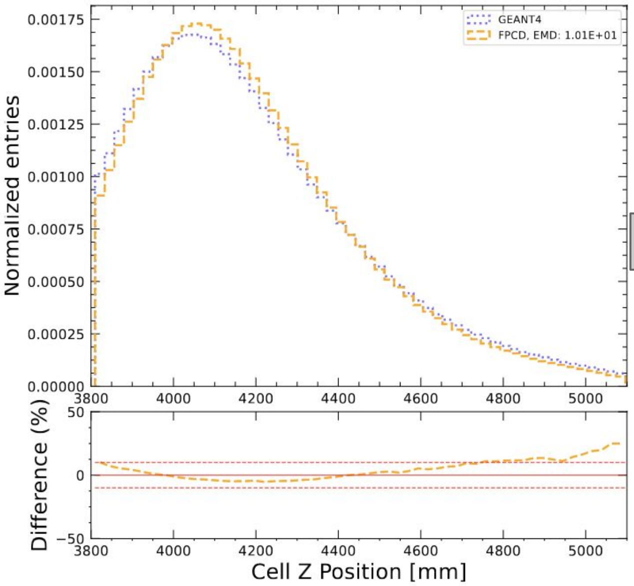


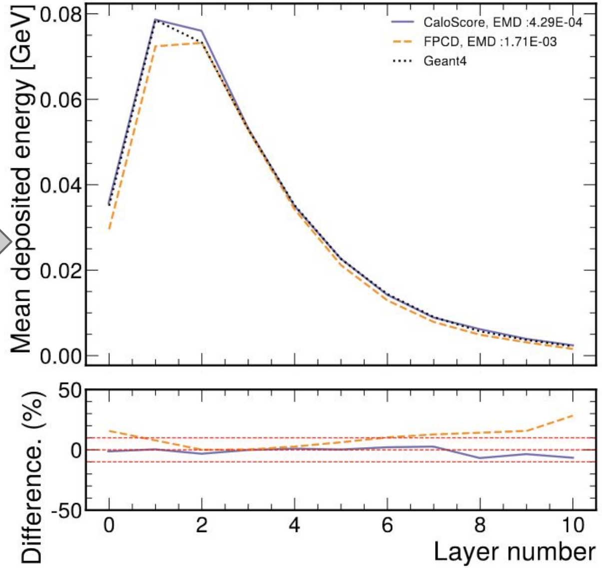


Calorimeter Simulation

F. T. Acosta, **V. Mikuni**, *et al* 2024 *JINST* **19** P05003

- Projection to voxel space is done only for comparison





Calorimeter Simulation

F. T. Acosta, **V. Mikuni**, *et al* 2024 *JINST* **19** P05003

- Point cloud model also requires less disk space and is faster to generate

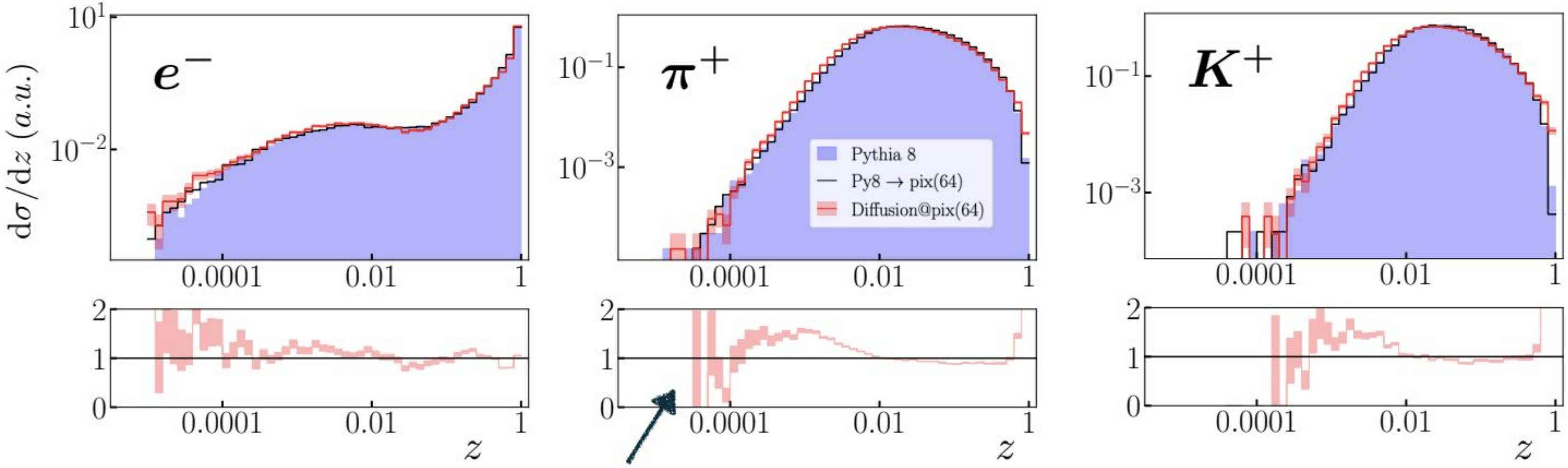
Model	# Parameters	Disk Size (Full)	Sample Time
Image	2,572,161	1016 MB (62 GB)	8036.19 s
Point Cloud	620,678	509 MB	2631.41 s

Point Cloud Simulation

Diffusion models for the EIC

Devlin, Qiu, FR, Sato `23

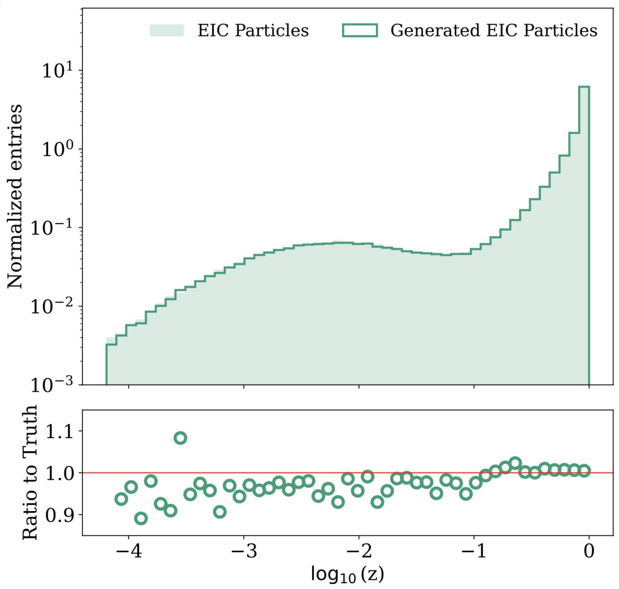
- Momentum distributions

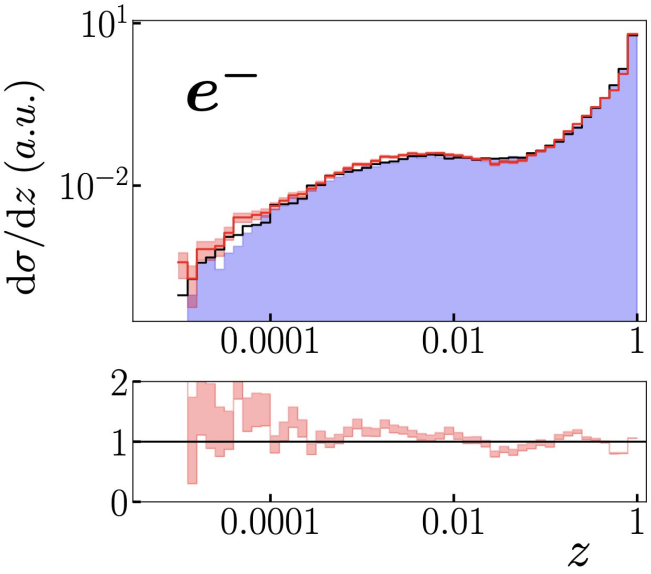


1σ Statistical uncertainties only

Point Cloud Description of the data

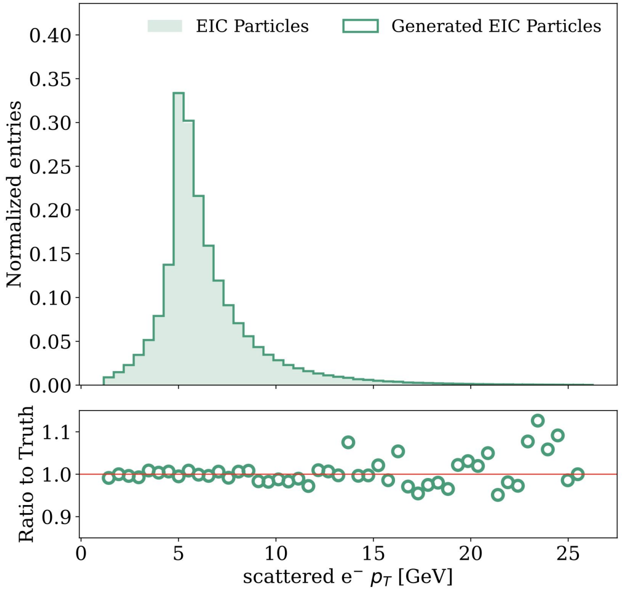
- Use point clouds instead of images: up to 50 particles
- Able to reproduce the z cutoff without additional transformations

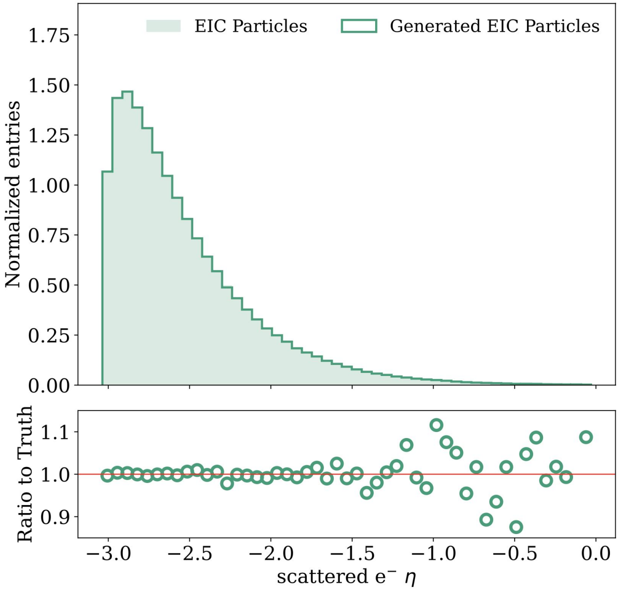




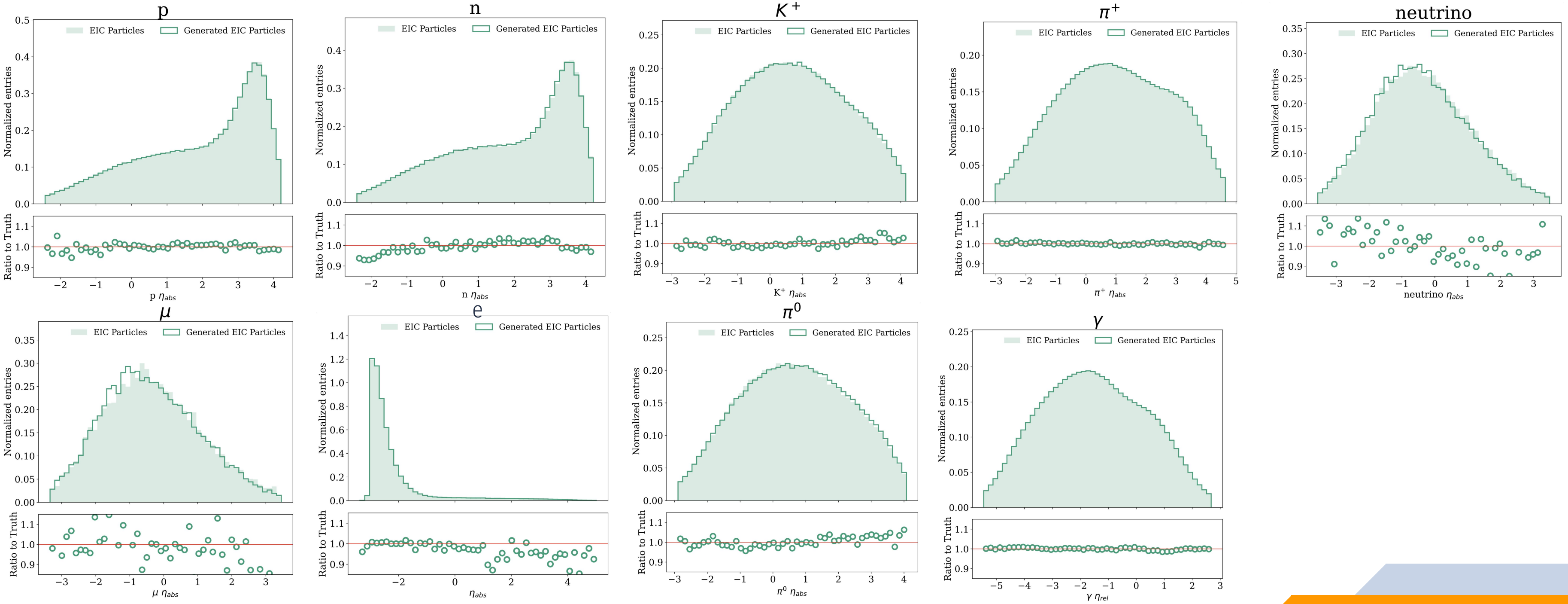
Point Cloud Description of the data

- Use the scattered electron as a reference and generate other particles conditioned on the electron kinematics



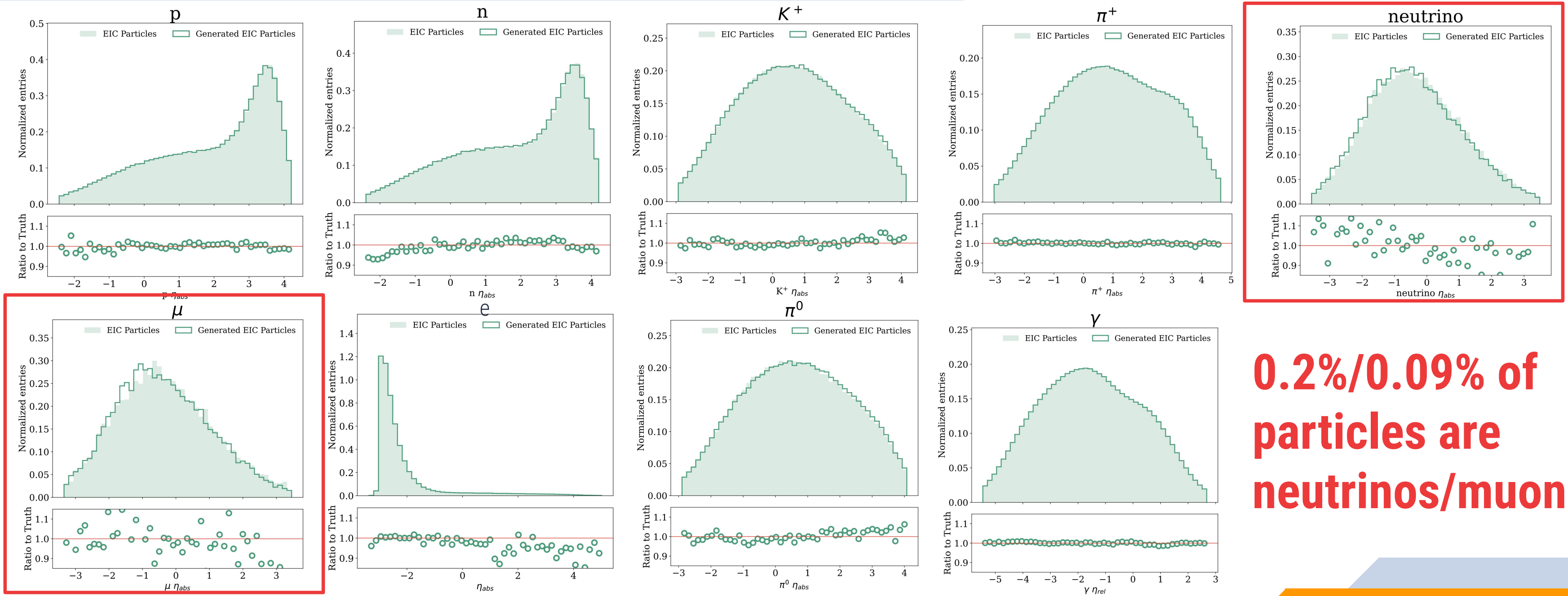


Point Cloud Description of the data



- Good agreement for all particles

Point Cloud Description of the data

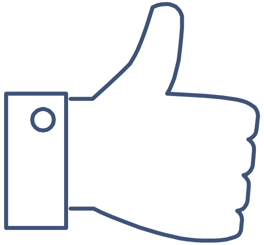


0.2%/0.09% of particles are neutrinos/muons

- Good agreement for all particles

Conclusion

- Diffusion Models are accurate generative models
- Initial image models were used for detector simulation
- Point cloud description of the data is more efficient:
  - Helps with data sparsity
  - Reduces the dimensionality of the inputs
- Compared to other generative models:
  - Flows: Diffusion is more flexible and can also get the data likelihood
  - VAES: Diffusion is able to learn sharp distributions more easily
  - GANS: Diffusion is easier to train



THANKS!

Any questions?

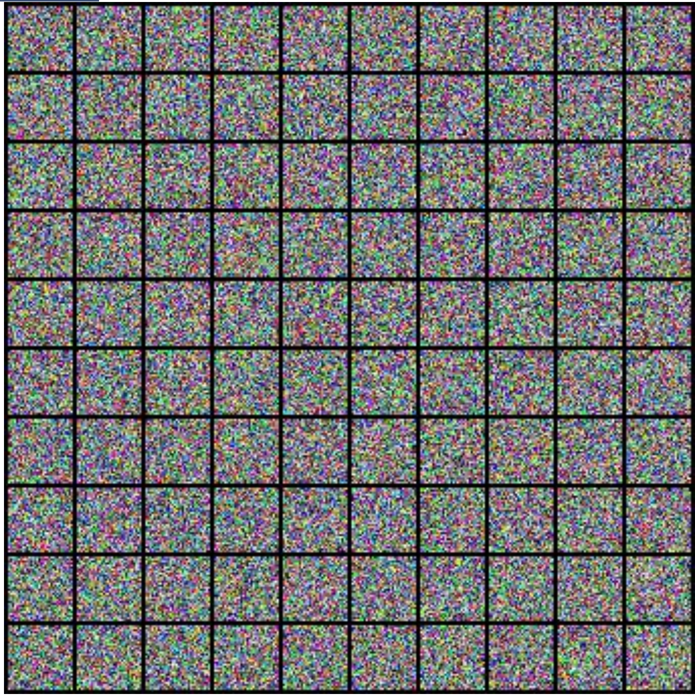
Generative models

- **GANS:**
  - Modern GAN architectures haven't really been explored in HEP, mostly the vanilla ones with ok results
- **VAE:**
  - KL Divergence can behave poorly when generator output changes too fast during training, often needs regularization.
  - Reconstruction loss is often taken as MSE, which learns only averages and makes sharp distributions blurry. For images there are other tailored losses that improve this behaviour
- **NF:**
  - Since the transformation needs to be invertible, bottleneck layers cannot be used, requiring very large networks for even small problems. Can still be improved by splitting into multiple smaller networks
  - Autoregressive flows are one of the best density estimators but alone are very slow either to train or to sample (O(d^2) in the slowest direction), but can still be overcome with distillation models

	Training Stability	Scalability	Fast inference	Fidelity	Expressivity
Diffusion	Yes	Yes	No	Yes	Yes
GANS	No	Yes	Yes	Maybe	Yes
VAE	Maybe	Yes	Yes	Maybe	Kinda
NF	Yes	Maybe	Maybe	Yes	Kinda

BERKELEY LAB

Score matching/denoising/diffusion



Denoise diffusion models are the newest state-of-the-art generative models for image generation.

Pros:

- **Stable training**: convex loss function
- **Scalability**: Network complexity is more sensitive to the architecture than the dimensionality
- **Access to data likelihood after training**: similar to NFs, but overall normalization is not required during training

Cons:

- **Slow sampling**: Possibly **1000s** of model evaluations to generate realistic images

Score-matching

- The common choice for **λ(t) is σ(t)^2** resulting in the loss function

$$\frac{1}{2}\mathbb{E}_t\mathbb{E}_{p_t(\tilde{x})}\left[\|\sigma(t)s_\theta(\tilde{x},t)+\epsilon(0,1)\|_2^2\right]$$

- Another important result is when **λ(t) is g(t)^2** that represents an upper bound of the data likelihood

$$\mathbf{KL}(p_0(\mathbf{x})\|p_\theta(\mathbf{x}))\leq\frac{T}{2}\mathbb{E}_{t\in\mathcal{U}(0,T)}\mathbb{E}_{p_t(\mathbf{x})}[\lambda(t)\|\nabla_\mathbf{x}\log p_t(\mathbf{x})-\mathbf{s}_\theta(\mathbf{x},t)\|_2^2]$$
$$+\mathrm{KL}(p_T\parallel\pi).$$

- Allowing the **maximum-likelihood** training of diffusion models!

Likelihood estimation?

- Data generation can also be achieved by solving the **associated ODE**
  - Often leads to **worse** samples compared to Langevin dynamics generation
- On the other hand, we can also use the deterministic ODE recover the **data density!**

SDE
$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]dt + g(t)d\mathbf{w}$$

ODE
$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right]dt$$

$$d\mathbf{x} = \tilde{\mathbf{f}}(\mathbf{x}, t)dt,$$
$$\log p_0(\mathbf{x}(0)) = \log p_T(\mathbf{x}(T)) + \int_0^T \nabla \cdot \tilde{\mathbf{f}}_{\boldsymbol{\theta}}(\mathbf{x}(t), t)dt,$$