

# The Calorimeter Pyramid

Are you interested in a Pyramid Scheme?

Simon Schnake, Dirk Krücker, Kerstin Borrás

29.04.24

EuCAIFCon24

HELMHOLTZ

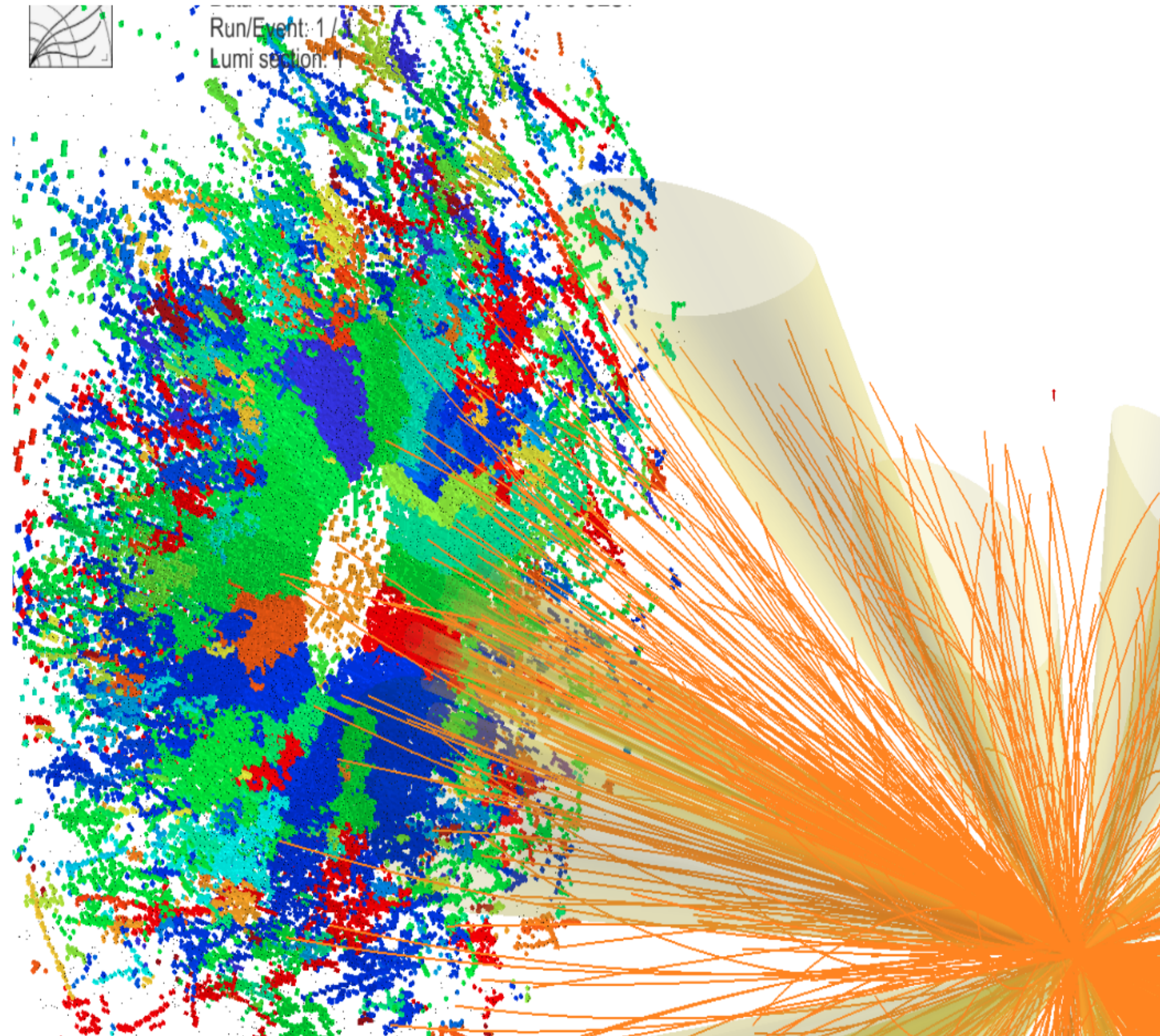


**RWTH**AACHEN  
UNIVERSITY



# The Challenge

Modern calorimeters have millions of channels



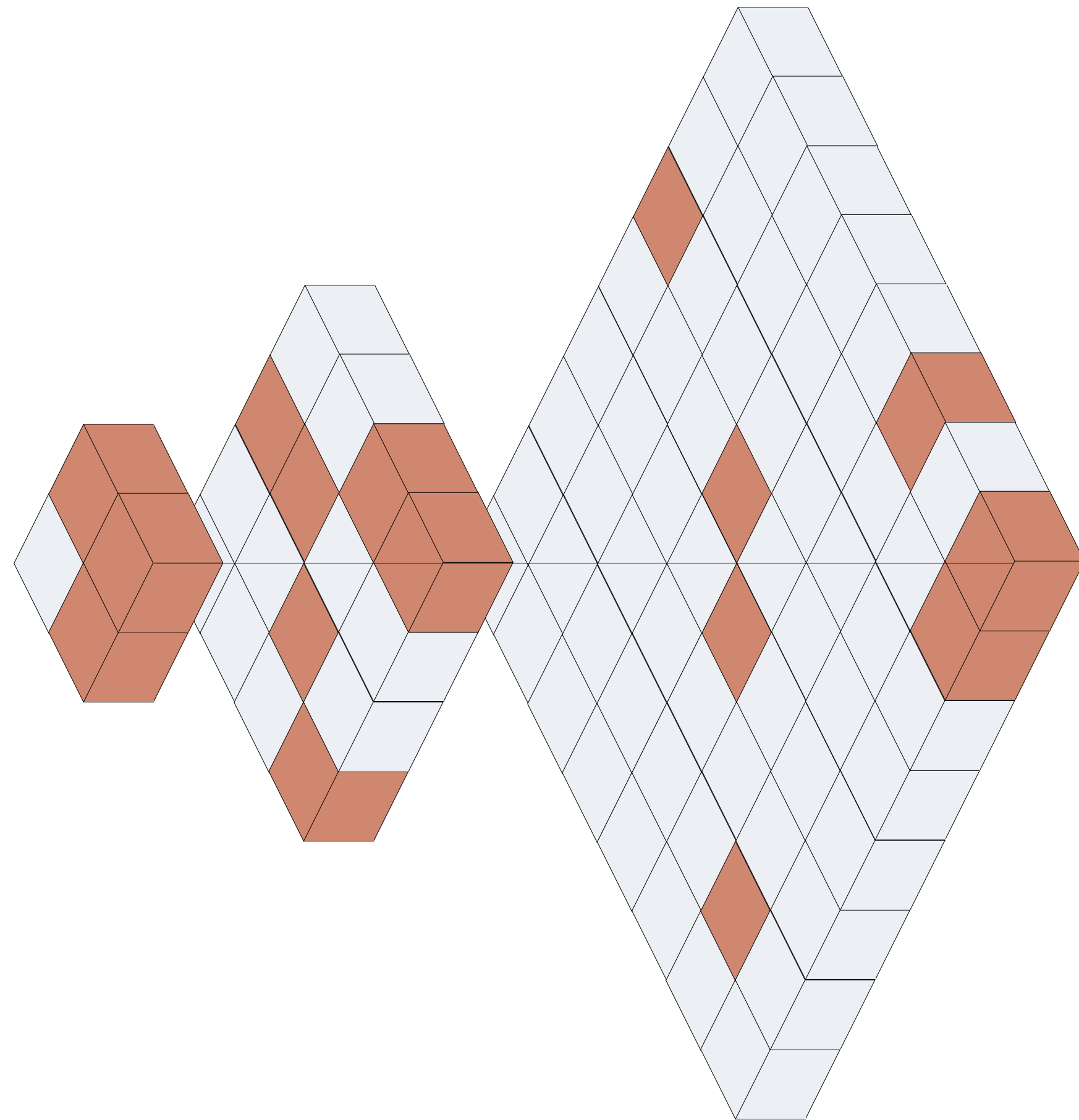
Simulated view of one HGCal endcap, containing particles from the nominal 140 pileup interaction expected at the HL-LHC  
[D. Newbold - [The High-Luminosity Upgrade of the CMS Detector](#)]

How do you scale generative models to millions of cells?



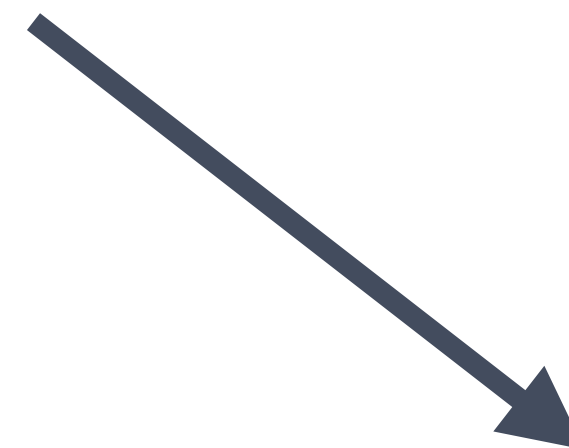
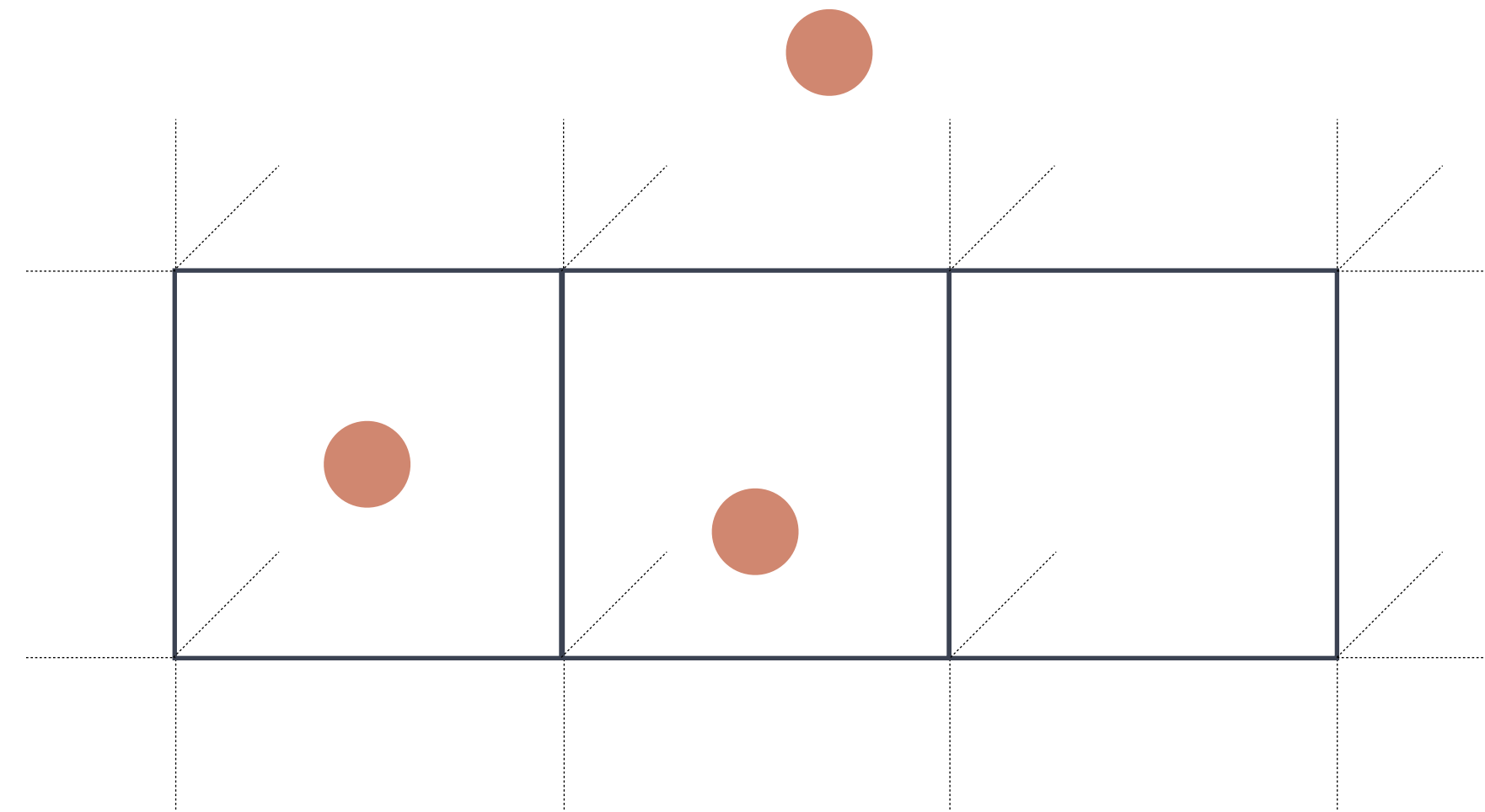
# Two Strategies

Super Resolution



or

Point Clouds

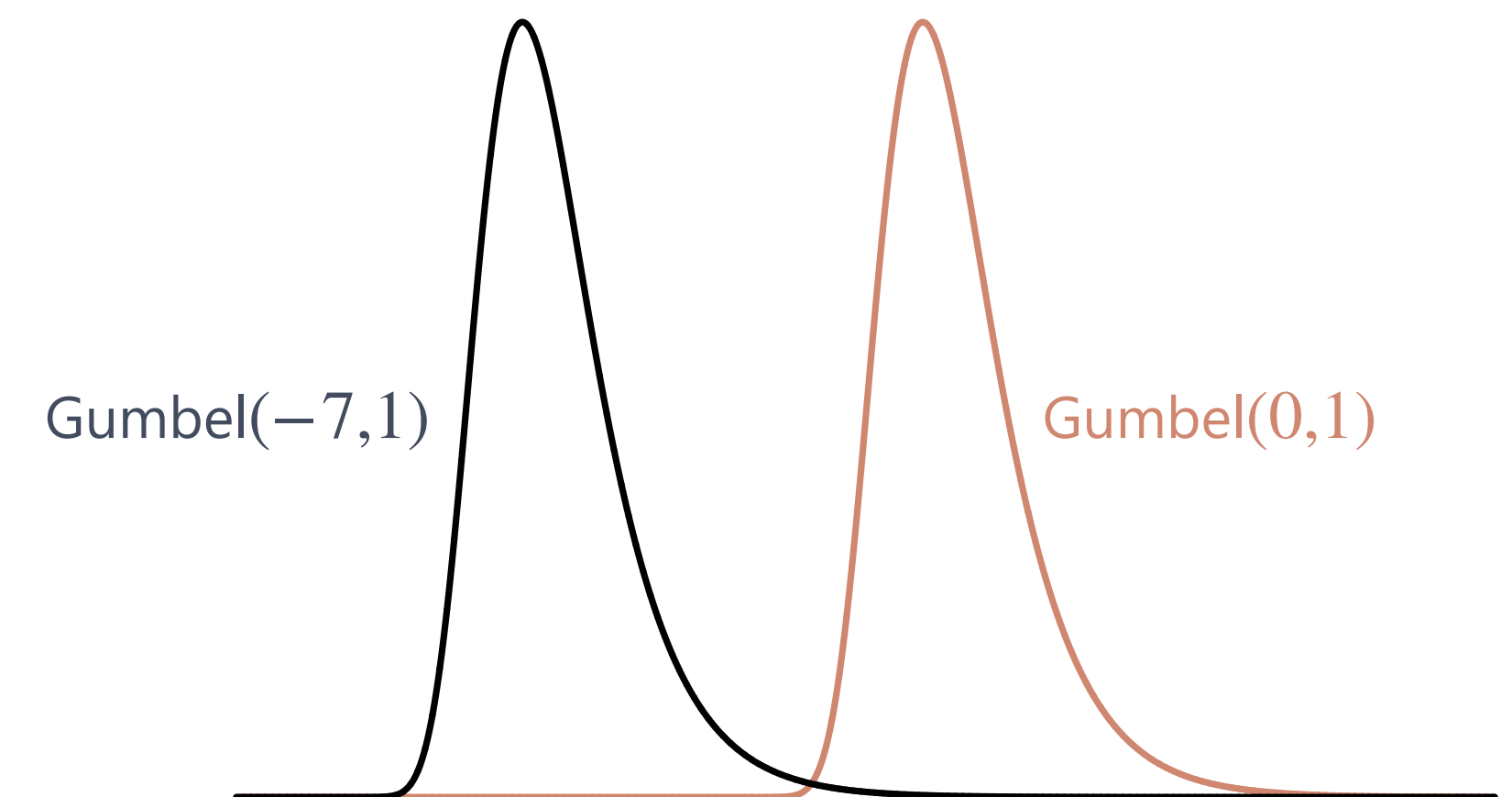
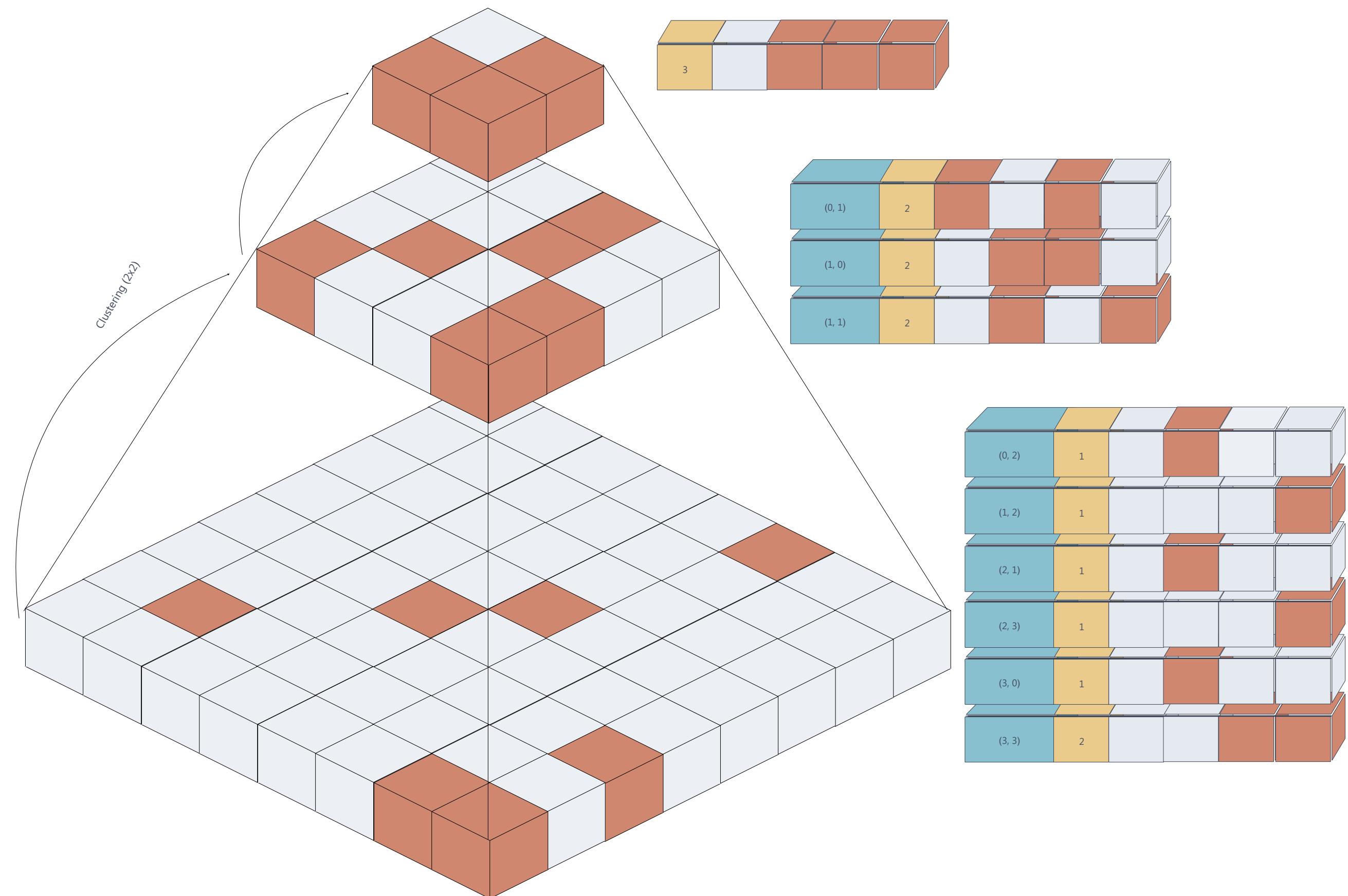


**Sparse Super Resolution**

# The Calorimeter Pyramid

## Sparse Super Resolution

First, learn all hit cells  $\rightarrow$  Second, learn the energies of the hits





# Choose your Diffusion

Efficient and flexible ways to accelerate ~~diffusion~~ (DM/CFM) in HEP

---

Cheng Jiang, Sitian Qian, Huilin Qu

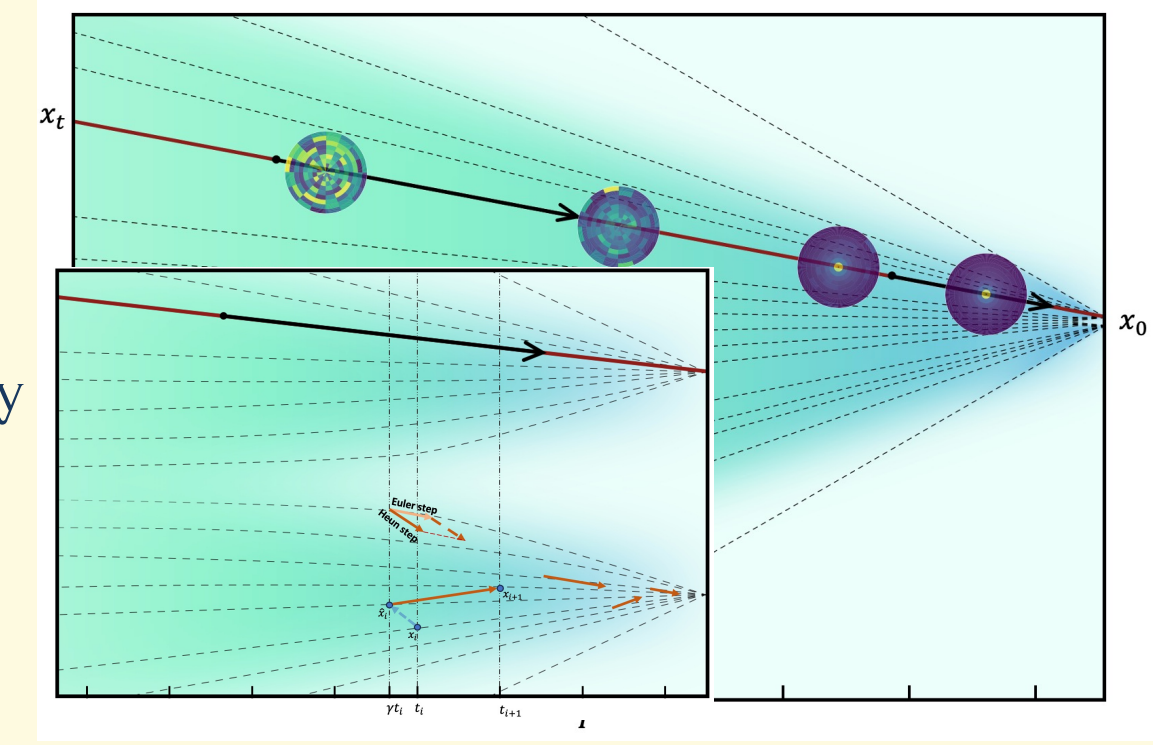
2401.13162



THE UNIVERSITY  
*of* EDINBURGH

# Motivation

- The study focuses mostly on Score Matching, in which the score function is solved by different choices of SDE/ODE. How we could effectively accelerate the generative model, by replacing only parts of that.



- Backward process (training-free):

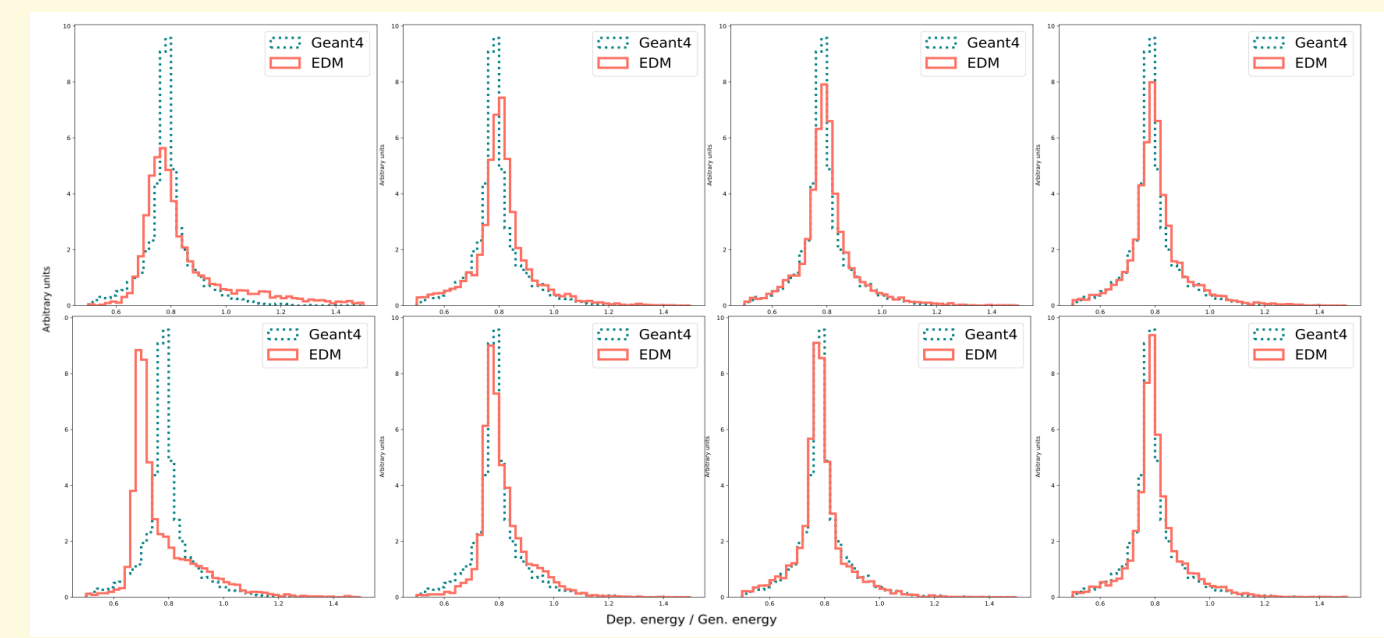
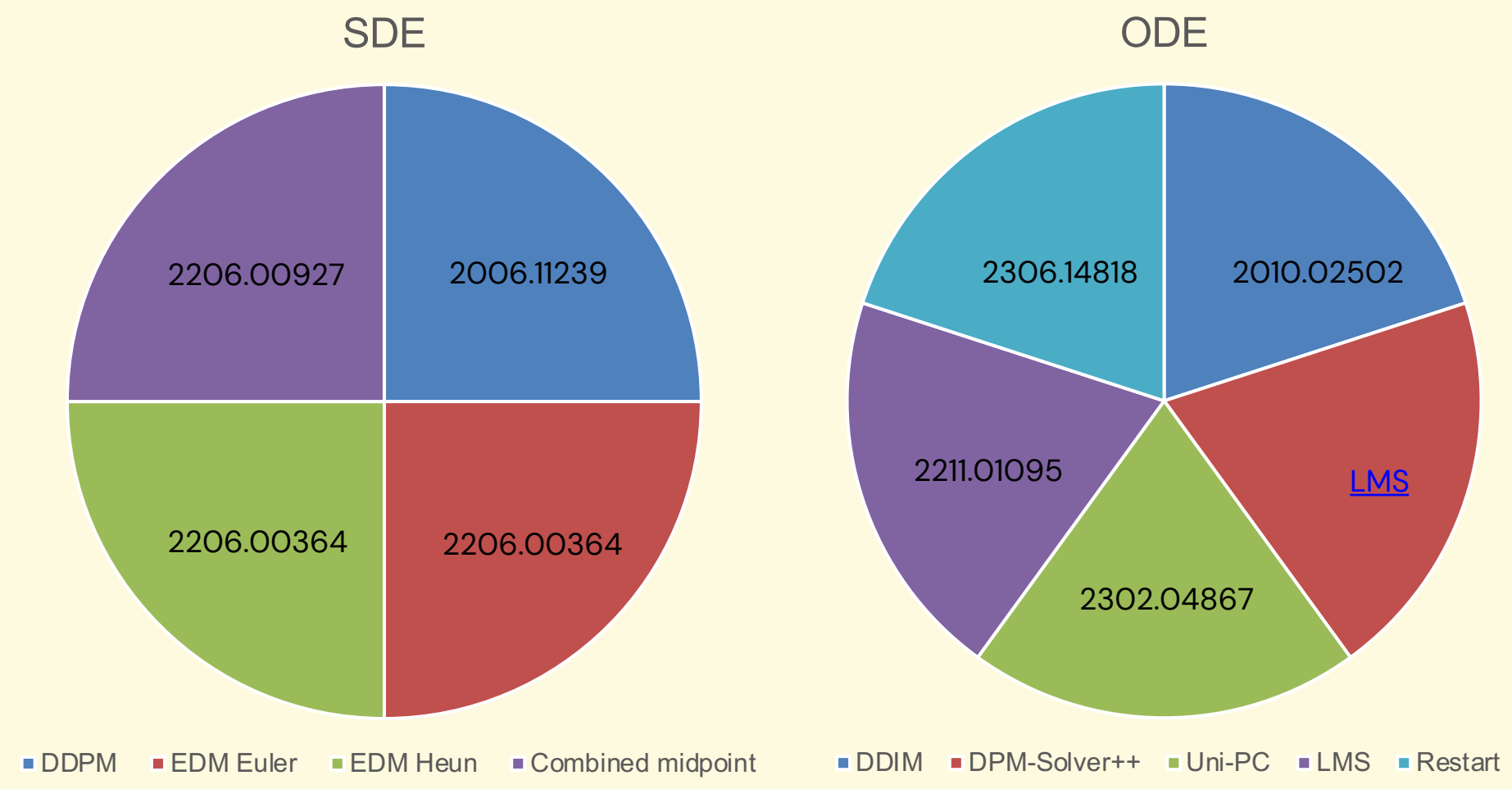
We have adopted almost all mainstream samplers/schedulers to do comprehensive comparisons on both shower cells (*CaloChallenge*) and jet constituents (*JetNet*)

- Forward process (faster divergence):

Effective way to mitigate the challenging optimization: Denoiser function with preconditioning parameters, weighted by min-Signal-to-Noise ratio (min-SNR)

$$\mathcal{L} = \mathbb{E}_{t,\epsilon} [\mathbf{w}(t) \| F(\mathbf{c}_{in} \mathbf{x}_t, t) - \frac{1}{\mathbf{c}_{out}} (\mathbf{x}_0 - \mathbf{c}_{skip} \mathbf{x}_t) \|^2]$$

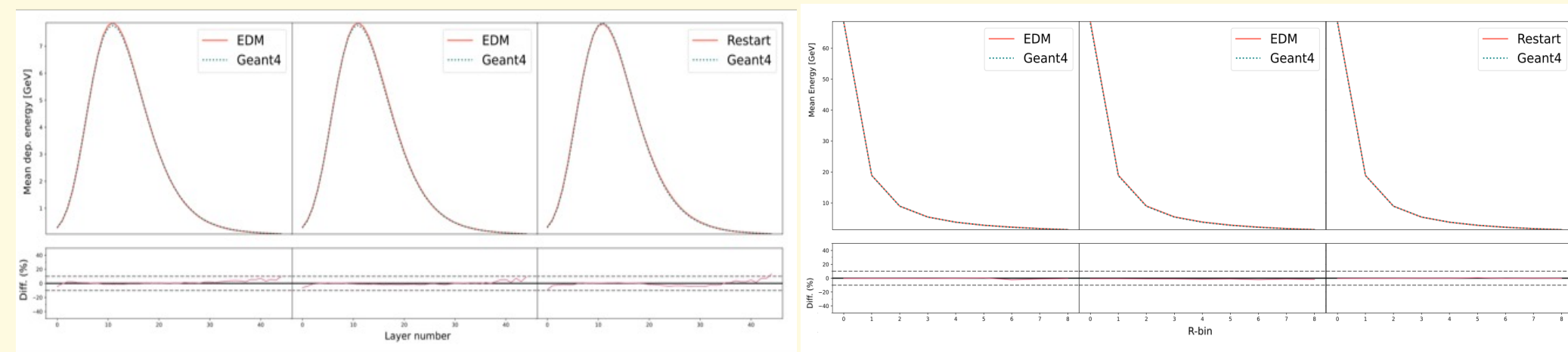
$$\mathbf{w}(t) = \frac{(t * \sigma_c)}{(t^2 + \sigma_c^k)^2}$$





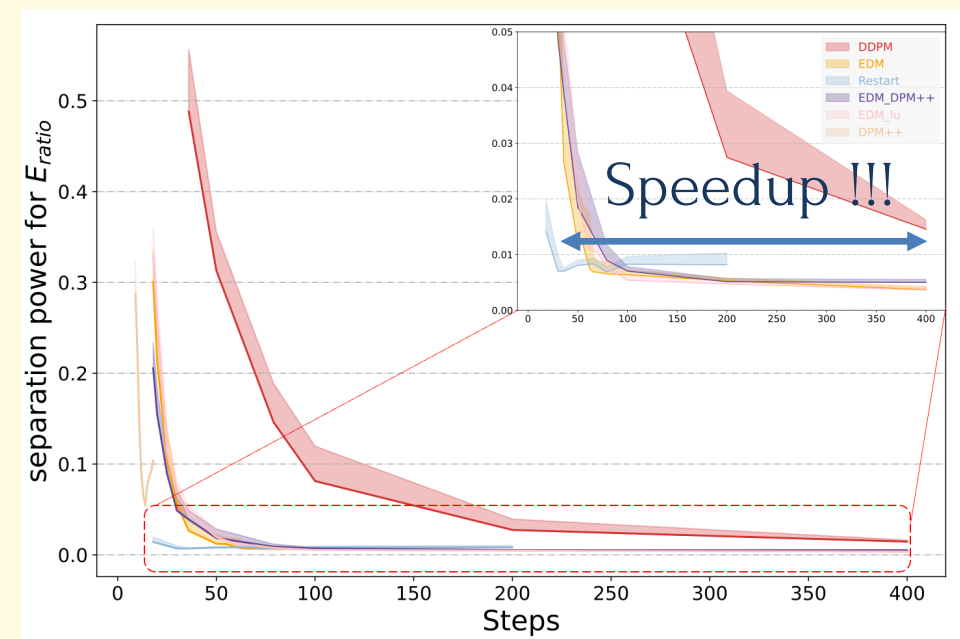
# Results & More (Wed Loc #45)

Indistinguishable high-level features for shower from cell-level generations



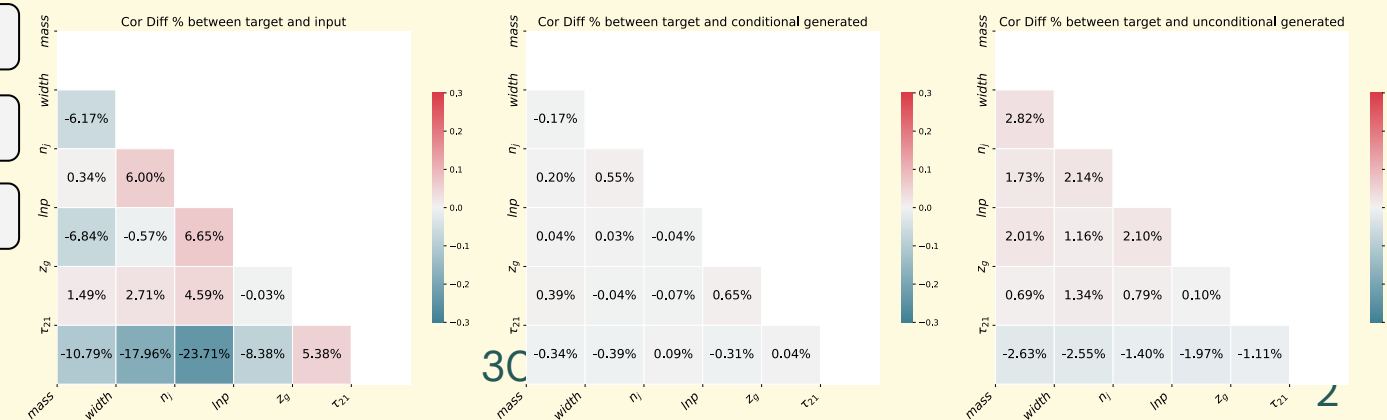
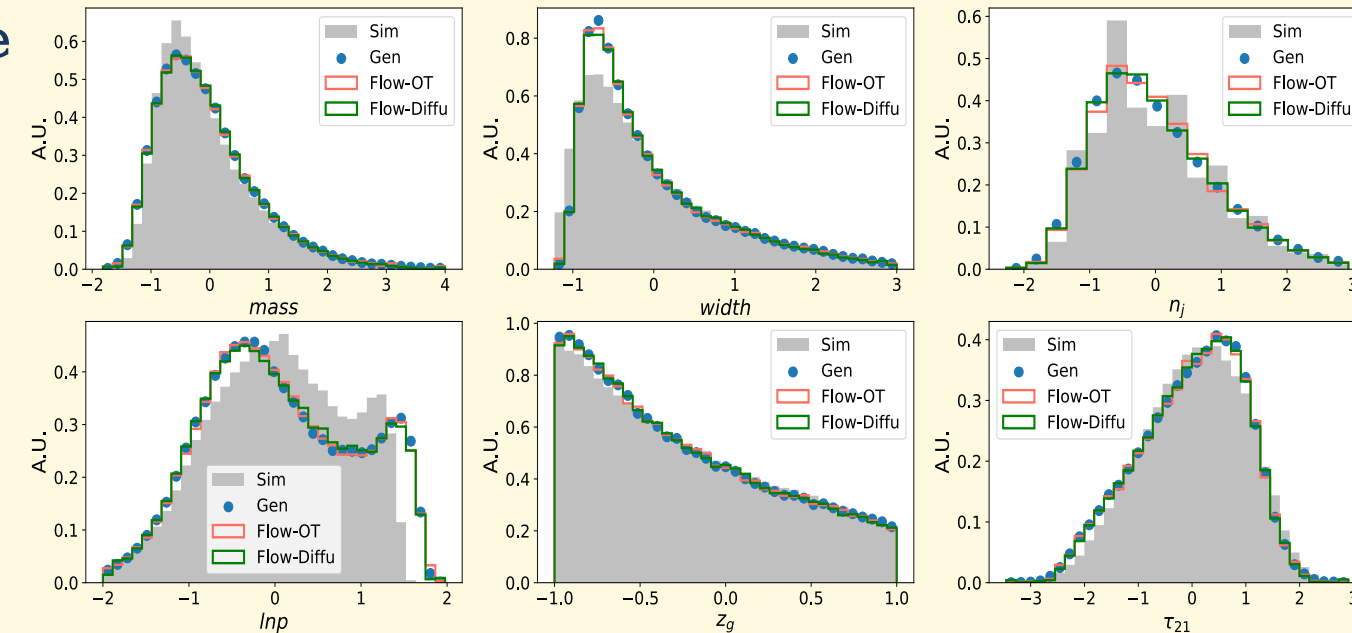
How about replacing the backbone for the model? **Changing flow matching with Unet/Transformer backbone to GBDT, which latter has much faster training and inference time. Is this even possible?? YES! (BUFF: BDT based-ultra-fast flow matching.)** Few mins training, below millisecond generation time, could replace most flow-based model. E.g. Unfolding, huge improvement on correlation

Achieve  $O(10)$  acceleration with comparable performance for current benchmark models



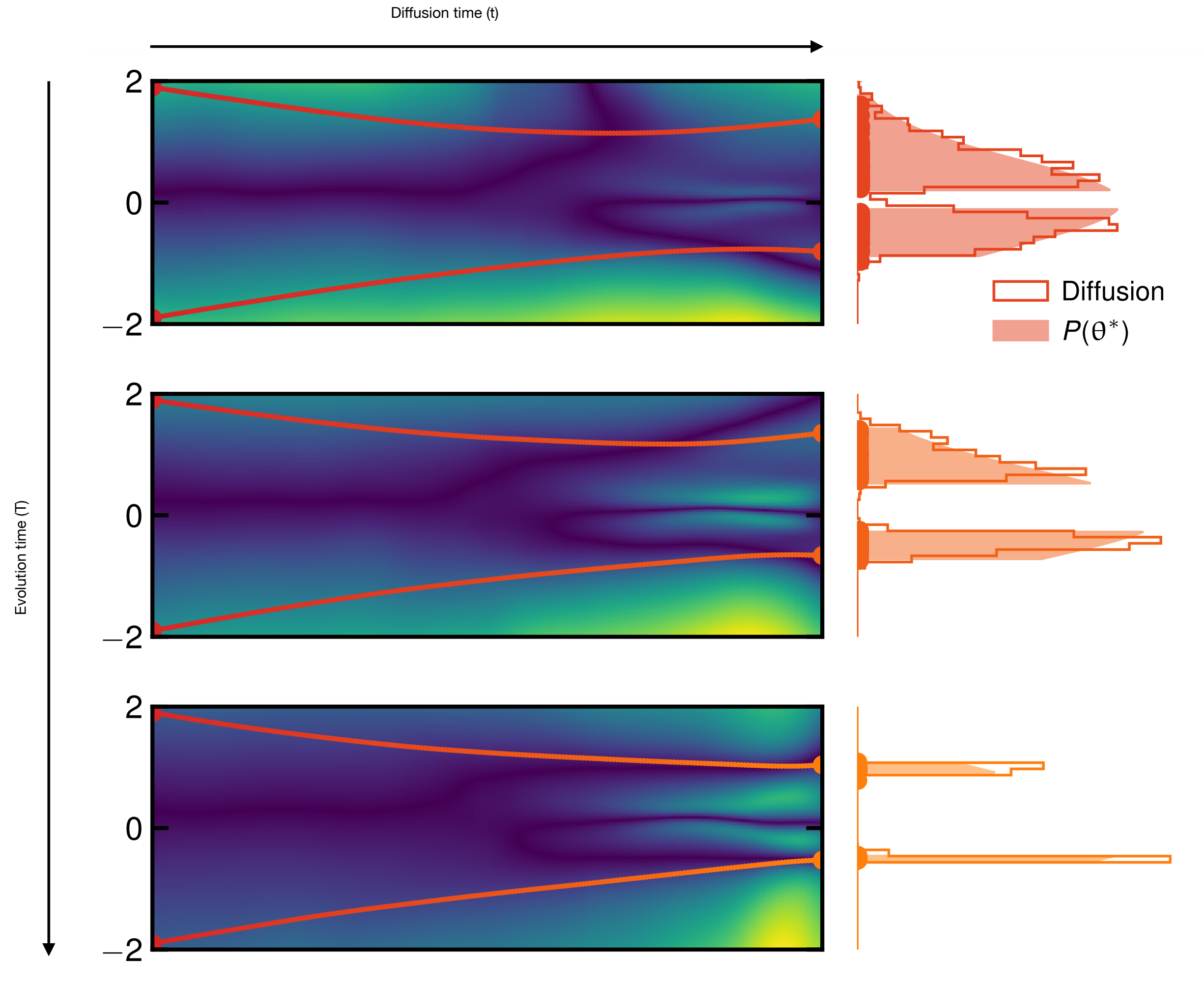
	AUC	Separation	FPD
<b>DDPM</b> 79/200/400	55.3/53.2/52.6	0.0810/0.0344/0.0155	0.074/0.046/0.043
<b>EDM</b> 39/79/200	54.1/52.0/51.5	0.0256/0.0076/0.0055	0.035/0.027/0.023
<b>EDM_DPM++</b> 79	52.3	0.0103	0.026
<b>EDM_Lu</b> 79	52.2	0.0086	0.026
<b>Restart</b> 18/36/79	55.2/52.0/51.8	0.0169/0.073/0.0057	0.059/0.025/0.022
<b>LMS</b> 36	53.8	0.0305	0.095
<b>DPM++</b> 20	59.8	0.0534	0.146
<b>Uni-PC</b> 20	60.3	0.1304	0.152

difference



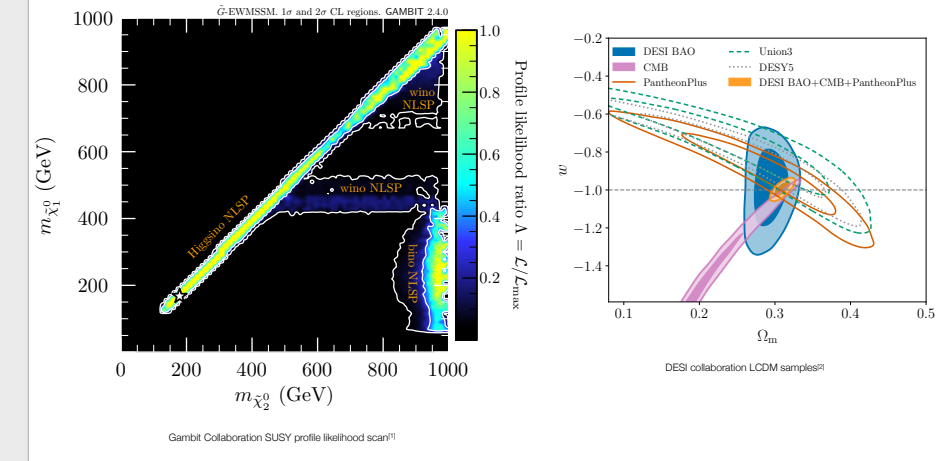
# fusions

Can we bring the latest developments in score based generative modelling to a nested sampling paradigm?



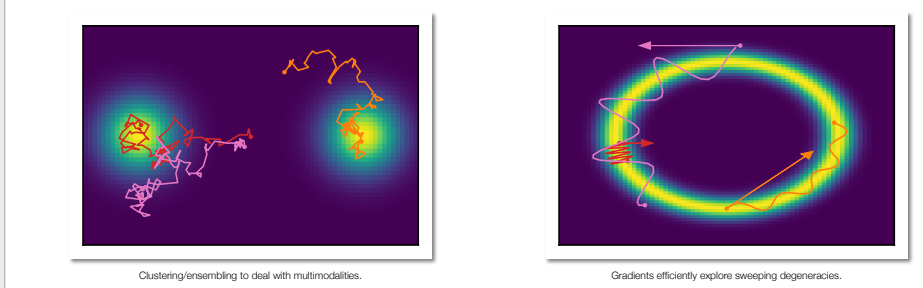
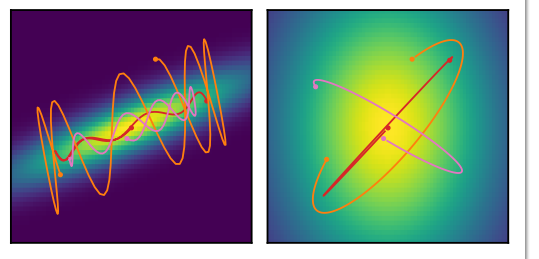
## INFERENCE

Fundamental physics is full of hard inference problems. Our optimization or sampling algorithms have to be able to navigate complex geometry



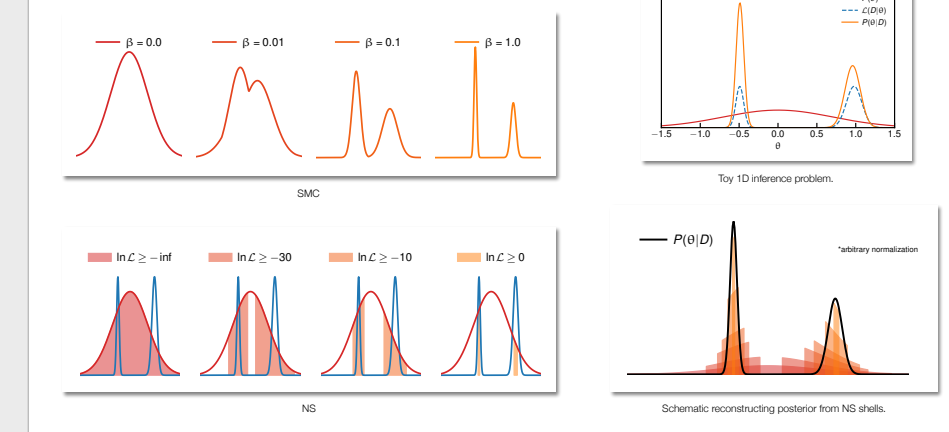
## GEOMETRY

Bad geometry in inference problems comes in many guises, and intuition gets progressively less clear in high dimension. Machine learnt neural mappings offer us a new tool to approach this.



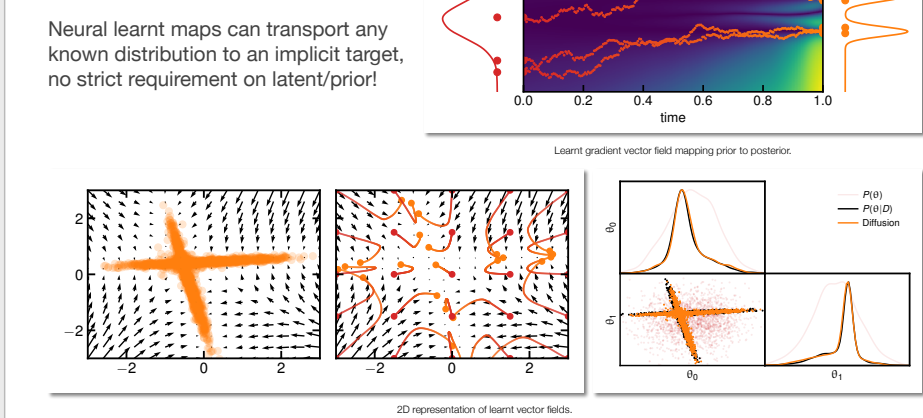
## BRIDGING DISTRIBUTIONS

Population Monte Carlo methods — particle filters — form bridges from known (prior) to complex unknown (posterior) distributions. Sequential Monte Carlo (SMC) and Nested Sampling (NS) are two variants evolving populations of points. Both give us access to the normalizing constant Z.



## DIFFUSION

Diffusion models learn the gradient of the implicit density of a point cloud. Solving evolution through this field with Stochastic Differential Equation (SDE) or Ordinary Differential Equation (ODE) solvers yields Diffusion<sup>17</sup> or Continuous flows<sup>18</sup>.



# DIFFUSION MEETS NESTED SAMPLING

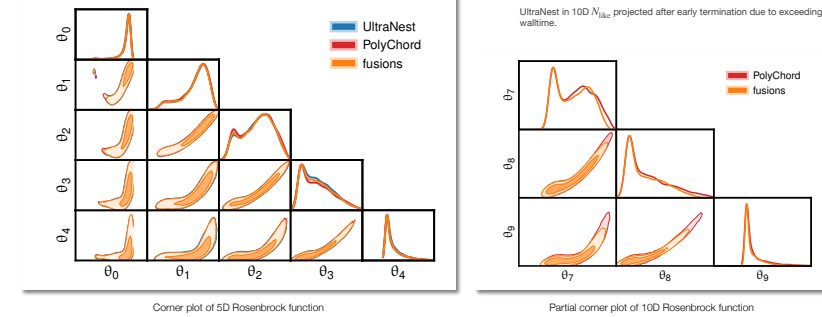
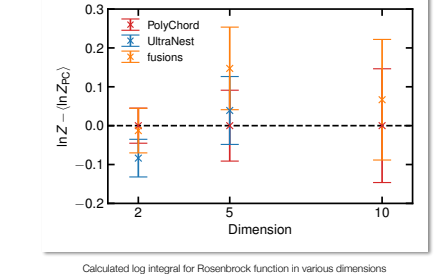
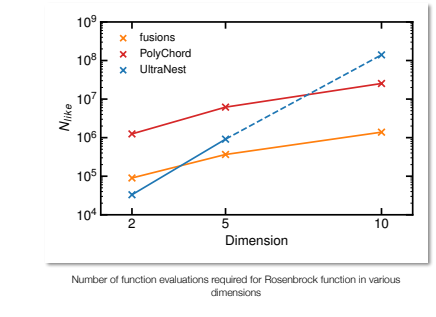
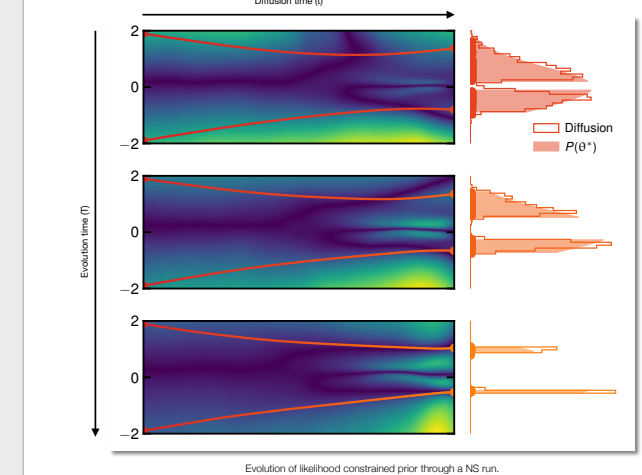
NEUTRALISING BAD GEOMETRY IN BRIDGING INFERENCE PROBLEMS

DAVID YALLUP

[yallup/fusions](https://github.com/yallup/fusions) [dy297@cam.ac.uk](mailto:dy297@cam.ac.uk) [yallup@github.io](https://github.com/yallup)

## RESULTS\*

Diffusion models introduce time axis to the problem, bridging algorithms have another time axis we can efficiently evolve by fine tuning the score estimate.



Comparison to standard (non-neural) tools<sup>9,10</sup> shows promising scaling, comparable to step samplers despite using rejection sampling, whilst maintaining accurate predictions on benchmark challenging problems.

Algorithm demonstrated uses zero classical methods, treating the geometry of the problem solely with neural networks and score based models.

\* Work in progress, comparison to other neural methods<sup>4,5,11,12</sup>, plenty left on the table to tune in the algorithm.



# Calculating entanglement entropy with generative neural networks

Dawid Zapolski, Piotr Białas, Piotr Korcyl,  
Tomasz Stebel, Mateusz Winiarski  
Jagiellonian University in Kraków

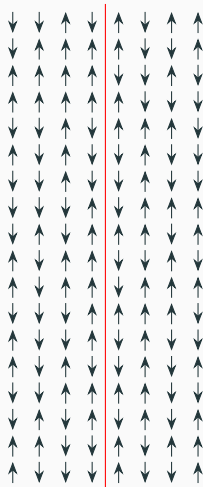
---

# Calculating entanglement entropy with generative neural networks



Quantum 1D Ising

$$S_n(A) = \frac{1}{1-n} \log \text{Tr} \rho_A^n$$

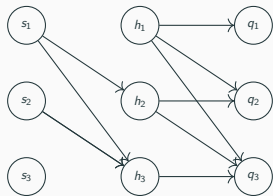


Classical 2D Ising



# Calculating entanglement entropy with generative neural networks

Autoregressive neural network



$$q_{\theta}(\mathbf{s}) = \prod_{i=1}^N q_{\theta}(s_i | s_1, \dots, s_{i-1})$$

Entropy as a function of the subsystem size

