Simulating the CMS High Granularity Calorimeter with ML

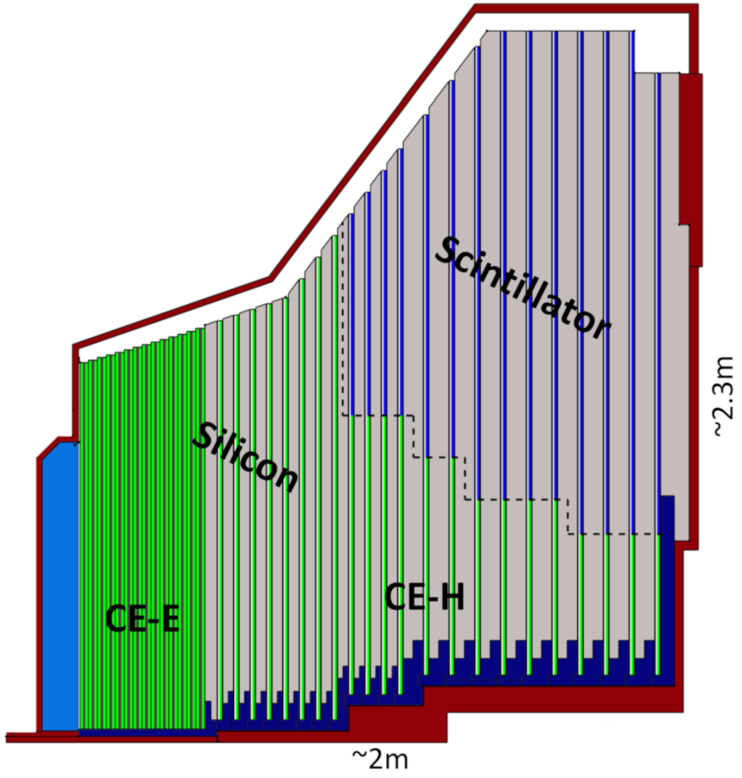
Oz Amram (FNAL), **Kevin Pedro** (FNAL)
on behalf of the CMS Collaboration

October 22, 2024

FERMILAB-SLIDES-24-0278-CMS-CSAID-PPD

Introduction

- CMS detector upgrades for HL-LHC include a new high granularity calorimeter (HGCal) in the endcap region
  - Inner layers made of silicon (green) w/ copper, tungsten, lead absorbers
  - Outer layers made of plastic scintillator (blue) w/ copper, steel absorbers
  - Granularity varies between electromagnetic (CE-E) and hadronic (CE-H) sections
  - Total of **~6M** readout channels
- Challenge: this calorimeter takes ~*2× longer* to simulate than existing calorimeters (~91K readout channels)

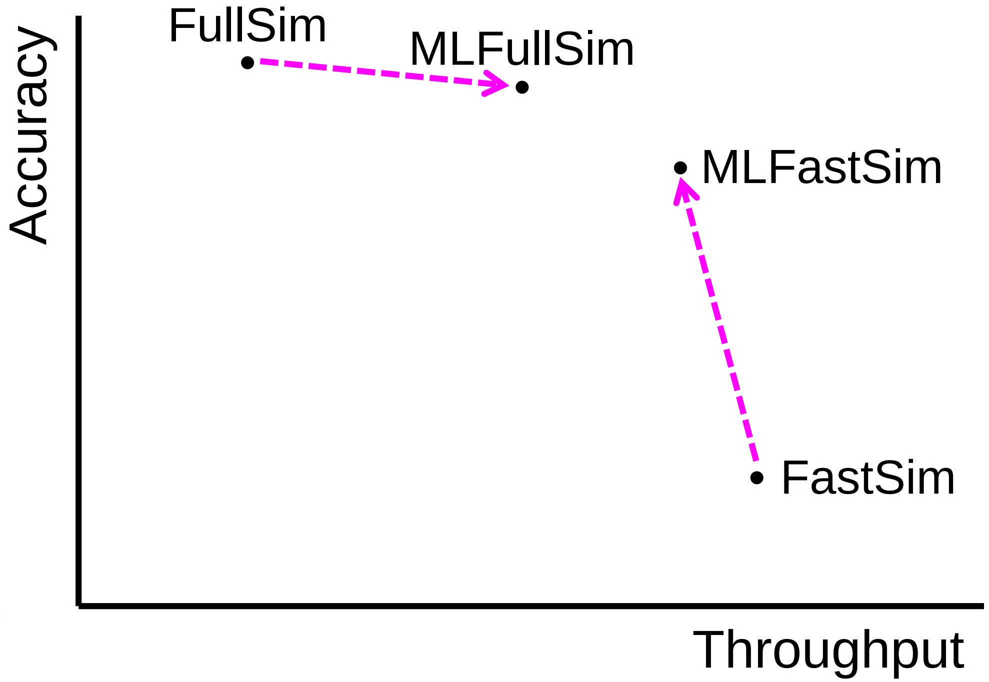


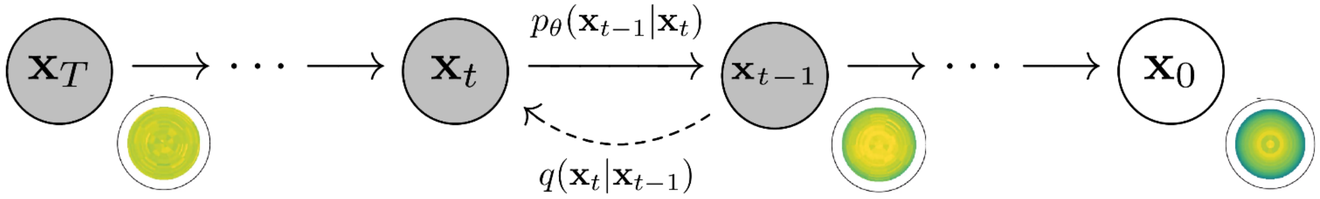
ML for Simulation

- Replace costly simulation components with generative ML
    - FullSim (Geant4): increase throughput, preserve accuracy
    - FastSim (parametric): increase accuracy, preserve throughput

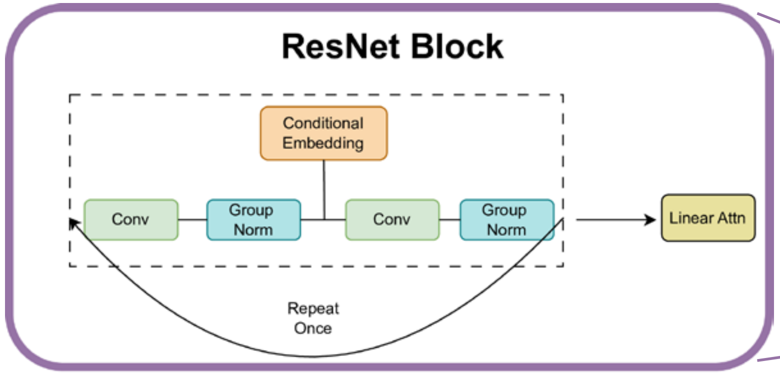
➢ Generative ML: learn *probability density* of simulated hits from particle showers

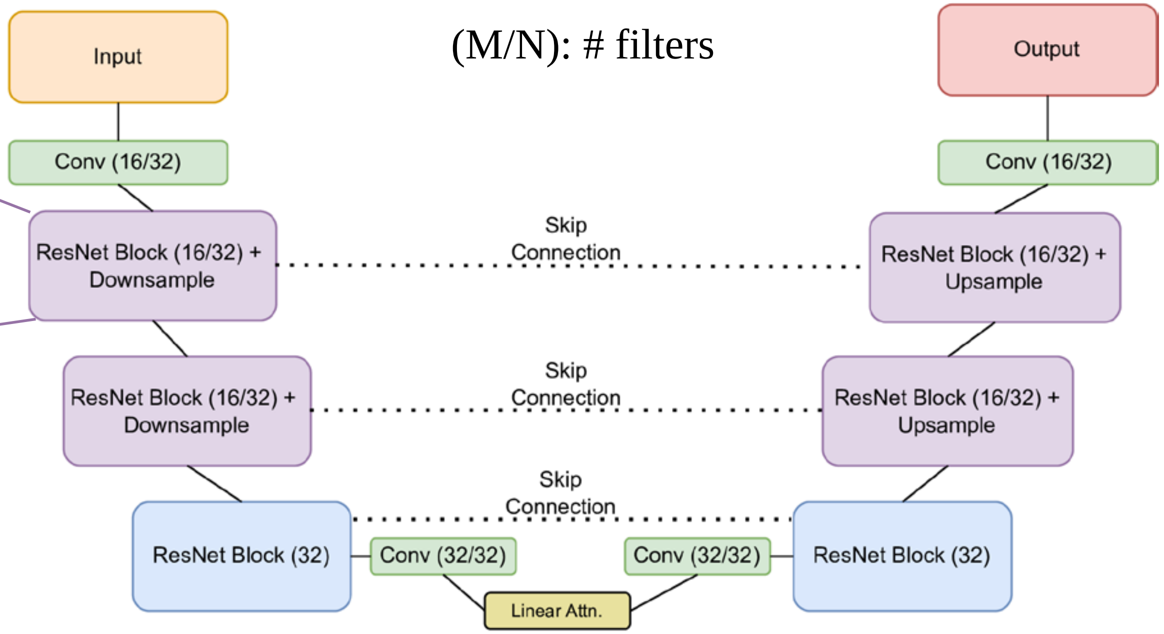
- Highest quality: *diffusion models*
    1. Add known amount of random noise to input training data
    2. Learn to predict noise in training data
    3. Starting from pure random noise, remove predicted noise iteratively $\rightarrow$ create a new sample from distribution of training data





CaloDiffusion



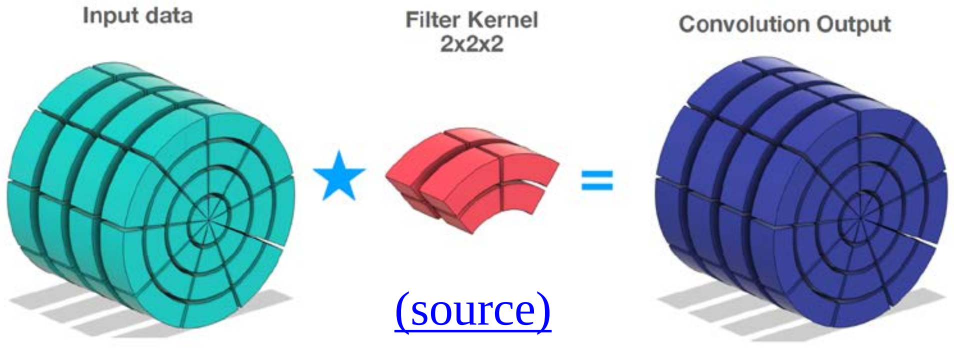


- Base architecture: U-net
  - Skip connections ensure no loss of information
- Linear self-attention layers applied to each convolutional ResNet block
  - Allows dimensionality reduction in *z* to handle longitudinal correlations in showers
- + several geometric innovations (next slides)
- Cosine noise schedule for training
- Stochastic sampling algorithm for generation

- Objective:
  - Predict weighted average of noise and denoised image
- Aim for highest achievable quality first
  - Then focus on improving speed
  - Wrong answers can be obtained infinitely fast

Geometric Innovations

- Particle showers are invariant & periodic in φ
  - Pad in φ so convolutions "wrap around"



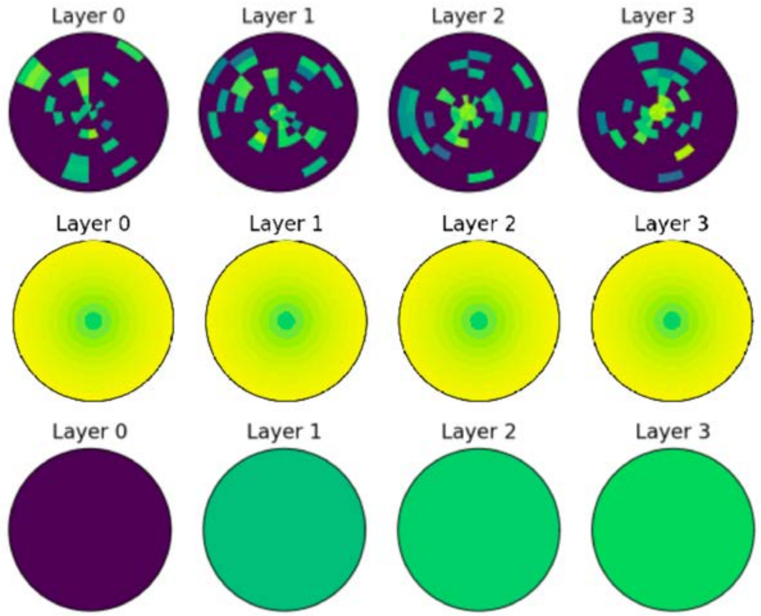
(source)

- Particle showers are *not* invariant in *r* or *z*
  - Provide *r* and *z* (layer) as extra per-pixel channels (input features)
  - Convolutions become *conditional*

Shower image

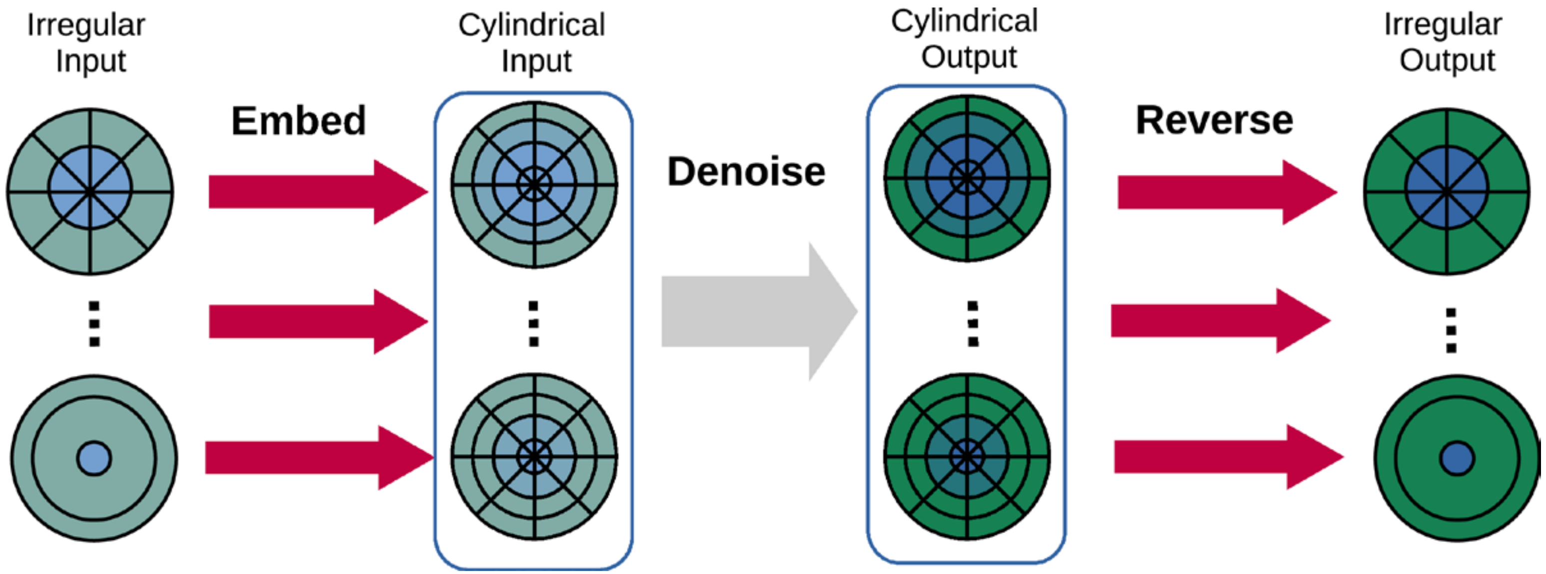
Radial image

Layer image



- ➢ *Conditional cylindrical convolutions*
  - Handle inherent features of particle detector geometry, distinct from rectangular images

Geometry Latent Mapping: GLaM

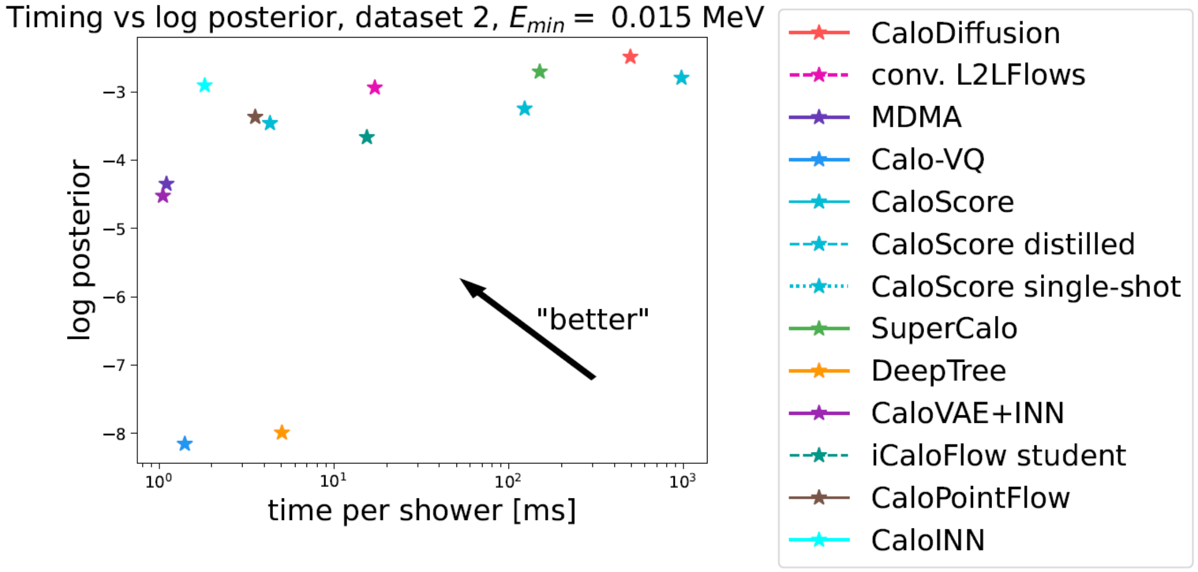


- Some calorimeter geometries have different radial/angular bins in each layer
  - Can't directly apply convolutions, which require regular neighbor structure
- Learn forward and reverse embeddings to and from a regular geometry
  - Simple matrices C (NxM) and D (MxN)
    - C initialized to split or merge cells based on overlap between original and embedded geometries
    - D initialized as Moore-Penrose pseudoinverse of C
- Inspired by "latent diffusion" approach
  - But not necessarily lower-dimensional representation; can be higher-dimensional

CaloChallenge Performance

[C. Krause](#)

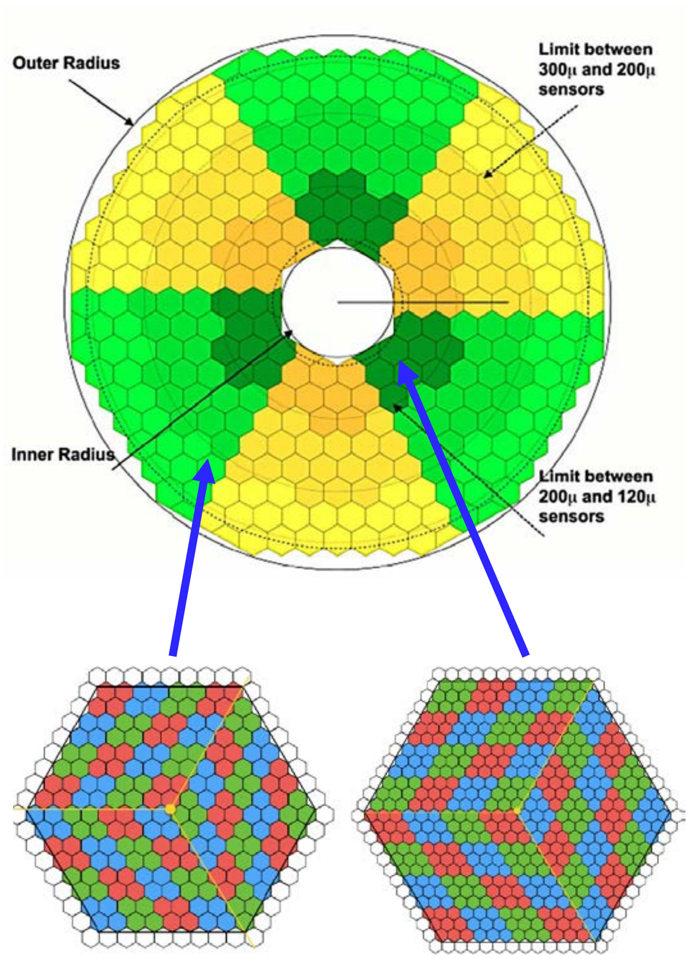
- CaloChallenge: community competition w/ three public datasets (~200K events each):
  1. Low granularity (368/533 voxels), irregular geometry (based on ATLAS calorimeter), photon & pion showers
  2. Medium granularity (6480 voxels), silicon-tungsten sampling calorimeter, electron showers
  3. High granularity (40500 voxels), otherwise same as #2



- CaloDiffusion (*Phys. Rev. D* 108 (2023) 072014): leading performance in accuracy
  - But slower inference time, typical of diffusion models (multiple steps required)
  - This first version required 400 diffusion steps
    - Subsequent versions incorporate improvements to reduce steps while maintaining quality

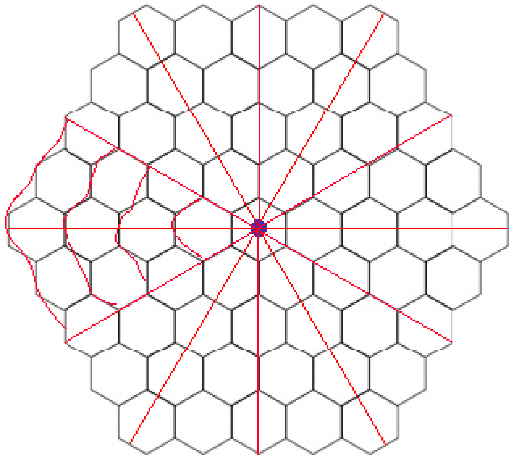
HGCal Dataset

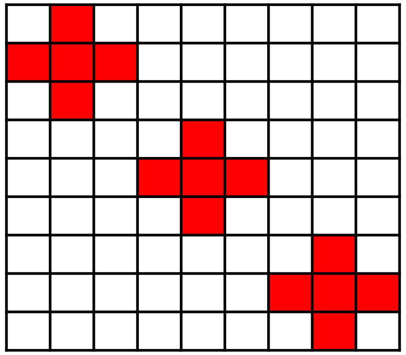
- 500,000 photon showers
  - o $\eta = 2.0$, $\varphi = \pi/2$, $E = 50–100$ GeV
  - o Train: 400K, test: 100K
- Geometry:
  - o HGCal version 11 from 2019 with 50 total layers (CMS-TDR-022)
  - o CMSSW_11_3_X, Geant4 version 10.7.1
- Voxelization:
  - o 20 “rings” of hexagonal cells around generated photon trajectory
  - o 28 layers (CE-E) × 1988 cells ≈ 56K voxels
- Preprocessing: (E_i = voxel energy)
  - o Logit transform: $u_i = \log(x/(1-x))$, $x \equiv \delta + (1 - 2\delta)E_i$
  - o Standardization: $u_i' = (u_i - \bar{u})/\sigma_u$



HGCaloDiffusion

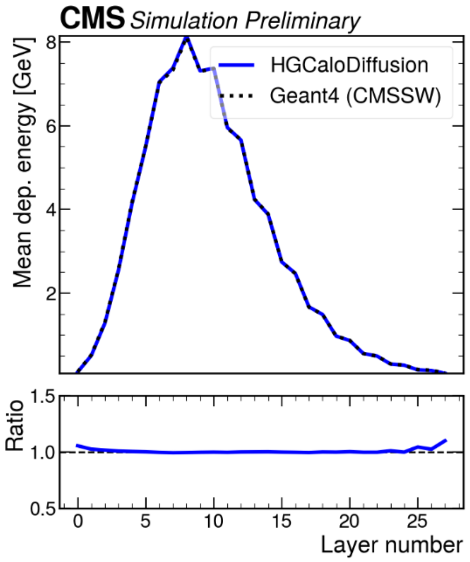
- CaloDiffusion model *plus*:
  - Scaled-up U-net (32, 64, 96 filters), 4M params
  - LayerDiffusion: separate lightweight diffusion model (5 dense layers with residual connections, 680K params) to predict total deposited energy per layer
    - Improves modeling of global quantities and reduces # steps in inference
  - Minimum signal to noise ratio weighting during training (arXiv:2401.13162) with improved noise schedule ("EDM", arXiv:2206.00364)
  - Deterministic sampling algorithm ("DDIM", arXiv:2010.02502)
  - GLaM adjustments:
    - Map to cylindrical geometry w/ 12 angular bins × 21 radial bins = 252 bins
      - Compression by a factor of ~7
    - Full embedding matrix would be 252 bins × 1988 voxels, per layer
    - ➢ Fix most elements to zero, only local entries learnable (5×5, ~10K per layer)
  - Diffusion steps for generation: 200

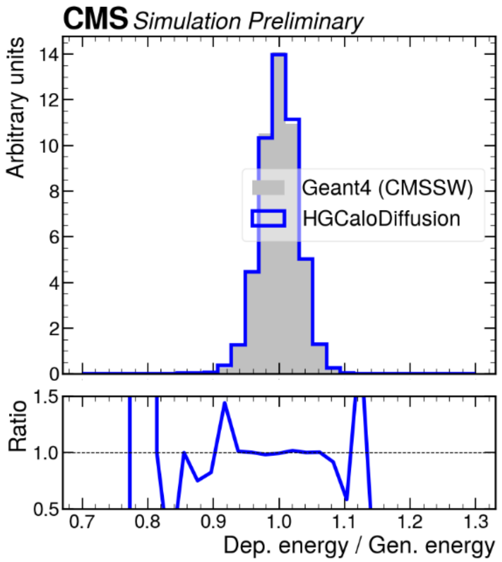




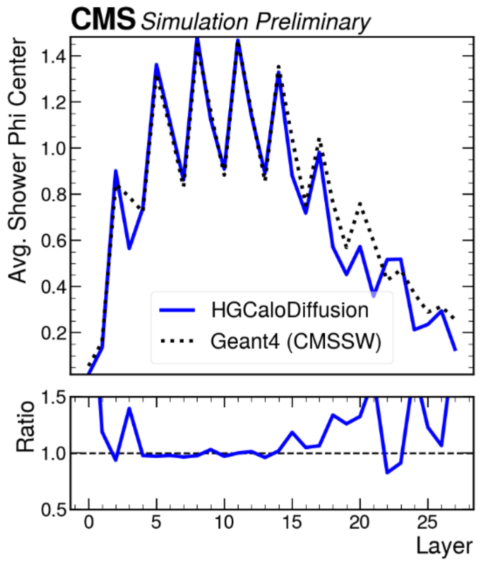
Successes

Global quantities well-modeled (thanks to LayerDiffusion)

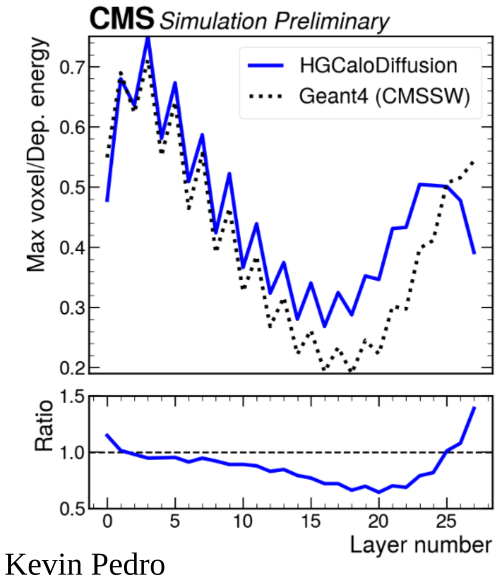


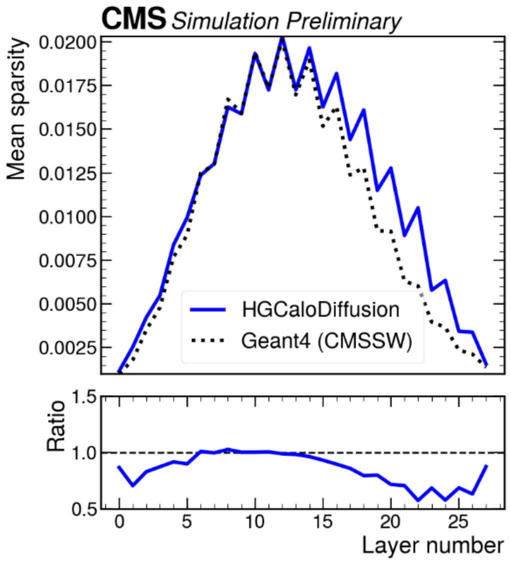


GLaM allows reproduction of “sharp” features in original geometric space

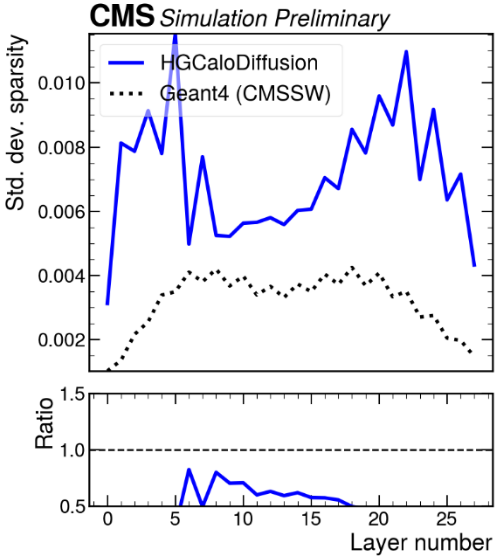


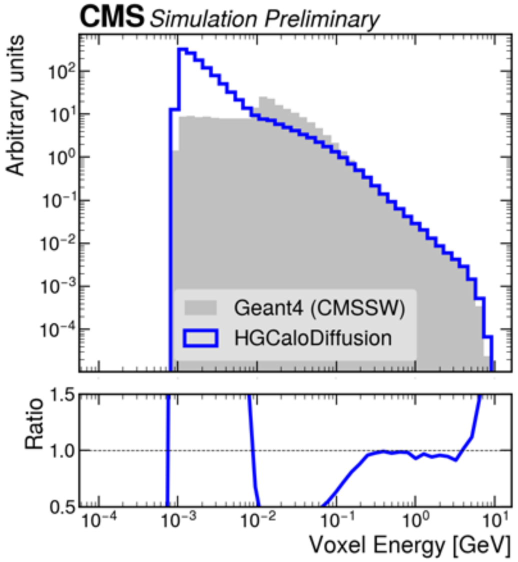
Reasonable agreement in maximum voxel per layer and average sparsity ($n_{E > 1\ \mathrm{MeV}}/n_{\mathrm{hits}}$)



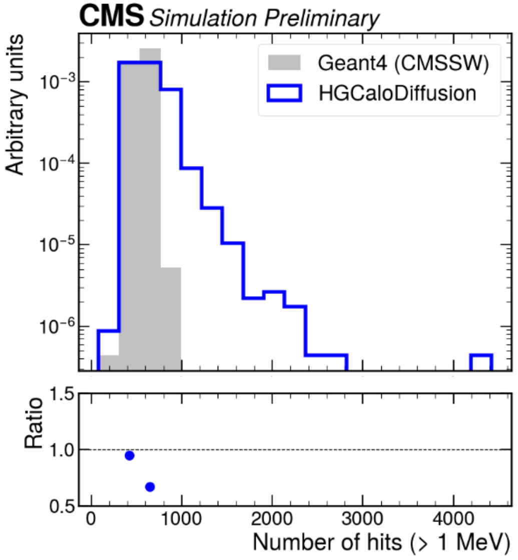


Opportunities for Improvement

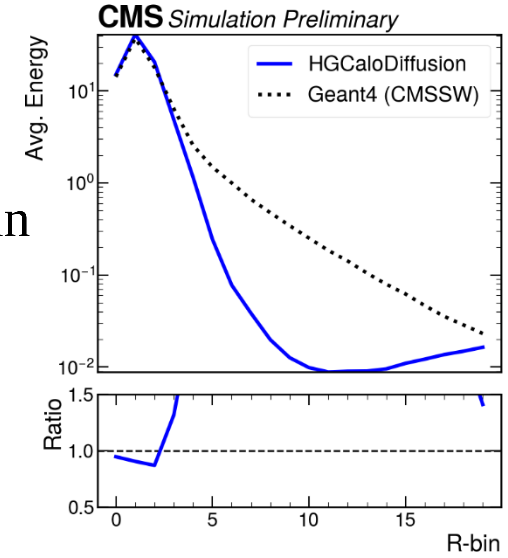




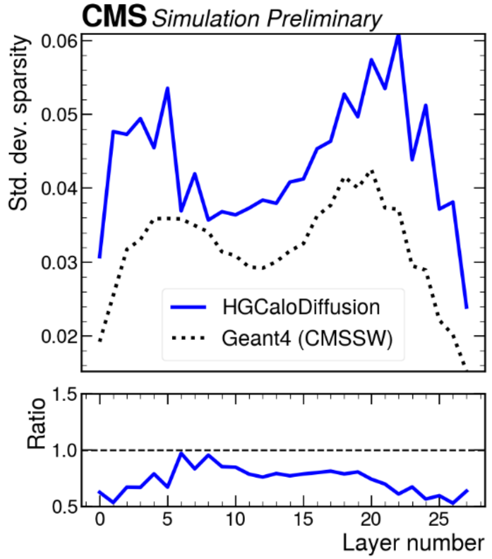
Deficits in sparsity modeling from:
1. “splitting” energy among multiple cells (first usage of GLaM for compression)
2. “leftover” noise in cells that should be empty

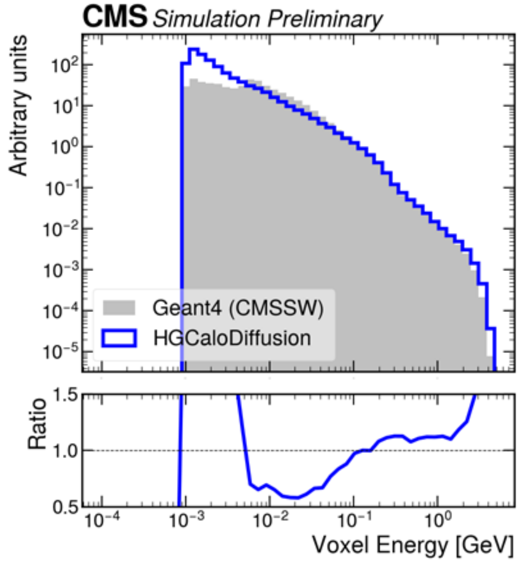


Leads to discrepancies in width-related variables (still being understood)

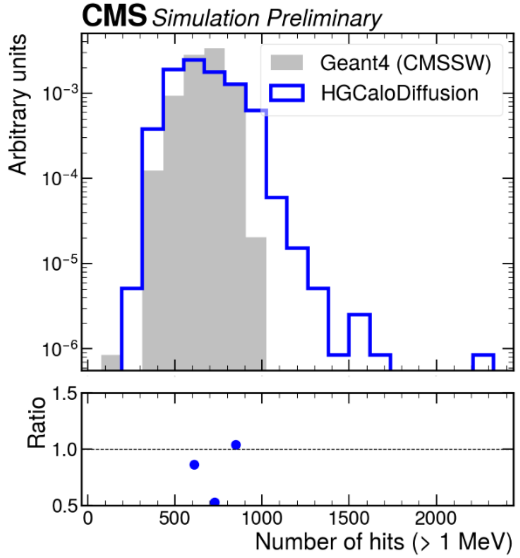


In GLaM Space

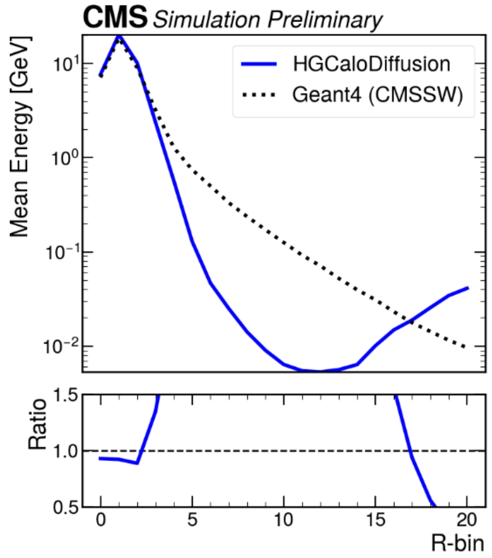




Examining the same quantities in “GLaM space” (compressed cylindrical geometry) shows improved modeling



Discrepancies remain in a few variables (still being understood)



Performance

- Classifier score: 0.995
  - Train a classifier to distinguish between “real” and generated showers
  - Look at area under receiver-operator characteristic curve: 0.5 means indistinguishable
  - Inputs: high-level features, such as plots shown previously
- Frechét particle distance: 0.726 (0.002 for Geant4 vs. itself)
  - W_2 distance between Gaussian fits to high-level feature space
- Kernel particle distance: 0.014 (0.000002 for Geant4 vs. itself)
  - Maximum mean discrepancy in high-level feature space

➢ Discrepancies in some features (e.g. energy vs. R):
noticeable enough to distinguish most generated showers

  - Expect improvements in metrics when these discrepancies are resolved

Outlook

- CaloDiffusion: bleeding-edge industry models and techniques + particle physics domain knowledge
  - Denoising diffusion architecture; sophisticated objectives, training schedule, sampling algorithm
  - Conditional cylindrical convolutions and GLaM for irregular geometries
- *Leading performance* on virtually every CaloChallenge metric assessed so far
- Scaling up to CMS HGCal:
  - Increases in both dimensionality and irregularity
  - Potential solutions to challenges in modeling sparsity and related quantities:
    - Reduce GLaM compression or use autoencoder-based latent diffusion
    - Dedicated add-on to predict sparsity, similar to LayerDiffusion
- Inference can be improved by reducing number of steps
  - Modifying sampler and/or the model
  - Batched inference on GPU will naturally provide higher throughput than CPU

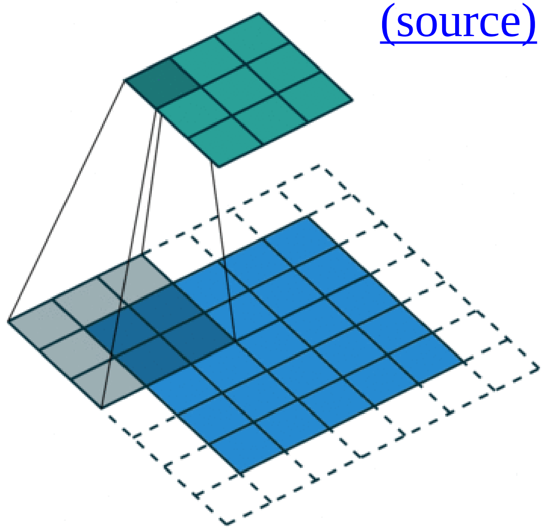
Acknowledgments

- This work was performed with support of the U.S. CMS Software and Computing Operations Program under the U.S. CMS HL-LHC R&D Initiative.
- Additional support provided by the Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy

Backup

Why Convolutions?

(source)



- Convolutions started the modern machine learning revolution (AlexNet, 2012)
  - *Spatial locality* and translational invariance
  - Shared weights → fewer parameters, *better scaling*
  - Highly *efficient* on GPUs: spatial locality implies memory locality
- Ideally suited for computer vision with rectangular images
  - Application to irregular geometries requires innovations
- Graph neural networks?
  - **Pro**: natural representation for irregular geometries
  - **Cons**: adjacency matrices consume substantial memory; operations less local/efficient; hard to generate arbitrary output (masking technique exists, but difficult to scale)
- Point clouds or transformers?
  - **Pro**: no adjacency matrix consuming memory
  - **Con**: discards useful geometric information, which then must be learned from (often sparse) inputs

➢ For generative applications, convolutions still have a lot to offer!
  - And they can keep up with transformers when trained properly… arXiv:2310.16764

Metrics

- Speed only matters if needed accuracy is achieved
  - Wrong answers can be obtained infinitely fast
- Looking at 1D histograms: not good enough!
  - Can miss high-dimensional correlations
- Best category: **integral probability metrics**

$$D_{\mathcal{F}}(p_{\text{real}}, p_{\text{gen}}) = \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{\mathbf{x} \sim p_{\text{real}}} f(\mathbf{x}) - \mathbb{E}_{\mathbf{y} \sim p_{\text{gen}}} f(\mathbf{y}) \right|$$

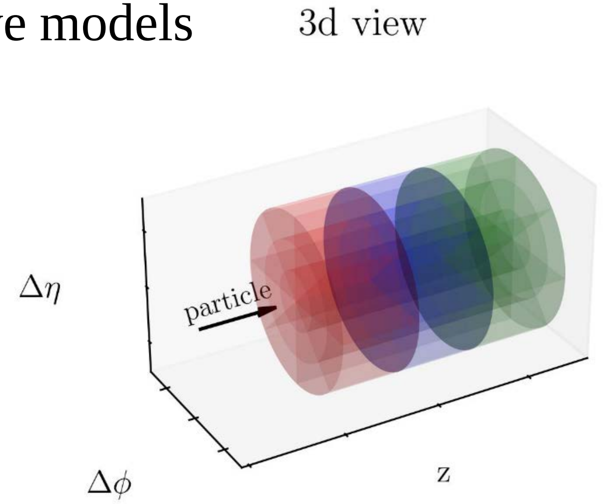
  - *Wasserstein distance* W_1: F is set of all K-Lipschitz functions
    - Only works well in 1D, biased in high-D
  - *Maximum mean discrepancy* (MMD): F is unit ball in reproducing kernel Hilbert space
    - Depends on choice of kernel
  - *Fréchet distance*: W_2 distance between Gaussian fits to (high-D) feature space
    - Features can be hand-engineered or obtained from NN activations
- Another interesting category: *classifier scores*
  - Train NN to distinguish real vs. generated
  - AUC score ranges from 0.5 to 1.0
- *Fréchet Particle Distance* most clearly distinguishes between two similar approaches (message passing GAN and generative adversarial particle transformer)

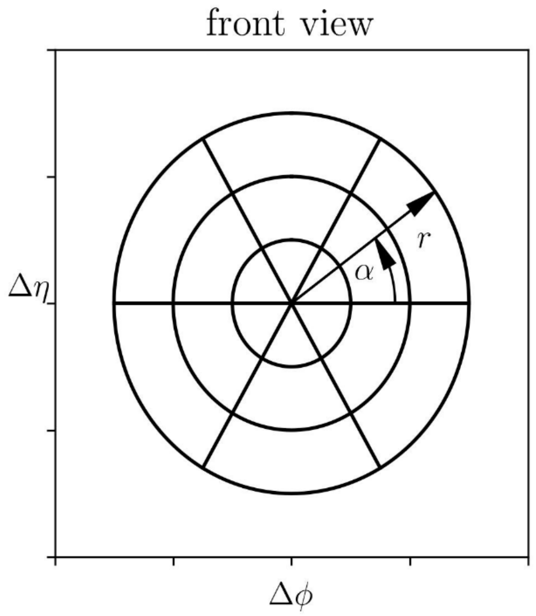
	FPD $\times 10^3$	KPD $\times 10^3$	W_1^M $\times 10^3$
Truth	0.08 ± 0.03	-0.006 ± 0.005	0.28 ± 0.05
MPGAN	$\mathbf{0.30 \pm 0.06}$	$\mathbf{-0.001 \pm 0.004}$	$\mathbf{0.54 \pm 0.06}$
GAPT	0.66 ± 0.09	0.001 ± 0.005	0.56 ± 0.08

arXiv:2211.10295

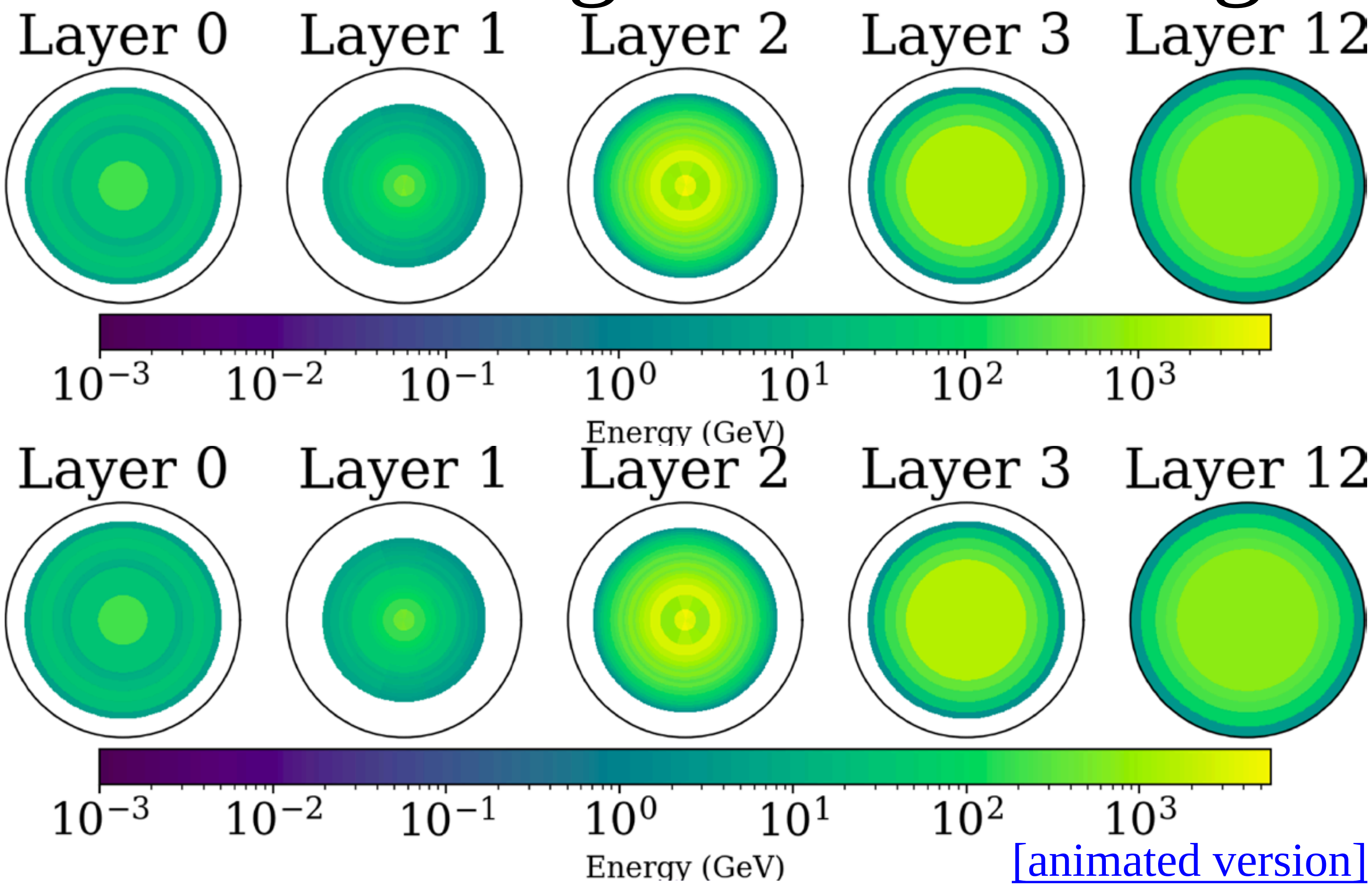
CaloChallenge Datasets

- CaloChallenge: common datasets for evaluation & comparison of generative models
  - Dataset 1: ATLAS calorimeter, irregular
    - Photons (368 voxels), 242K events
    - Pions (533 voxels), 241.6K events
  - Dataset 2: silicon-tungsten, 45 layers
    - Electrons (6480 voxels), 200K events
  - Dataset 3: silicon-tungsten, 45 layers
    - Electrons (40500 voxels), 200K events
- Preprocessing: (E_i = voxel energy)
  - Logit transform: $u_i = \log(x/(1-x))$, $x \equiv \delta + (1 - 2\delta)E_i$
  - Standardization: $u_i' = (u_i - \bar{u})/\sigma_u$





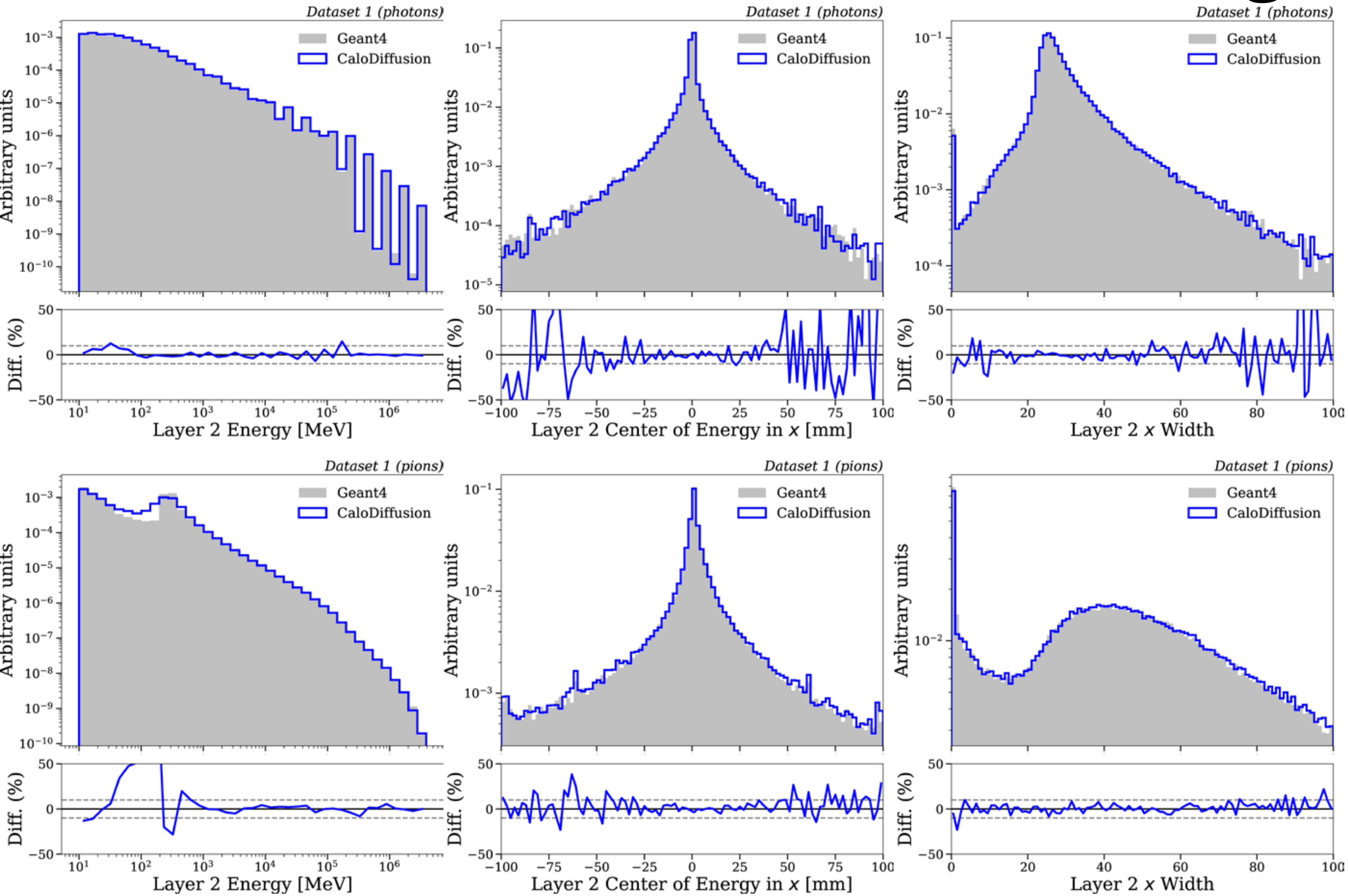
CaloDiffusion: Average CaloChallenge Showers



[animated version]

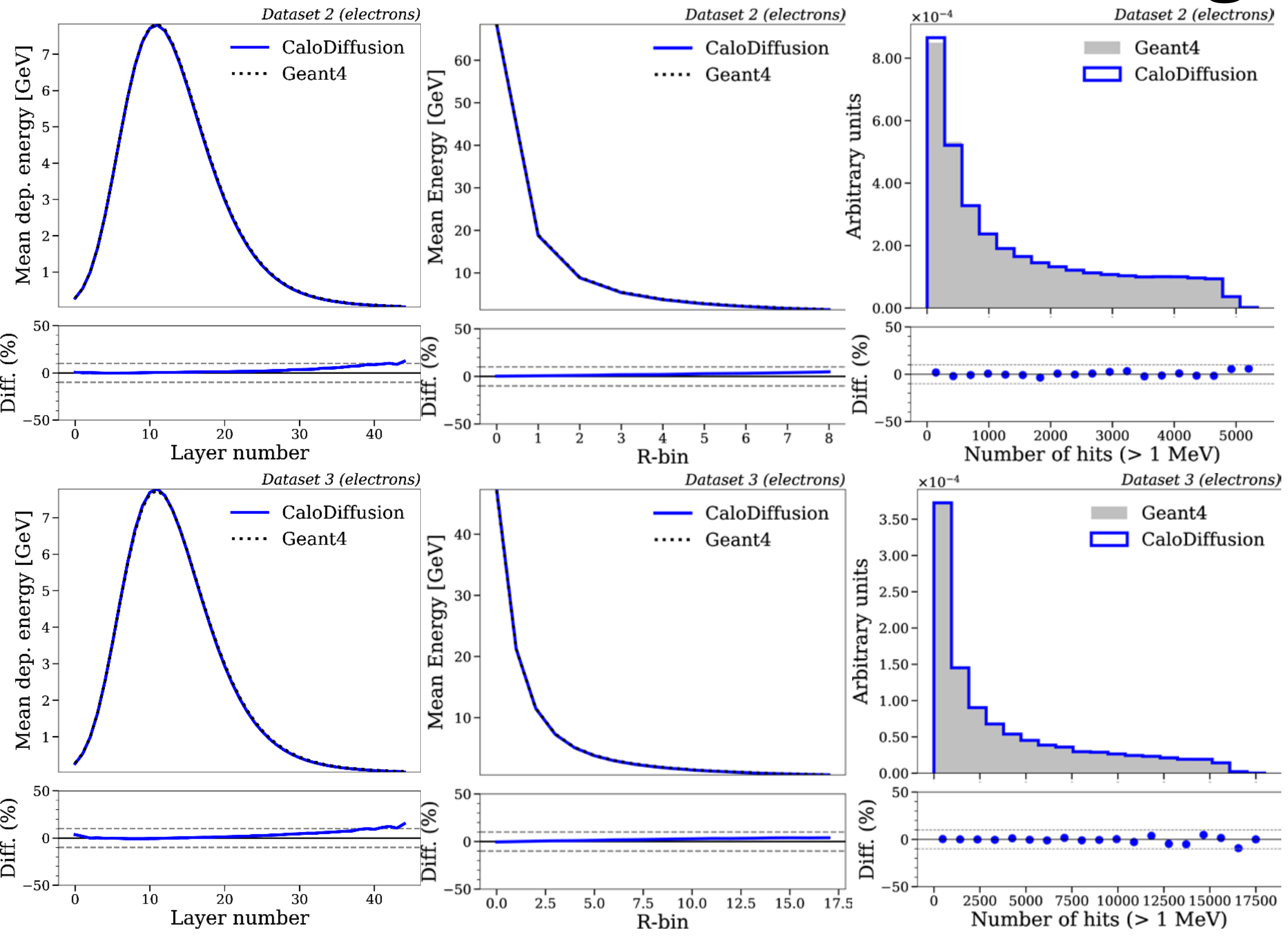
- Top: Geant4; bottom: CaloDiffusion (dataset 1, photons)
  - … or is it the other way around? Can you tell?

CaloDiffusion: CaloChallenge Dataset 1



- Excellent modeling for photon showers
- Some mismodeling of low-energy pions
  - Could be resolved by dedicated training/conditioning
  - No significant impact on shower shape variables

CaloDiffusion: CaloChallenge Datasets 2 & 3



- Very good agreement in shower shapes and physically important quantities
- So far, have only shown 1D comparisons
- Next: further and higher-dimensional quantification

CaloDiffusion: CaloChallenge Metrics

- Classifier AUC: train a binary classifier to distinguish between Geant4 and generative model
  - 2 hidden layers, 2048 neurons each; 20% dropout after each layer
  - Two flavors w/ different inputs: (incident particle energy included in both)
    - Low-level: full showers (all voxels)
    - High-level: energy in each layer, center of energy and shower width in η and φ
  - Compared to CaloScore v2 (undistilled), (i)CaloFlow (teacher)
- Integral probability metrics: Fréchet Particle Distance (FPD), Kernel Particle Distance (KPD)
  - High-level shower features used as input

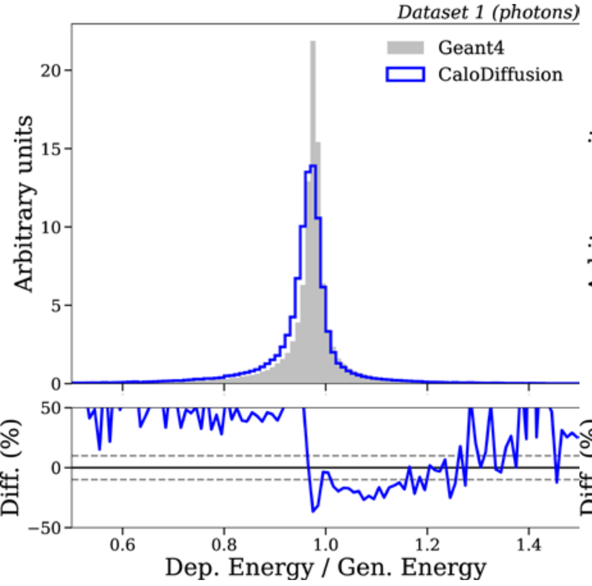
	Classifier AUC (low / high)		
Dataset	CaloDiffusion	CaloFlow	CaloScore v2
1 (photons)	**0.62** / 0.62	0.70 / **0.55**	0.76 / 0.59
1 (pions)	**0.65** / **0.65**	0.78 / 0.70	- / -
2 (electrons)	**0.56** / **0.56**	0.80 / 0.80	0.60 / 0.62
3 (electrons)	**0.56** / **0.57**	0.91 / 0.95	0.67 / 0.85

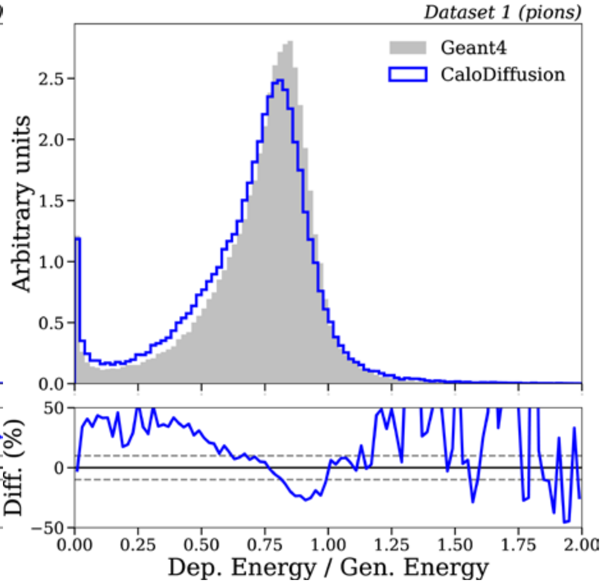
Dataset	FPD†	KPD
1 (photons)	0.014(1)	0.004(1)
1 (pions)	0.029(1)	0.004(1)
2 (electrons)	0.043(2)	0.0001(2)
3 (electrons)	0.031(2)	0.0001(1)

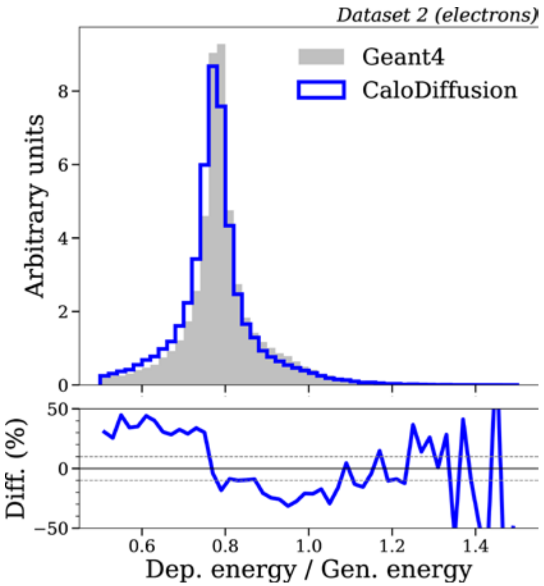
- CaloDiffusion wins in almost all comparisons, with very small distance values
  - Generated showers almost indistinguishable from Geant4
  - Further comparisons to come in CaloChallenge summary

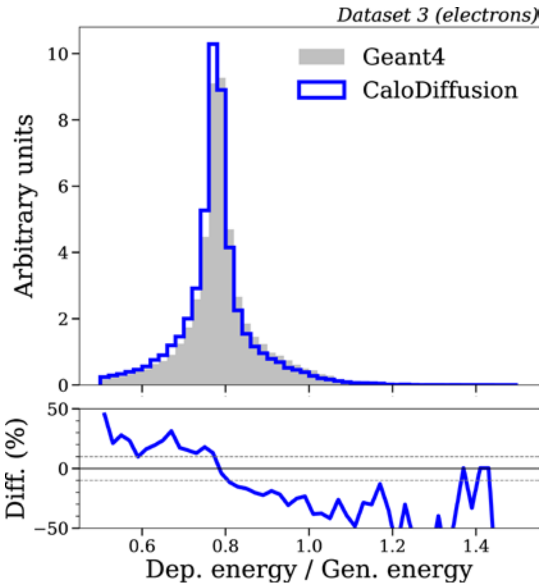
† Geant4 self-comparison values subtracted (0.008, 0.0005, 0.008, 0.011)

CaloDiffusion: Areas for Improvement

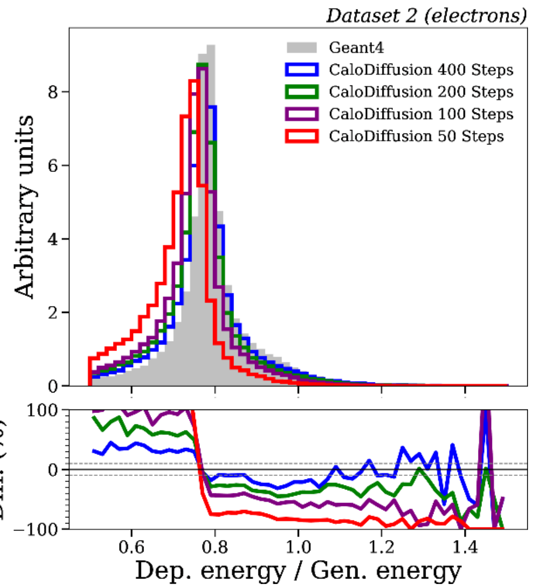








- Deficit in total energy modeling
- Need 400 diffusion steps to get acceptable quality
  - o Still faster than Geant4 (~100s) w/ batching on GPU
- Fewer steps:
  - o Linear speed improvement
  - o But even less accurate in this quantity

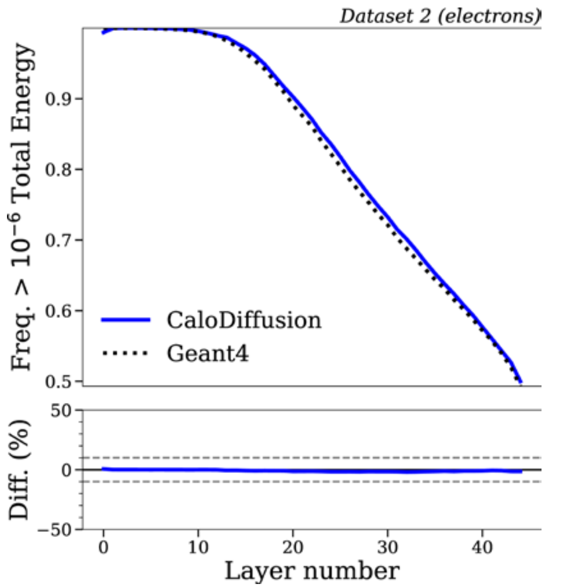


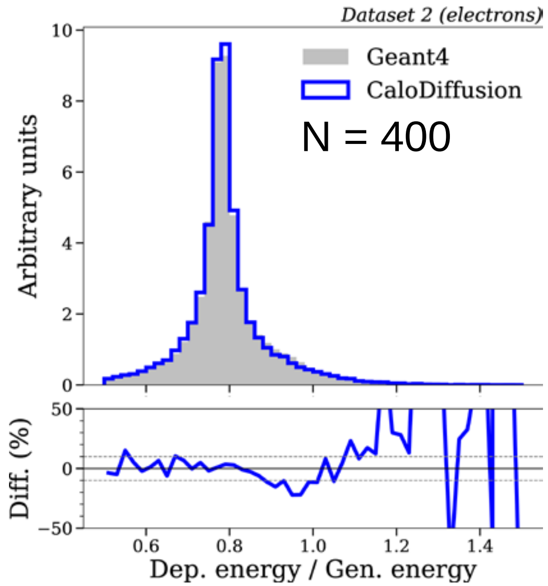
Dataset	Batch Size	Time/Shower [s] CPU	Time/Shower [s] GPU
1 (photons)	1	9.4	6.3
(368 voxels)	10	2.0	0.6
	100	1.0	0.1
1 (pions)	1	9.8	6.4
(533 voxels)	10	2.0	0.6
	100	1.0	0.1
2 (electrons)	1	14.8	6.2
(6.5K voxels)	10	4.6	0.6
	100	4.0	0.2
3 (electrons)	1	52.7	7.1
(40.5K voxels)	10	44.1	2.6
	100	-	2.0

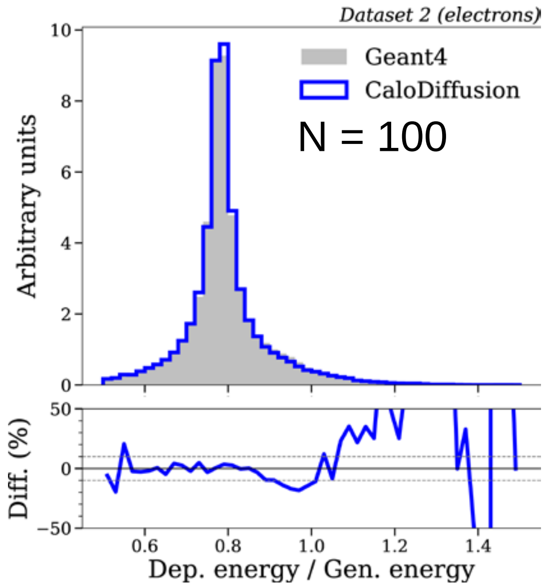
Num. Steps	Classifier AUC (low / high)	FPD	E Ratio Sep. Power
400	0.56 / 0.55	0.043(1)	0.011
200	0.61 / 0.56	0.046(1)	0.036
100	0.69 / 0.59	0.065(3)	0.079
50	0.83 / 0.67	0.110(4)	0.251

Improvement: More Diffusion!

- Train LayerDiffusion to predict energy deposited per layer (1D diffusion)
  - Negligible inference time (200 steps) compared to CaloDiffusion
- Normalize CaloDiffusion output based on LayerDiffusion
  - Only if both models predict sufficiently non-zero deposited energy in a layer
- ➢ Substantial improvement in total energy modeling
- Number of CaloDiffusion steps can be reduced with no loss of quality
  - 4× speedup for Dataset 2! (8× for Dataset 1 & improves low-energy pions)

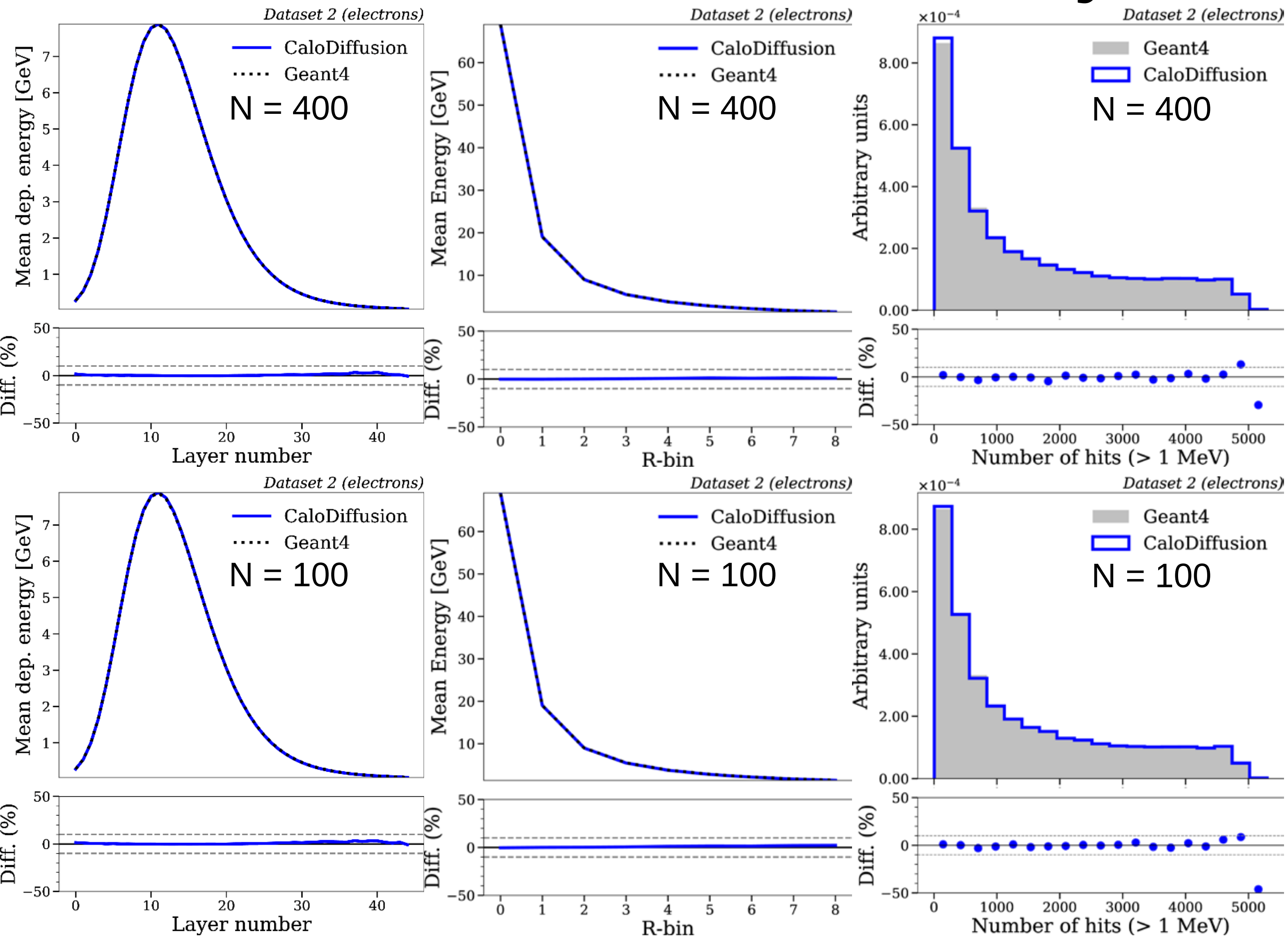






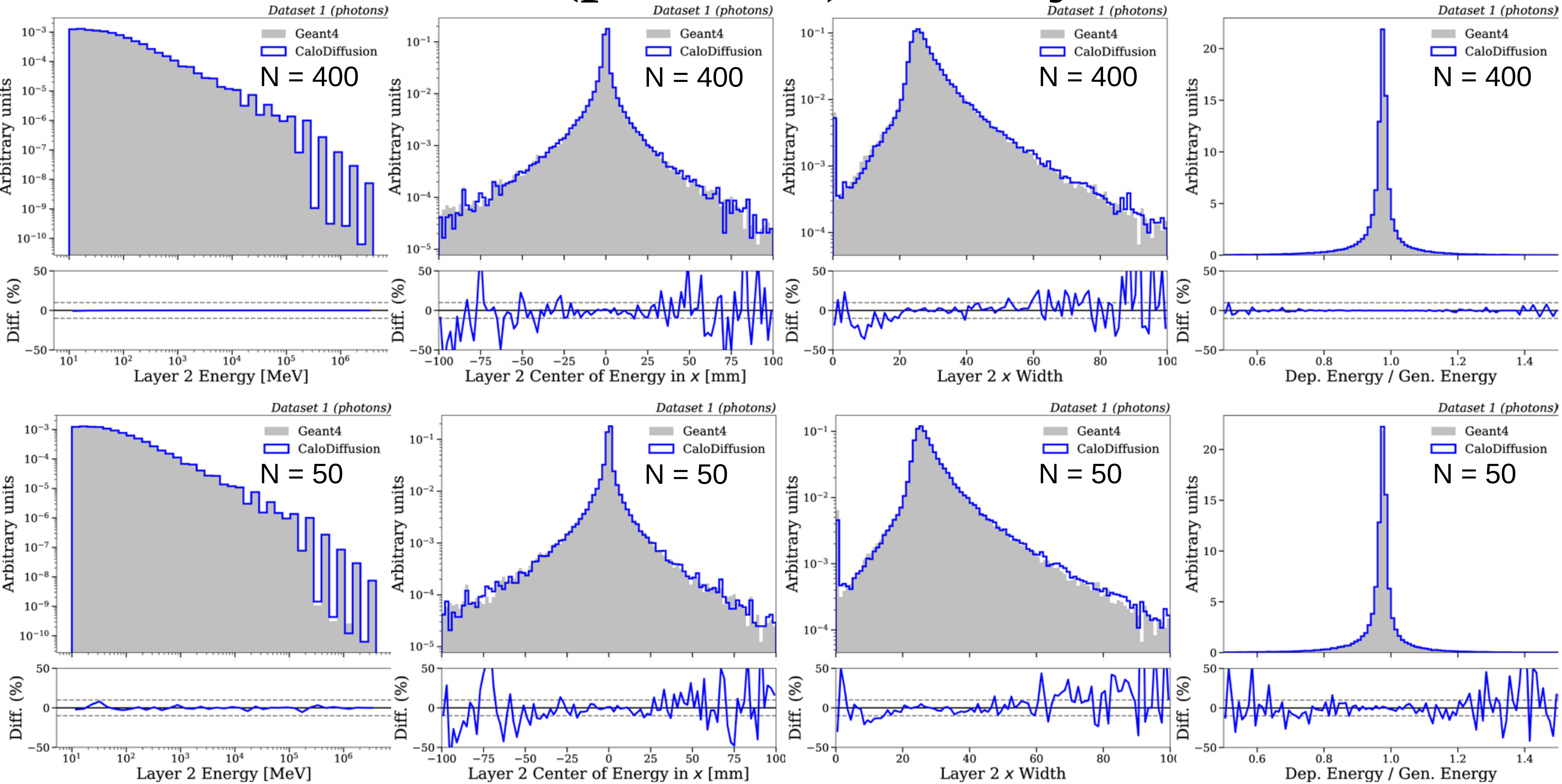
Model (2, electrons)	AUC (low / high)	FPD	KPD	E Ratio Sep. Power
Orig. (N = 400)	0.56 / 0.56	0.043	0.0001	0.011
Layer (N = 400)	0.54 / 0.58	0.045	0.00005	0.0017
Layer (N = 100)	0.54 / 0.60	0.076	0.0003	0.0017

Dataset 2 w/ LayerDiffusion



- Virtually indistinguishable for 4× fewer diffusion steps
- Improved agreement vs. original CaloDiffusion

Dataset 1 (photons) w/ LayerDiffusion



Dataset 1 (pions) w/ LayerDiffusion