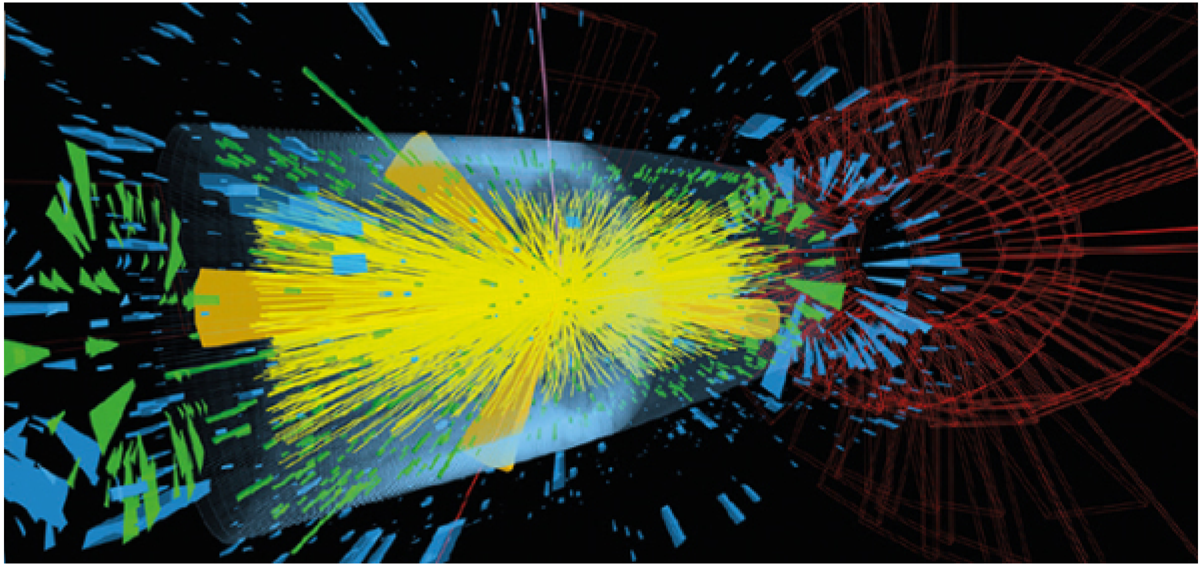
Fast Calorimeter Simulation



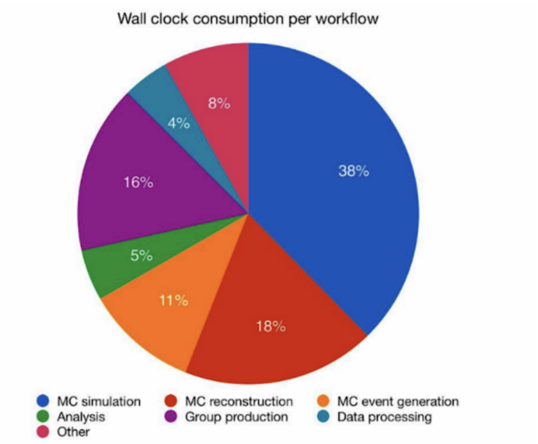
Oz Amram

Fermilab

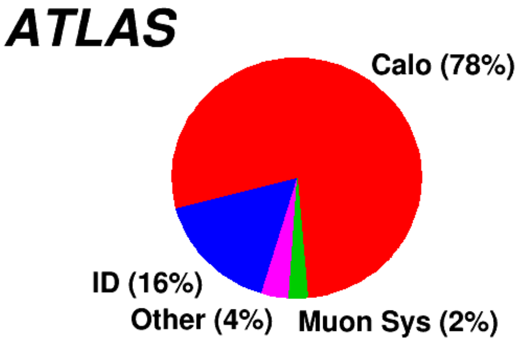
Based on 2308.03876

FastML
Oct. 15, 2024

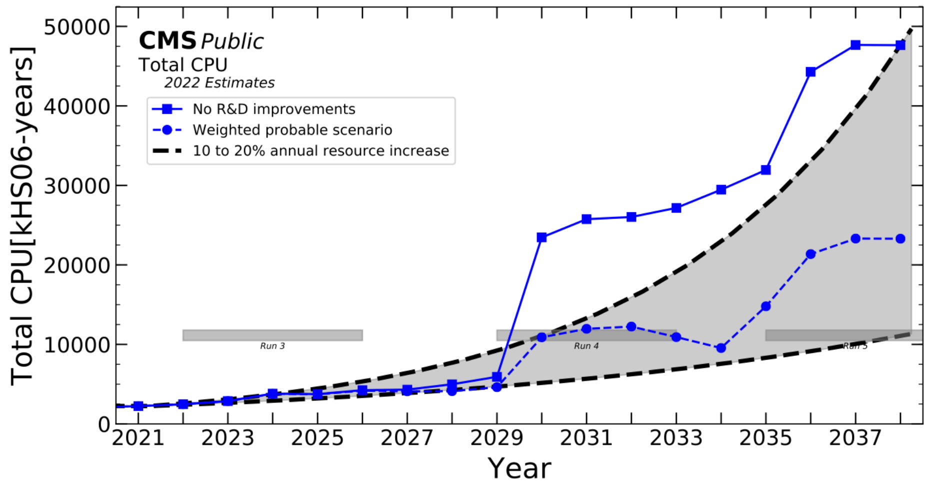
The Need for Fast Simulation



ATLAS CPU hours used by various activities in 2018

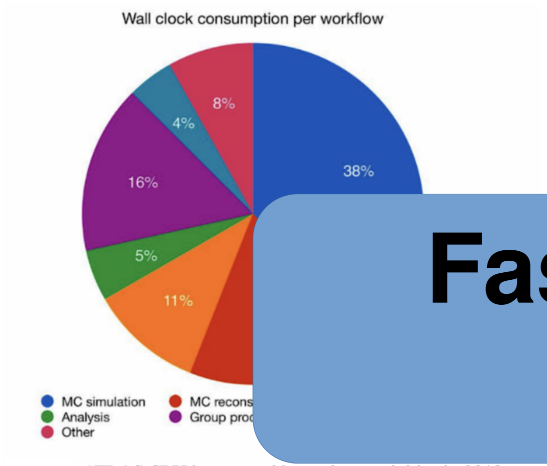


Subdetector CPU fraction for 50 ttbar events
MC16 Candidate Release

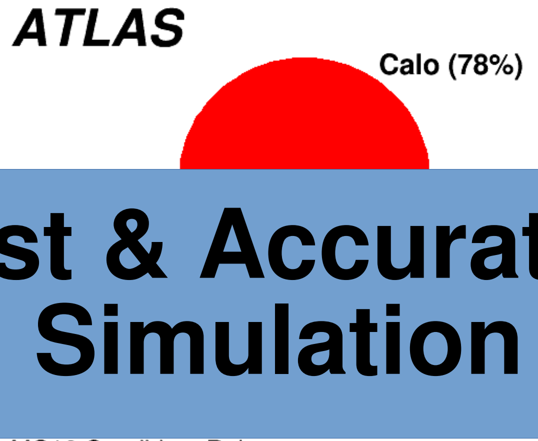


- Calorimeter simulation is a significant part of LHC computing
- For HL-LHC, computing simulation more crunched

The Need for Fast Simulation



ATLAS CPU hours used by various activities in 2018

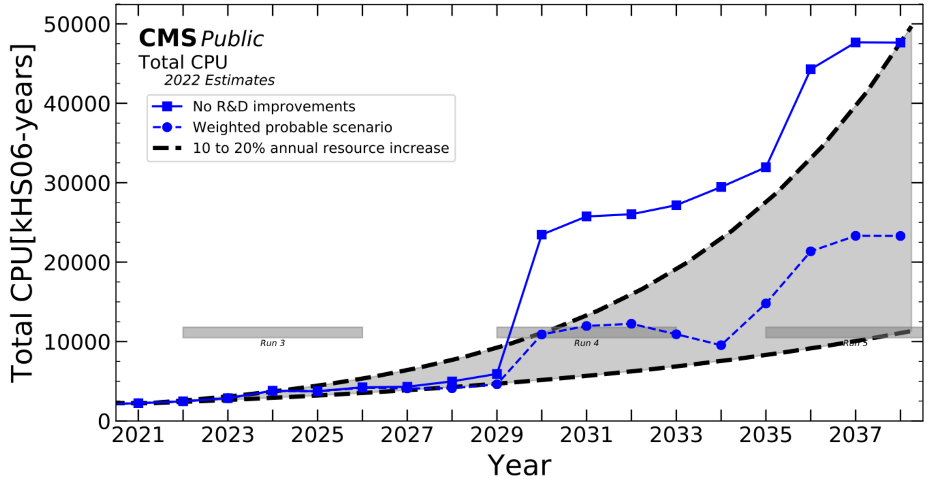


MC16 Candidate Release

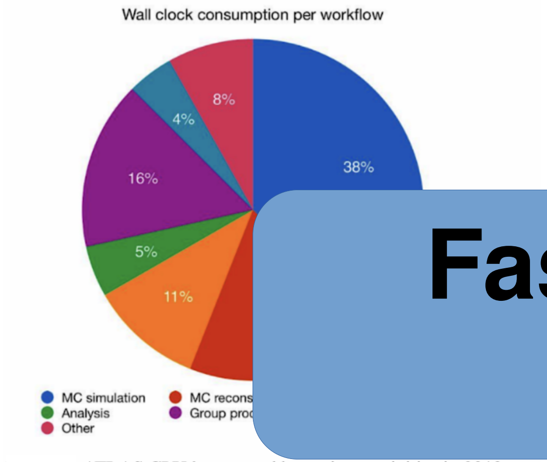
- Calorimeter simulation is a significant part of

Fast & Accurate Calorimeter Simulation is Needed!

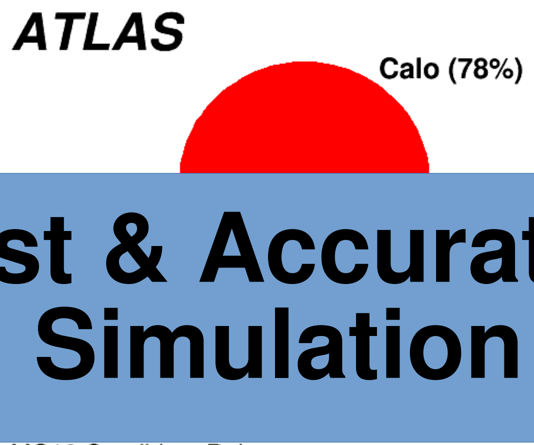
- For HL-LHC, computing simulation more crunched



The Need for Fast Simulation



ATLAS CPU hours used by various activities in 2018

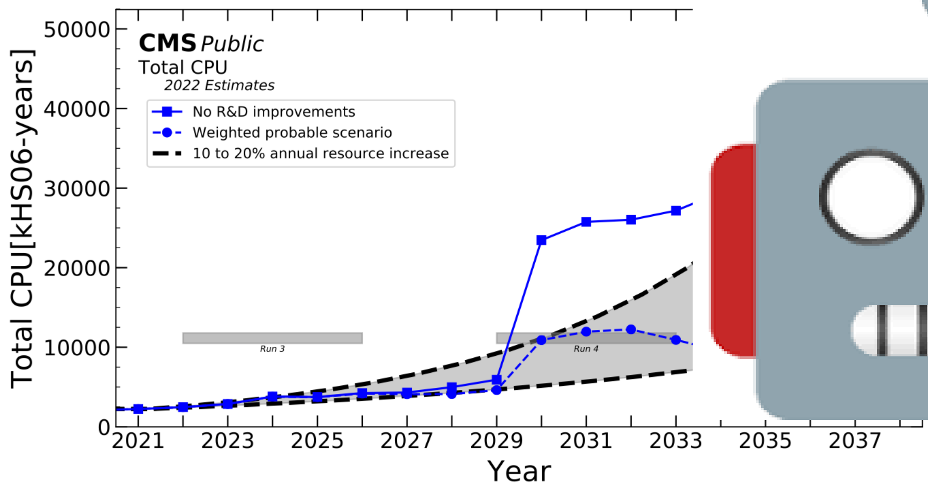


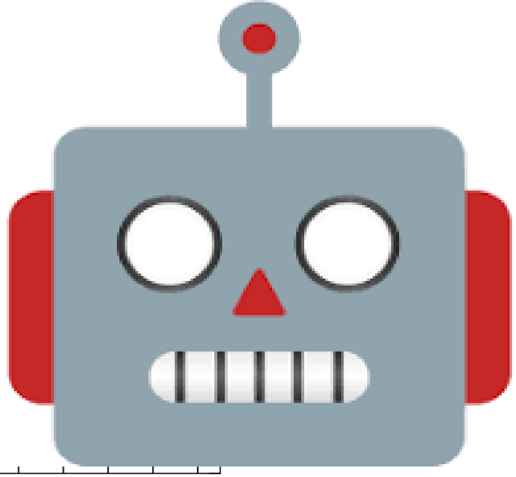
MC16 Candidate Release

- Calorimeter simulation is a significant part of

Fast & Accurate Calorimeter Simulation is Needed!

- r HL-LHC, computing ulation more nched

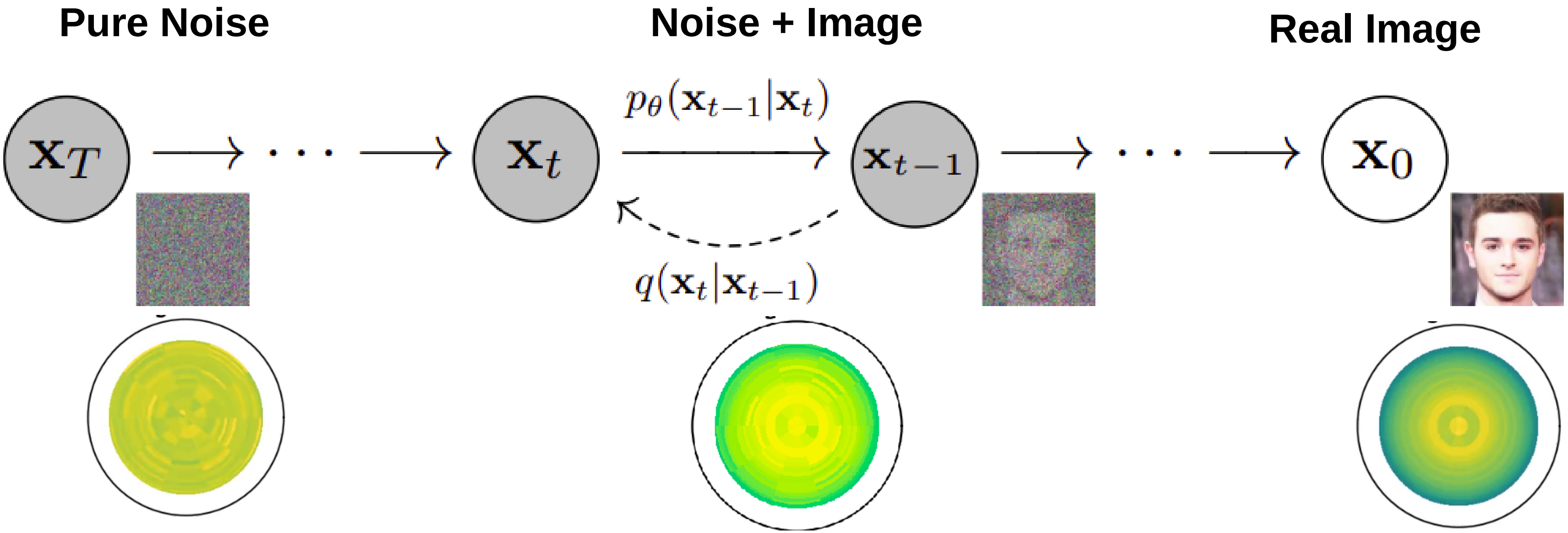




Model

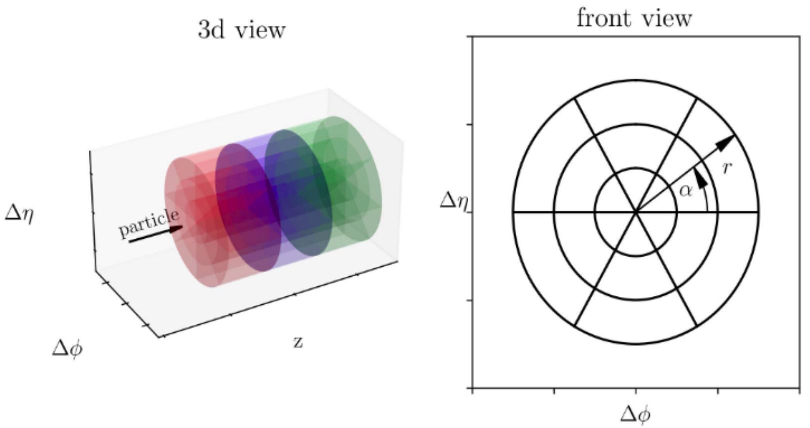
Goal : Train a generative ML model to mimic physics based simulation (Geant) with high accuracy & significant speedup

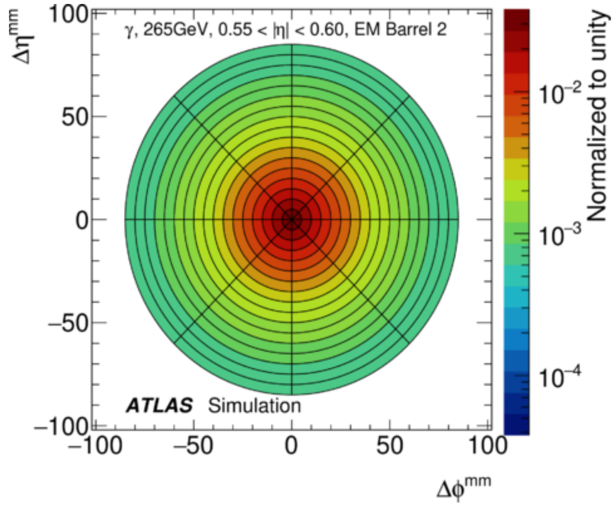
'CaloDiffusion'



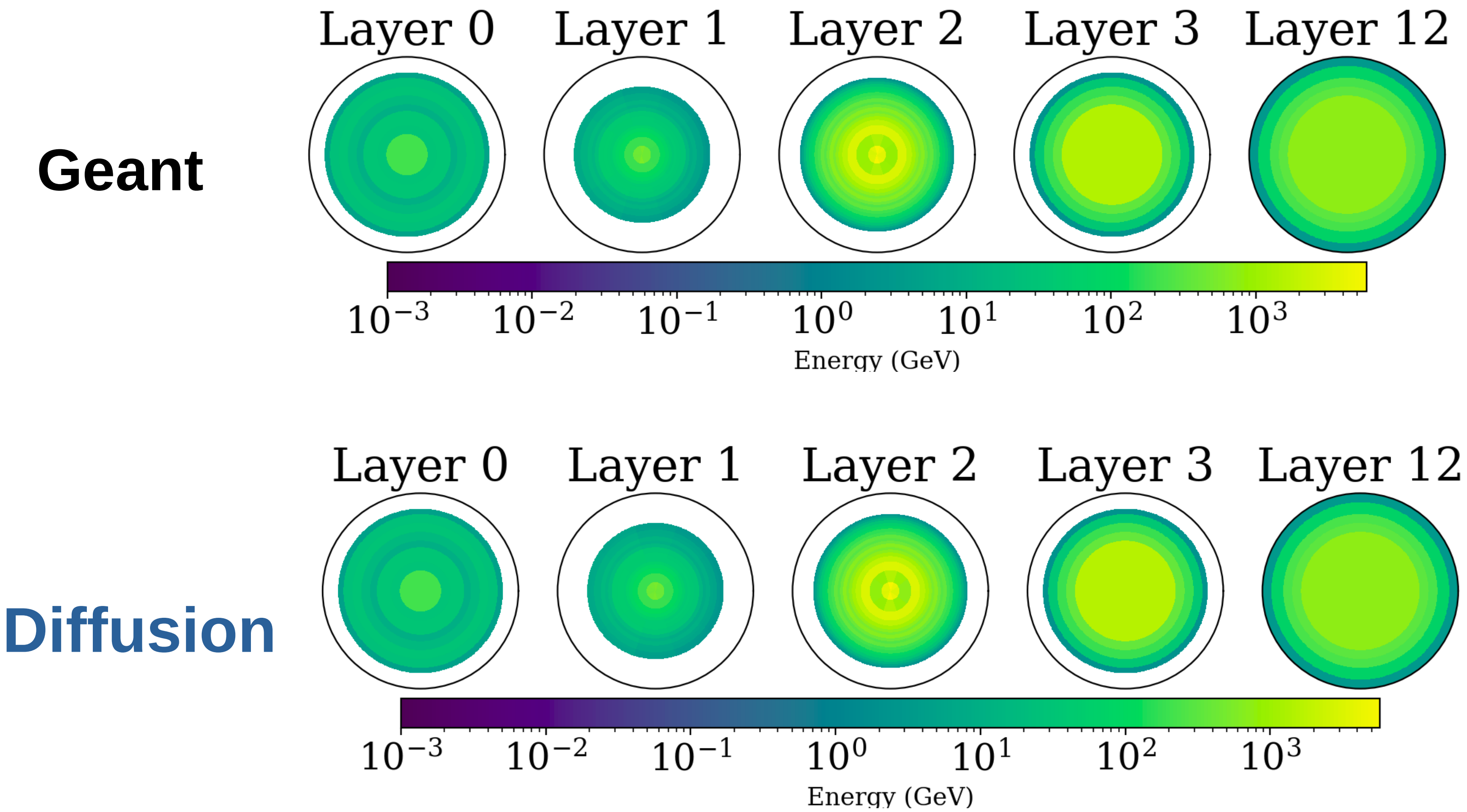
Dataset: Calo Challenge

- Community challenge to compar generative models for Calorimet simulation
- Standard datasets to allow comparison
  - Dataset1: ATLAS-like geometry
  - Dataset2: 45 layers, 6480 total voxels
  - Dataset3: 45 layers, 40,500 total voxels

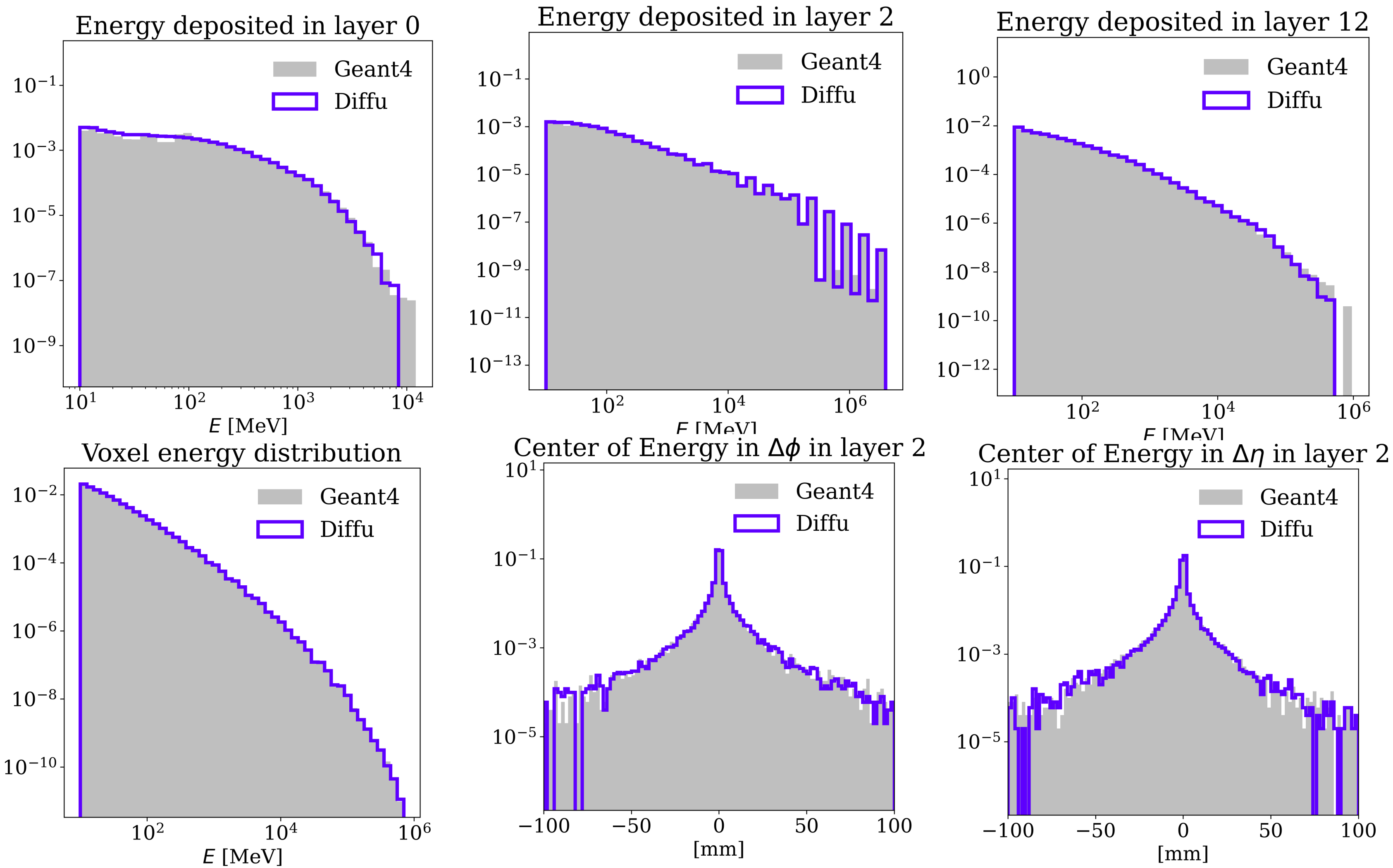




Average Showers

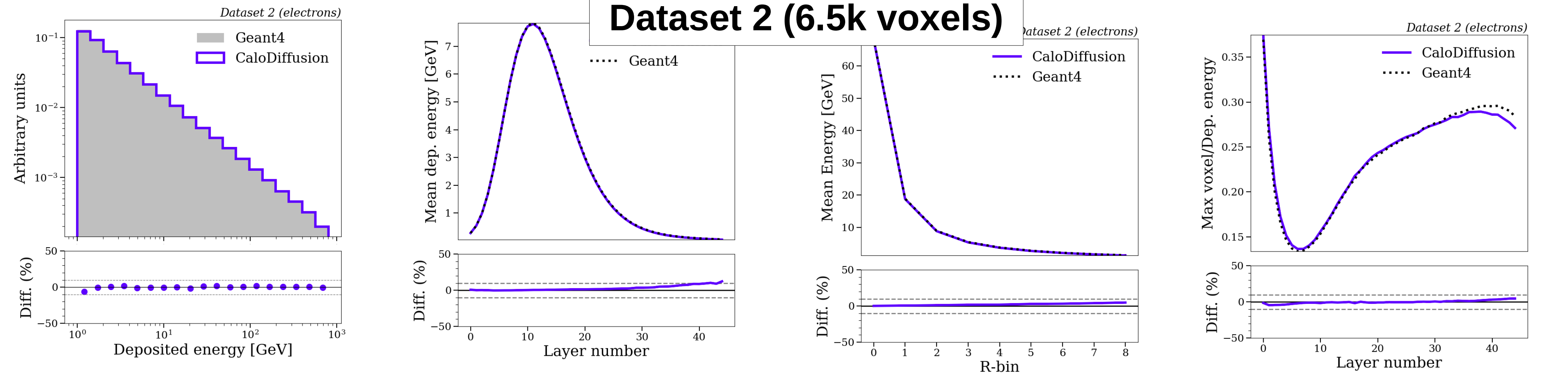


Dataset 1 Results

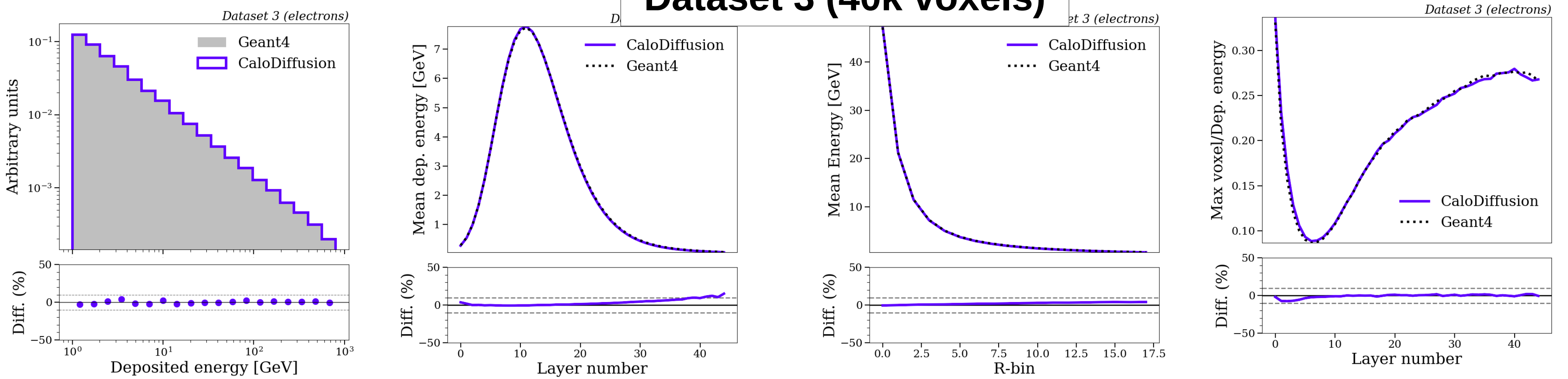


Results: Datasets 2 & 3

Dataset 2 (6.5k voxels)

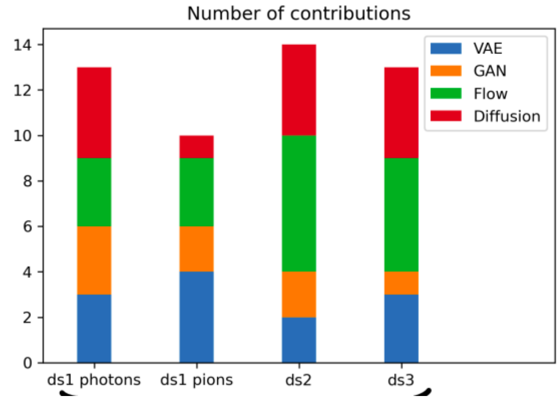


Dataset 3 (40k voxels)



Comparisons to Other Approaches

- CaloChallenge now concluded → comparisons of all approaches



50 total submissions

- CaloDiffusion achieved the highest quality results

Some complicated evaluation metric...

CaloDiffusion	-2.4895 ± 0.1255	1.
conv. L2LFlows	-2.9404 ± 0.1424	5.
CaloINN	-2.9061 ± 0.1459	4.
MDMA	-4.3483 ± 0.4573	11.
Calo-VQ	-8.1542 ± 0.3924	14.
CaloScore	-2.7967 ± 0.0569	3.
CaloScore distilled	-3.2482 ± 0.1576	7.
CaloScore single-shot	-3.4612 ± 0.2108	9.
iCaloFlow teacher	-3.1657 ± 0.1583	6.
iCaloFlow student	-3.6658 ± 0.2291	10.
SuperCalo	-2.7070 ± 0.1180	2.

Full results here

Fast?

- Geant ~100s/shower
- Diffusion time per shower depends on batch size
- Up to **1000x speedup** if run on GPU's
- Further work improving this even more
  - Better model → less diffusion steps → linear speedup

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

Outlook

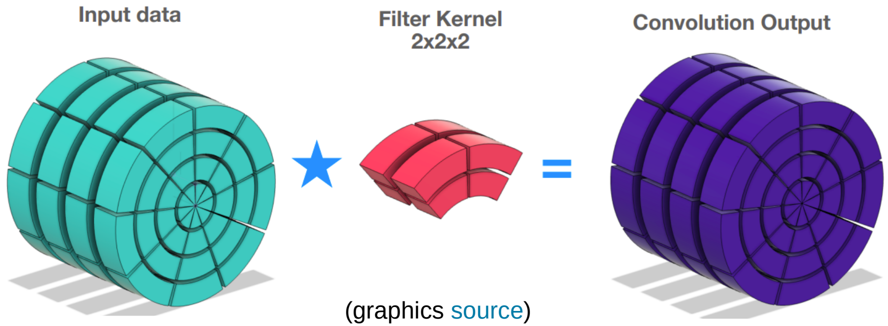
- ML models can be a fast & accurate substitute for calorimeter simulations!
- **CaloDiffusion** leading is leading approach
- Public results on CMS HGCal soon!

Backup

Optimizing for Cylindrical Data

- Regular convolutions assume pure translation symmetry
- Our data : phi is **periodic**, and R & Z **not translation invariant**

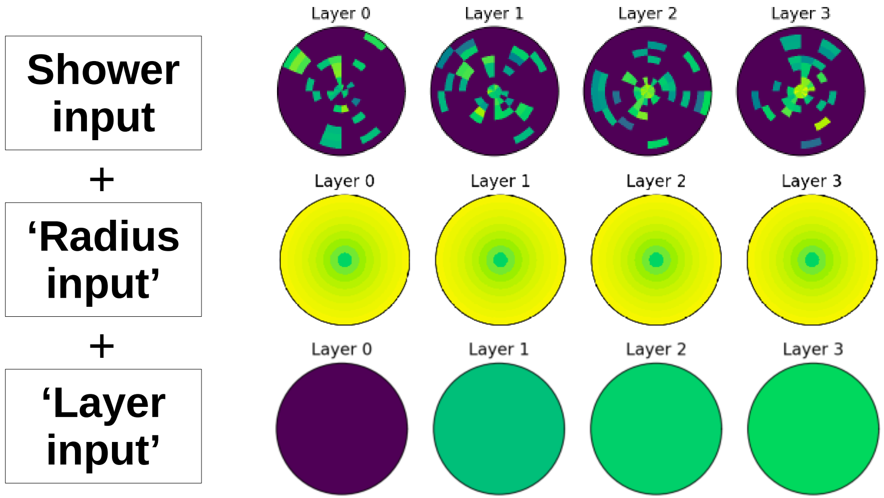
Implement **cylindrical convolutions** to respect periodic boundary of phi



(graphics source)

'Circularly' pad phi dimension before 3D conv

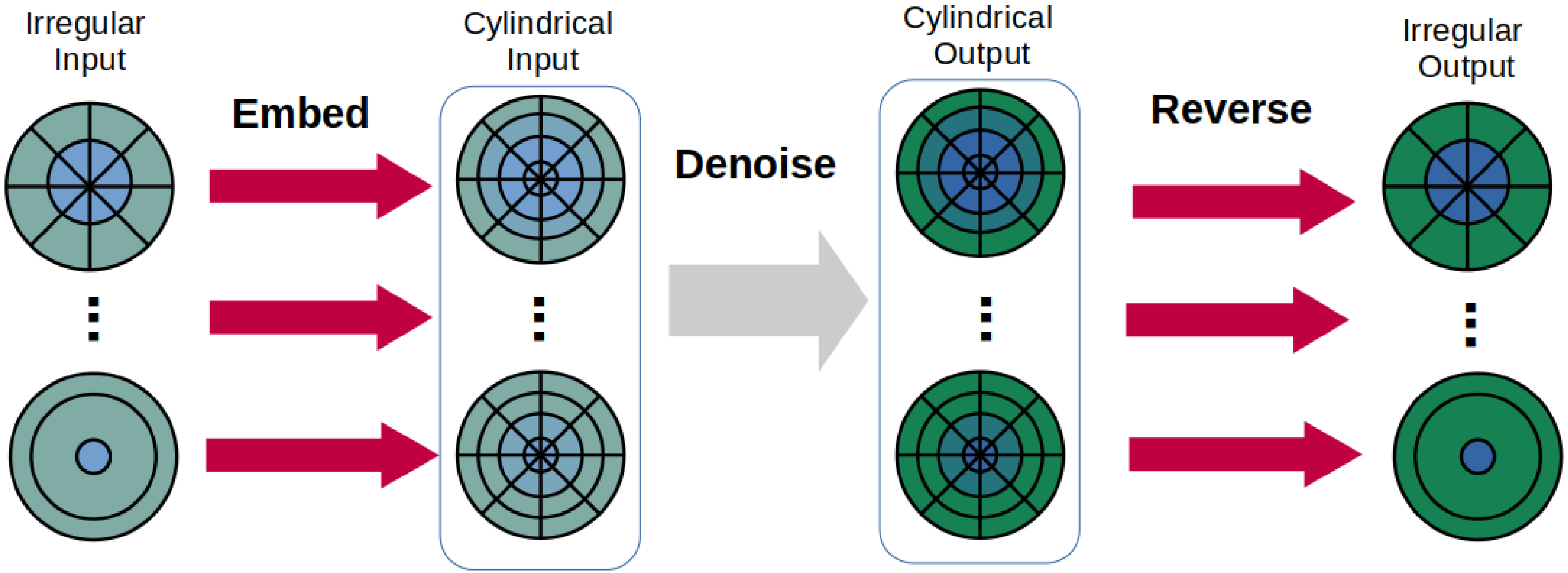
Allow convolutions to be **conditional on R & Z** by using additional channels



Additional input channels

Embedding Irregular Geometries

- Dataset 1 (ATLAS detector) is cylindrical but has **irregular structure** in layers
  - Different radial / angular bins in each layer → can't apply cylindrical convolutions
  - Previous approaches have used fully connect networks or very large 1D CNN's
- Learn an **embedding** that maps input into **regular cylindrical structure**



Quantifying Performance

- Train a NN classifier to distinguish between Geant showers and CaloDiffusion showers
- Quantify sample quality based on AUC on holdout set
  - AUC → 0.5 means synthetic showers are indistinguishable from 'truth'

	Dataset 1 (ATLAS-like)	Dataset 2	Dataset 3
Classifier AUC *	**~0.65**	**~0.55**	**~0.55**

AUC much less than 1 → Very similar showers!

Additional metrics in backup

Outlook

- Using diffusion models we can generate synthetic calorimeter showers
- Utilized several optimizations for **cylindrical** calorimeter geometries & new **embedding** approach for irregular shapes
- Classifiers struggle to distinguish between Geant & **CaloDiffusion** showers
- Future work: improve generation time, more complicated geometries
- See paper for more details,
  - Code and pretrained models available on github

Technical Details

- ‘logit’ transformation of voxel energies and then standard scale to zero mean and unit variance
  - Correct preprocessing important for diffusion process, related to scale of added noise
- Denoising network uses ‘U-net’ architecture with cylindrical convolutions
  - Two conditional inputs : shower energy and diffusion step
  - ~400k params for dataset1 and 2, 1.1M for dataset3
- 400 diffusion steps, ‘cosine’ noise schedule (2102.09672)
- Choices for training objective:
  - Datasets 1 and 2 : Network is trained to predict noise component of image
  - Dataset 3 : Network trained to predict weighted average of noise component and un-noised image,
    - More stable, recommended by 2206.00364
- Sampling uses DDPM algorithm (2006.11239)

Additional Metrics

- Distance metrics:
  - Frechet Particle Distance and Kernel Particle Distance (proposed in 2211.10295)
    - Use implementation proposed for CaloChallenge, based on high level shower features
  - We find that the computation of FPD is slightly biased, ie non-zero values even comparing different random samples of Geant to each other
  - Compare scores for Diffu-Geant (D-G) vs Geant-Geant (G-G)

	Dataset 1 (ATLAS-like)	Dataset 2	Dataset 3
FPD (D-G / G-G)	**0.035 / 0.008**	**0.095 / 0.008**	**0.275 / 0.011**
KPD (D-G / G-G)	**0.007 / 0**	**0.0001 / 0**	**0.0007 / 0**

Embedding Details

- First find superset of all radial/angular bins → embedding space
- For each layer, embedding in radial dimension is an M_i x M_* matrix
  - M_i (M_*) is number of radial bins in layer i (embedding space)
  - Initialize weights be proportional to area overlap of bins + 10^-3 * Gaussian noise
- Reverse matrix is M_* x M_i, initialized to pseudo-inverse of embedding matrix
- For now, enforcing phi symmetry, energy is split evenly among phi bins (not learnable)
- Found small benefits of conditioning on phi in addition to R & Z
  - There is slight non-uniformity in phi in the energy distributions of dataset1