Attention to Mean-Fields for Calorimeter Simulation

Artwork by DALL – E · 2

CaloChallenge - 30.06.2023

Benno Käch, Isabell Melzer-Pellmann, Moritz Scham, Simon Schnake, Alexi Verney-Provatas, Lucas Wiens, Frederic Engelke, Valle Varo, Soham Bhattacharya, Dirk Krücker

HELMHOLTZAI

UH

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

DESY.

Preliminary Note

- Started working on this 7 days before the deadline of the abstract submission → please excuse any lack of knowledge about calorimeters
- Large improvement possible by more sophisticated choice of data representation
- Model mostly based on recent publication on JetNet

My unsolicited & controversial opinion on generative models for detector simulation:

- Point Clouds crucial to handle sparsity and irregularity of detector
- VAE's: No meaningful reconstruction loss for point clouds
- Flows: Isomorphic constraints not viable for point clouds → padding not an option
- Diffusion models (amateur opinion): no meaningful matching between points in denoising steps + speed
- **Only GANs remain!**

Particle Cloud Representation of Calorimeter Hits

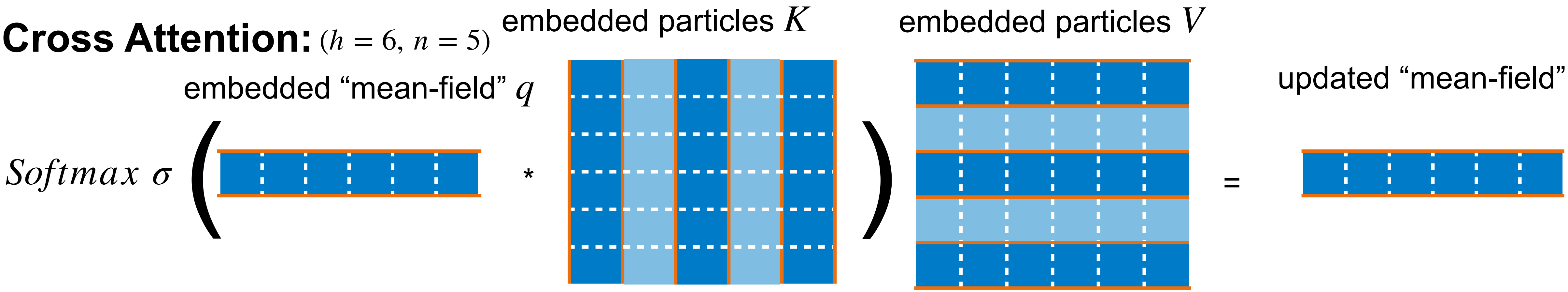
- Handle sparsity in the detector by representing hits as Particle Cloud [1]
- Reduces batch dimension to (batch size,~4k hits,4) → fits on one GPU
- Model learns physics ~decoupled of detector geometry
- Mapping detector cells to point clouds:

```
hits=detector[E>0]
for coordinate  in (z, alpha, R):
    for cellnumber in enumerate(cells):
            coordinate(cellnumber) x=cellnumber + U(0,1)
    x=MinMaxScaler(x)
    x=Logit(x)
    x=StandardScale(x)
for hit in hits:
    E(hit)=BoxCoxTransform(E(hit))
```

- Gives about Gaussian distribution for variables (except α)
- α periodcity → problematic
- Volume of space **not** respected by particle cloud definition

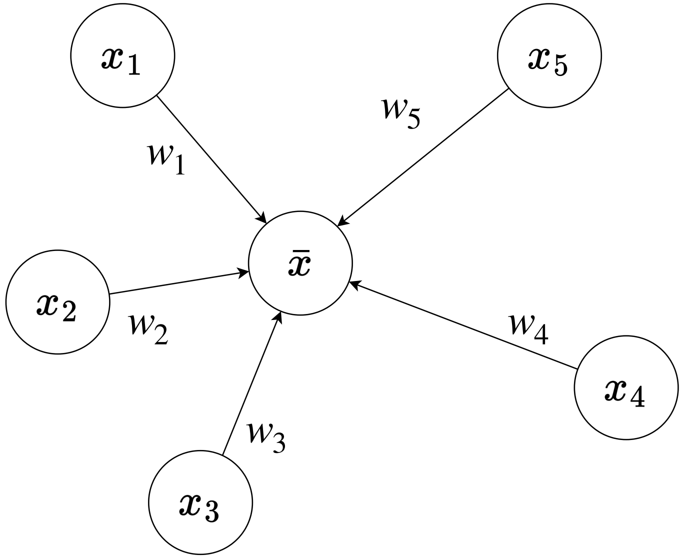
Main Information Aggregation Mechanism: Cross Attention

Cross Attention: $(h = 6,\ n = 5)$



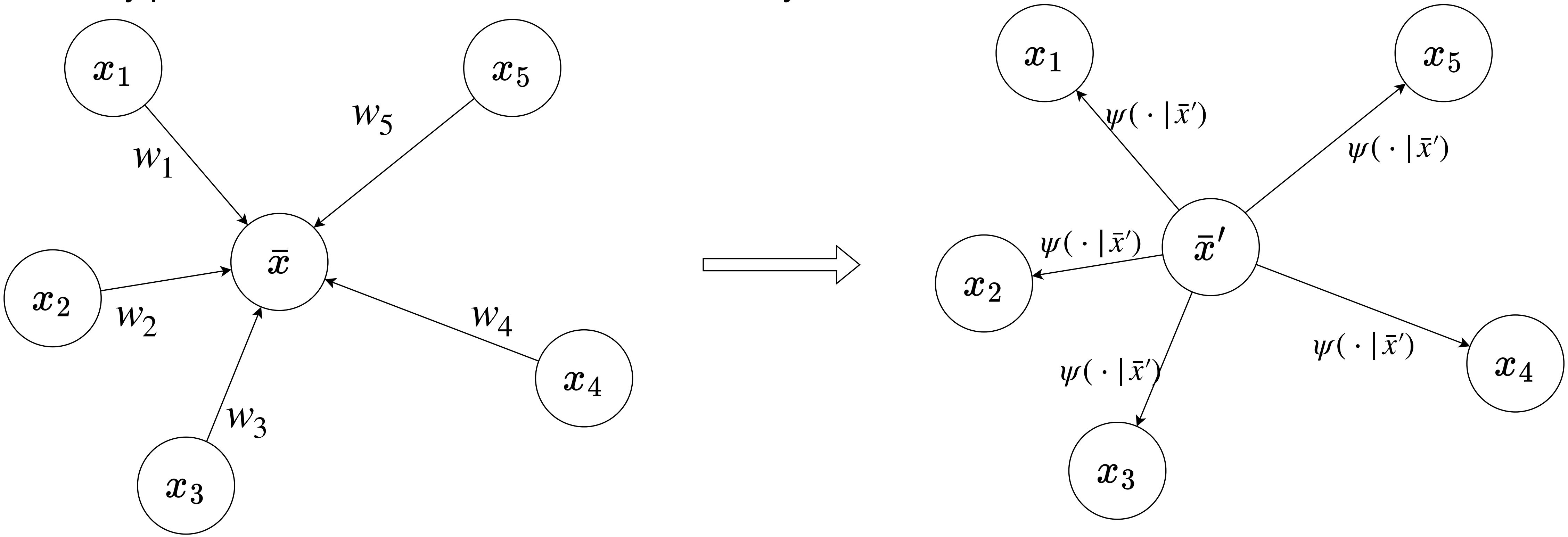
- "mean-field" $q \in \mathbb{R}^{1\times h}$, h hidden dimension
- K: n embedded particles $K = (W_K(x))^T, x \in \mathbb{R}^{n\times 4}, W_K \in \mathbb{R}^{h\times n}$
- V: n embedded particles, $V = W_V(x), x \in \mathbb{R}^{n\times 4}, W_V \in \mathbb{R}^{h\times n}$

$$\bar{\mathbf{x}}' = \sigma\left((\mathbf{q}\cdot K)/\sqrt{h}\right) V = \sum_{i=1}^{n} w_i W_V \mathbf{x}_i$$



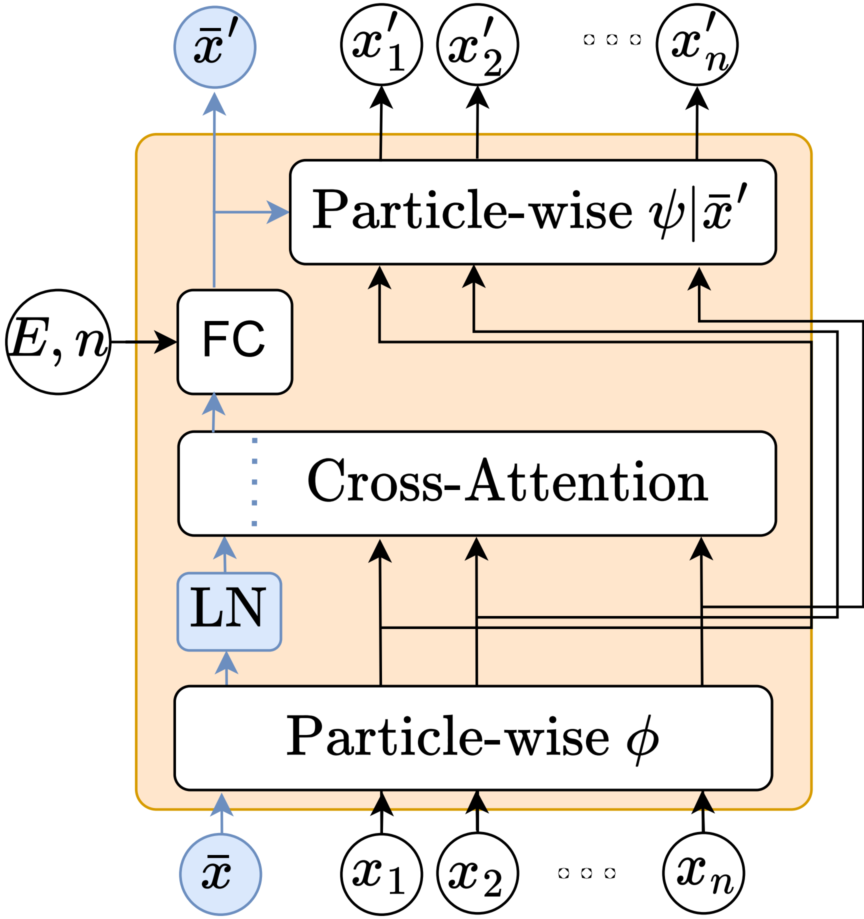
Mean-Field Interaction

- Every particle updates mean-field with dynamic contribution
- Mean-field updates particles via particle-wise layer, conditioned on mean field
  → Every particle interacts with mean-field individually



Main Block of Architecture

- Architecture motivated by Transformer Encoder architecture used for Particle Cloud Generation [2]
- IN: embedded particles $x_i \in \mathbb{R}^l$, embedded mean-field $\bar{x} \in \mathbb{R}^l$
- OUT: embedded particles $x_i \in \mathbb{R}^l$, embedded mean-field $\bar{x} \in \mathbb{R}^l$
  1. Particle-wise $\phi : \mathbb{R}^l \rightarrow \mathbb{R}^h$
  2. Layer Norm applied to mean-field
  3. Multi Headed Cross-Attention between mean-field and particles
  4. Energy & shower multiplicity conditioned fully-connected layer updates mean-field
  5. Particle-wise FC $\psi : \mathbb{R}^l \rightarrow \mathbb{R}^h$ conditioned on mean-field proxies particle-particle interaction
- Not shown here: residual connection between in/out particles & mean-field
- Permutation equivariant

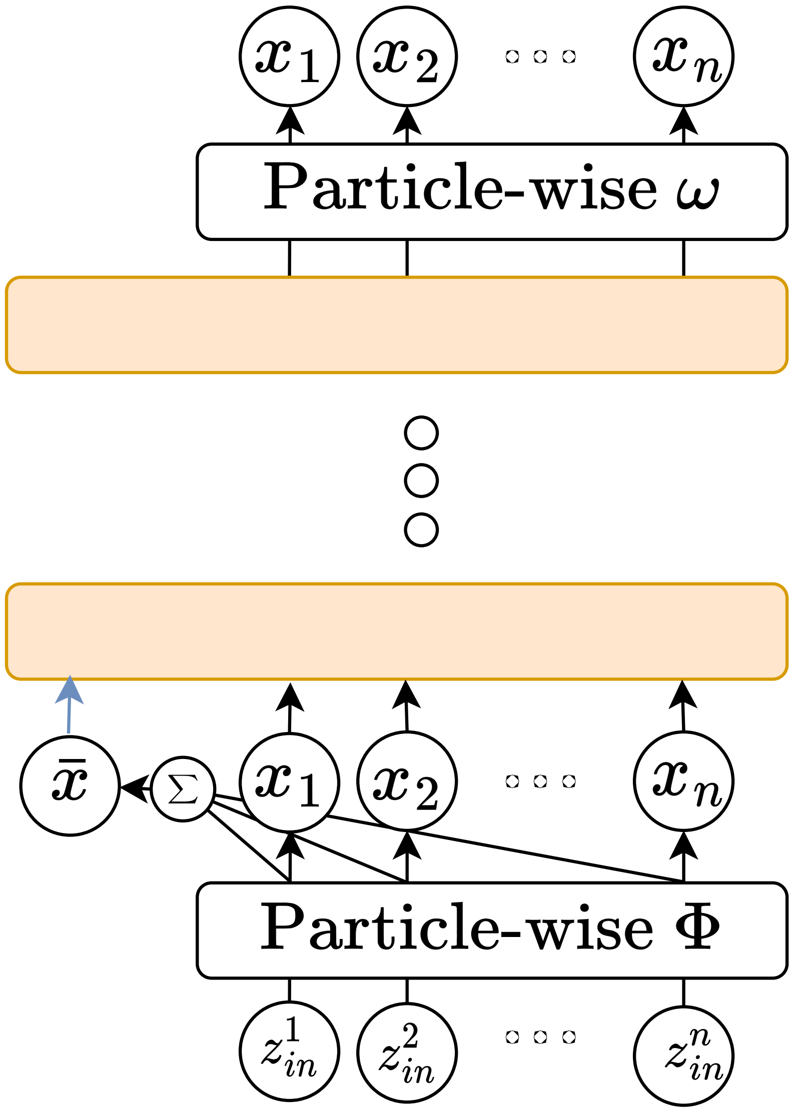


 [2] Kaech et al., 2022

Generator Architecture

- IN: $x_i = \Phi(z) \in \mathbb{R}^l, z \sim N(0,1) \in \mathbb{R}^{n\times4}$
- OUT: $x' \in \mathbb{R}^{n\times4}$
- Before first block:
    - Noise embedded to latent space with Φ to dimension $\mathbb{R}^l$, $l = 16$
    - Mean-field initialised as $\bar{x} = \frac{1}{c_N}\sum_{i=1}^{n}\Phi(z_i)$

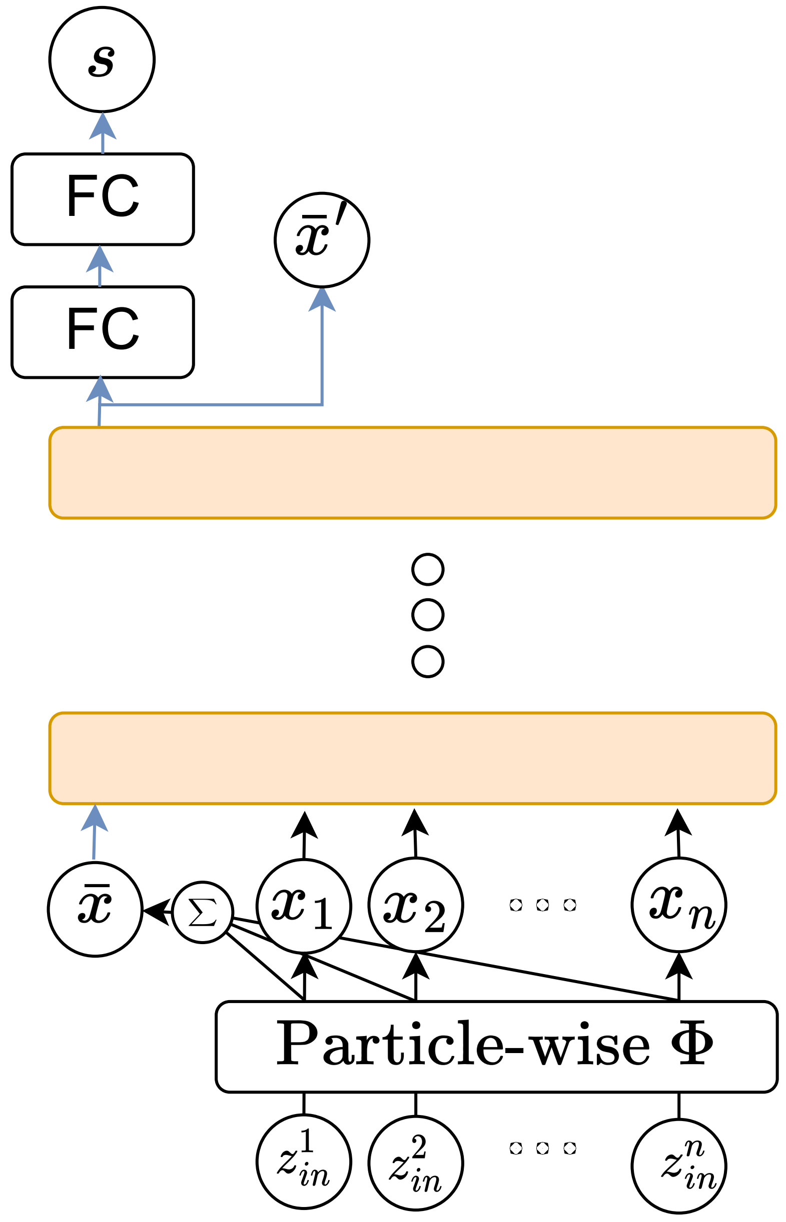
      $\rightarrow c_N$ average shower multiplicity
- 5 blocks used, hidden dimension $h = 48$, 16 heads
- After last block: particle-wise FC ω projects down to 4 dimensions
- ~66 k parameters



Critic Architecture

- IN: particles $x_i \in \mathbb{R}^{n\times4}$
- OUT: $s \in \mathbb{R}^1, \bar{x}' \in \mathbb{R}^l$
- Particles embedded to latent space with Φ to dimension $\mathbb{R}^l$, $l = 16$
- Mean-field initialised as $\bar{x} = \frac{1}{c_N}\sum_{i=1}^{n}\Phi(z_i)$

  → c_N average shower multiplicity
- 2 Layer Perceptron gives scores s
- Outgoing mean-field used for feature matching
- Permutation invariant architecture
- 6 blocks, hidden dim $h = 32$, 16 heads
- ~40 k parameters



Training

Mean FielD Matching Attentive (MDMA) - GAN

- WGAN GP Loss: $\begin{cases} L_C = -C(x_{real}) + C(x_{gen}) + GP & \text{Critic} \\ L_G = -C(G(z)) & \text{Generator} \end{cases}$
- Gradient Penalty: $GP = (\nabla_{\hat{x}}(C(\hat{x}) - 1)^2, \quad \begin{cases} \hat{x} = \lambda x_{real} + (1-\lambda) x_{gen} \\ \lambda \sim U(0,1) \end{cases}$

  → Only interpolate between same sized clouds due to masking
- Additional loss terms in Generator
  - Feature Matching: Generator L2 loss between mean-field in last critic layer for real and fake showers

    $$L_{MF} = \left| \bar{x}'_{real} - \bar{x}'_{fake} \right|^2$$
  - Response Matching: Detector Response $\eta = \frac{\sum_{cells} E}{E_{inc}} \rightarrow L_E = \left| \eta(x_{real}) - \eta(x_{fake}) \right|^2$
  - $L_G^{tot} = -C(G(z)) + L_{MF} + L_E$
- Checkpointing $\mathbb{E}_{f \in [R,\alpha,z,E]} \int_x \left| \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{f_i^{real} < x} - \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{f_i^{fake} < x} \right|$
- Bucketing: Data loader groups similar sized showers to batches to reduce padding

Failures

- Least Squares GAN, Non-Saturating GAN, Vanilla GAN loss
- Transferring information from mean-field to particles by addition or with Gated Linear Unit (GLU)
- Momentum in ADAM ($\beta_1 > 0$) (Generator only, Critic only, both) (→ RmsProp)
- Two-Time scale Update Rule (TTUR) GAN training
- Shared Batch Norm on particles
- Layer Norm also on particles before Cross Attention
- Additive/multiplicative noise to real/generated showers
- Multiple Critic steps per Generator Step
- Spectral Norm
- (One-)Cycle LR scheduling
- Progressively growing showers by only modelling hardest hits first

Tips & Tricks

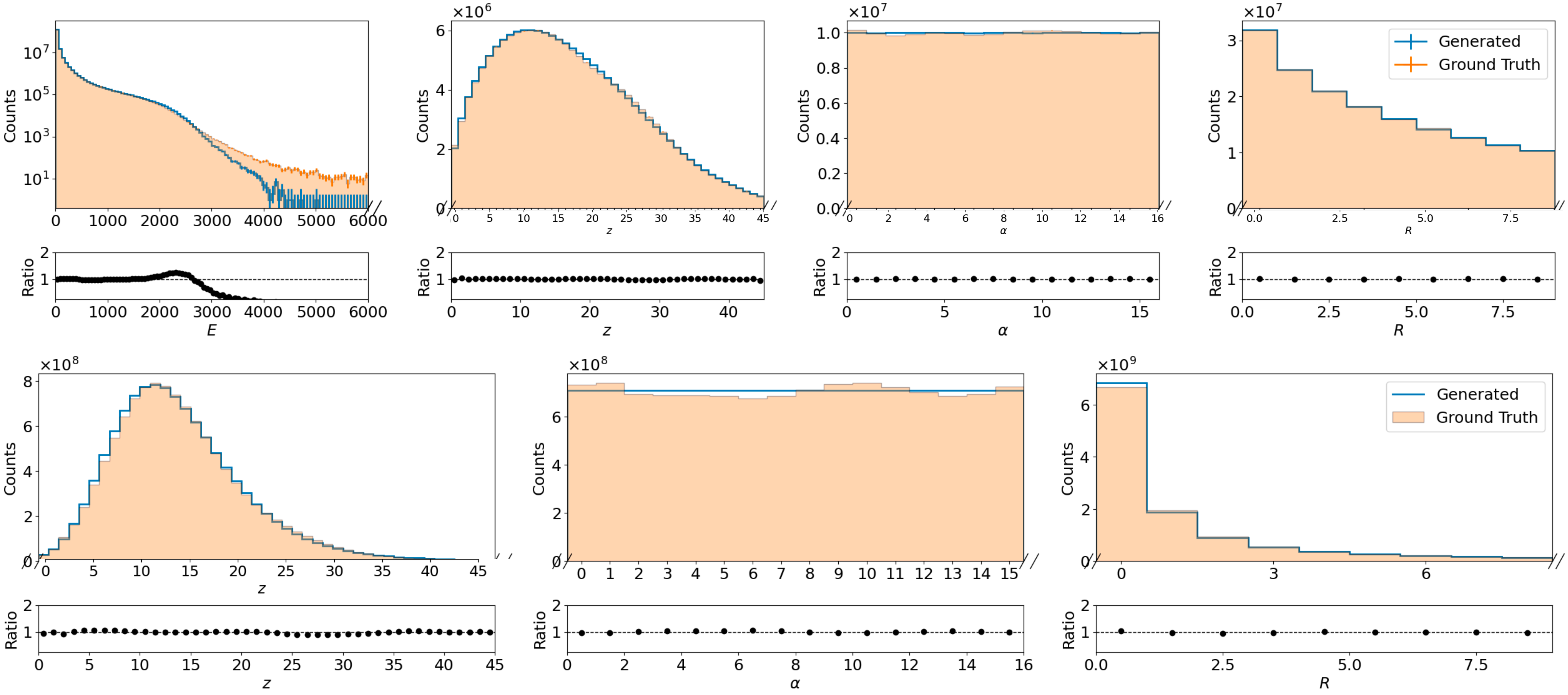
- Mean-field matching: makes generator converge more quickly
- Explicit conditioning on the number particles **crucial** for performance
- Special Weight Normalisation [3] for GANs in critic makes training more stable
- WGAN-GP loss necessaryLearning rate scheduled with Cosine Annealing for both Generator and Critic
- Gradient Accumulation takes some of bias induced by bucketing away
- Layer Norm stabilises training, although takes away physical interpretation of mean-field
- Reading and understanding losses makes GAN training easier
  - "how does loss saturate?"
  - "what happens if Generator/Critic is training only?"

[3] Xiang et al., 2017

Middle Dataset

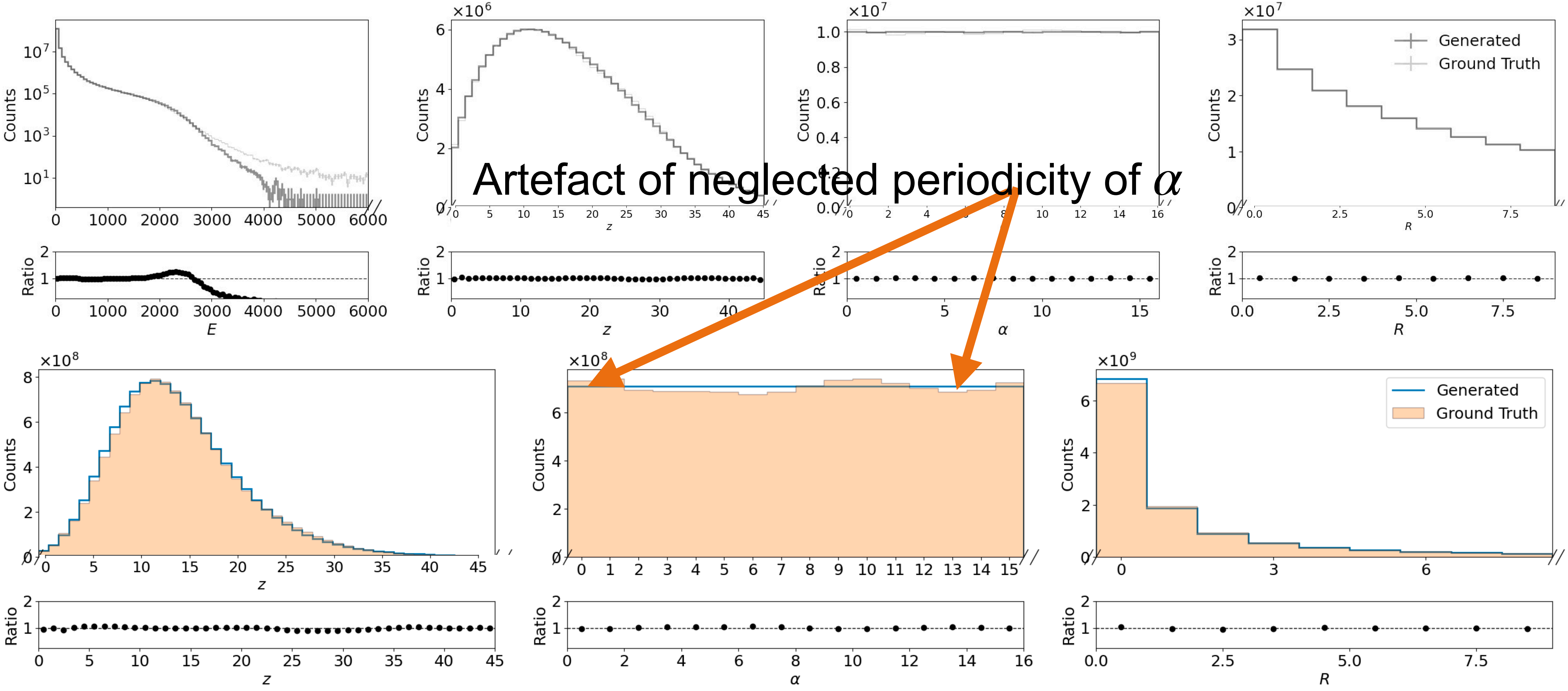
Marginal Variables & Energy Weighted Representation

Agreement in Marginals good



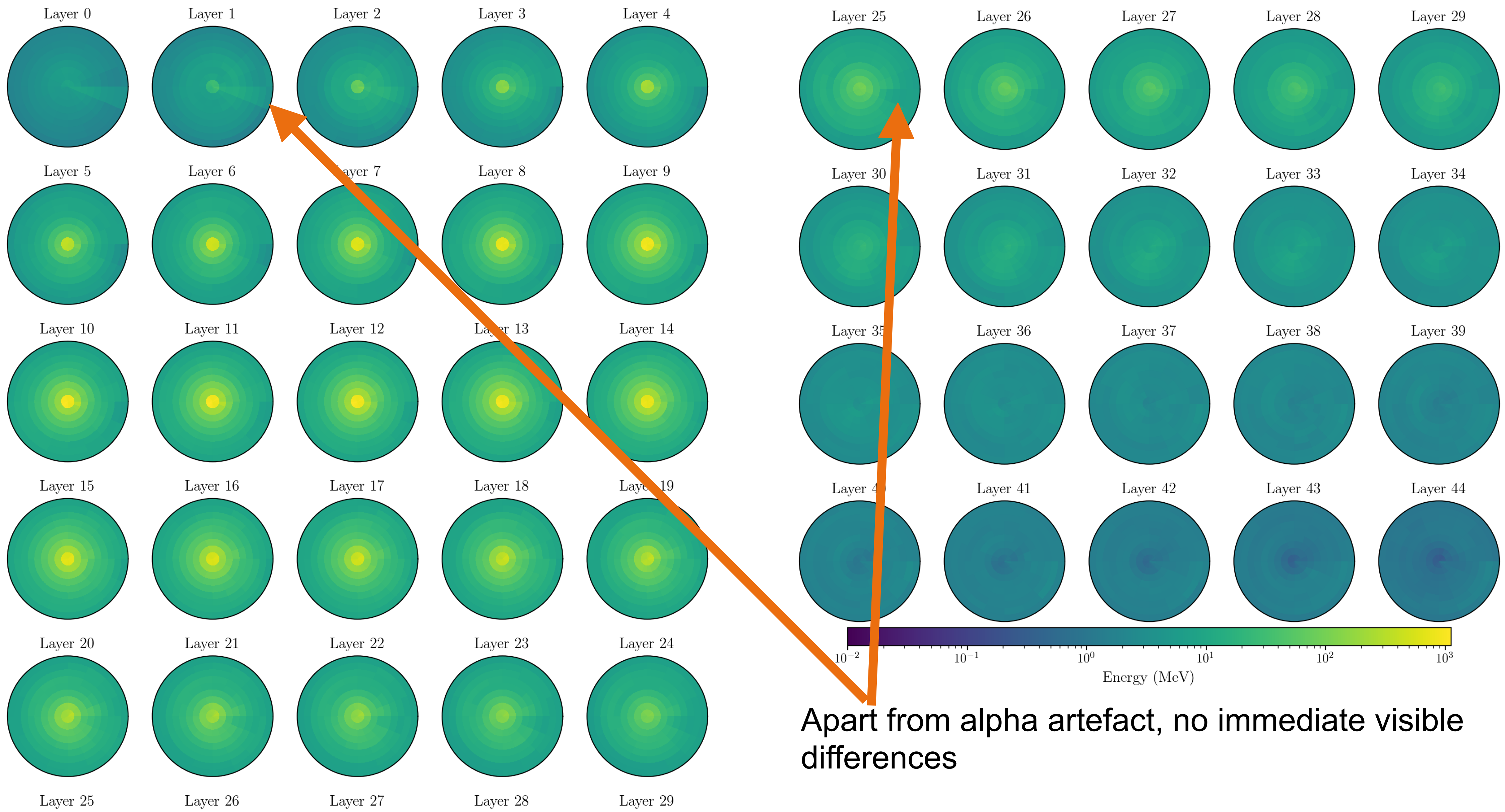
Marginal Variables & Energy Weighted Representation

Results of naive α dequantisation!



Shower Profiles

Thanks Claudius for these nice visualisations!



Apart from alpha artefact, no immediate visible differences

Classifier Metric & Detector Response

<u>Dataset 2</u>

High-level:

Accuracy: 81 % on testing set

AUC: 0.89 on testing set

JSD: 0.39

Low-Level:

Accuracy: 93 % on testing set

AUC: 0.97 on testing set

JSD: 0.71

Low-Level (normed):

Accuracy: 89 % on testing set

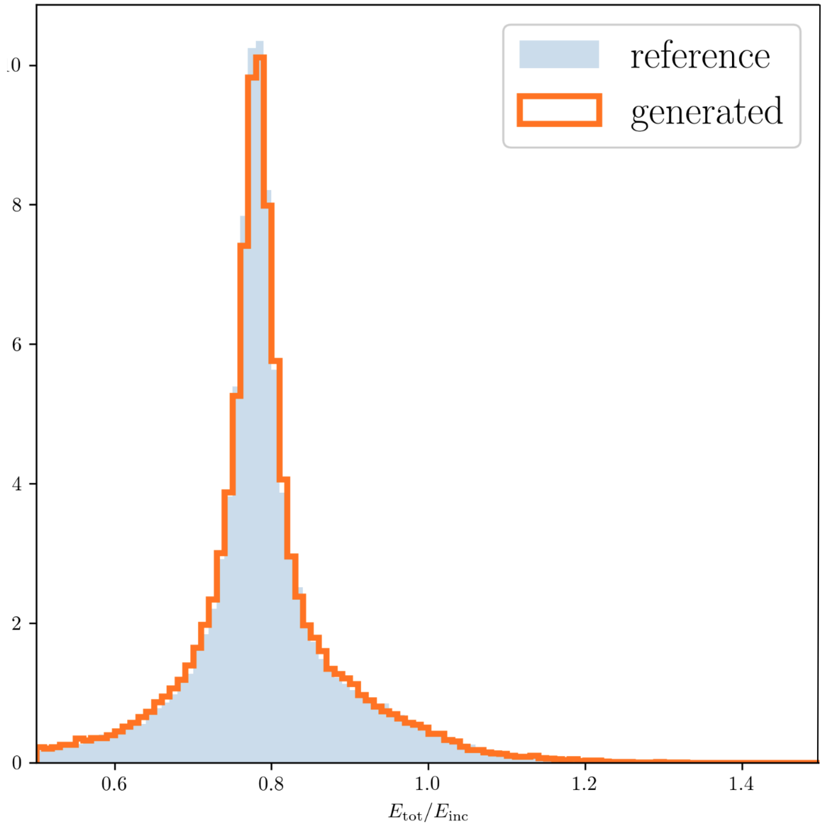
AUC: 0.96 on testing set

JSD: 0.62

Timing:

Generation only: 0.45 ms/shower

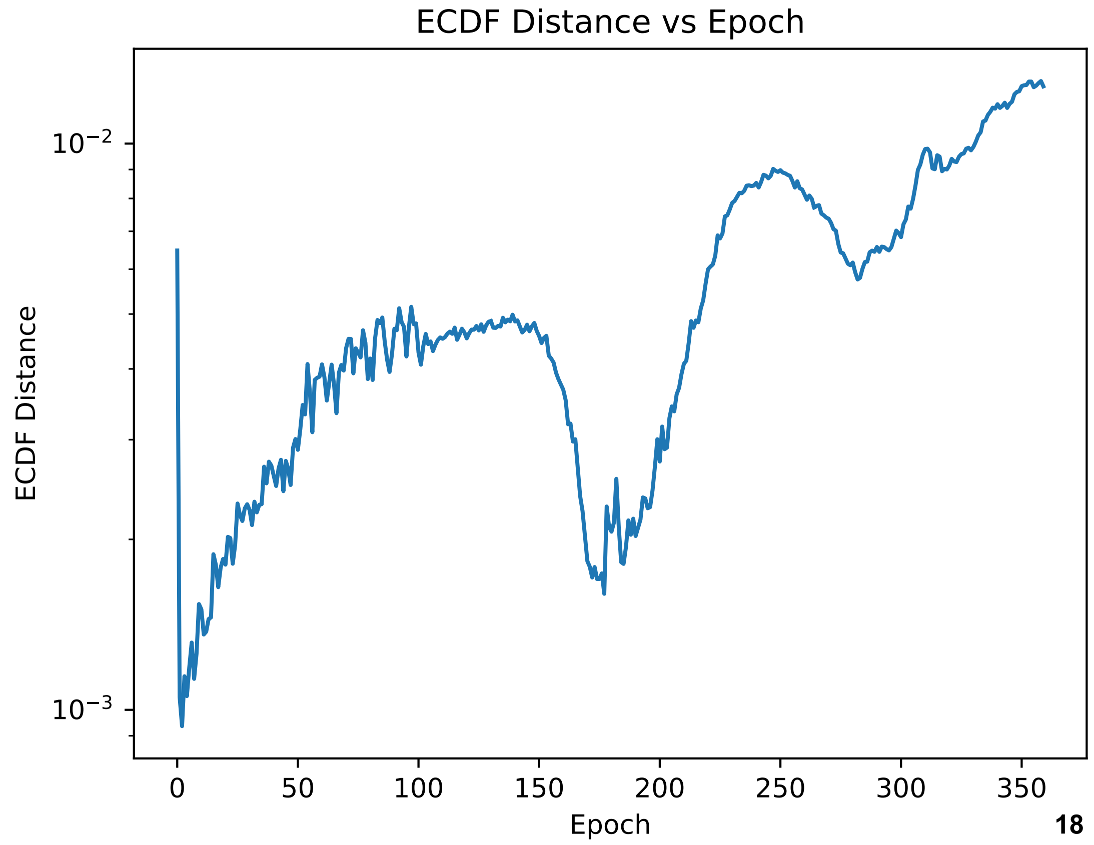
w/ Voxelization: 2.2 ms/shower



Big Dataset

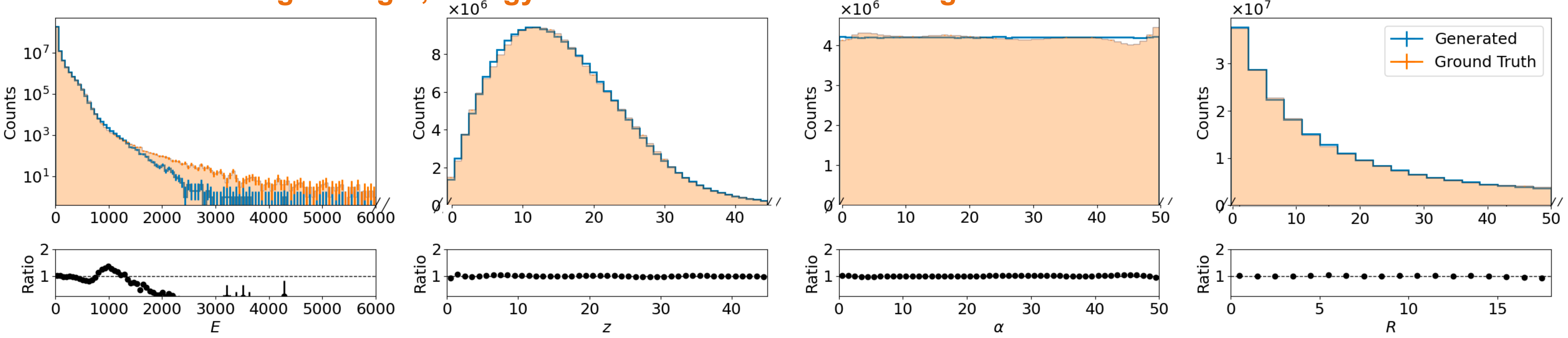
Transfer Learning

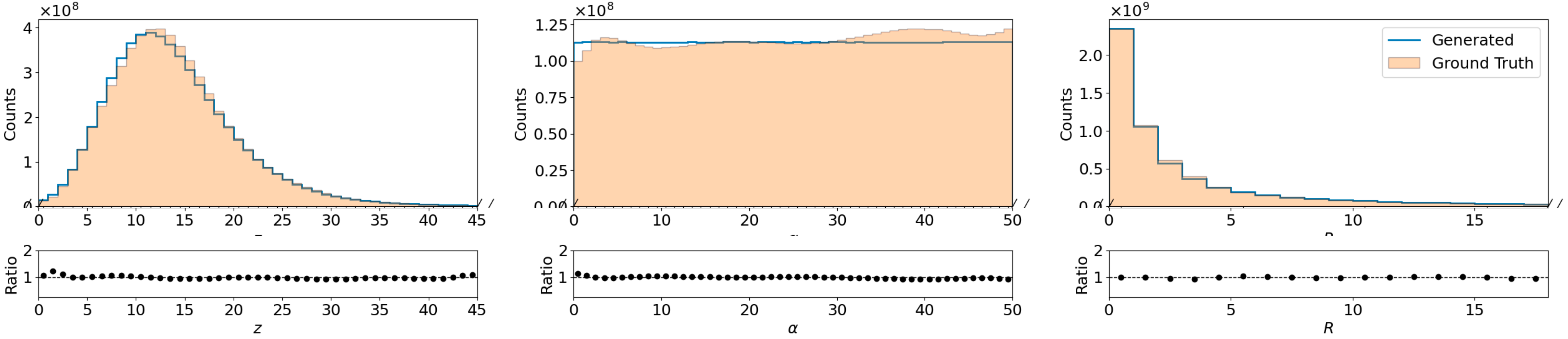
- Number detector hits significantly higher on dataset 3 → Training significantly slower
- Physics in detector the same → only representation changing
- Use model pretrained on dataset 2 → fine-tune on dataset 3
- Convergence significantly faster → one epoch
- Interestingly diverges after one Epoch - yet unclear why
- Proves power of point cloud representation



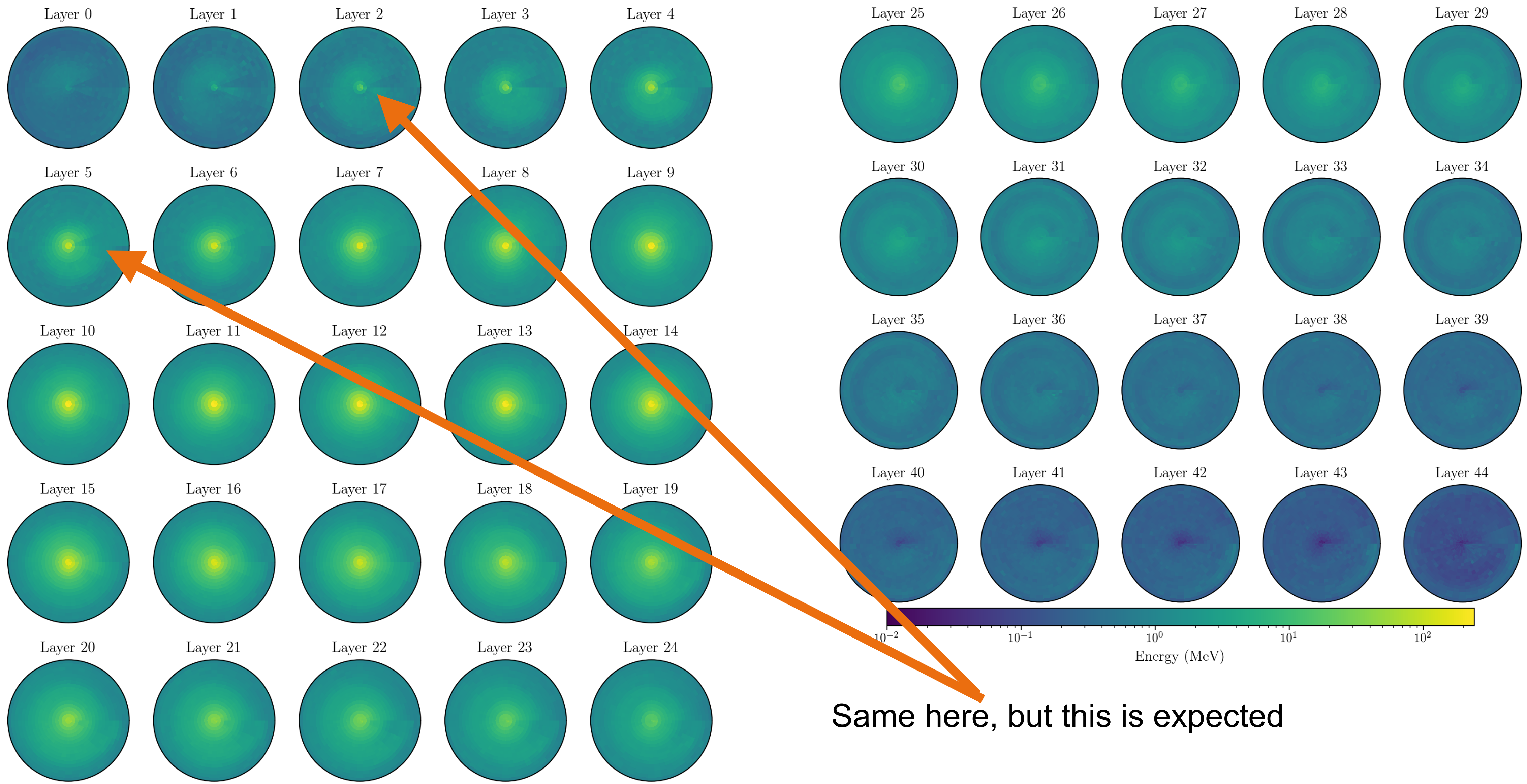
Marginal Variables & Energy Weighted Representation

α miss-modelling stronger, Energy also not modelled well enough





Marginal Variables & Response



Same here, but this is expected

Classifier Metric & Response

Dataset 3

High-level:

Accuracy: 87 % on testing set

AUC: 0.93 on testing set

JSD: 0.51

Low-Level:

Accuracy: 91 % on testing set

AUC: 0.96 on testing set

JSD: 0.65

Low-Level (normed):

Accuracy: 89 % on testing set

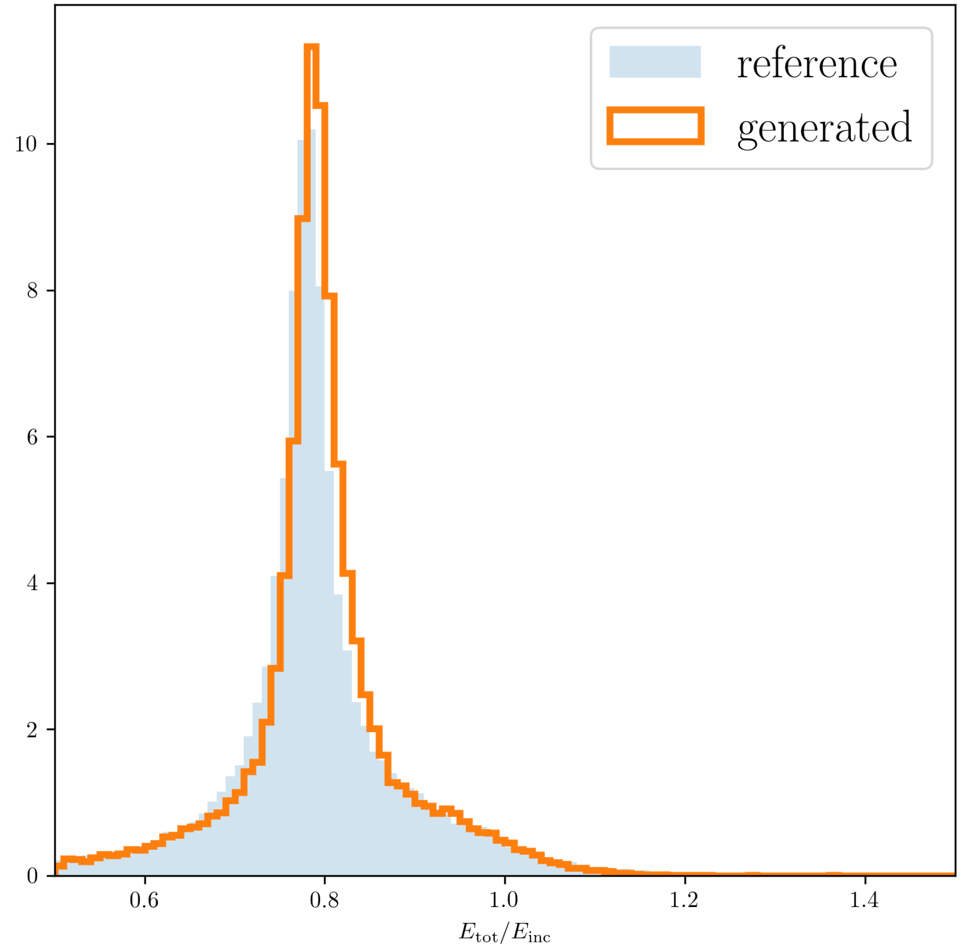
AUC: 0.95 on testing set

JSD: 0.60

Timing:

Generation only: 1.1 ms/shower

w/ Voxelization: 3.2 ms/shower



Conclusion & Further Plans

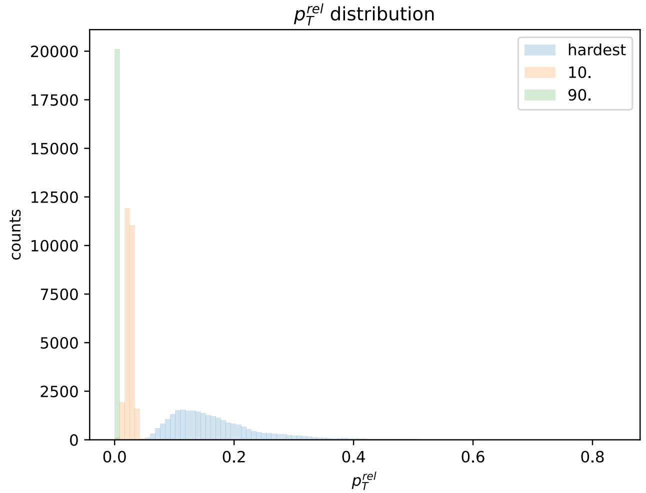
- Switch data representation to be more euclidean like
- Separate conditional model for particle multiplicity given energy needed $p(n \mid E)$
- Multi-GPU training needed (model constrained for now to ~$100'000$ parameters
  $\rightarrow$ due to memory issues batch size < 64
- Move on to more realistic detector - regularity of detector only needed during naive dequantisation
- Up for discussion: switch to sim-hits?
  $\rightarrow$ more natural, no dequantisation needed, arbitrary detector geometry possible
- Learn at low "resolution" first by grouping neighbouring hits
  $\rightarrow$ increase gradually during training

Backup

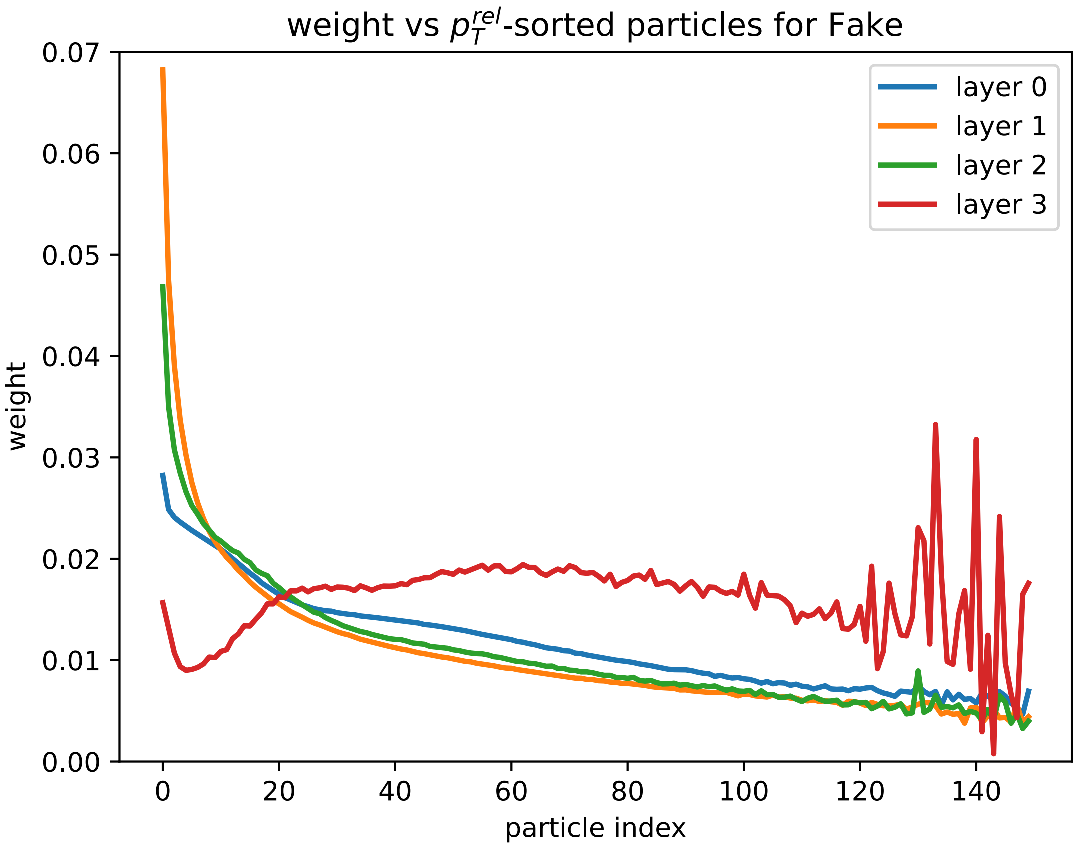
Model Interpretability

- Attention weights tell us which particles are “important” for the generator and critic
- Following plots: order particles by either of $(p_T^{rel}, \eta^{rel}, \phi^{rel})$ and calculate average weight per particle index
- Expectation: Harder particles more important that soft ones, no ordering import in others

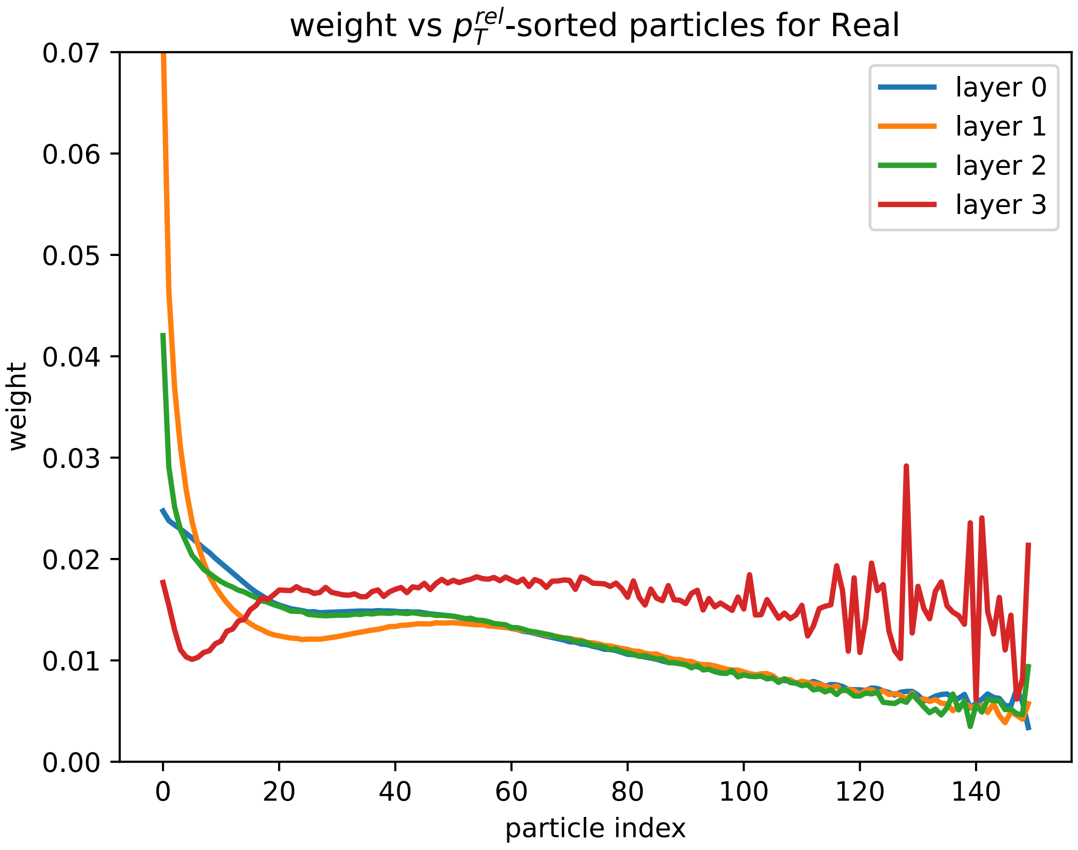
Top Quark



p_T^{rel} distribution
hardest
10.
90.
counts
p_T^{rel}

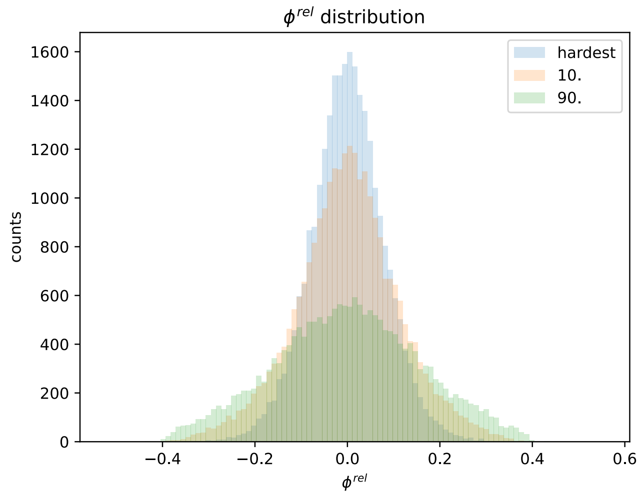


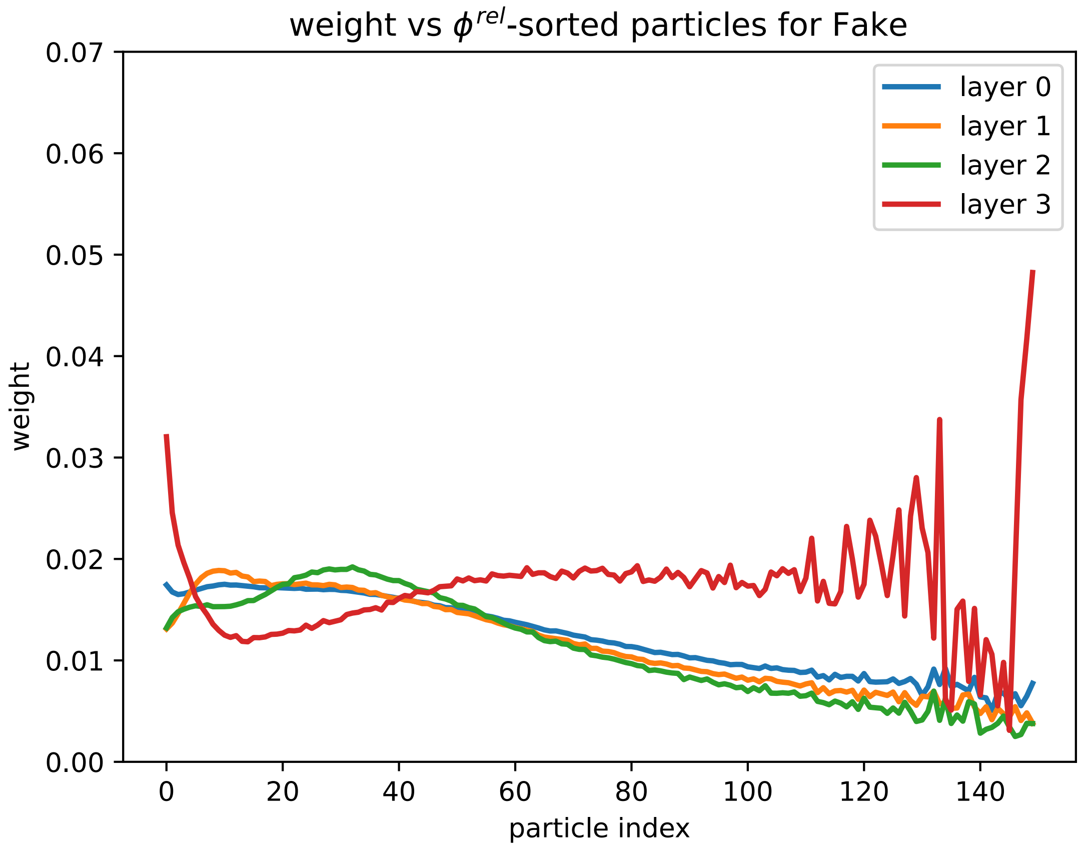
weight vs p_T^{rel}-sorted particles for Fake
layer 0
layer 1
layer 2
layer 3
weight
particle index

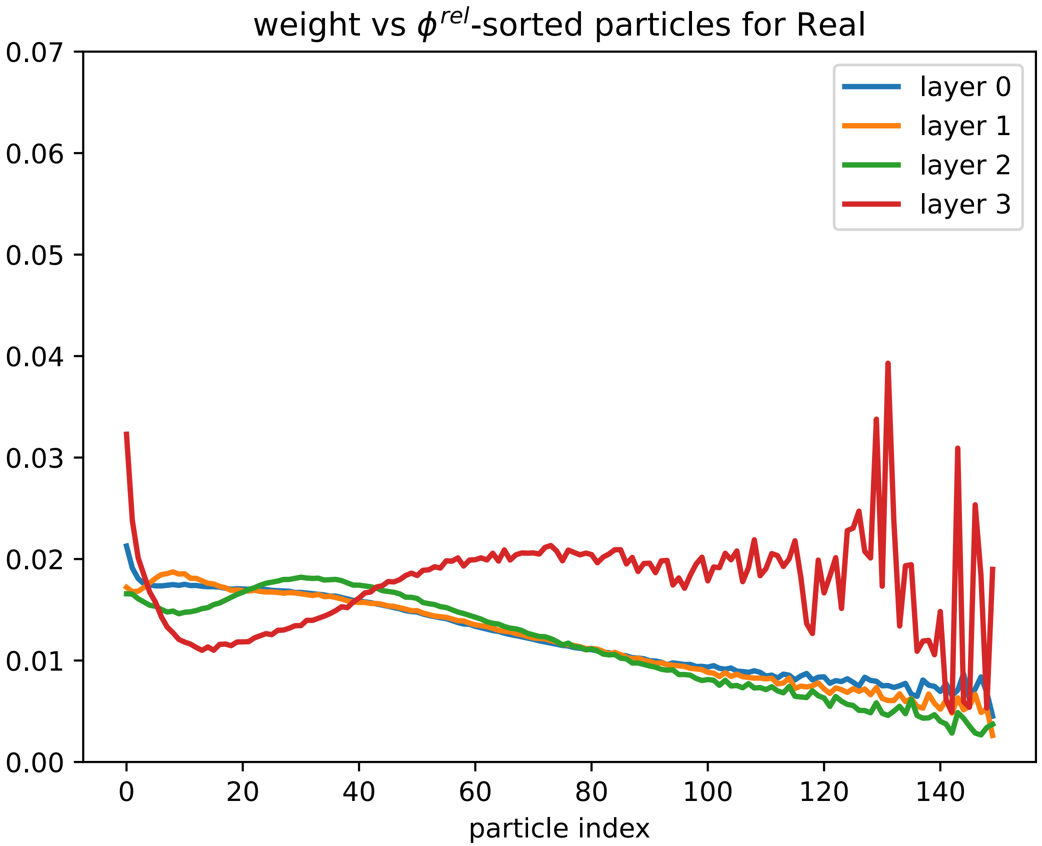


weight vs p_T^{rel}-sorted particles for Real
layer 0
layer 1
layer 2
layer 3
weight
particle index

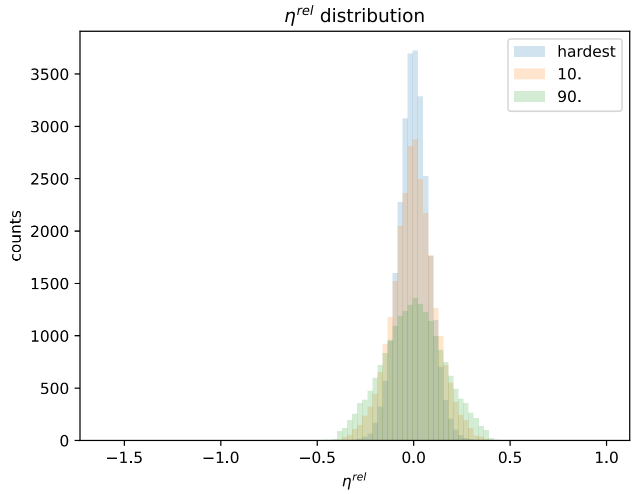
Top Quark


ϕ^{rel} distribution
counts
ϕ^{rel}
hardest
10.
90.
0
200
400
600
800
1000
1200
1400
1600
−0.4
−0.2
0.0
0.2
0.4
0.6

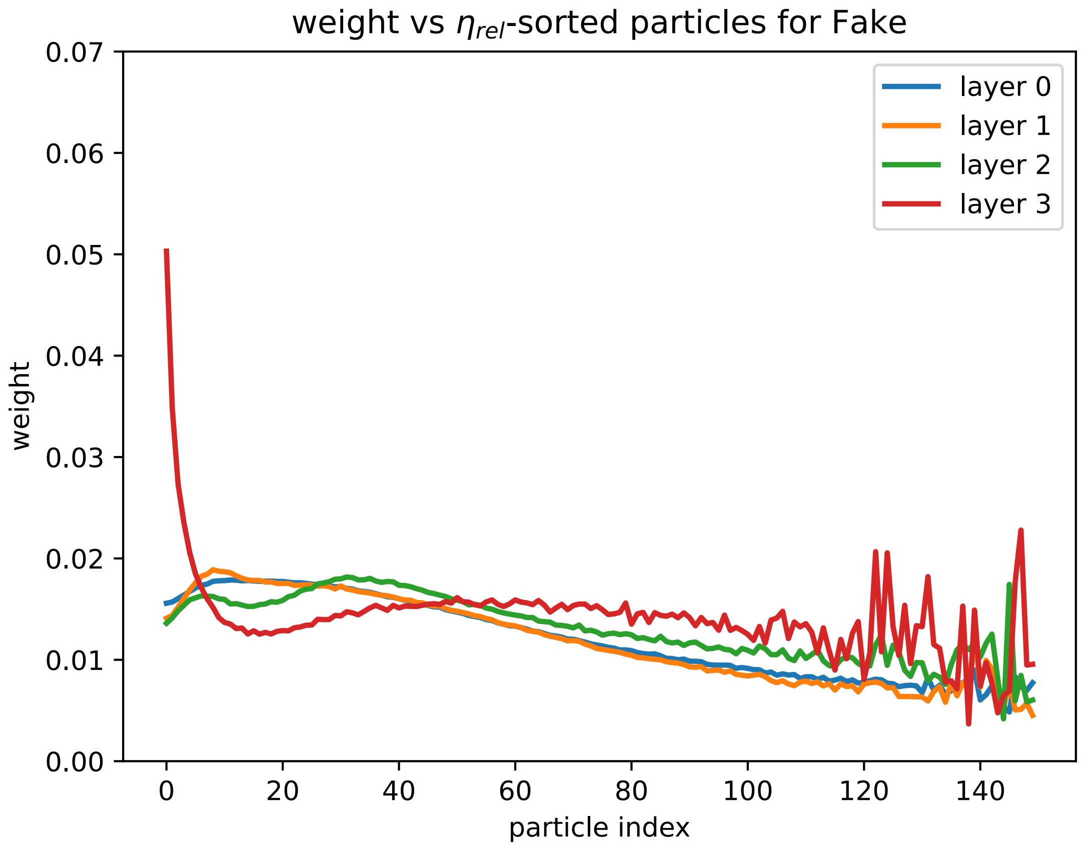

weight vs ϕ^{rel}-sorted particles for Fake
weight
particle index
layer 0
layer 1
layer 2
layer 3
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0
20
40
60
80
100
120
140


weight vs ϕ^{rel}-sorted particles for Real
particle index
layer 0
layer 1
layer 2
layer 3
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0
20
40
60
80
100
120
140

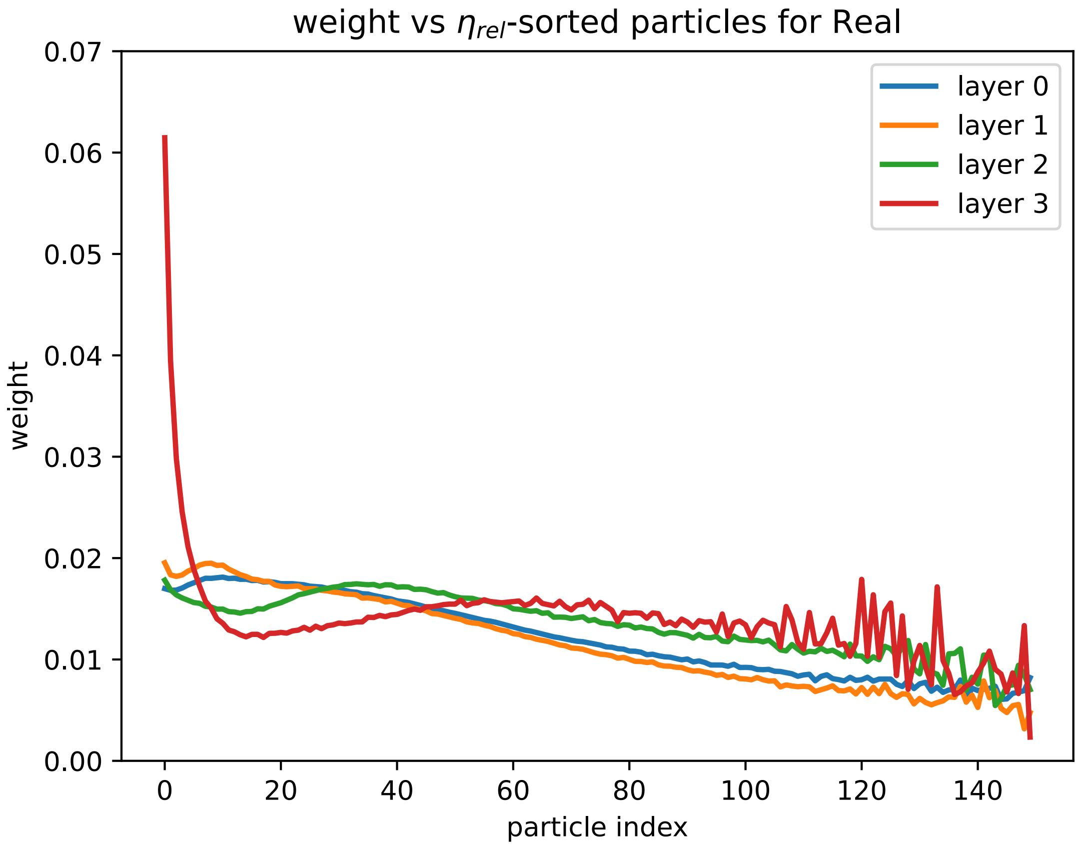
Top Quark



η^{rel} distribution
hardest
10.
90.
counts
η^{rel}



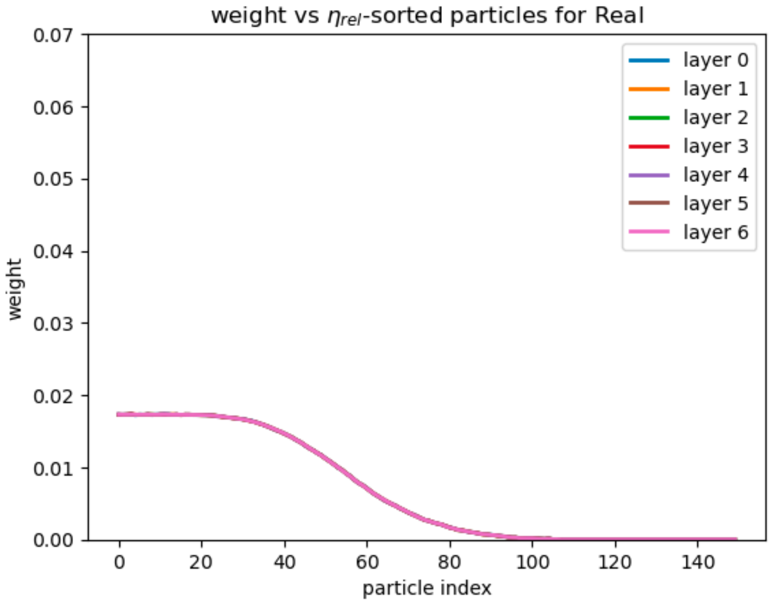
weight vs η_{rel}-sorted particles for Fake
layer 0
layer 1
layer 2
layer 3
weight
particle index



weight vs η_{rel}-sorted particles for Real
layer 0
layer 1
layer 2
layer 3
weight
particle index

Generator

Intuition: higher norm in latent space means higher importance



Conclusion & Outlook

- Linear Model seems to work
- Needs funny name
- Need to decide how much to optimise models
- Harder particles more important
- Gluon, light-quarks unexpectedly hard

Generative Modelling for Detector Simulation

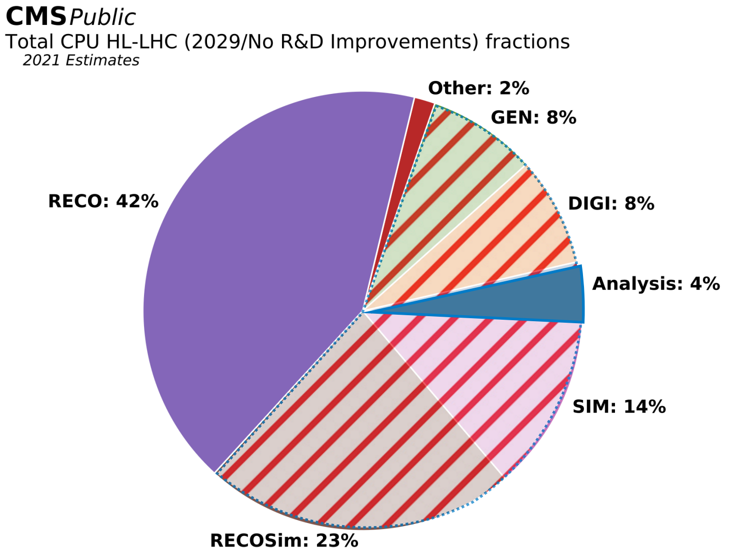


Figure from CMS Offline Computing Results

- Rely on experiment simulation in High Energy Physics (Digital Twin)
- Classically generated with Monte Carlo simulation
  → slow and computing intense
- Already > 50 % of computing budget
- Coming High Luminosity upgrades makes MC approach challenging

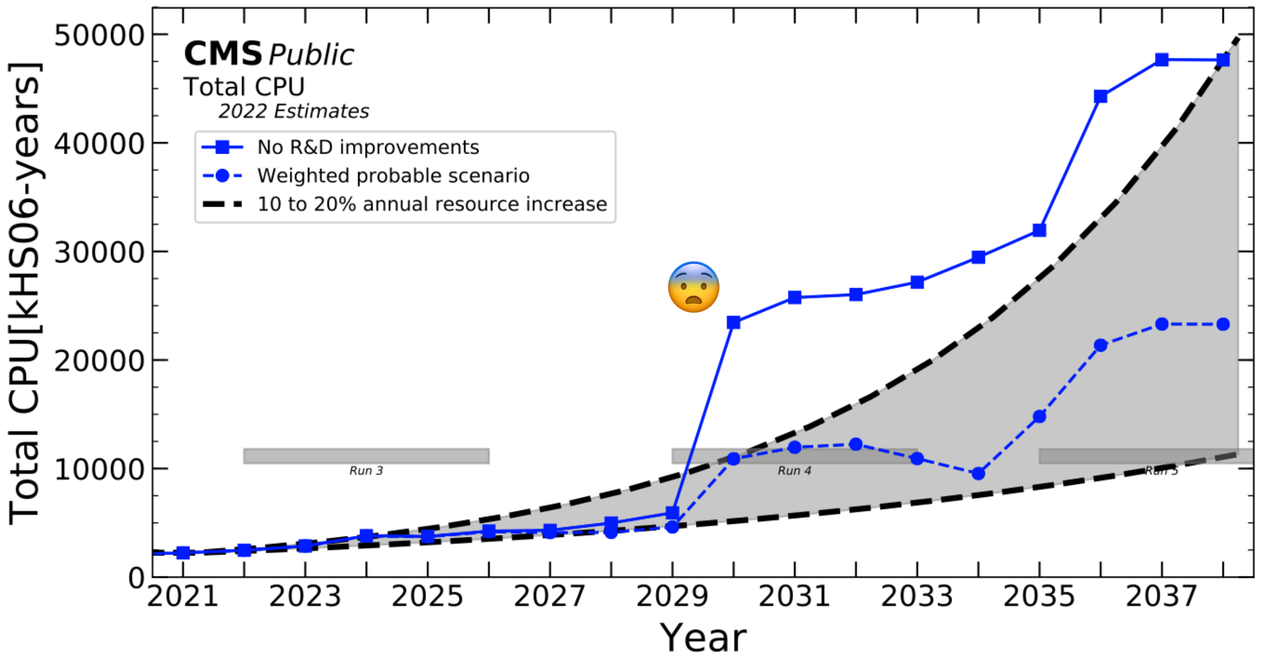
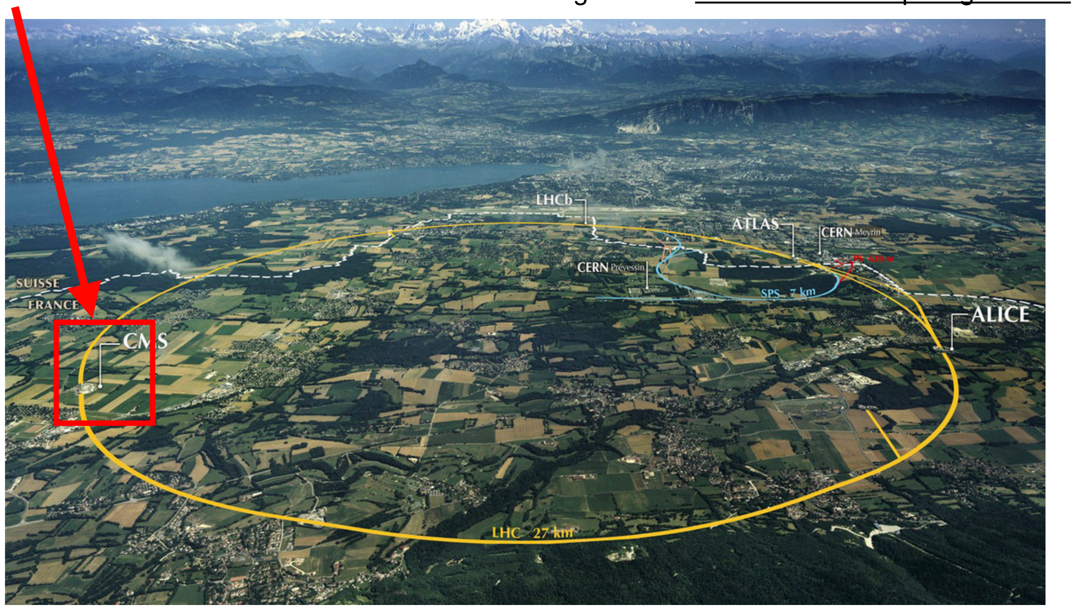


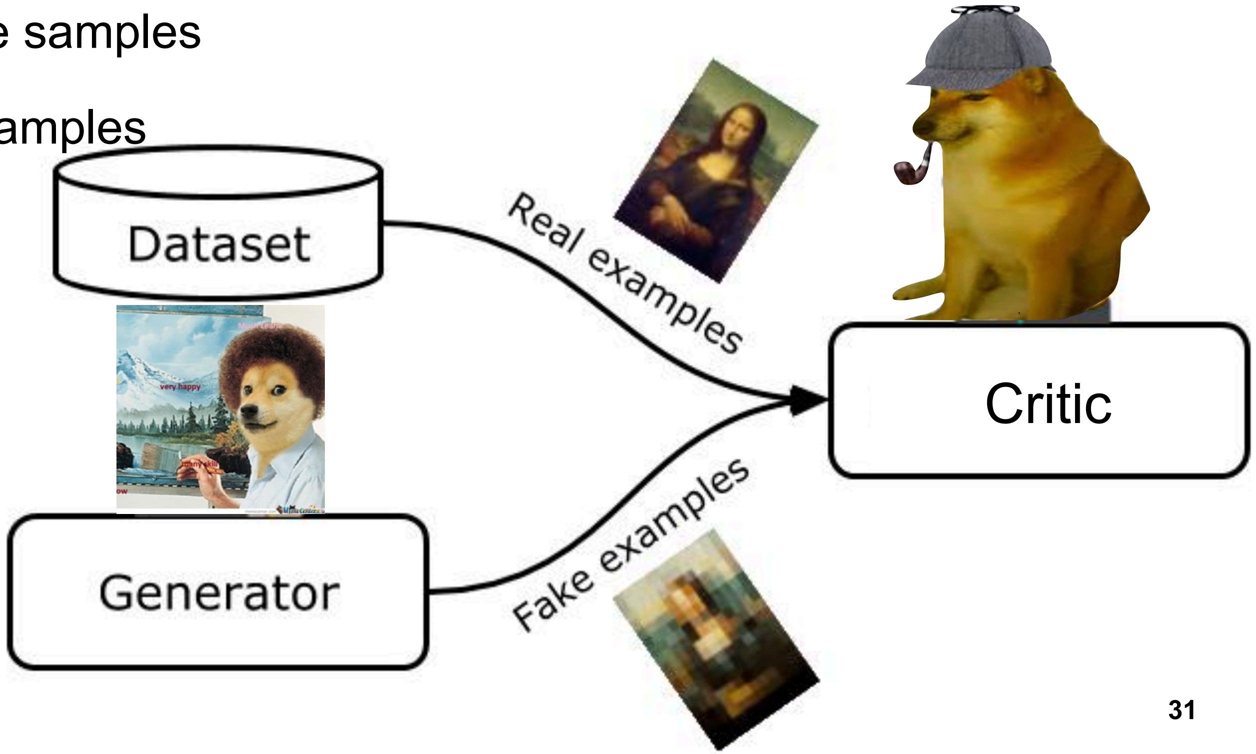
Figure from CMS Offline Computing Results

CMS Experiment



Generative Adversarial Networks

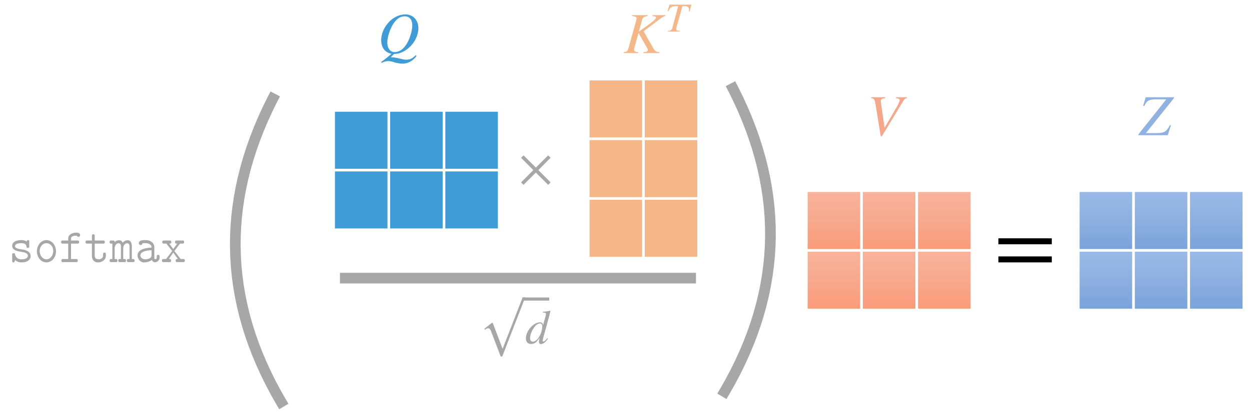
- Consists of Generator & Critic → 2 models
- Generator: generates fake samples $\boldsymbol{X}_{Fake} = \boldsymbol{G}(\boldsymbol{Z}),\ \boldsymbol{Z} \sim \boldsymbol{N}(0,1)$
- Critic: rates "realness" of fake/real samples with score $s = \boldsymbol{D}(\boldsymbol{X})$, with $\boldsymbol{X} = \begin{cases} \boldsymbol{X} = \boldsymbol{G}(\boldsymbol{Z}) \\ \boldsymbol{X} = \boldsymbol{X}_{\text{Real}} \end{cases}$
- Critic optimised to give $s = 0$ for fake and $s = 1$ for true samples
- Generator optimised to make critic inaccurate on fake samples
- 2 adverse models to train → unstable



Self-Attention

Attention is all you need! [5]

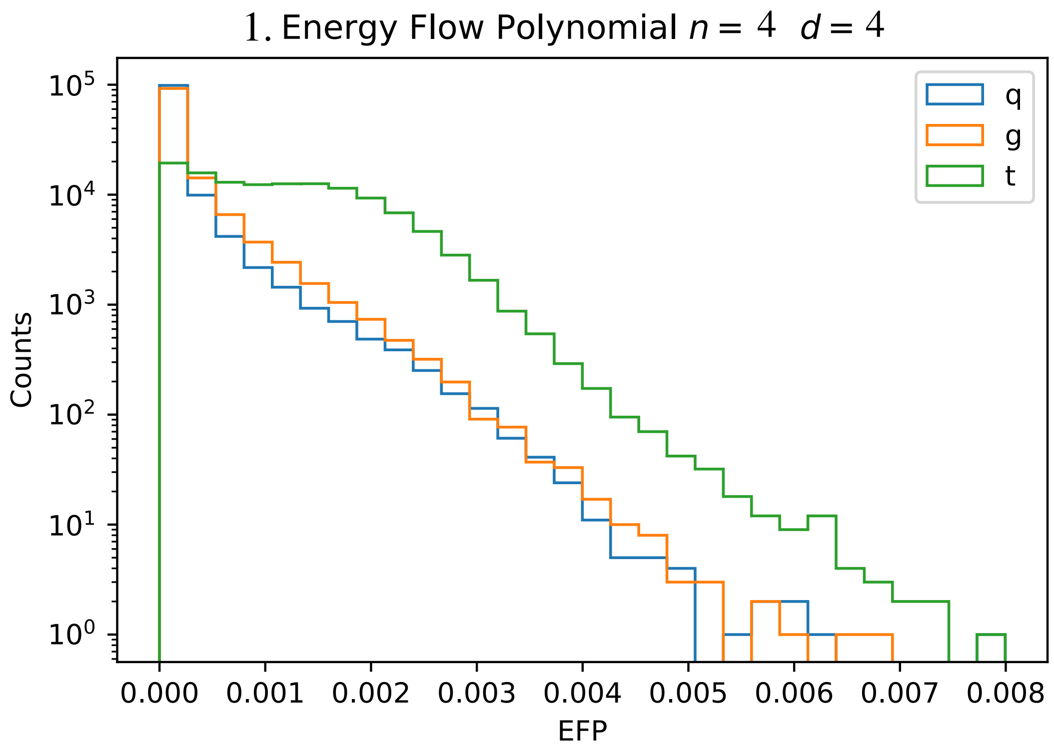
- Commonly used in NLP
- Permutation invariant
- Self-Attention: n inputs, n outputs - interaction between inputs
- Particles attend to other particles with strength: $\texttt{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \frac{\texttt{softmax}(\boldsymbol{Q} \cdot \boldsymbol{K}^{\mathrm{T}} + \boldsymbol{M} \cdot (-\infty))}{\sqrt{\texttt{d}}} \boldsymbol{V}$
- $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ Linear embeddings of input $\rightarrow \boldsymbol{Q} = \boldsymbol{W}_Q \boldsymbol{x}, \boldsymbol{K} = \boldsymbol{W}_K \boldsymbol{x}, \boldsymbol{V} = \boldsymbol{W}_V \boldsymbol{x}$
- $\boldsymbol{M} = 1$ mask for jets with < 30 particles $\rightarrow$ No influence



[5] A. Vaswani et al., "Attention Is All You Need", https://arxiv.org/abs/1706.03762

Wasserstein Distance

- Metric on probability distributions
- Formally: $W_1(\mathbb{P}_r, \mathbb{P}_g) := \inf_{\gamma \in \Gamma(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\gamma}[\,||\boldsymbol{x}-\boldsymbol{y}||\,]$
- Not tractable for $\dim(\boldsymbol{X} \sim \mathbb{P}_g) > 1$
  - W_1^P: average of W_1 over (η, ϕ, p_T)
  - W_1^M: invariant jet mass
  - W_1^{EFP}: 5 Energy Flow Polynomials [4] (n=4,d=4)



<u>In-sample distances</u>

Parton	$W_1^M(\times 10^{-3})$	$W_1^P(\times 10^{-3})$	$W_1^{EFP}(\times 10^{-5})$	FPND	COV $\uparrow$	MMD
Gluon	0.5 ± 0.1	0.4 ± 0.2	0.4 ± 0.4	0.01	0.56	0.036
Light Quark	0.42 ± 0.09	0.6 ± 0.4	0.5 ± 0.5	0.01	0.55	0.024
Top Quark	0.5 ± 0.1	0.6 ± 0.4	1.1 ± 0.4	0.03	0.56	0.072

[4] Komiske et al., Energy flow polynomials: A complete linear basis for jet sbstructure, arxiv.org/abs/1712.07124

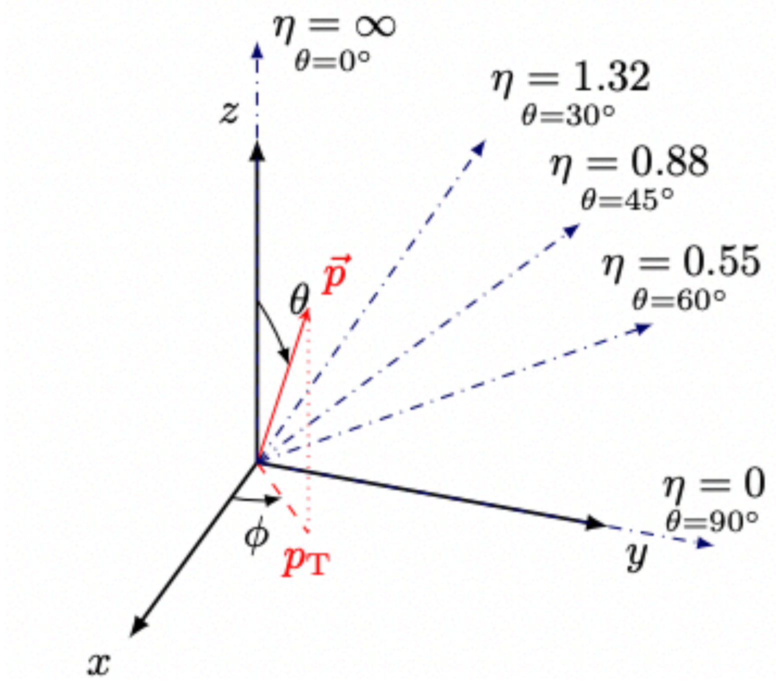
Particle Cloud Generation

JetNet [1] Datasets

- Jets: unordered sprays of particles
- Particles: tuples of $(\eta^{\mathrm{rel}}, \phi^{\mathrm{rel}}, p_T^{\mathrm{rel}})$ relative to jet axis
- Constrained to max 150 particles/jet

  → Goal: generate $\boldsymbol{X} = \left\{ \left(\eta_{(i)}^{\mathrm{rel}}, \phi_{(i)}^{\mathrm{rel}}, p_{T,(i)}^{\mathrm{rel}} \right) \right\}_{(i \leq n)} \sim p_{data}(\boldsymbol{X})$

- Invariant jet mass: $m_{rel}^2 = \left(\sum_{i=1}^{n} |\boldsymbol{p}_i| \right)^2 - \left(\sum_{i=1}^{n} \boldsymbol{p}_i \right)^2$
- Size $\sim 178'000$ Samples
- 70% used for training
- Benchmarking possible



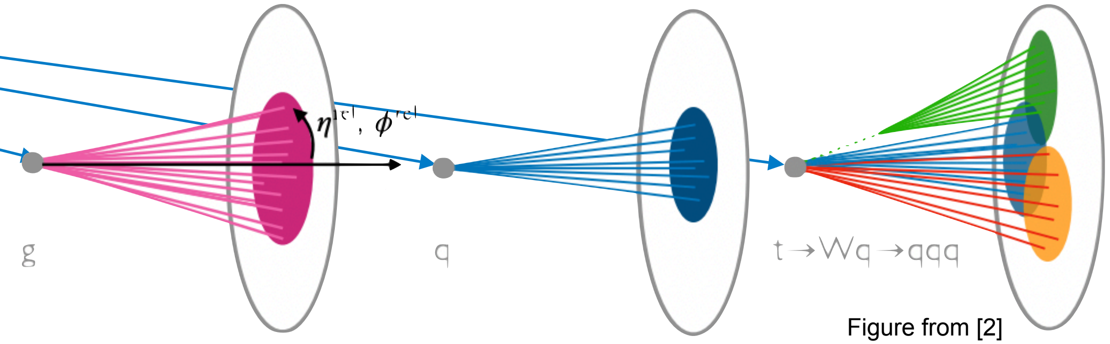


Figure from [2]

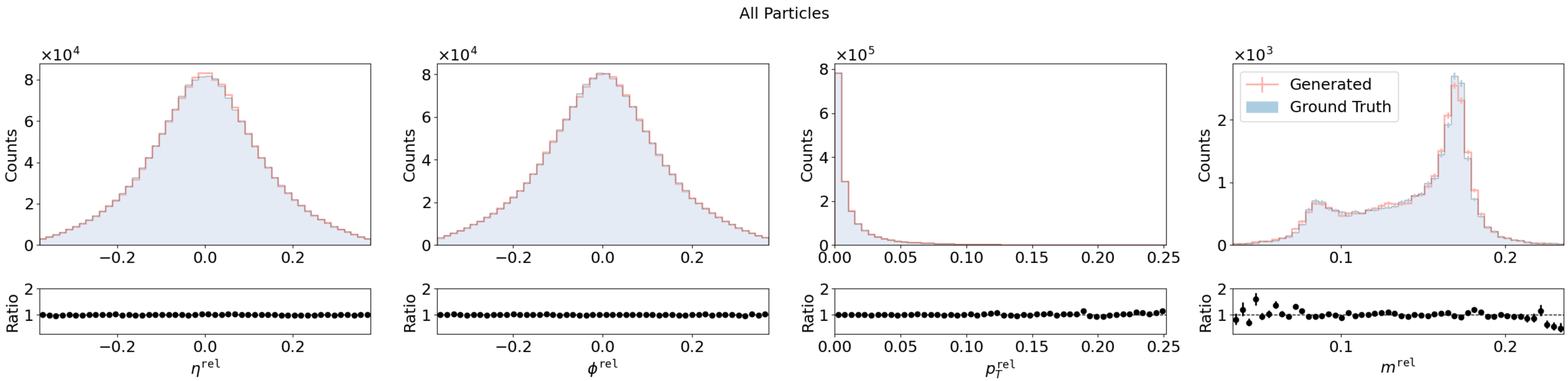
[1] Kansal et al, JetNet, PyPi
[2] Kansal et al., Particle Cloud Generation with Message Passing Generative Adversarial Networks, arxiv.org/abs/2106.11535

Scores

Same hyperparameters for all models, 2000 Epochs training

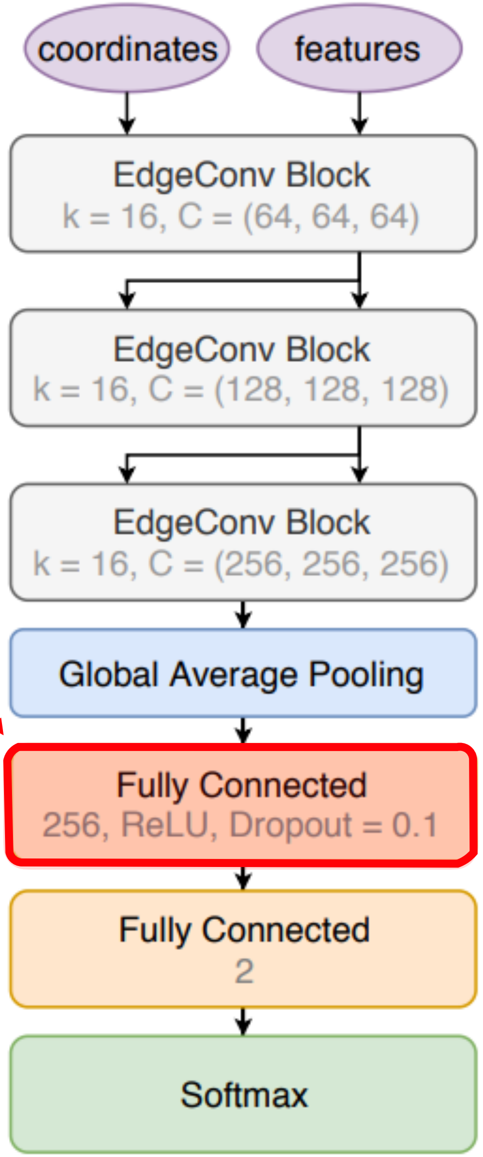
Jet Class	Model	$W_1^M (\times 10^3)$	$W_1^P (\times 10^3)$	$W_1^{EFP} (\times 10^5)$	COV ↑	MMD	KPD $(\times 10^5)$	FPD$(\times 10^5)$
Light Quark	MF	0.50 ± 0.06	$\mathbf{1.39 \pm 0.04}$	$\mathbf{0.75 \pm 0.04}$	**0.54**	**0.022**	$\mathbf{-0.03 \pm 0.05}$	$\mathbf{3.9 \pm 0.9}$
	EPiC	$\mathbf{0.42 \pm 0.06}$	3.84 ± 0.09	0.83 ± 0.08	0.52	**0.022**	-0.0 ± 0.1	5.4 ± 0.8
Gluon	MF	0.81 ± 0.07	$\mathbf{0.45 \pm 0.04}$	1.4 ± 0.2	**0.55**	**0.032**	$\mathbf{-0.04 \pm 0.09}$	4.3 ± 0.8
	EPiC	$\mathbf{0.5 \pm 0.1}$	3.19 ± 0.05	$\mathbf{1.01 \pm 0.08}$	0.53	0.036	0.06 ± 0.05	$\mathbf{3.6 \pm 0.3}$
Top Quark	MF	$\mathbf{0.58 \pm 0.08}$	$\mathbf{0.45 \pm 0.06}$	$\mathbf{1.7 \pm 0.2}$	0.57	**0.058**	$\mathbf{-0.1 \pm 0.2}$	$\mathbf{1.2 \pm 0.5}$
	EPiC	0.62 ± 0.03	3.80 ± 0.06	2.6 ± 0.2	**0.59**	0.068	2 ± 1	21.1 ± 0.6
W	MF	$\mathbf{0.25 \pm 0.02}$	$\mathbf{0.21 \pm 0.03}$	$\mathbf{0.29 \pm 0.02}$	**0.57**	**0.023**	$\mathbf{-0.001 \pm 0.008}$	$\mathbf{4 \pm 1}$
Z	MF	$\mathbf{0.20 \pm 0.01}$	$\mathbf{0.54 \pm 0.04}$	$\mathbf{0.23 \pm 0.03}$	**0.56**	**0.026**	$\mathbf{-0.00 \pm 0.03}$	$\mathbf{5 \pm 1}$

Plots


All Particles
×10^4
Counts
Ratio
η^{rel}
×10^4
ϕ^{rel}
×10^5
p_T^{rel}
×10^3
Generated
Ground Truth
m^{rel}

Fréchet ParticleNet Distance (FPND) [2]

- Inspired from Fréchet Inception Distance (FID) for image generation [5]
- *Wasserstein-2 distance between Gaussians fitted to activations in last FC layer of ParticleNet* [6] *of MC & ML generated jets*
- Sensitive to output quality & mode collapse



<u>In-sample distances</u>

Parton	$W_1^M (\times 10^{-3})$	$W_1^P (\times 10^{-3})$	$W_1^{EFP} (\times 10^{-5})$	FPND	COV ↑	MMD
Gluon	0.5 ± 0.1	0.4 ± 0.2	0.4 ± 0.4	0.01	0.56	0.036
Light Quark	0.42 ± 0.09	0.6 ± 0.4	0.5 ± 0.5	0.01	0.55	0.024
Top Quark	0.5 ± 0.1	0.6 ± 0.4	1.1 ± 0.4	0.03	0.56	0.072

[2] Kansal et al., Particle Cloud Generation with Message Passing Generative Adversarial Networks, arxiv.org/abs/2106.11535
[5] Heusel et al., GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, arxiv.org/abs/1706.08500
[6] Qu et al., ParticleNet: Jet Tagging via Particle Clouds, arxiv.org/abs/1902.08570