Improving neural generative models with density ratio estimation and Monte-Carlo sampling

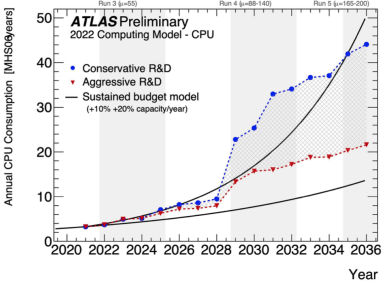
ML4Jets 2026

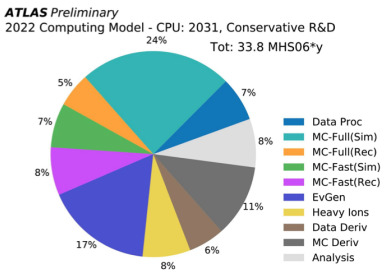
Minh-Tuan Pham, Corentin Allaire, and David Rousseau

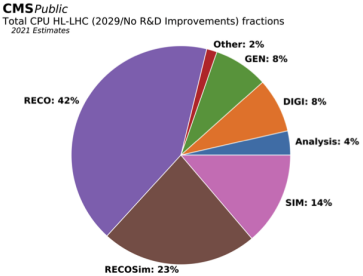
Laboratoire de Physique des 2 Infinis Irène Joliot-Curie (IJCLab) – University Paris-Saclay

Vienna, September 2026

Calorimeter simulation in HEP







(Left) Projected ATLAS CPU budget for HL-LHC [1]. CPU consumption in (centre) ATLAS [1] and (right) CMS [8].

- HEP experiments rely on accurate and fast detector simulations. Constitutes a large fraction of computing budgets. HL-LHC increases demands by an order of magnitude
- GEANT4 is very accurate, but slow and CPU-intensive
- Generative DNNs are promising alternatives, competitive quality and orders of magnitude faster, but not a silver bullet

Motivation

- Training performant generative models is non-trivial
  - ▷ Complex distributions require deep models, large datasets and careful tuning
  - ▷ Training instabilities, mode collapse, etc.
- Quality vs. speed trade-off

Illustrative model performance on CaloChallenge dataset 2. [5]

Model	AUC	Latency (ms/shower)
CaloINN	0.784	$\mathbf{0.20 \pm 0.01}$
CaloDiffusion	**0.680**	27.0 ± 0.2

Latency is measured in batch size 10k. Flow model is faster than diffusion at a cost of quality.

Improve a fast but lower-quality generator?

2 propositions:

- Better architecture, larger model, more data (hard)
- Improve existing model predictions with correction (easier)

Some basic facts

1. Given 2 prob. densities p_1 and p_2, if we (1) **sample** $x \sim p_1$, (2) **accept** with prob. $\boxed{p(x) \propto w(x) = \frac{p_2(x)}{p_1(x)}}$, the accepted sample follow p_2: $P(X = x | p(X) > U(0,1)) = p_2(x)$
2. A classifier $D(\cdot; \phi) : \Omega_X \to (0,1)$ trained by binary cross-entropy loss

$$L(\phi) = \mathbb{E}_{X \sim p_2}[\log D(X; \phi)] + \mathbb{E}_{X \sim p_1}[\log(1 - D(X; \phi))]$$

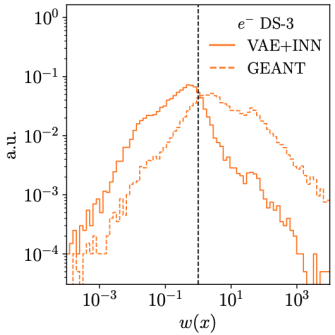
has a unique minimum at $D(x; \hat{\phi}) = P(x \sim p_2 | x; \hat{\phi})$,

$$\boxed{\frac{D(x; \hat{\phi})}{1 - D(x; \hat{\phi})} = \frac{P(x | x \sim p_2; \hat{\phi})}{P(x | x \sim p_1; \hat{\phi})} = w(x; \hat{\phi})}$$

Refine DNN generative modelling

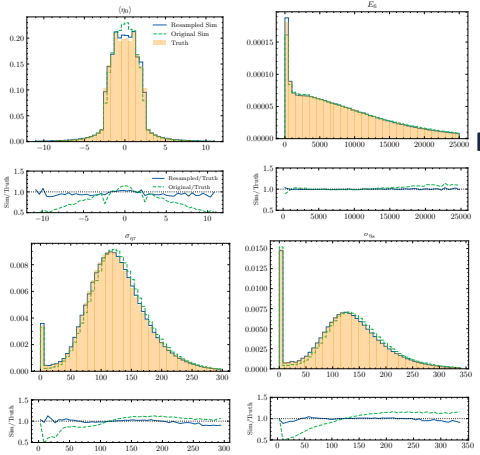
- A generative model maximizes $\mathbb{E}_{X\sim p_d}[\log p_m(X;\theta)]$ [a]. Sampling: $x \sim p_m(\cdot, \hat{\theta})$.
- Quality check: Use density ratio $\boxed{w(x) = \dfrac{p_d(x)}{p_m(x,\hat{\theta})}}$ [b], often estimated by a classifier $D(\cdot;\phi)$.
- Monte-Carlo sampling (1) $x \sim p_m$, (2) accept with probability $\boxed{p \propto w(x;\hat{\phi}) \approx \dfrac{D(x;\hat{\phi})}{1 - D(x;\hat{\phi})}}$.

[a] $p_d(\cdot)$: data distribution, $p_m(\cdot;\theta)$: parametrized model distribution
[b] Most powerful statistic to test $H_0 : x \sim p_m$ against $H_1 : x \sim p_d$



Illustrative distribution of estimated likelihood ratios in data and generated samples [3]

Monte-Carlo sampling in data space



We've shown (CHEP2026) that

- MC in data space **improves** agreement with truth
- Improve features not in density ratio estimator (DRE) training input

However...

- MC with $p_m(\cdot, \theta)$ can be **expensive**:

$$z \sim \mathcal{N}(0,1) \to x = f_\theta(z) \to \mathrm{MC}$$

- **Solution**: MC sample in latent space then decode

$$z \sim \mathcal{N}(0,1) \to \mathrm{MC} \to x = f_\theta(z)$$

Drawing and rejecting Gaussian samples is **cheap**; expensive decoding done **once**

Practical implementation

Given a data set $S_d = \{x_i | x_i \sim p_d\}$, $x =$ shower energy deposits:

A. Train generative model with negative log-likelihood[1]; obtain $p_m(\cdot; \hat{\theta})$

B. DRE training $D(\cdot; \phi)$

1. $\boxed{Z_m = \{(z_i, 0) | z_i \sim \mathcal{N}(0,1)\}}$, $\boxed{Z_d = \{(f_{\hat{\theta}}(x_i), 1) | x_i \in S_d\}}$, $z =$ latent space representation of x.
2. Optimize binary cross-entropy loss on $Z_d + Z_m$[2], amplify the gaussian "noise" by $\mathbf{M} = \frac{|Z_m|}{|Z_d|}$.

C. Monte-Carlo sampling

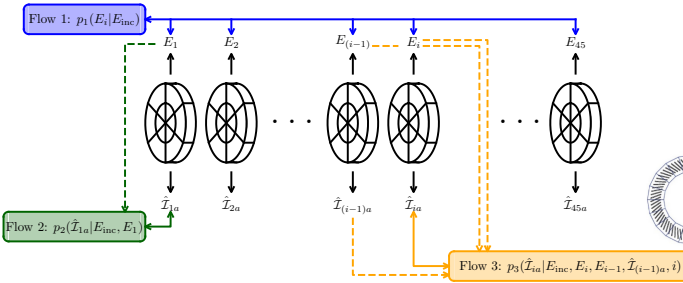
1. Draw $z \sim \mathcal{N}(0,1)$, evaluate $w(z; \hat{\phi}) = \frac{D(z;\hat{\phi})}{(1 - D(z;\hat{\phi}))}$, normalize by $\mathbf{Z}$: $p(z) = \frac{w(z;\hat{\phi})}{\mathbf{Z}}$
2. Accept z if $p(z) >$ Uniform$(0,1)$ then decode $x = f_{\hat{\theta}}(z)$, else reject, repeat (1).

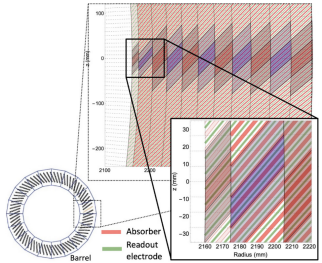
Parameters: $\mathbf{M}$ and $\mathbf{Z}$

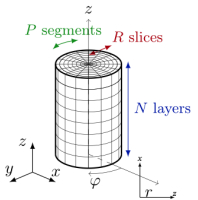
[1] NLL loss: $L(S_d; \theta) = -\sum_{S_d} \log p_m(x_i; \theta)$

[2] BCE loss: $L(Z_d, Z_m; \phi) = \sum_{Z_d} \log[D(z_i; \phi)] + \sum_{Z_m} \log[1 - D(z_i; \phi)]$ (Noise-Contrastive Density Estimation [4])

Generative model and training data



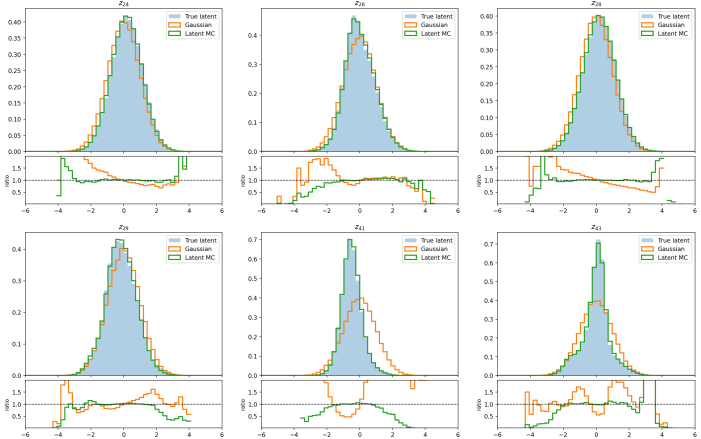




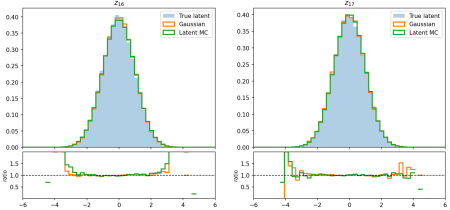
iCaloFlow [2]: Normalizing flow based on 3 Masked Autoregressive Flows [7]. Focus on **Flow 1**.
LEMURS [6]: Open calorimeter dataset for ML development.

Geometry	FCCee ALLEGRO
Inc. particle	γ
Inc. energy	$1 - 100$ GeV
$N_z \times N_r \times N_\phi$	$45 \times 9 \times 16$
N. showers	10^6
η	$(-0.76, 0.76)$

Latent space Monte-Carlo

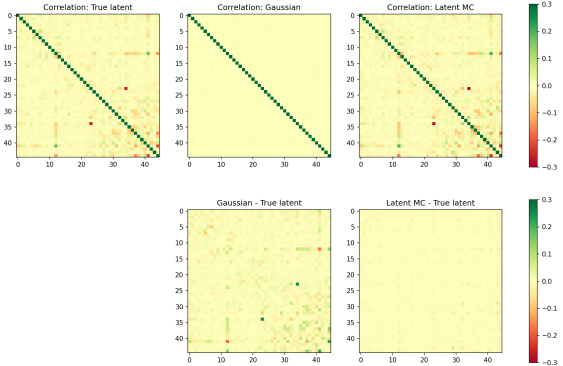


- Latent space showers not strictly Gaussian
- Better agreement with data after MC sampling



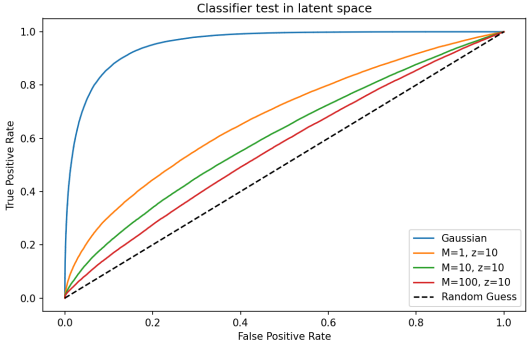
- Does not degrade agreement in dimensions already Gaussian

Multivariate metrics: Correlation and classifier test



(Upper) Correlation matrices of latent variables. (Lower) Difference from data correlation matrices.

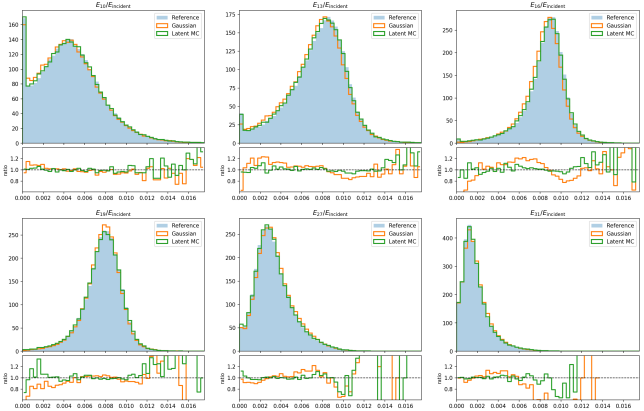
$\mathcal{N}(0,1)$ does **not** reflect the correlation in truth. MC fixes this $\Rightarrow$ DRE learns multivariate patterns.



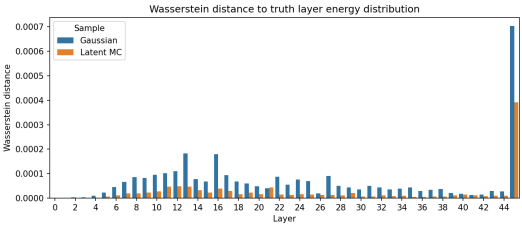
Classifier test on gaussian and MC latent samples (closer to diagonal is better)

ROC AUC $\downarrow$, more difficult to distinguish MC from truth. MC works.

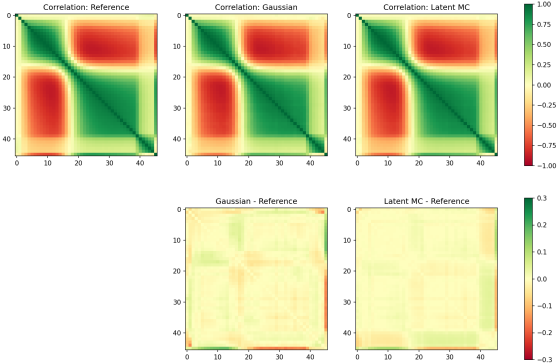
Data space marginal distributions: Layerwise energy deposits

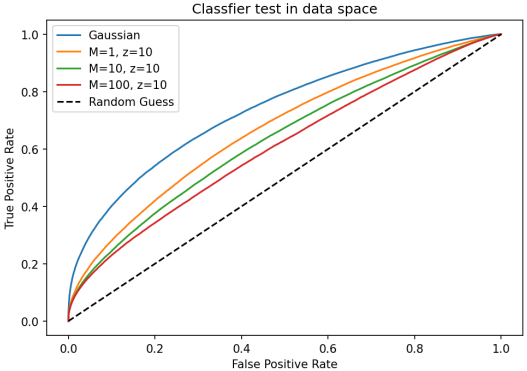


- Better latent sample improves decoded data-space sample
- Both visual and quantitative quality improve



Multivariate metrics: Correlation and classifier test in data space





Both correlation and classifier test show better agreement with data in showers decoded from MC latent.

Effect of noise amplifier $\mathbf{M}$ and normalization factor $\mathbf{Z}$

Recall $\mathbf{M}$ controls noise-to-signal ratio in DRE training[3]. Infinite noise converges to direct maximum likelihood estimation [4]. $\mathbf{Z}$ determines acceptance rate of MC sampling.[4] Large Z = lower rate but more "faithful" to density ratio distribution.

Space	$\mathbf{Z}$	$\mathbf{M}$	AUC
Latent	10	1	0.678 ± 0.002
Latent	5	10	0.625 ± 0.002
	10		0.603 ± 0.002
	100		$\mathbf{0.566 \pm 0.001}$
Latent	10	100	$\mathbf{0.564 \pm 0.002}$
Gaussian Latent			0.9554 ± 0.0007

Space	$\mathbf{Z}$	$\mathbf{M}$	AUC
Data	10	1	0.671 ± 0.002
Data	5	10	0.674 ± 0.002
	10		0.630 ± 0.006
	100		$\mathbf{0.604 \pm 0.005}$
Data	10	100	$\mathbf{0.606 \pm 0.006}$
Decoded Gaussian			0.736 ± 0.007

Best physics performance at high noise and normalization. Need to balance with acceptance rate. Empirically estimate normalization at a given noise-level by $F^{-1}_{w(x)}(0.99)$.

[3] $L(Z_d, Z_m; \phi) = \sum_{Z_d} \log[D(z_i)] + \sum_{Z_m} \log[1 - D(z_i)]; \quad \mathbf{M} = \frac{|Z_m|}{|Z_d|}$

[4] z accepted if $U(0,1) < \frac{D(z)}{\mathbf{Z}(1-D(z)))}$

Summary

- A classifier trained to distinguish generated and data samples is an effective density ratio estimator.
- Monte-Carlo sampling latent space of a normalizing flow driven by a DRE cheaply improves data-space agreement.
- Promising approach to improve generative modelling when data/model is limited.

Outlook

- Is one-step DRE sufficient ? Is Gaussian noise appropriate ?
- Can these results hold up in large latent space ?

Thank You

Tuan Pham

tuan.pham@ijclab.in2p3.fr

IJCLab – Laboratoire de Physique des 2 Infinis – Irène Joliot-Curie

References I

ATLAS Software and Computing HL-LHC Roadmap.
Technical report, CERN, Geneva, 2022.

M. R. Buckley, I. Pang, D. Shih, and C. Krause.
Inductive simulation of calorimeter showers with normalizing flows.
Physical Review D, 109(3), Feb. 2024.

F. Ernst, L. Favaro, C. Krause, T. Plehn, and D. Shih.
Normalizing flows for high-dimensional detector simulations.
SciPost Phys., 18:081, 2025.

M. U. Gutmann and A. Hyvärinen.
Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics.
Journal of Machine Learning Research, 13(11):307–361, 2012.

References II

C. Krause, M. Faucci Giannelli, G. Kasieczka, B. Nachman, D. Salamani, D. Shih, A. Zaborowska, O. Amram, K. Borras, M. R. Buckley, E. Buhmann, T. Buss, R. P. Da Costa Cardoso, A. L. Caterini, N. Chernyavskaya, F. A. G. Corchia, J. C. Cresswell, S. Diefenbacher, E. Dreyer, V. Ekambaram, E. Eren, F. Ernst, L. Favaro, M. Franchini, F. Gaede, E. Gross, S.-C. Hsu, K. Jaruskova, B. Kách, J. Kalagnanam, R. Kansal, T. Kim, D. Kobylianskii, A. Korol, W. Korcari, D. Krücker, K. Krüger, M. Letizia, S. Li, Q. Liu, X. Liu, G. Loaiza-Ganem, T. Madula, P. McKeown, I.-A. Melzer-Pellmann, V. Mikuni, N. Nguyen, A. Ore, S. Palacios Schweitzer, I. Pang, K. Pedro, T. Plehn, W. Pokorski, H. Qu, P. Raikwar, J. A. Raine, H. Reyes-Gonzalez, L. Rinaldi, B. L. Ross, M. A. W. Scham, S. Schnake, C. Shimmin, E. Shlizerman, N. Soybelman, M. Srivatsa, K. Tsolaki, S. Vallecorsa, K. Yeo, and R. Zhang.
Calochallenge 2022: a community challenge for fast calorimeter simulation.
Reports on Progress in Physics, 88(11):116201, nov 2025.

P. McKeown, P. Raikwar, and A. Zaborowska.
Lemurs dataset: Large-scale multi-detector electromagnetic universal representation of showers, 2025.

References III

G. Papamakarios, T. Pavlakou, and I. Murray.
Masked autoregressive flow for density estimation, 2018.

C. O. Software and Computing.
CMS Phase-2 Computing Model: Update Document.
Technical report, CERN, Geneva, 2022.

Rejection sampling

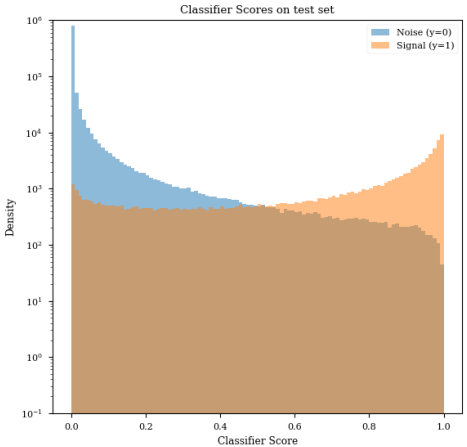
Problem: Given $S_1 = \{x_i\}, x_i \sim p_m$, how can we resample it such that the subsample S_2 follows $P(X = x_k) = p_d(x_k) \forall x_k \in S_2$? Instead of accepting S_1, for each $x_i \in S_1$, accept with probability $w(x)$. The resulting sample follows a density given by:

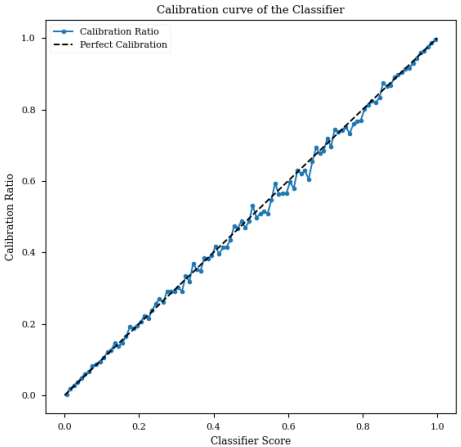
$$P_X(X = x) = \frac{w(x)p_m(x)}{\int w(x')p_m(x')dx'} = p_d(x) \rightarrow w(x) \propto \frac{p_d(x)}{p_m(x)}$$

In practice, the normalization factor can be set to ensure $w(x) \leq 1 \, \forall x \in S_1$. Given a classifier $D(x)$ trained with BCE loss to distinguish between p_d $(y = 1)$ and p_m $(y = 0)$:

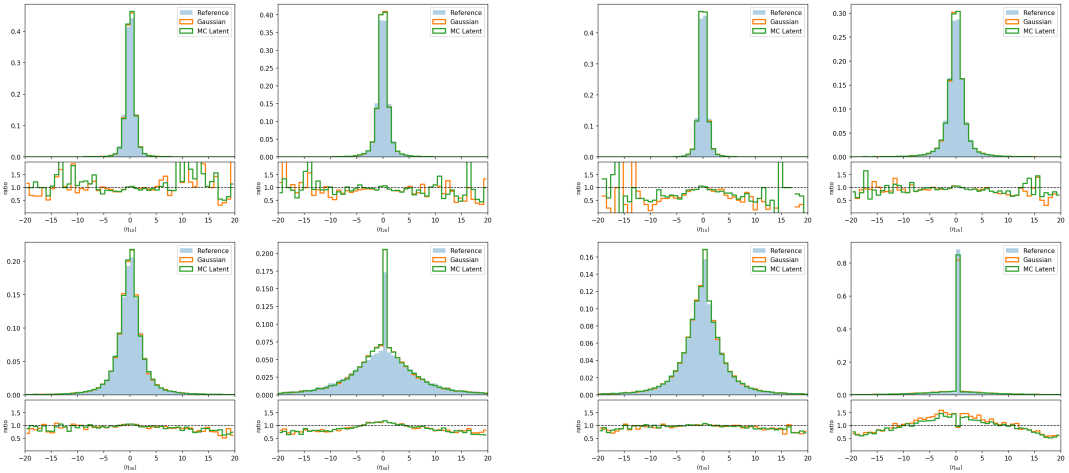
$$\begin{aligned} \frac{D(x)}{1 - D(x)} &\approx \frac{P(Y = 1|X = x)}{P(Y = 0|X = x)} = \frac{P(Y = 1, X = x)}{P(Y = 0, X = x)} \\ &\approx \frac{P(X = x|Y = 1)P(Y = 1)}{P(X = x|Y = 0)P(Y = 0)} = \frac{p_d(x)}{p_m(x)} \propto w(x), \text{ (assuming } P(Y = 1) = P(Y = 0)) \end{aligned}$$

Calibration curve

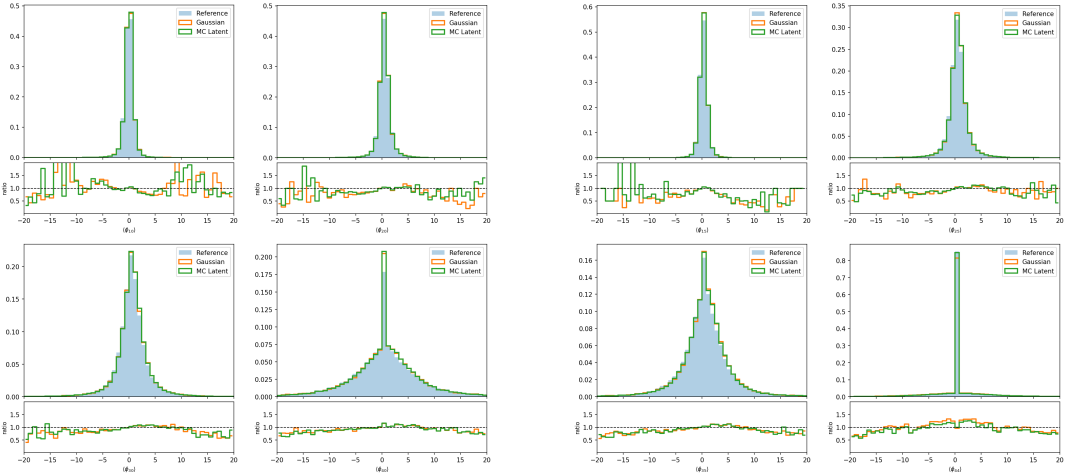




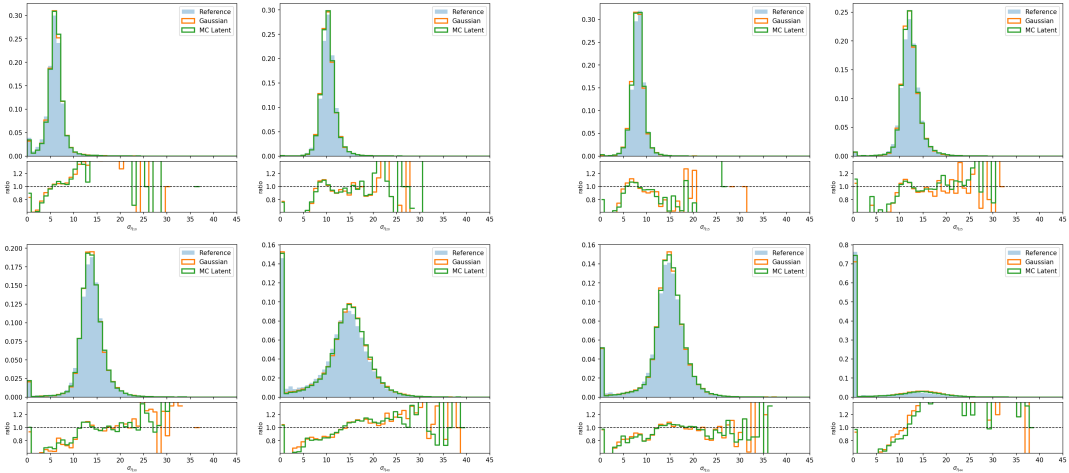
Shower η center



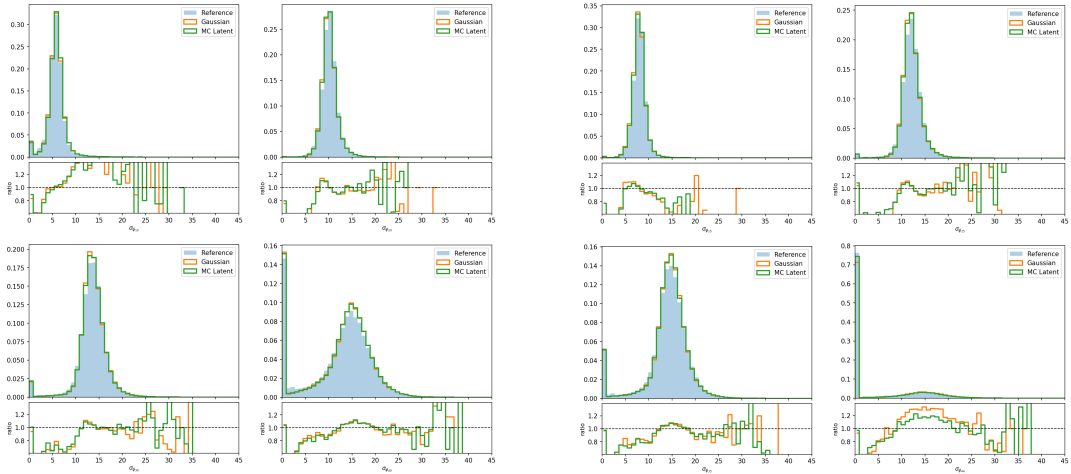
Shower ϕ center



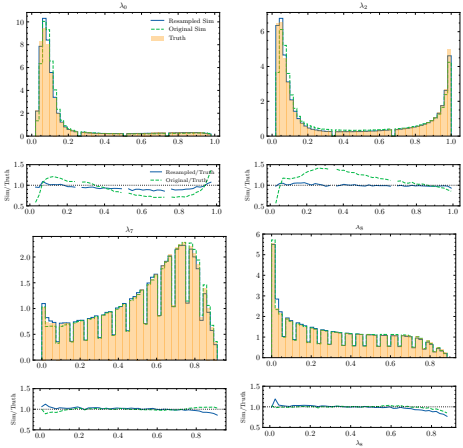
Shower η width



Shower ϕ width



High-level features: Sparsity



Sparsity is the fraction of voxels with non-zero energy deposit. It is not included in the input to the classifier.

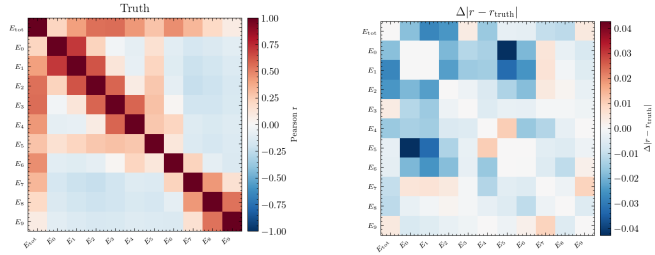
Key Observations

- Improvement in sparsity distribution with resampling
- Possible explanation: sparsity is correlated with other features (e.g. shower width) that are included in the classifier input

Pearson Correlation Coefficient (PCC)

PCC between features X_i and X_j:

$$r_{ij} = \frac{\mathrm{Cov}(X_i, X_j)}{\sigma_{X_i}\sigma_{X_j}} = \frac{\mathbb{E}[(X_i - \mu_{X_i})(X_j - \mu_{X_j})]}{\sigma_{X_i}\sigma_{X_j}}$$

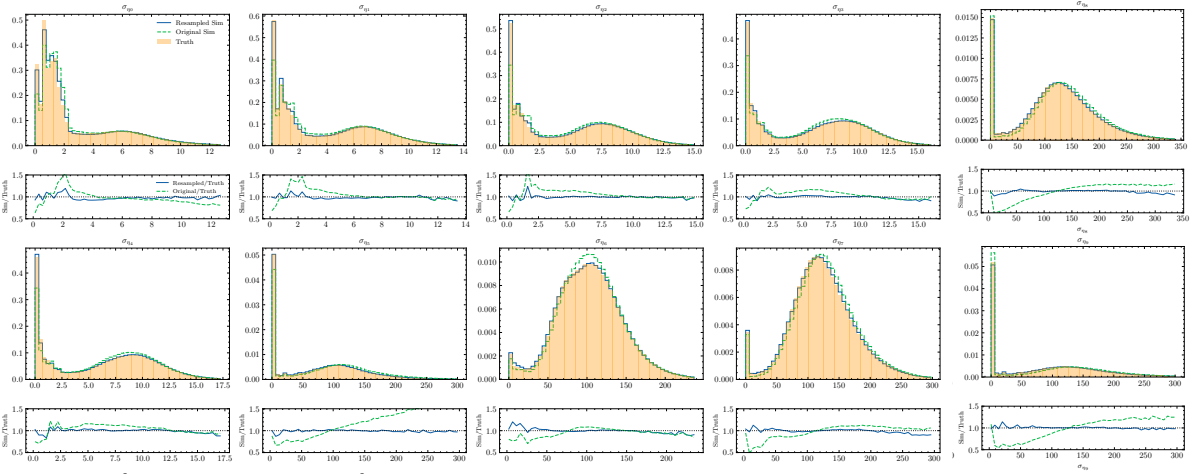


(Left) PCC of total and per-layer energy from GEANT4.
(Right) $|r_{\mathsf{res}} - r_{\mathsf{truth}}| - |r_{\mathsf{sim}} - r_{\mathsf{truth}}|$ (blue is better).

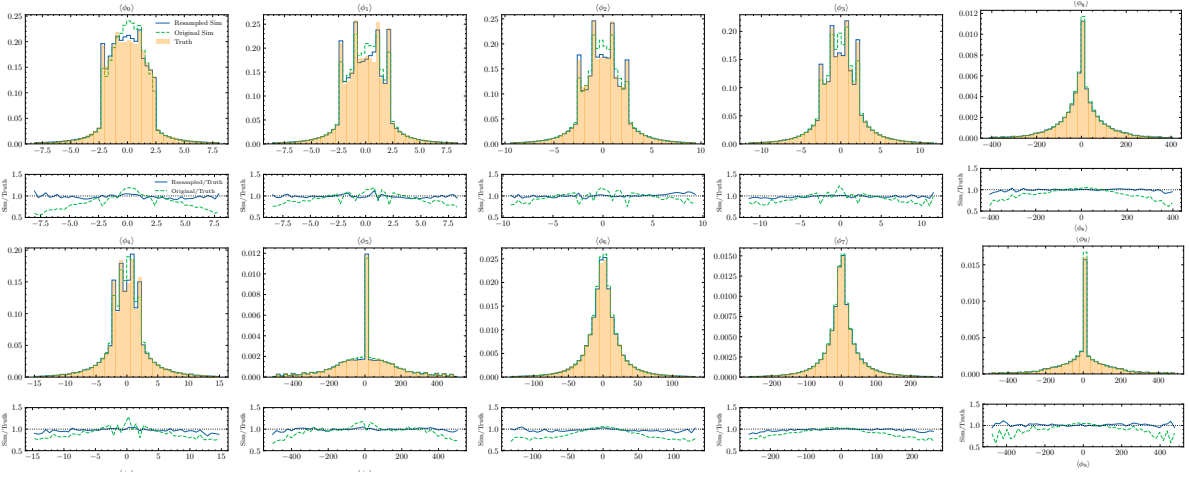
Key Observations

- For most pairs (X_i, X_j), a smaller difference (better agreement) between sample and truth PCC is observed, up to -0.04
- Small degradations observed, magnitude < 0.01
- The classifier is effective at capturing the correlation between features

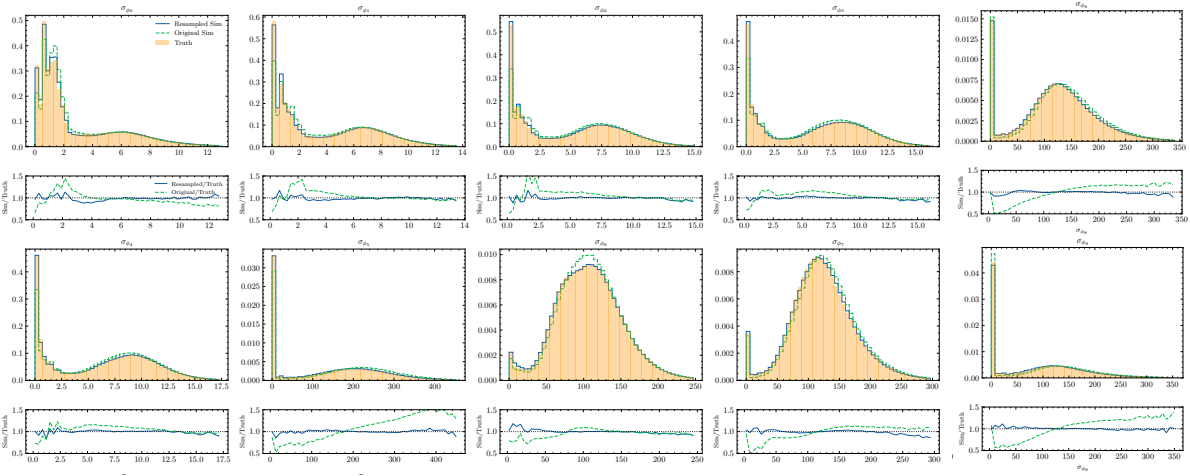
Shower η width



Shower ϕ center



Shower ϕ width



Shower sparsity