CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds

Jesse Cresswell

Layer 6 AI

Nov. 2, 2022

ML4Jets 2022 at Rutgers University

Team

About me:
PhD in T-HEP at U.Toronto on AdS/CFT and entanglement.
Now Sr. Scientist at Layer 6 AI, TD's ML research lab.

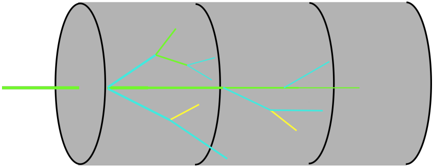
Work done in collaboration with:
Anthony Caterini, Layer 6 AI
Brendan Ross, Layer 6 AI
Gabriel Loaiza-Ganem, Layer 6 AI
Humberto Reyes-González, University of Genoa
Marco Letizia, University Of Genoa

Fast Calorimeter Simulation Challenge 2022

Physics-based simulations of calorimeter showers are slow.
Challenge: train a **surrogate model** that can generate realistic showers quickly and from the correct distribution.

Deep generative models trained on shower data can learn the distribution of showers, and enable fast sampling.

Challenge presents 3 datasets of EM showers and standardized metrics for evaluating model performance.



In the standard scenario for DGMs we are given IID samples $\{x_n\}_{n=1}^{N} \subset \mathbb{R}^D$ from a distribution $\mathbb{P}^*$.

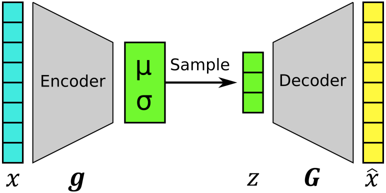
For a specific calorimeter experiment $\mathbb{P}^*$ could be determined in principle from the laws of physics, but is intractable in practice.

We aim to estimate $\mathbb{P}^*$ with a DGM.

Popular DGMs include

- Variational Autoencoders
- Normalizing Flows
- Generative Adversarial Networks
- Score-based Generative Models
- Diffusion Models

Variational Autoencoders (VAEs) (Kingma & Welling; ICLR 2014) learn a probabilistic **encoder** $g : \mathbb{R}^D \to \mathbb{R}^d$ mapping data to a latent space, and a probabilistic **decoder** $G : \mathbb{R}^d \to \mathbb{R}^D$ mapping latents to data.



Often g_θ and G_θ are parameterized by neural networks that output μ and σ for multivariate Gaussians.

Training is based on **maximum likelihood estimation**

$$\theta^* = \arg\max_\theta \left(\prod_{n=1}^{N} \log p_\theta(x_n) \right) \tag{1}$$

where $p_\theta(x_n)$ is the likelihood of the data under the modelled density.

Manifold Hypothesis

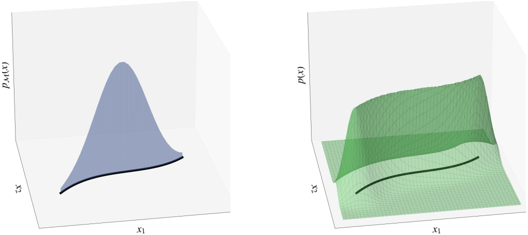
The **Manifold Hypothesis** states that high-dimensional real-world data is supported on a low-dimensional embedded submanifold $\mathcal{M} \subset \mathbb{R}^D$.
(Bengio, Courville, Vincent; IEEE TPAMI 2013)

EM calorimeter showers are highly structured.
Constraints of QED processes $\implies$ shower data has manifold structure.

Hence, the target distribution $\mathbb{P}^*$ is supported on $\mathcal{M}$, not $\mathbb{R}^D$.

What happens when we try to model $\mathbb{P}^*$ with a DGM that learns a density $p_\theta(x)$ on $\mathbb{R}^D$?

Maximum likelihood estimation can fail when the dimensionalities of $p_\theta(x)$ and $\mathbb{P}^*$ differ. **Manifold overfitting** can occur where $\mathcal{M}$ is learned but not the distribution $\mathbb{P}^*$ on it. (Loaiza-Ganem, Ross, Cresswell, Caterini; TMLR 2022)



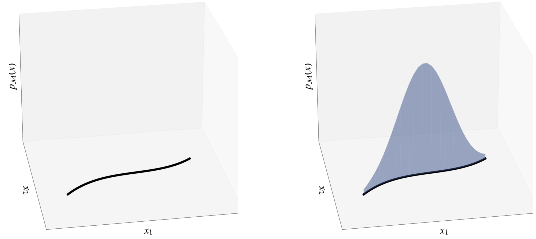
To maximize the likelihood of the data, the density is sent to infinity around $\mathcal{M}$, where $\mathcal{M}$ is a set of measure zero wrt Lebesgue measure.

This does not happen when $p_\theta(x)$ and $\mathbb{P}^*$ have the same dimensionality because $p_\theta(x)$ must remain normalized.

How can we prevent manifold overfitting?

Though commonly used, adding full-dimensional noise to the data changes $\mathbb{P}^*$, and destroys manifold structure.

The simple solution is a **two-step approach**: first learn the data manifold, then estimate the distribution on it.



Two-Step Generative Models

1) Learn $\mathcal{M}$ with a *generalized autoencoder* - any model that constructs a low-dimensional encoding $z = g(x)$, and can reconstruct data with a decoder $x = G(z)$.

Examples: autoencoder, VAE, Wassertein autoencoder, bi-directional GAN.

2) Perform density estimation on the manifold, obtaining the low-dimensional density $p(z)$.

DGMs that explicitly construct $p(z)$: VAEs, normalizing flows, energy-based, auto-regressive, score-based, and diffusion models.

Calorimeter Shower Manifolds

The first-step model learns a latent space of fixed dimensionality d.

Use a statistical estimator of intrinsic dimension (Levina & Bickel; NeurIPS 2004)

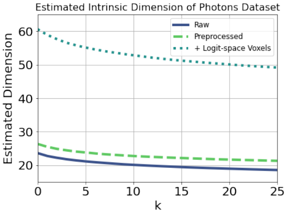
$$\hat{d}_k = \left(\frac{1}{n(k-1)} \sum_{i=1}^{n} \sum_{j=1}^{k-1} \log \frac{T_k(x_i)}{T_j(x_i)} \right)^{-1}, \tag{2}$$

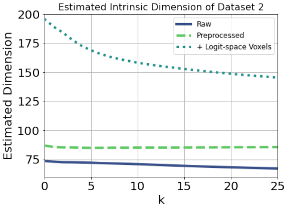
$T_k(x_i)$ - Euclidean distance between x_i and its kth nearest neighbour.
k - scale at which the manifold is probed.

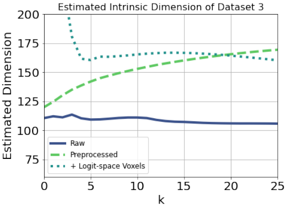
Estimator is derived from the expected number of neighbours per unit volume as dimension increases.

Although calorimeter shower data is high-dimensional, its intrinsic dimension is estimated to be much lower.

Photons dataset showers have 368 voxels, but $\hat{d}_k = 20$.







Electron datasets 2 & 3 have the same layout, but 3 has higher resolution, 6,480 and 40,500 voxels respectively. Estimates of $\hat{d}_k$ are 75 and 110.

Photons Dataset - Preliminary Results

Two-step models **reduce dimension**, so that training and sampling are **extremely fast** compared to full-dimensional models.

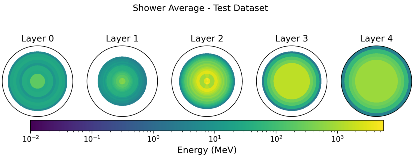
1) VAE parameterizes the encoder and decoder as MLP networks with 3 hidden layers of 512 units - output the parameters of diagonal Gaussians.

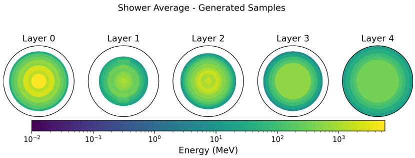
2) NF trained on 20-dimensional latent space is a 4-layer rational-quadratic neural spline flow. We used `nflows`: github.com/bayesiains/nflows.

VAE and NF trained for 200 epochs each, requiring 1 GB memory and 110 minutes on a Titan V GPU.

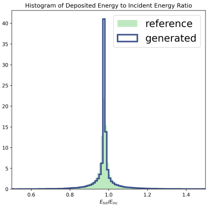
Code for training two-step models is available at github.com/layer6ai-labs/two_step_zoo.

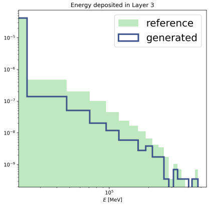
VAE was trained without conditioning. Adding conditioning improves shower averages, but encourages a segmented latent representation that may not generalize.

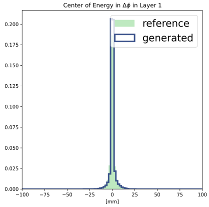


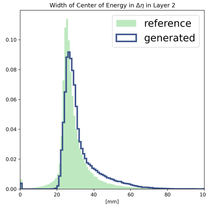


Comparison of histograms between test set, and generated samples:









CaloChallenge metrics

Separation power of χ^2 test between histograms, and sampling times (includes time to undo pre-processing).

Feature	χ^2 Power
$E_{\text{dep}}/E_{\text{inc}}$	0.0535
E_{dep}, L0	0.0540
E_{dep}, L1	0.0304
E_{dep}, L2	0.0243
E_{dep}, L3	0.0045
E_{dep}, L4	0.0009
CE in η, L1	0.0376
CE in η, L2	0.0512
CE in ϕ, L1	0.0145
CE in ϕ, L2	0.0391
Width in η, L1	0.1548
Width in η, L2	0.0538
Width in ϕ, L1	0.1080
Width in ϕ, L2	0.0489

Batch Size	Number of showers	Time per shower (ms)
1,000	1,000	0.0598
1,000	100,000	0.0844
5,000	5,000	0.0532
5,000	100,000	0.0315
10,000	10,000	0.0265
10,000	100,000	0.0246
50,000	50,000	0.0216
50,000	100,000	**0.0201**

Binary classifier trained to distinguish real and sampled showers attains only 0.78 AUC. Despite imperfectly learning the distribution, showers are realistic.

Conclusion

Calorimeter showers have low-dimensional structure dictated by physics.

Although this structure has not been understood from first principles, we can incorporate it into our modelling assumptions.

Learning the manifold, then estimating the density on it is a more principled approach that avoids *manifold overfitting*.

Dimensionality reduction also speeds up the training of high powered density estimators, like NFs or diffusion models as in stable diffusion (Rombach et al.; CVPR 2022).

Learning topologically non-trivial manifolds without prior knowledge is also possible (Ross, Loaiza-Ganem, Caterini, Cresswell; 2206.11267).

D. P. Kingma and M. Welling, "Auto-encoding Variational Bayes," *ICLR* (2014) .

Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2013) .

G. Loaiza-Ganem, B. L. Ross, J. C. Cresswell, and A. L. Caterini, "Diagnosing and Fixing Manifold Overfitting in Deep Generative Models," *Transactions on Machine Learning Research* (2022) .

E. Levina and P. Bickel, "Maximum likelihood estimation of intrinsic dimension," *NeurIPS* (2004) .

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis With Latent Diffusion Models," *CVPR* (2022) .

B. L. Ross, G. Loaiza-Ganem, A. L. Caterini, and J. C. Cresswell, "Neural implicit manifold learning for topology-aware generative modelling," *2206.11267* .