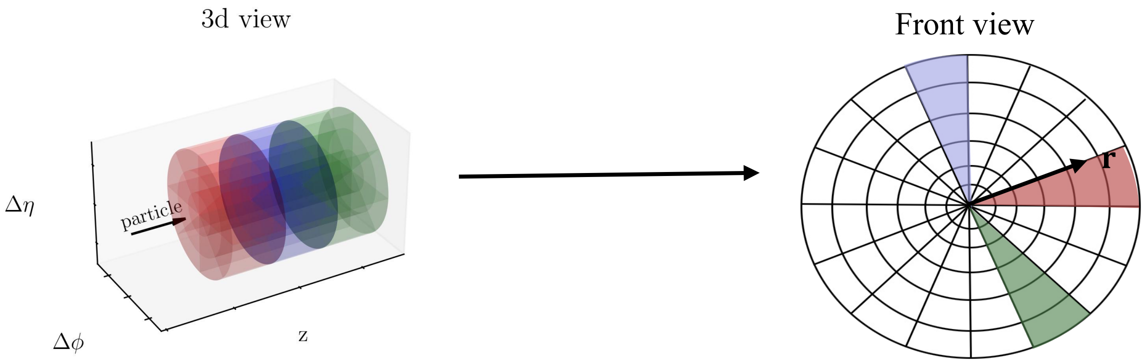
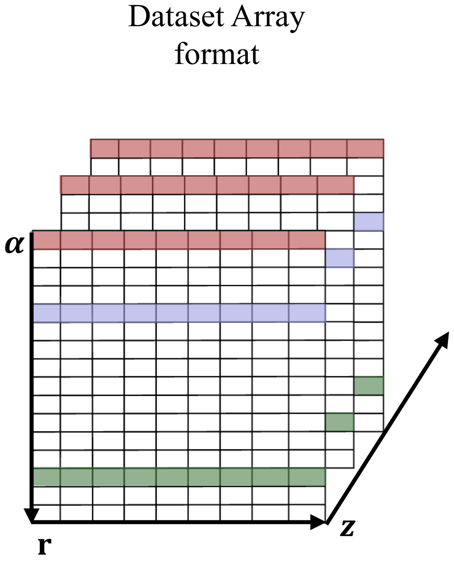
CaloLatent: Score-based Generative Modelling in the Latent Space for Calorimeter Shower Generation

Thandikire Madula: UCL

Vinicius M. Mikuni: NERSC

Surrogates for Calorimeter showers





Data Processing

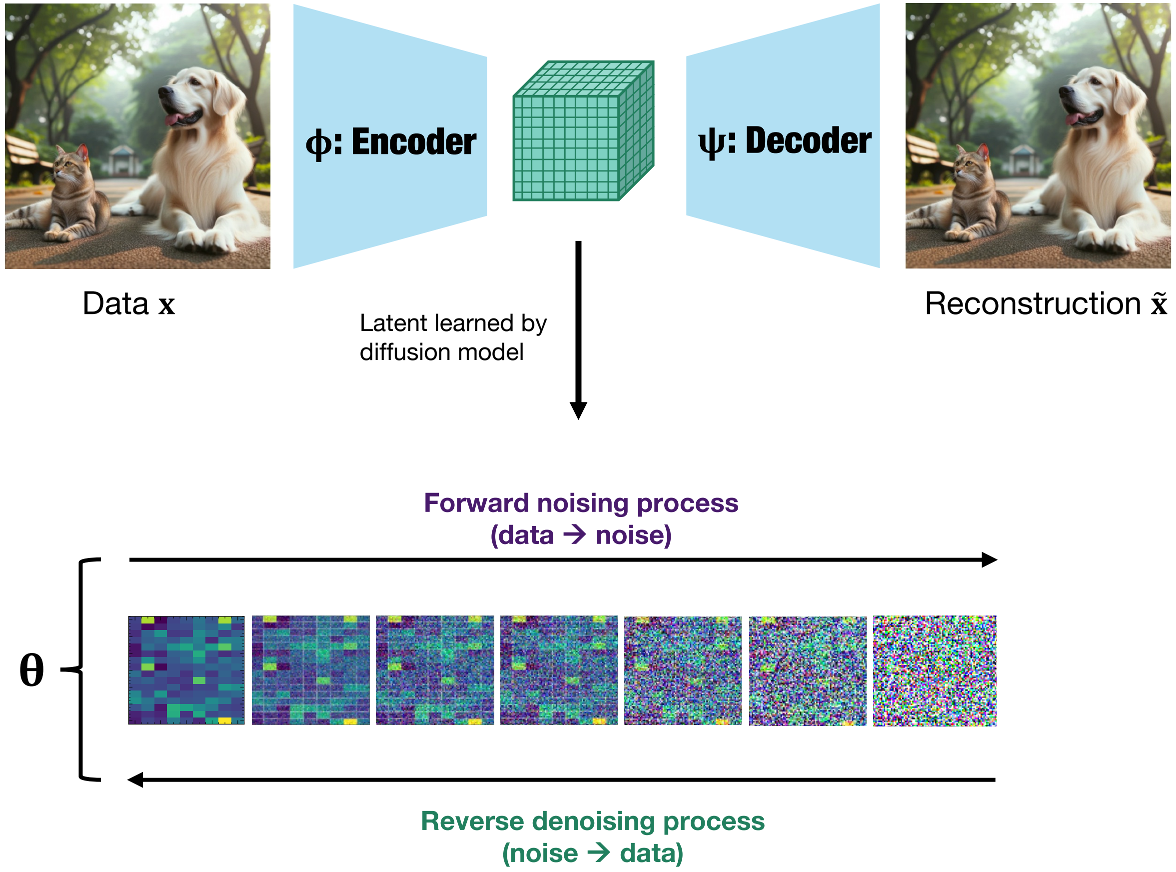
1. Normalise: $\frac{X}{E_{layer}}$
2. Minmax : $\frac{X - X_{min}}{X_{min} - X_{max}}$
3. Logit: $\frac{u - \alpha}{(1-2\alpha)u}$, $u = \frac{e^X}{1+e^X}$
4. Standardise: $\frac{X - \mu}{\sigma}$

- Calorimeter shower simulation is a costly step in the simulation pipeline
- As experiment luminosity and calorimeter granularity increase, this bottleneck worsens
- This motivates the development of fast surrogate models in a bid to alleviate this problem
- We **focus on the dataset 2** of the fast calorimeter challenge to evaluate our proposed surrogate model
- 45 x 16 x 9 = 6480

Fast Calorimeter Simulation Challenge

Latent Diffusion

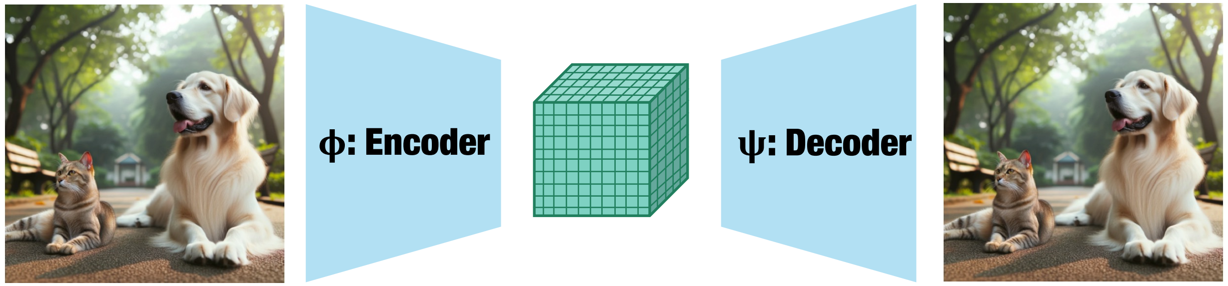
Rombach *et al*: arXiv:2112.10752v2 **[cs.CV]**



- **Main idea:**
  a) Map the data into a reduced latent space representation using a variational autoencoder
  b) Train a diffusion model in the VAE bottleneck
  c) Generate samples $\tilde{\mathbf{x}} = \boldsymbol{\psi}\big(\boldsymbol{\theta}(z)\big), \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- **Motivation:**
  - **Reduced data dimensionality**, diffusion models are in general slower to sample from than other generative models. If we only diffuse in the latent, we should speed up sample generation
  - **Better reconstruction,** we can generate samples using an approximation of the "true" latent instead of a multivariate normal
- Generation speed exclusively determined by diffusion model and decoder; encoder only relevant at train time

Variational AutoEncoders (VAEs)

VAEs: D. Kingma, M.Welling arXiv:1312.6114v11 **[stat.ML]**



Data $\mathbf{x}$: $\mathbb{R}^{\mathbf{D}}$

Latent: $\mathbb{R}^{\mathbf{d}}$

Reconstruction $\tilde{\mathbf{x}}$: $\mathbb{R}^{\mathbf{D}}$

$$\mathcal{L}_{VAE} = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[-\log p_\psi(\mathbf{x}\,|\mathbf{z})\right]}_{\text{Reconstruction term}} + \underbrace{\beta}_{\text{Regularisation weight}} \underbrace{\mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}_{\text{Regularisation term}}$$

- In latent diffusion, having a good VAE backbone is crucial
- The upper bound of the latent diffusion is set by the VAE i.e, latent diffusion model can only be as good as or worse than $\tilde{\mathbf{x}} = \boldsymbol{\psi}(\boldsymbol{\phi}(\mathbf{x}))$
- With VAE's there is a balancing act happening:
  - With minimal regularization (small β), the VAE excels in reconstruction, however, the latent representation may lack smoothness. This would necessitate a more elaborate diffusion model to capture the complexity. More elaborate = longer sampling time!
  - With $\beta = 1$ the latent space well be be more gaussian, a simpler diffusion model can be used since you are trying to map almost gaussian →gaussian. Simpler == shorter sampling time. However, with heavy regularisation we sacrifice reconstruction quality!

Score-based Generative Modelling

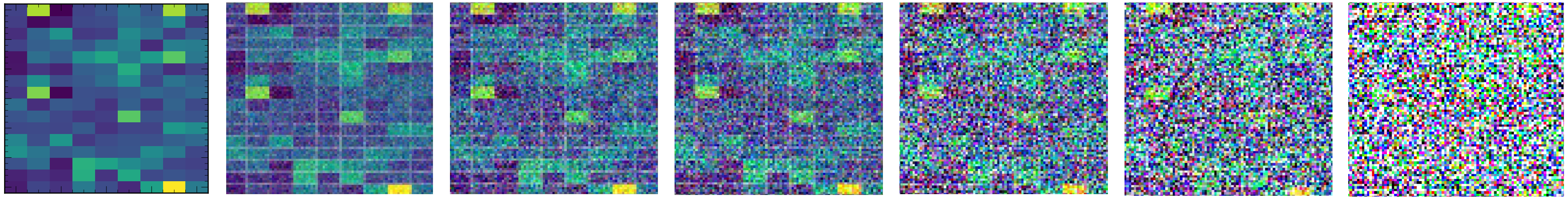
Score Models: Song *et al* arXiv:2011.13456v2 **[cs.LG]**

Drift coefficient

Diffusion coefficient

Wiener Process

$\mathbf{X}(\mathbf{0})$ $\longrightarrow$ $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$ $\longrightarrow$ $\mathbf{X}(\mathbf{T})$



$\mathbf{X}(\mathbf{0})$ $\longleftarrow$ $d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t)dt - g^2(t)\nabla_x \log p_t(\mathbf{x})]dt + g(t)d\bar{\mathbf{w}}$ $\longleftarrow$ $\mathbf{X}(\mathbf{T})$

Score Function

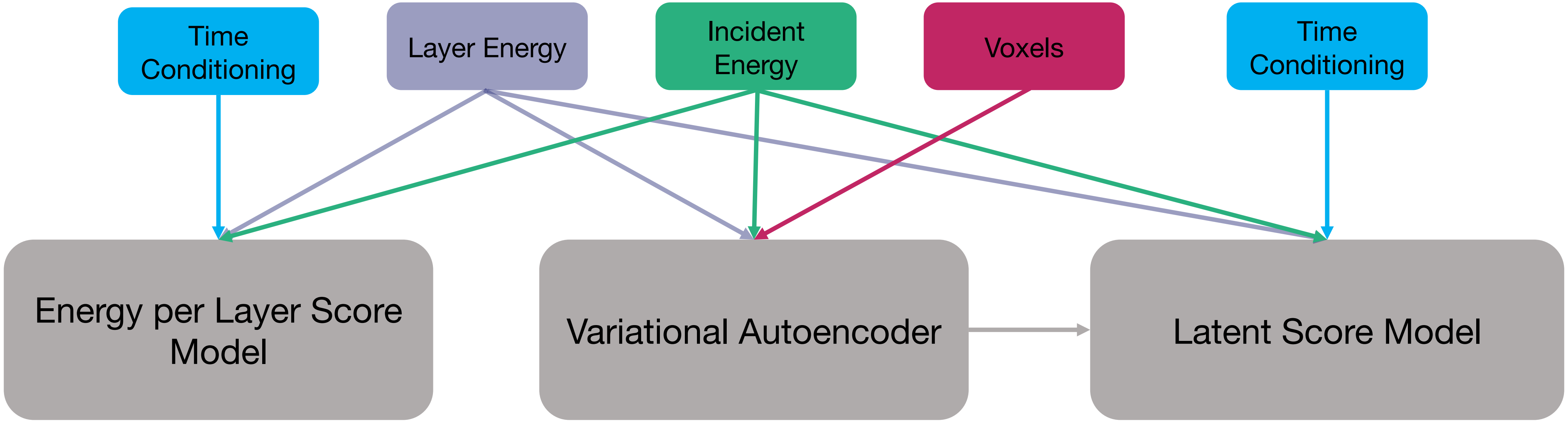
Wiener Process in reverse time

- Choose f and g so that the convolving kernel is a gaussian
- Generate samples from solving reserve SDE

Not real sample from latent space. Representational only!

CaloLatent: The Three Musketeers

ResNet: He *et al* arXiv:1512.03385v1 [cs.CV]



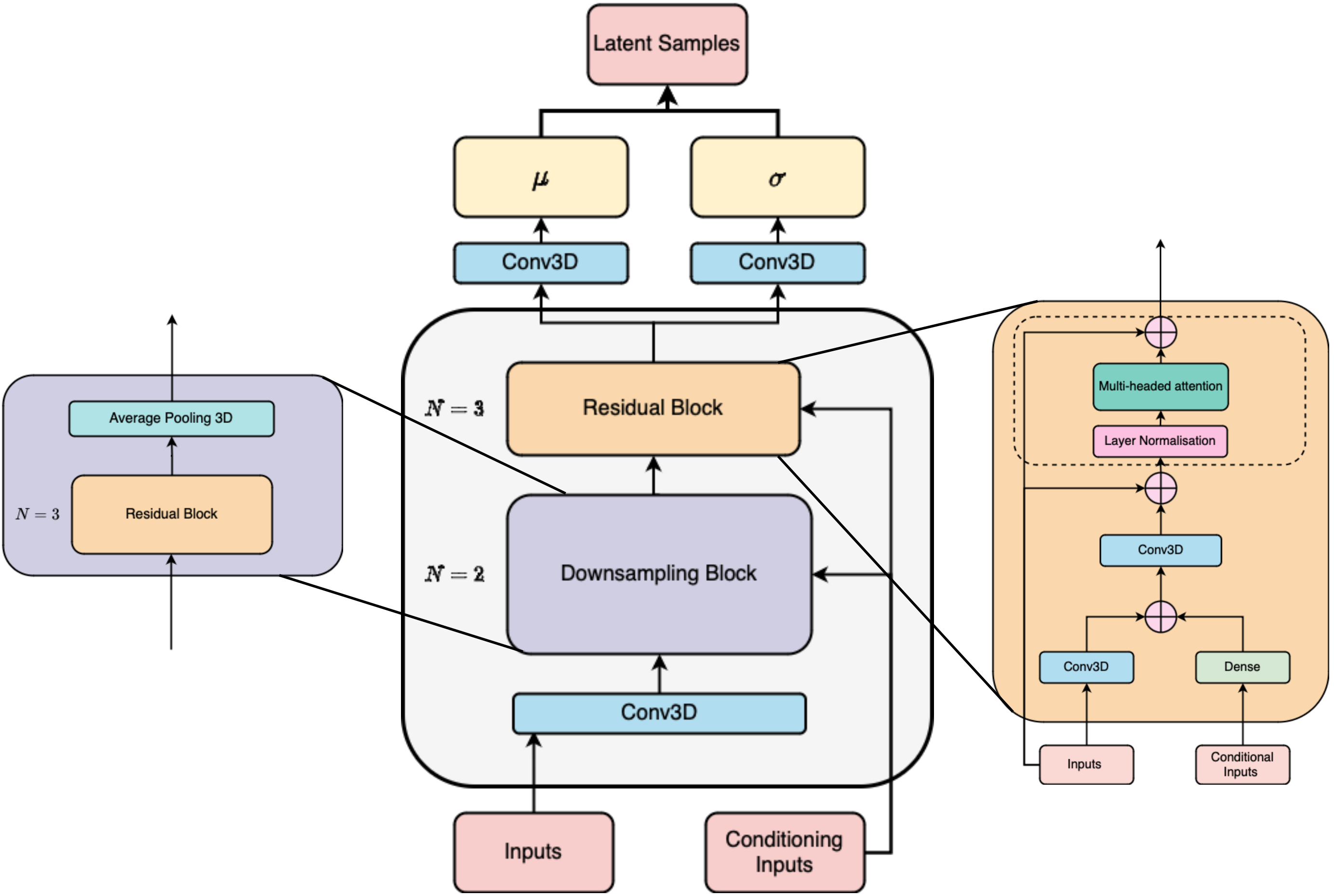
- Used to learn the energy per layer distribution
- **ResNet: 3 layers, 512 nodes**
- Trained independently for 500 epochs

- **Encoder: 8.8M params.**
- **Decoder: 1.9M params.**
- Trained for 500 epochs
- Minimal KL: $\boldsymbol{\beta} = \mathbf{1}\boldsymbol{e}^{-6}$

- **ResNet: 6.5M params.**
- Trained independently for 250 epochs after VAE and Layer model

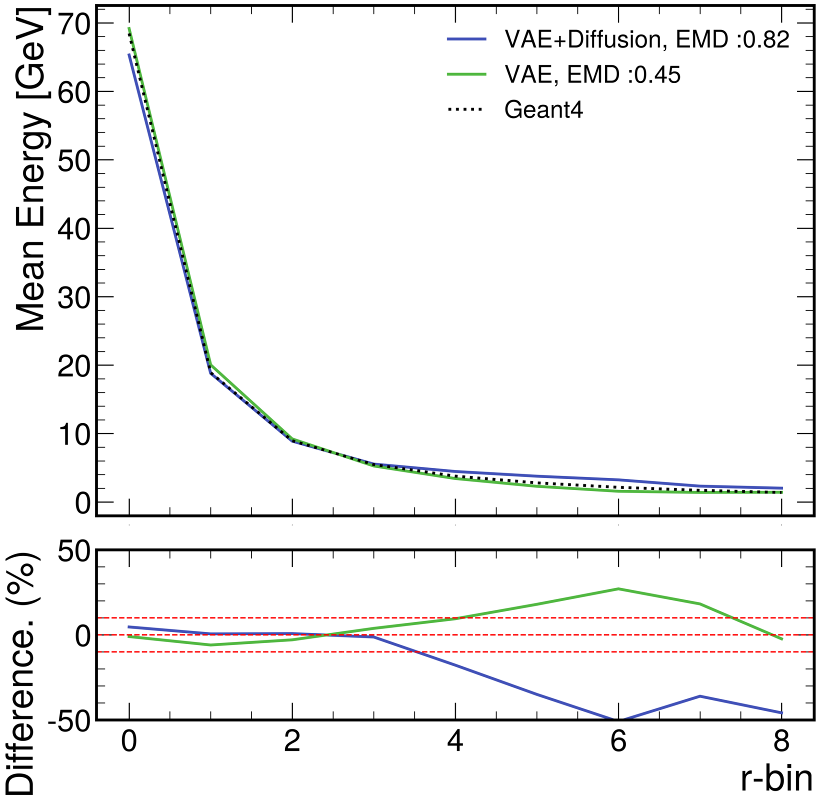
All models trained using 4 A100 GPUs, total training time < 36 hours

Model Architecture: VAE Encoder

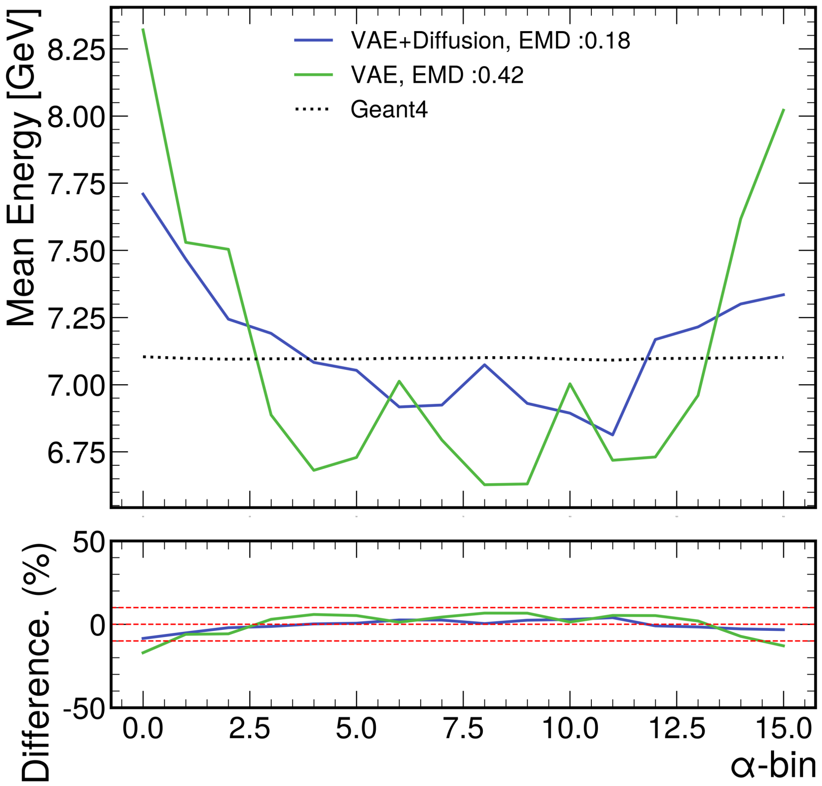


- The VAE backbone for CaloLatent is a 3D convolutional neural network with residual blocks.
- Reduce data dimensionality from **6408 → 1008**
- Increase the number of channels from **1 → 64 → 128 → 256** as we down sample to account for information loss
- Only apply attention in final residual blocks
- Decoder is inversion of Encoder, however, doesn't have residual blocks after up sampling.

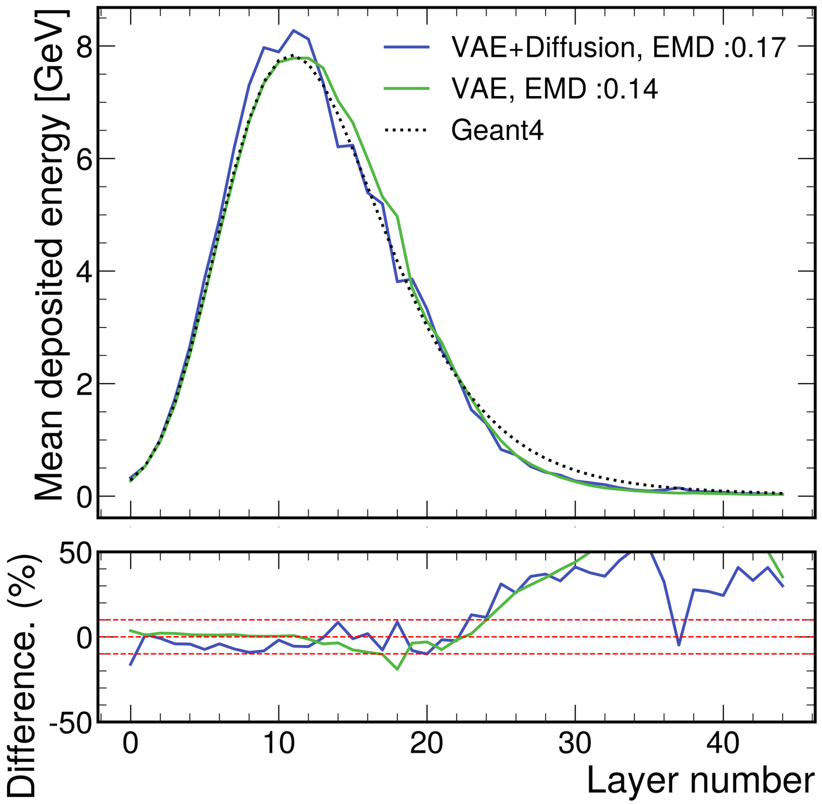
Results



Mean Energy [GeV]
VAE+Diffusion, EMD :0.82
VAE, EMD :0.45
Geant4
Difference. (%)
r-bin

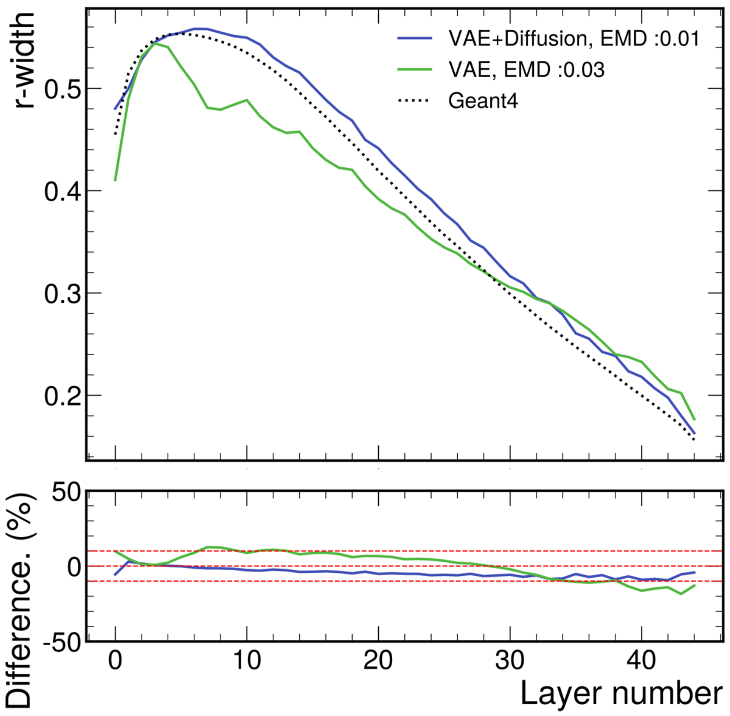


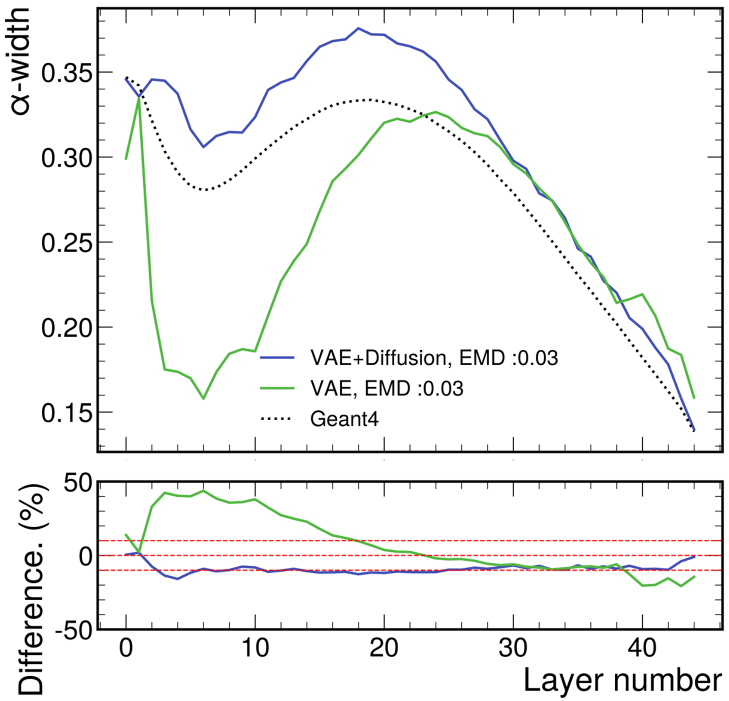
Mean Energy [GeV]
VAE+Diffusion, EMD :0.18
VAE, EMD :0.42
Geant4
Difference. (%)
α-bin

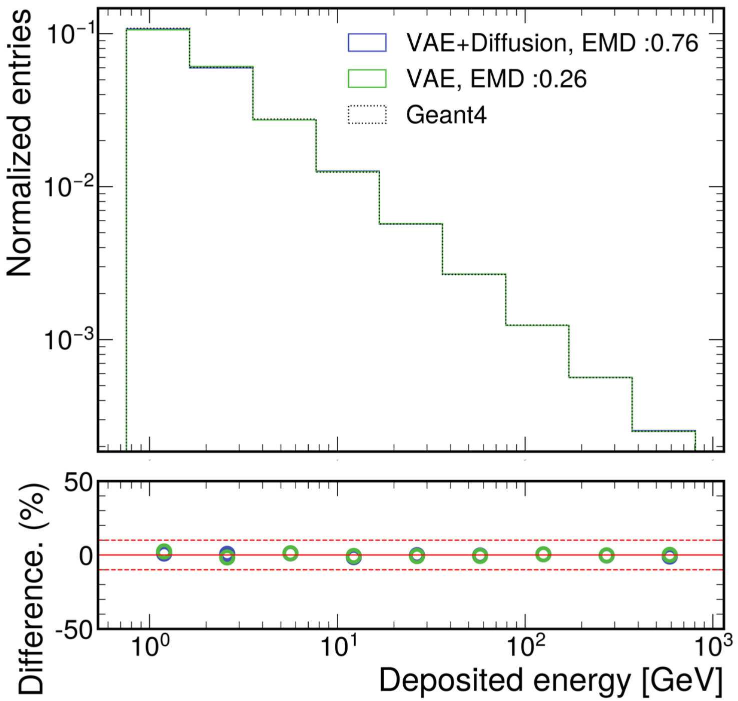


Mean deposited energy [GeV]
VAE+Diffusion, EMD :0.17
VAE, EMD :0.14
Geant4
Difference. (%)
Layer number

Results







$$\boldsymbol{\sigma_i = \sqrt{\langle x_i^2 \rangle - \langle x_i \rangle^2}} \qquad , \langle \mathbf{x_i} \rangle = \frac{\boldsymbol{\Sigma_j\, x_{i,j}\, E_j}}{\boldsymbol{\Sigma_j E_j}}$$

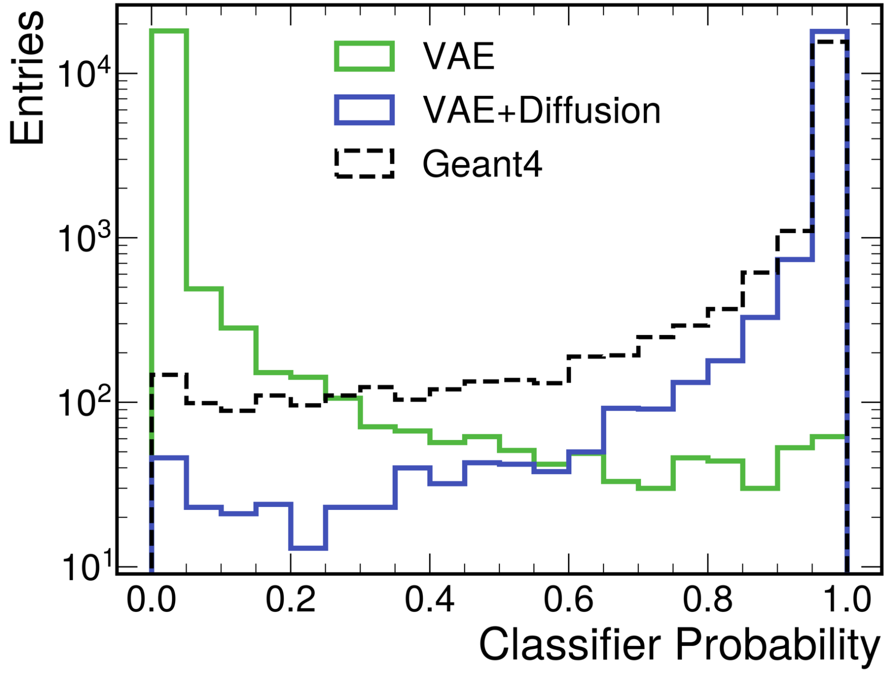
Model Evaluation: Sample Quality

Calorimeter Challenge Classifier Metrics (Model vs Geant4)

	AUC/JSD	
Classifier Inputs	**VAE**	**VAE + Diffusion**
Low-level	0.9951 / 0.8748	**0.9865 / 0.7868**
Norm. Low-level	0.9947 / 0.8907	**0.9808 / 0.7614**
High-level	0.9924 / 0.8462	**0.9662 / 0.6595**

- Optimal AUC/JSD: 0.5 / 0.0
- Low-level → Voxel level variables
- High-level → Histogram level Variable
- VAE + Diffusion → CaloLatent

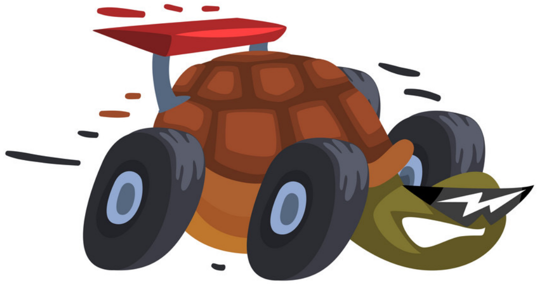
Alternative Classifier Formulation (VAE vs CaloLatent)



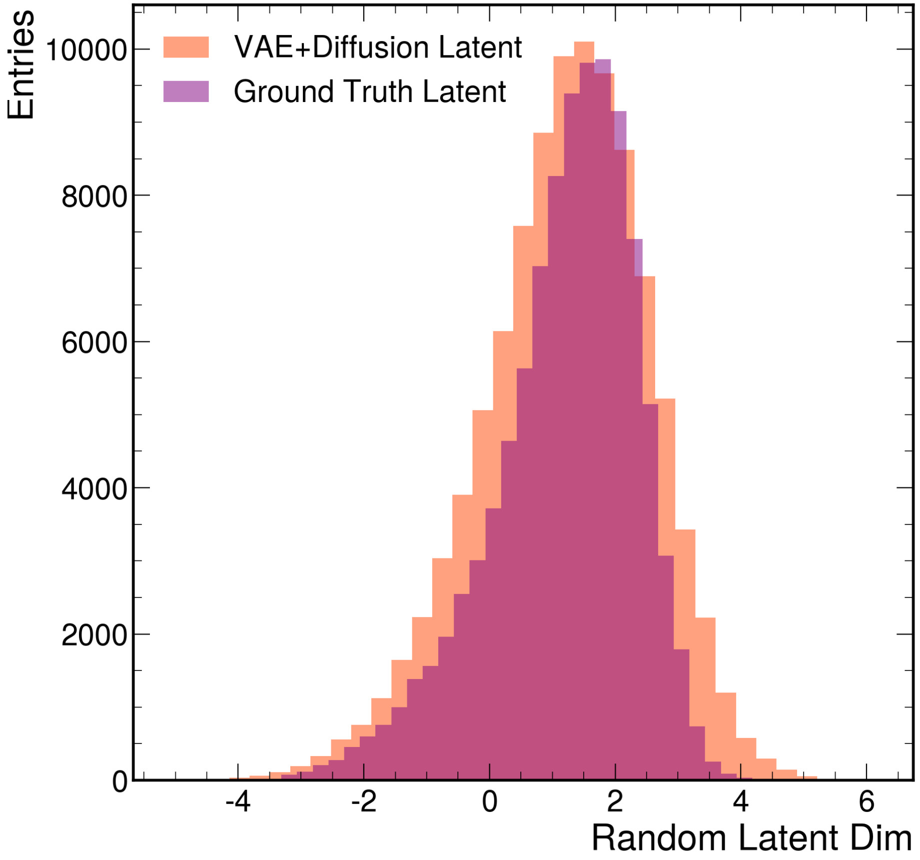
Model Evaluation: Sampling Time

Model	Time to 100 Showers [s]
Geant4	$\mathcal{O}(10^4)$
CaloScore	5.8
CaloScore V2*	27.8
CaloLatent	1.9

- All models evaluated using a single A100 GPU
- CaloScore V2 trains an additional student teacher model to reduce evaluation time to 0.01s for 100 showers



Conclusion



Looking at a random slice of the diffusion latent, compared to the ”true latent” produced by the encoder

Summary

- Have introduced CaloLatent, a latent diffusion-based model
- Demonstrated promising sample generating abilities, with plenty of room for improvement
- Demonstrated faster sampling time compared to similar published diffusion models

Further development

- Improve latent models' ability to capture latent distribution
  - Larger/different architecture
  - Hyperparameter optimisation
- Improve sample generation
  - Larger latent
  - Larger encoder