Pushing Normalizing Flows for higher-dimensional Detector Simulations

Florian Ernst,
in collaboration with
Luigi Favaro, Claudius Krause, Tilman Plehn and David Shih

ITP - Heidelberg University

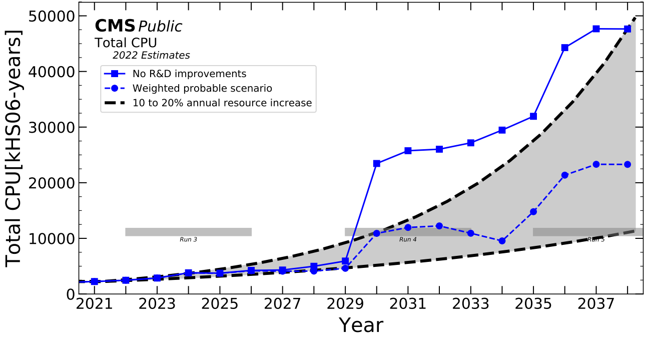
florian.ernst@uni-heidelberg.de

November 6, 2023

Structure

1. Introduction
2. Models
   - INN
   - VAE
3. Results
   - Classifier results
   - High level features
4. Backup Slides

CaloChallenge 2022

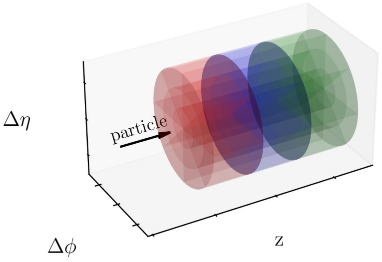


twiki.cern.ch

Datasets

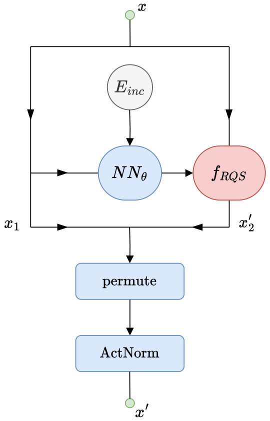
Three datasets of increasing dimensionality

1. Dataset 1
   - 368 voxels — Photons
   - 533 voxels — Pions
2. Dataset 2 — Electrons
   - $45 \times 16 \times 9 = 6480$ voxels
3. Dataset 3 — Electrons
   - $45 \times 50 \times 18 = 40500$ voxels

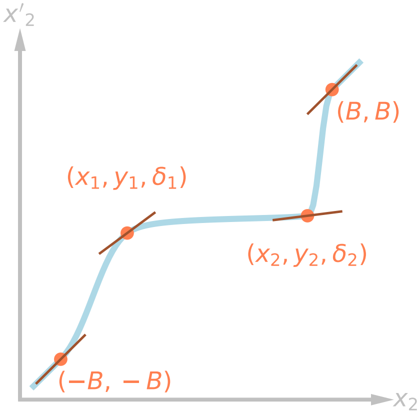


Generation with an INN

Architecture

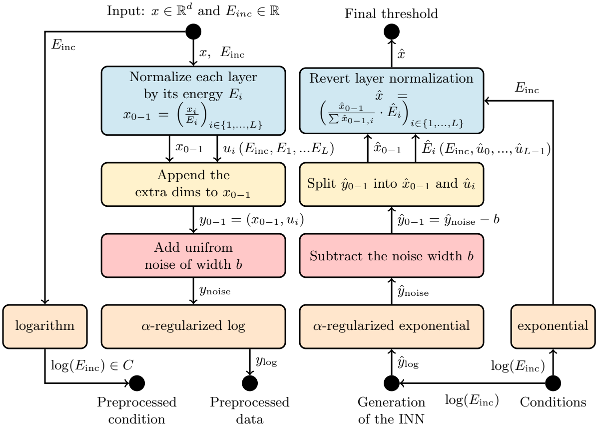


1410.8516



1906.04032

Preprocessing



$$u_0 = \frac{E_{\text{tot}}}{E_{\text{inc}}}$$

$$u_i = \frac{E_i}{\sum_{j=i}^{L} E_j}$$

$$\log_\alpha(x) = \log(x + \alpha)$$

Advantages and Disadvantages

Advantages

- Very accurate generations
- Fast in both directions

Disadvantages

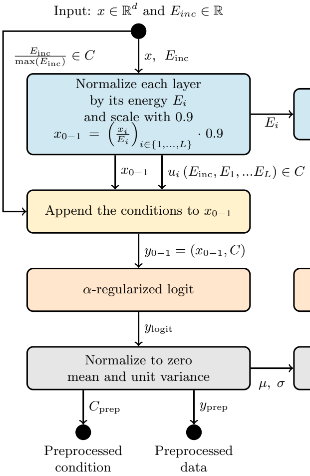
- Bad scaling (time and memory)

Compression with a VAE

Derivation

- Assume $\exists$ true joint distribution of data x and latent z.
- Minimize $D_{\mathrm{KL}}\left[E\left(z|x\right), D\left(z|x\right)\right]$.
- $\Rightarrow \mathcal{L} = -\sum_{x\in TS}\left\langle \log D\left(x|z\right)\right\rangle_{E(z|x)} + \beta\cdot D_{\mathrm{KL}}\left[E\left(z|x\right), p_{\mathrm{latent}}\left(z\right)\right]$

Preprocessing

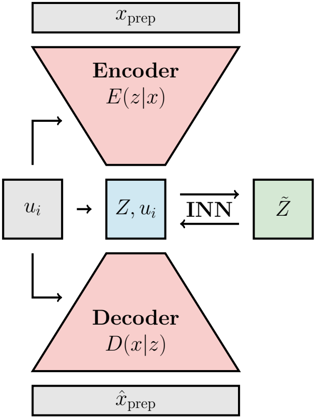


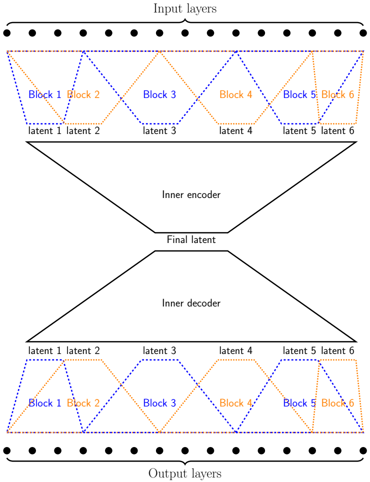
$$u_0 = \frac{E_{\mathrm{tot}}}{E_{\mathrm{inc}}}$$

$$u_i = \frac{E_i}{\sum_{j=i}^{L} E_j}$$

$$\mathrm{logit}_\alpha(x) = \mathrm{logit}((1-2\alpha)x + \alpha)$$

Architecture



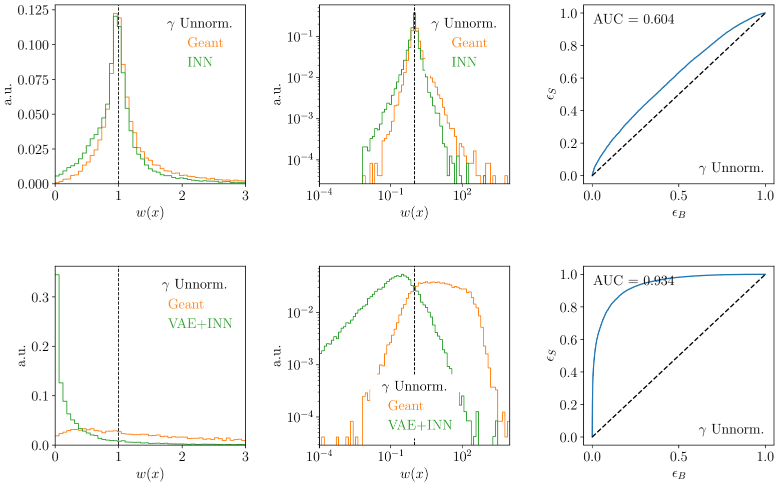


Results

Classifier — Dataset 1 (photons) 2305.16774

INN

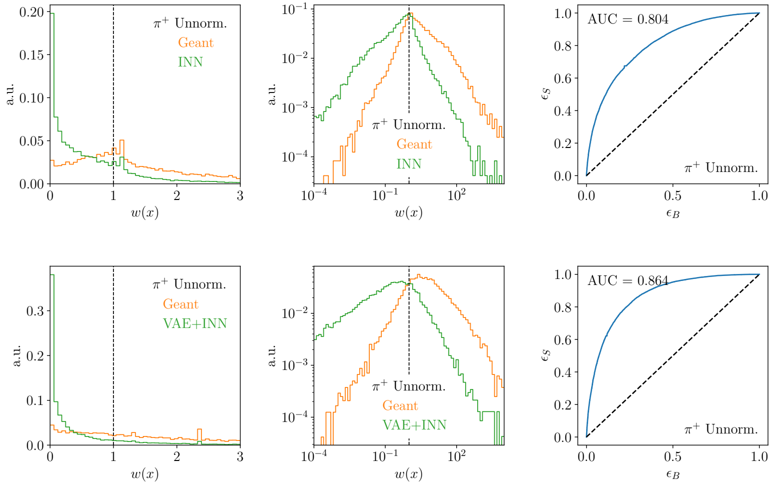
VAE+INN



Classifier — Dataset 1 (pions) 2305.16774

INN

VAE+INN

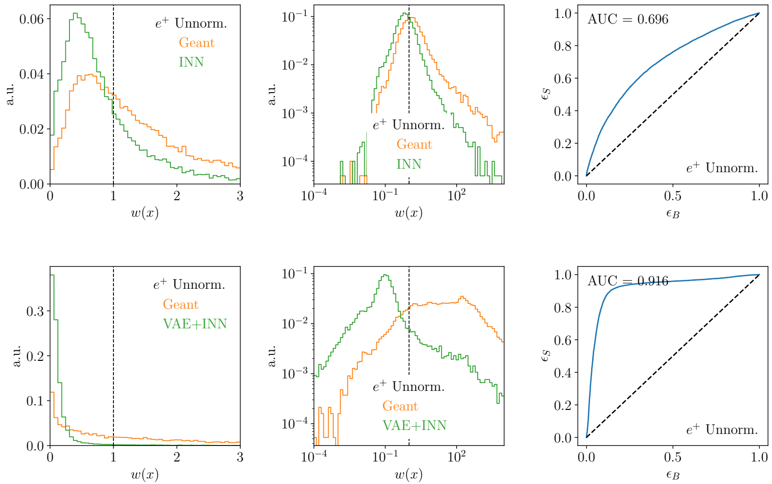


Classifier — Dataset 2

2305.16774

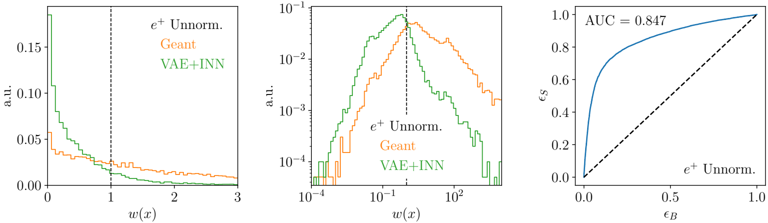
INN

VAE+INN

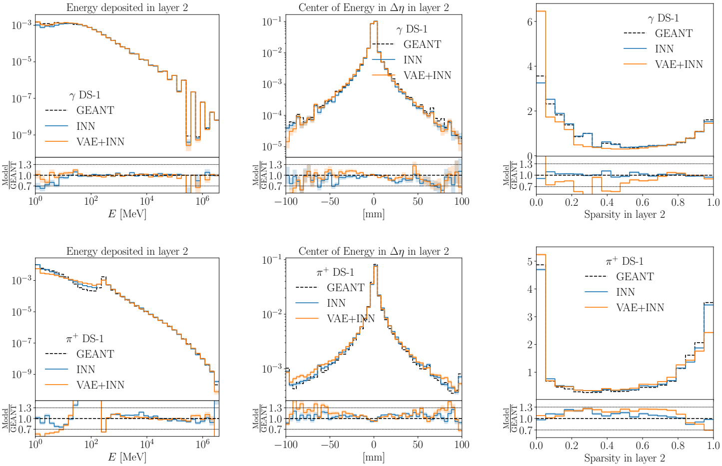


Classifier — Dataset 3 2305.16774

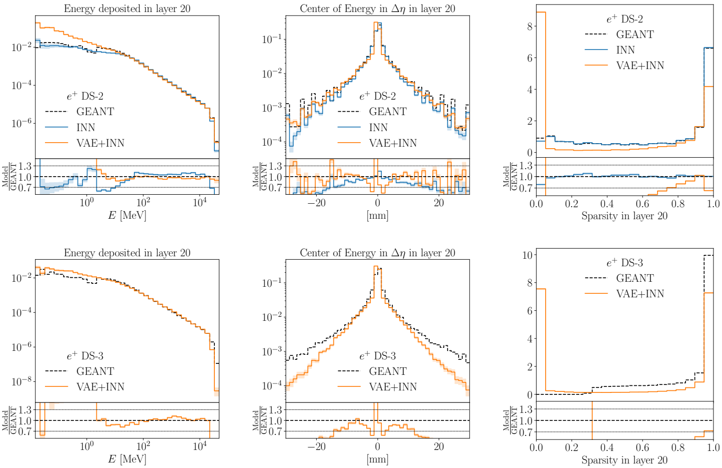
VAE+INN



High level features — Dataset 1



High level features — Dataset 2 & 3



Summary

- Coupling blocks are a viable and faster possibility for normalizing flows
- VAE compression is possible but results in worse samples
- VAEs perform best for hadronic showers, while pure INNs perform best for electromagnetic showers.

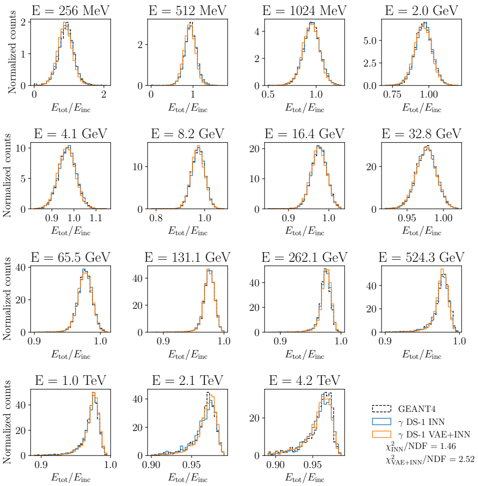
The End

Backup Slides

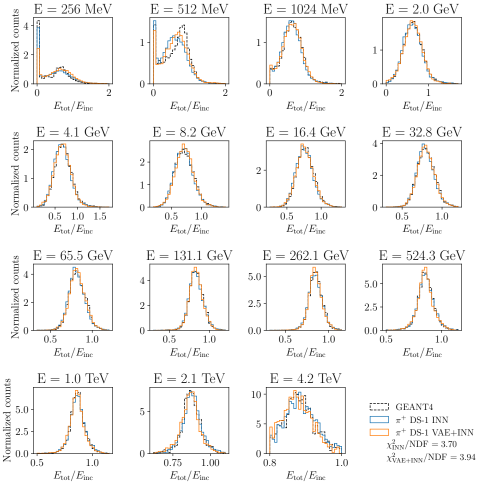
Parameter	INN ds1/ds2	INN (After VAE)
coupling blocks	RQS / Cubic	RQS
# layers	4 / 3	3
hidden dimension	256	32
# of bins	10	10
# of blocks	12/14	18
# of epochs	450 / 200	200
batch size	512 / 256	256
lr scheduler	"one cycle"	"one cycle"
max. lr	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
$\beta_{1,2}$ (ADAM)	$(0.9, 0.999)$	$(0.9, 0.999)$
b	$5 \cdot 10^{-6}$	/
α	$1 \cdot 10^{-8}$	$1 \cdot 10^{-6}$

Parameter	VAE	
lr scheduler	Constant LR	Inner VAE
lr	$1 \cdot 10^{-4}$	
hidden dimension	5000, 1000, 500 (Set 1)	
	1500, 1000, 500 (Set 2)	
	2000, 1000, 500 (Set 3)	
latent dimension	50 (Set 1,2) / 300 (Set 3)	
# of epochs	1000	
batch size	256	
β	$1 \cdot 10^{-9}$	
threshold t [keV]	2 (Set 1) / 15.15 (Set 2,3)	
hidden dimension	1500, 800, 300	Kernel
kernel size	7	
kernel stride	3 (Set 2), 5 (Set 3)	

$\frac{E_{tot}}{E_{inc}}$-photons



$\frac{E_{tot}}{E_{inc}}$-pions



INN derivation

$$
\begin{aligned}
& dx \; p_{\text{model}}\left(x \mid \theta\right) = dz \; p_{\text{latent}}\left(z\right) \\
\Leftrightarrow \quad & p_{\text{model}}\left(x \mid \theta\right) = p_{\text{latent}}\left(z\right) \left|\frac{\partial G_\theta(z)}{\partial z}\right|^{-1} = p_{\text{latent}}\left(\bar{G}_\theta(x)\right) \left|\frac{\partial \bar{G}_\theta(x)}{\partial x}\right| .
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \quad \mathcal{L}_{\text{INN}} &= - \left\langle \log p_{\text{model}}\left(x|\theta\right) \right\rangle_{p_{\text{data}} \sim TS} \\
&= - \left\langle \log p_{\text{latent}}\left(\bar{G}_\theta(x)\right) + \log \left|\frac{\partial \bar{G}_\theta(x)}{\partial x}\right| \right\rangle_{p_{\text{data}} \sim TS} .
\end{aligned}
$$