

# VAEs with Normalizing Flows for Robust Anomaly Detection

---

- Pratik Jawahar
- MS. in Robotics Engineering & AI at Worcester Polytechnic Institute (WPI, USA)
- Research Focus: Deep Learning, Fault Detection, Computer Vision, Audio Analytics
- DIANA-HEP Fellowship 2020
- Previously worked as a short-term intern with EP-CMX group (GEM Detectors) in 2018

## Baseline:

- Train one-class (SM events only) VAE with custom loss as in:  
<https://arxiv.org/pdf/1811.10276.pdf>
- Set a threshold on loss value and classify any events with a total loss above the threshold as Anomalies (Potential BSM events)

## Potential Downsides:

- VAE approximates latent space to a Gaussian distribution which might lead to inaccurate reconstruction (high negative log-likelihood)

## Methodology:

- Implement Normalizing Flows to change latent space distributions from Gaussian to another target distribution, which will help in more accurate reconstruction (<https://arxiv.org/abs/1505.05770>)
  - NICE
  - RealNVP
- Explore the use of other flows, with the same idea, such as:
  - Autoregressive Flows
  - Sylvester Normalizing Flows (SNF)
  - Continuous Normalizing Flows

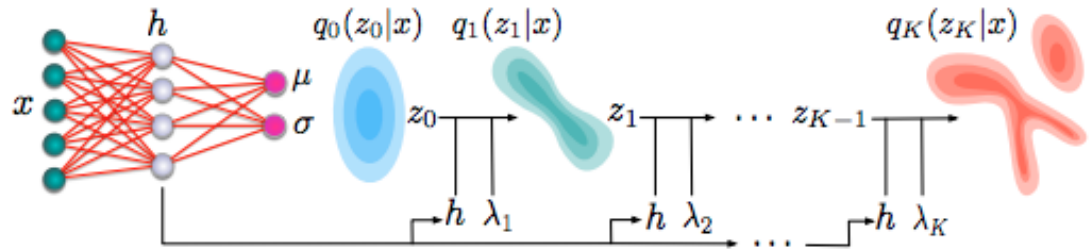
# Normalizing Flows:

- Invertible transformations that convert a given distribution ( $\mathbf{z}$ ) to a target distribution ( $\mathbf{z}'$ )
- State of the art algorithms in Variational Inference currently implement different types of flows

$$p_1(\mathbf{z}') = p_0(\mathbf{z}) \left| \det \left( \frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \right) \right|^{-1}$$

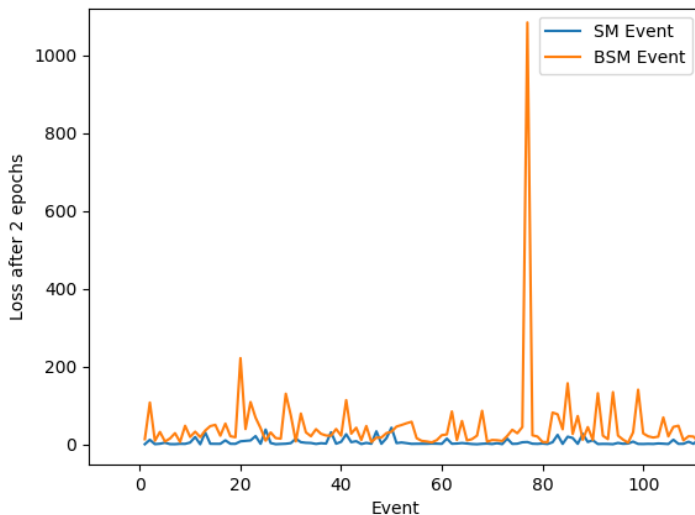
$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u}h(\mathbf{w}^\top \mathbf{z} + b),$$

$$\mathbf{z}_K = f_K \circ f_{K-1} \circ \dots \circ f_1(\mathbf{z})$$



# Current Status:

- Baseline VAE (ConvVAE) model using pyTorch (<https://github.com/mpp-hep/DarkFlow>)
- Inputs are not images, but an ordered 2D representation (columns ordered in decreasing pT) of the selection proposed last week, with suitable padding



## Current Status:

1. Flow implementation functions on pyTorch for
  - a. IAF, MAF
  - b. SNF
  - c. NICE, RealNVP
  - d. Neural Spline Flows

## Main Objectives:

1. Determine the best architecture for baseline VAE (ConvVAE/GraphNN)
2. Determine the flow algorithm that gives best performance
3. Determine best training process to incorporate flows into VAE