



Lecture 22: AI Monte Carlo

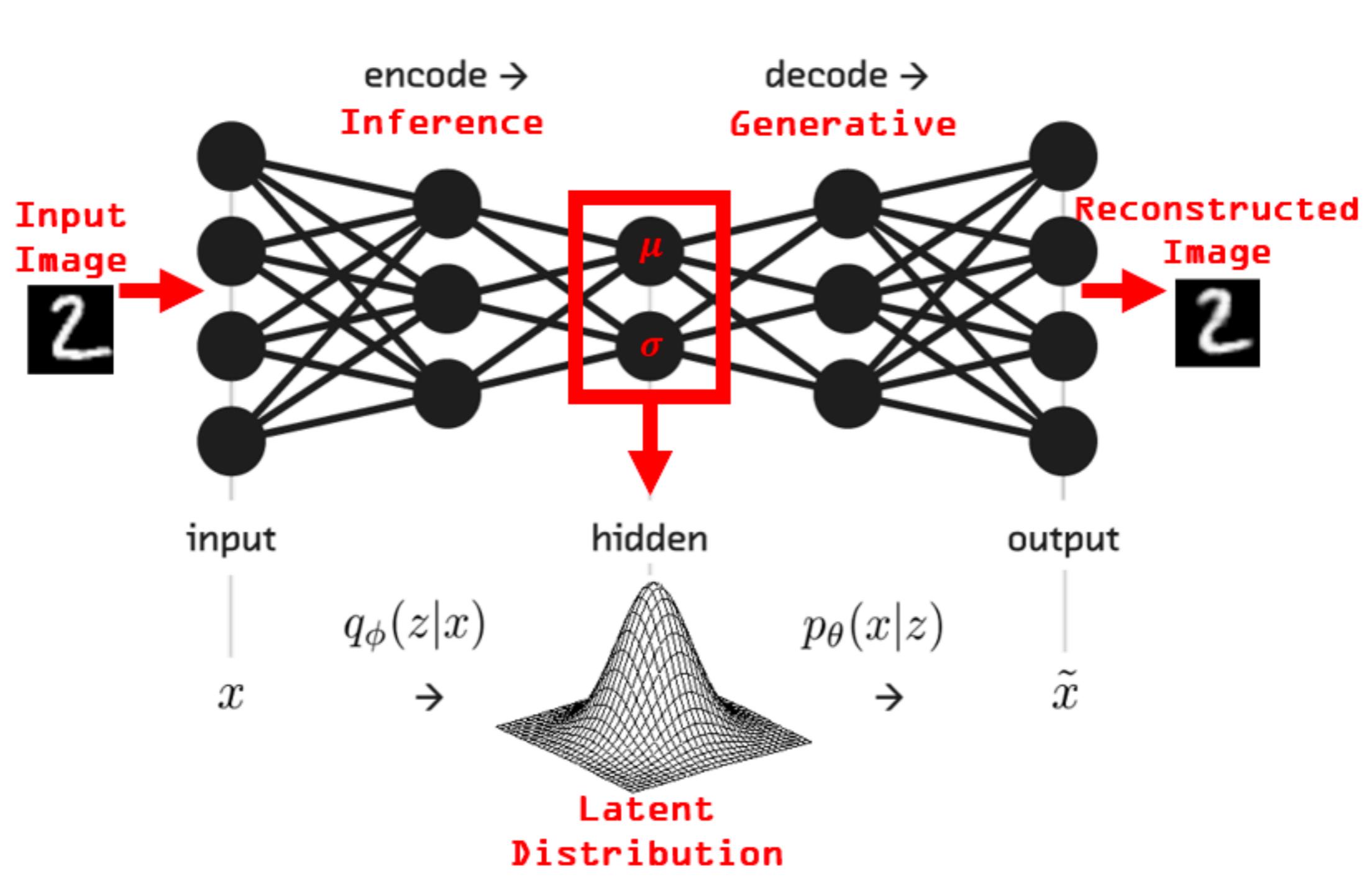
Proton Therapy



Proton Therapy Center at MGH

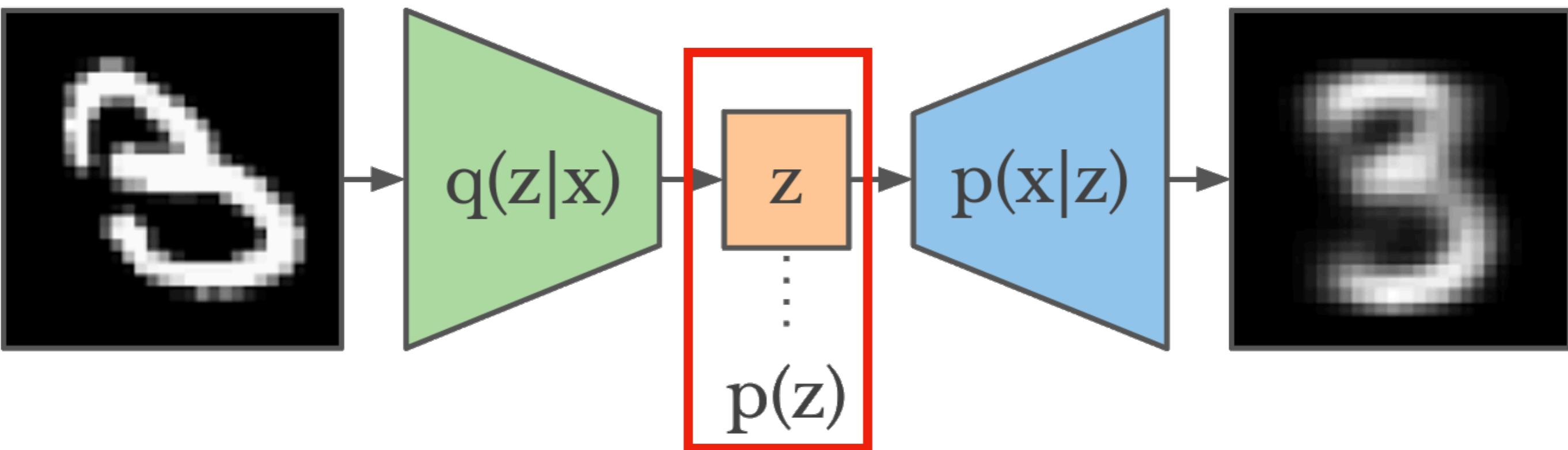
VAE

- Variational Autoencoder is a great way to model objects

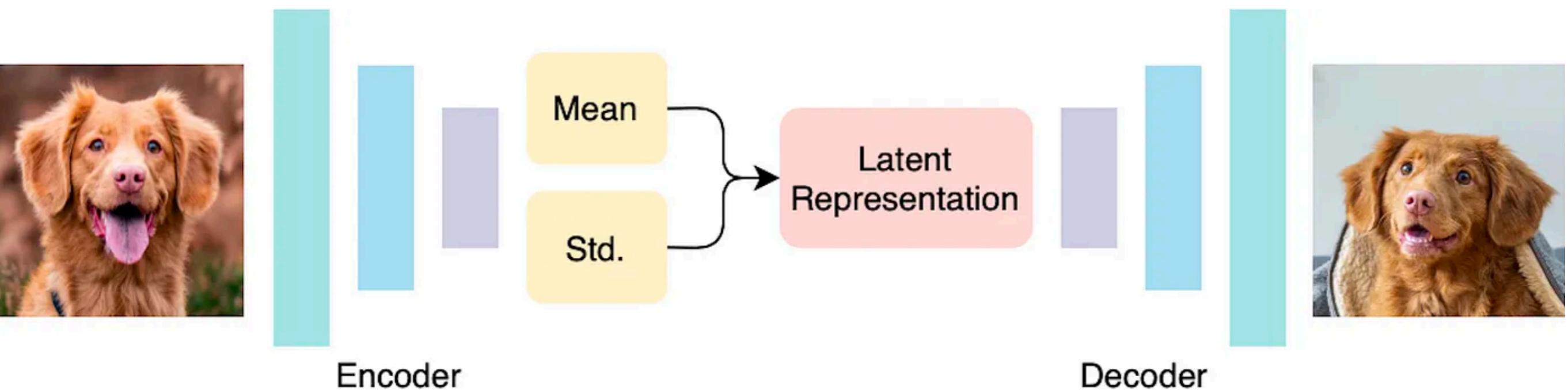


VAE

- Variational Autoencoder is a great way to model objects

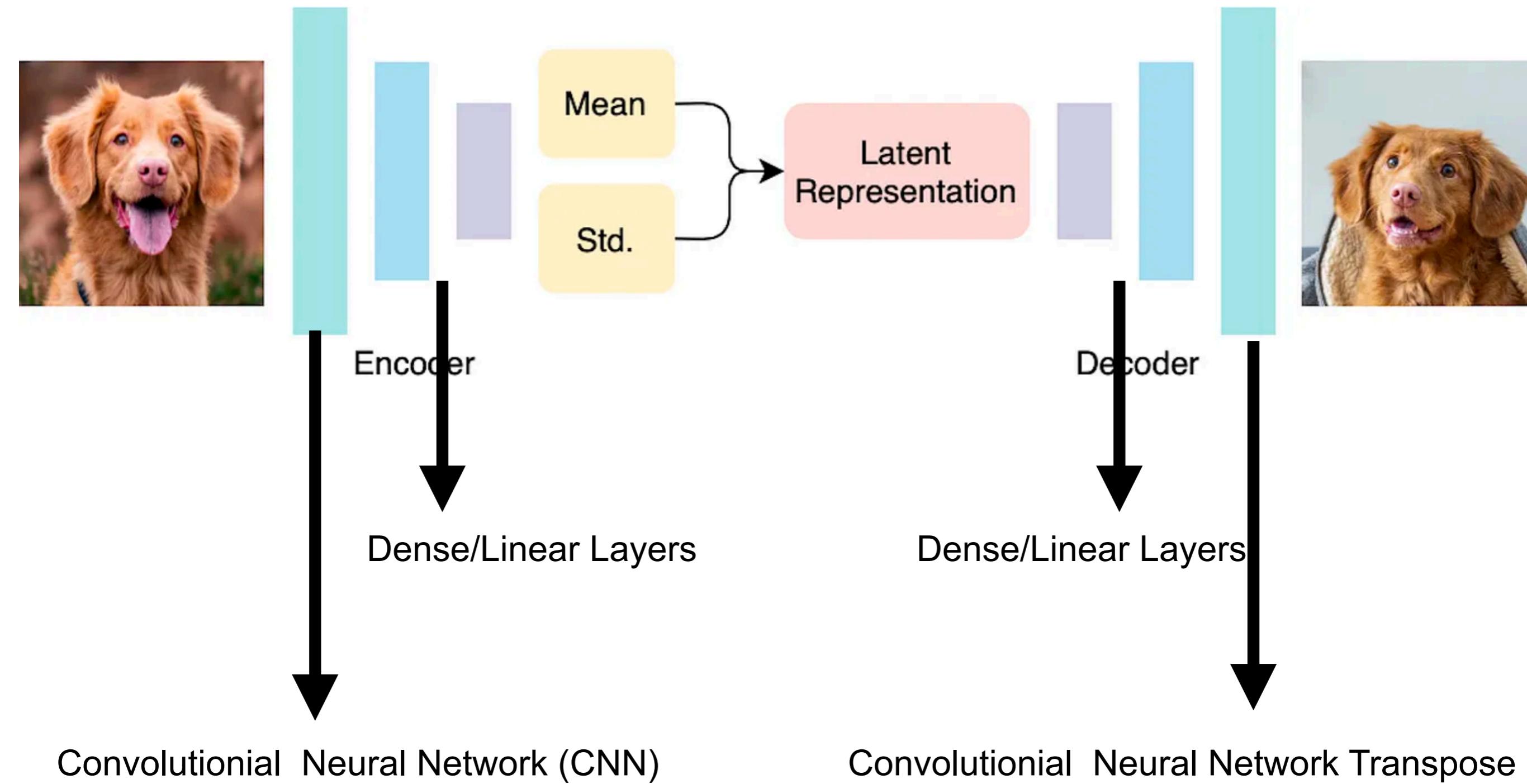


Randomly sample a normal distribution in this space

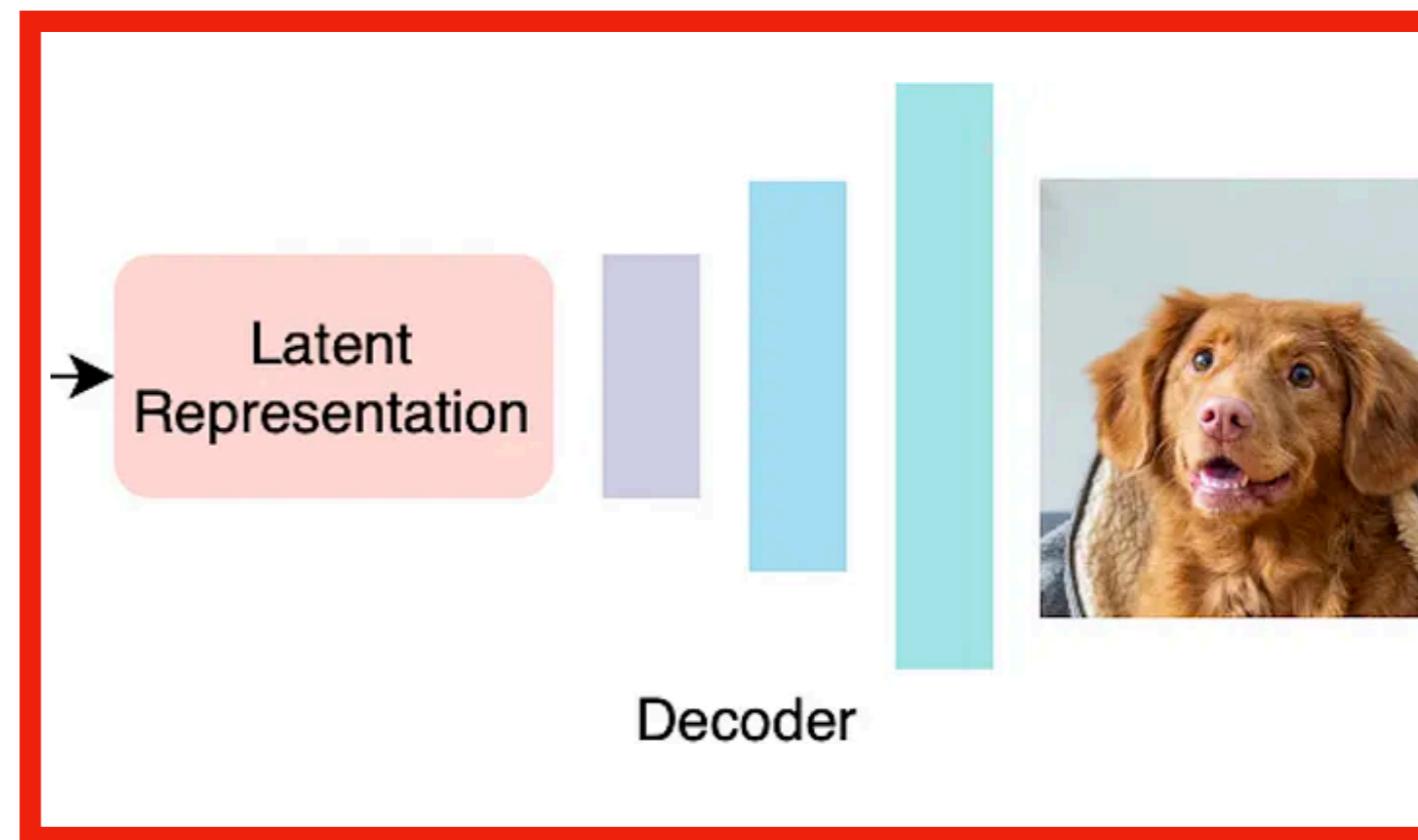
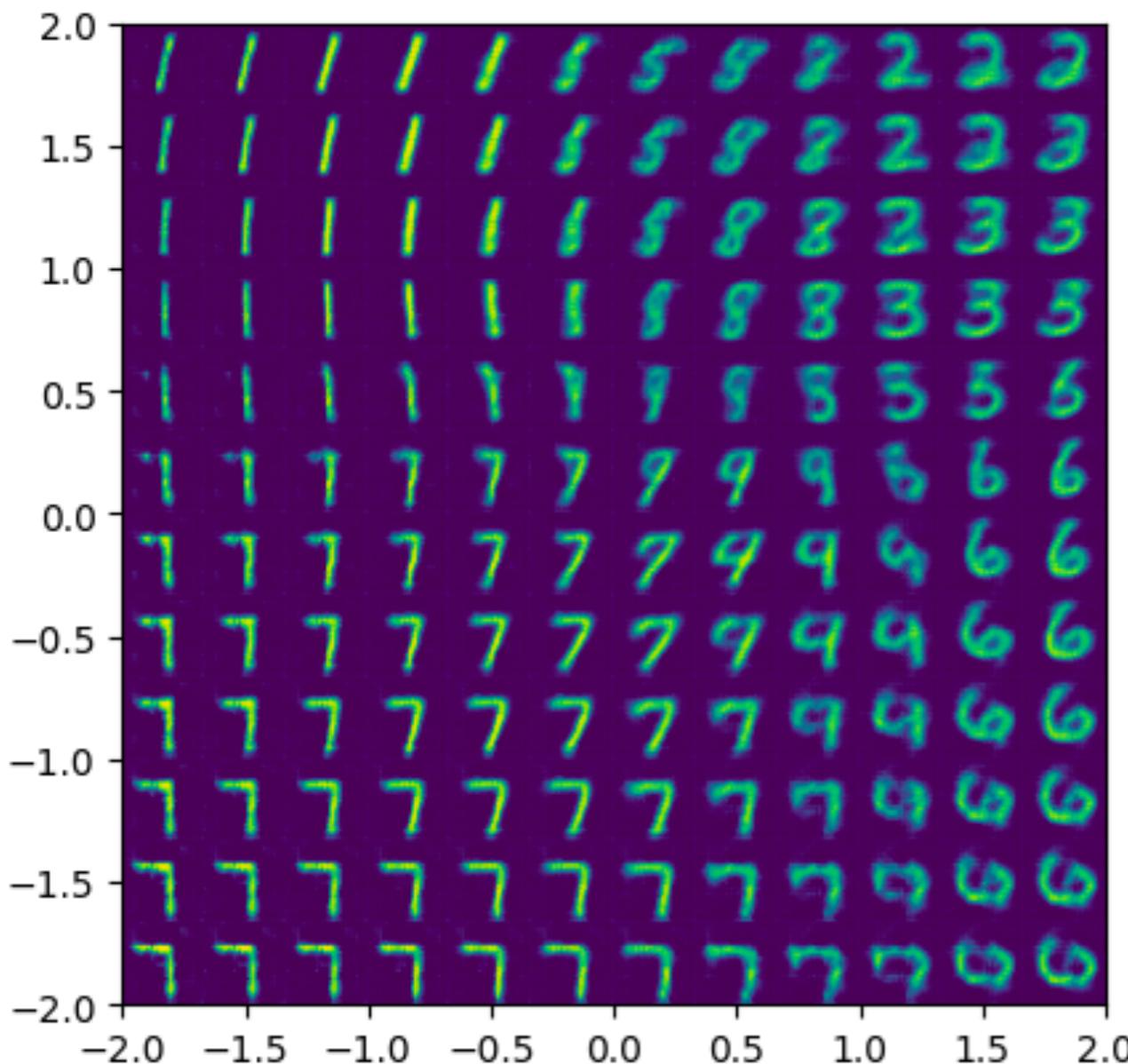


MNIST VAE encoder

- We will use a CNN to encode the data and process it

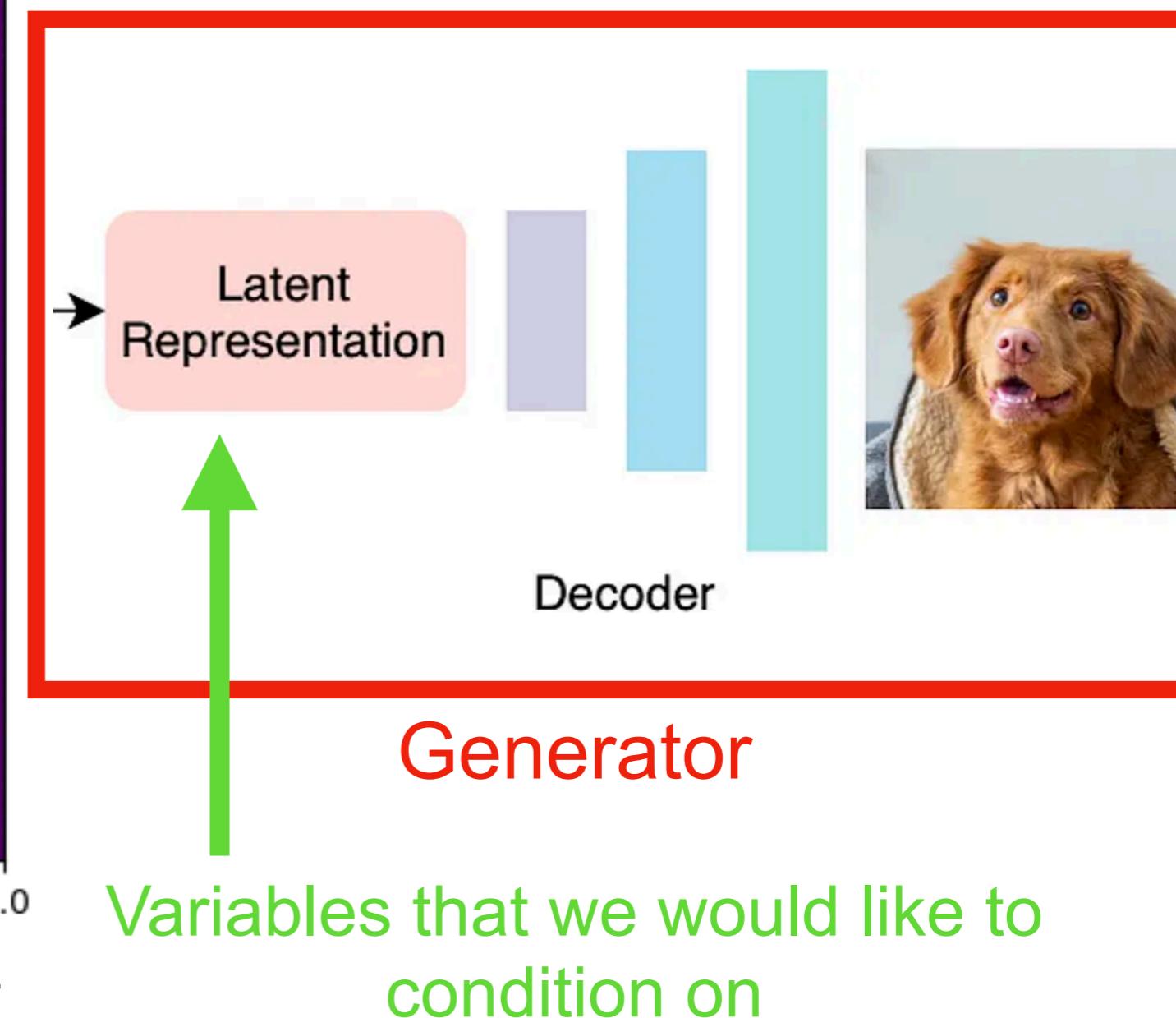
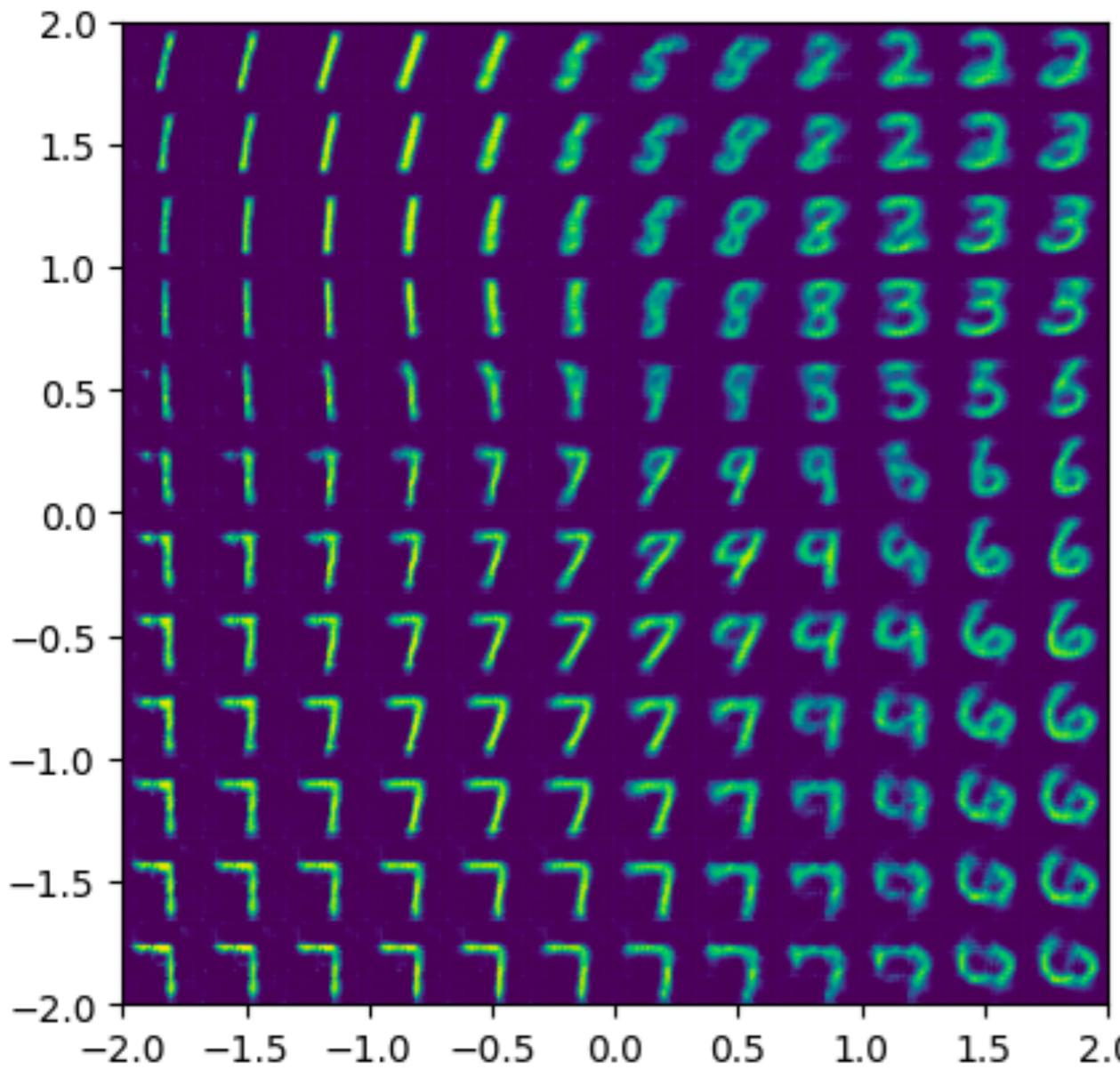


Exploring the latent space?



- We can sample the latent space as a generator

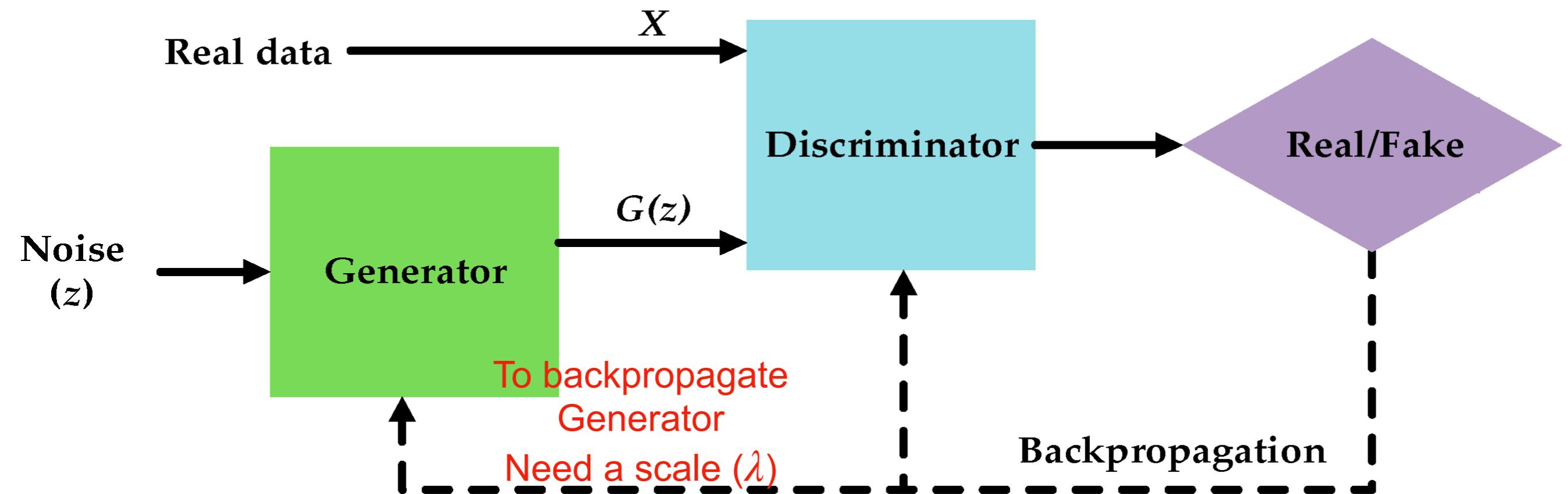
Conditional VAE



- Force known inputs into the VAE
 - That way our latent space has explicit knowledge of what is going on

GAN

- The other AI way of generating events uses
 - Generative Adversarial Networks (GANs)
 - I personally do not like these networks
 - They need too much tuning, **don't recommend them**

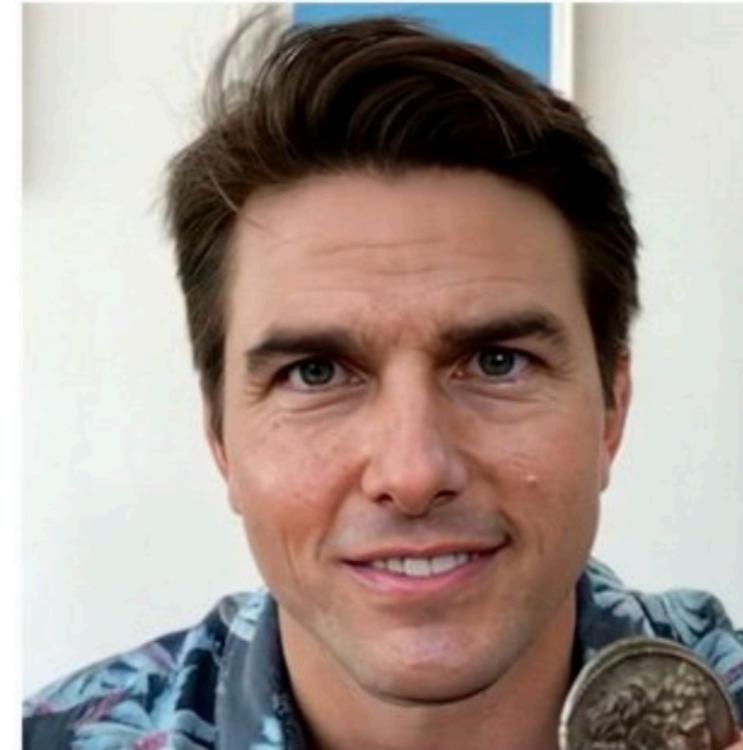


Power of GANs

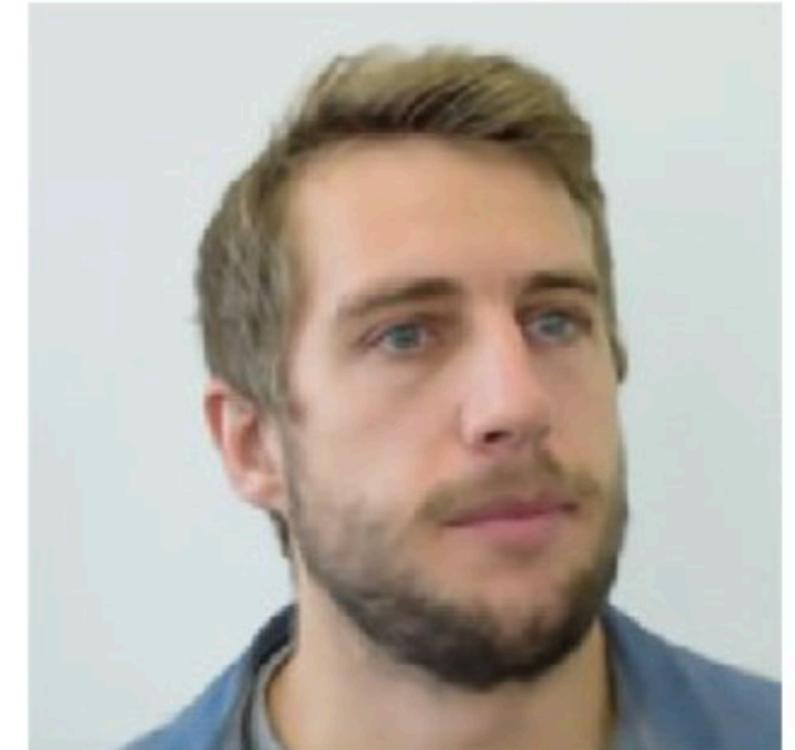
- GANs became popular for generating fake human faces
 - Not as a hard a problem as typical physics problems
 - Our proton problem is probably already too hard for it



GAN



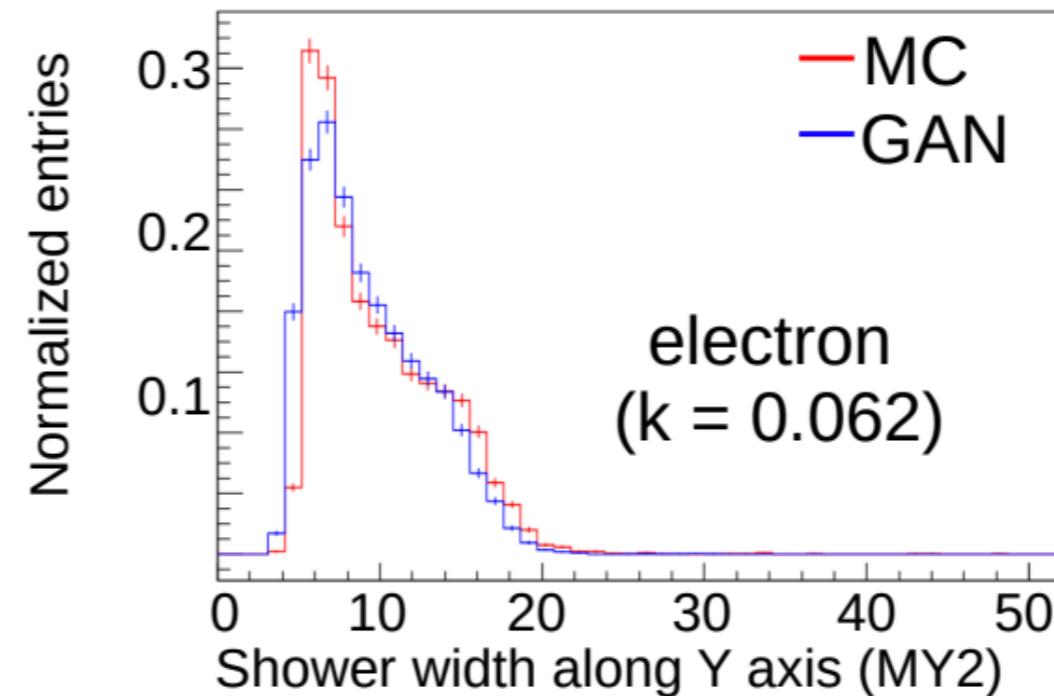
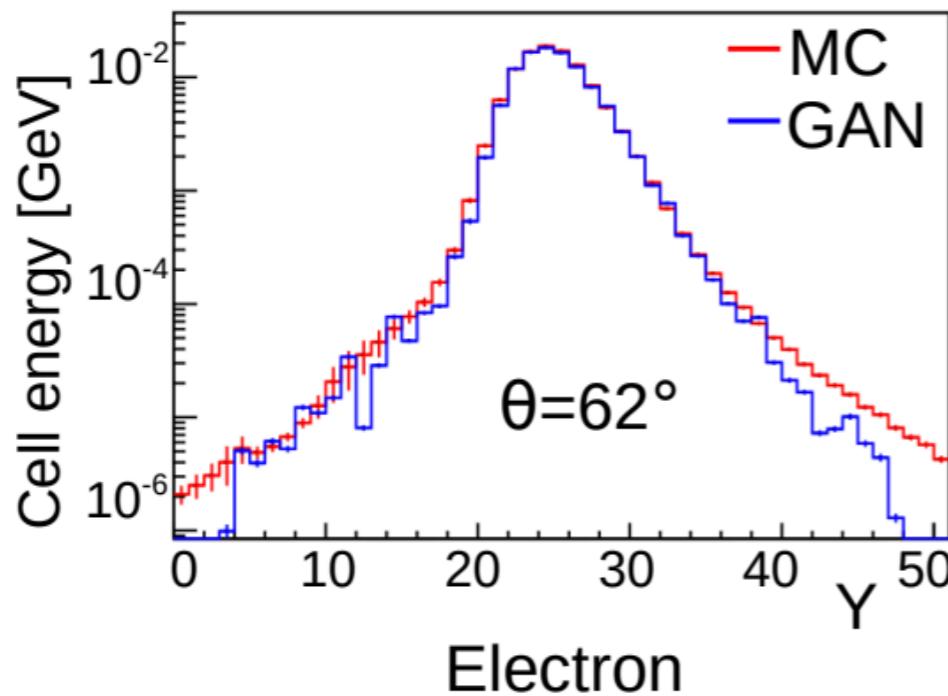
Autoencoder



NeRF

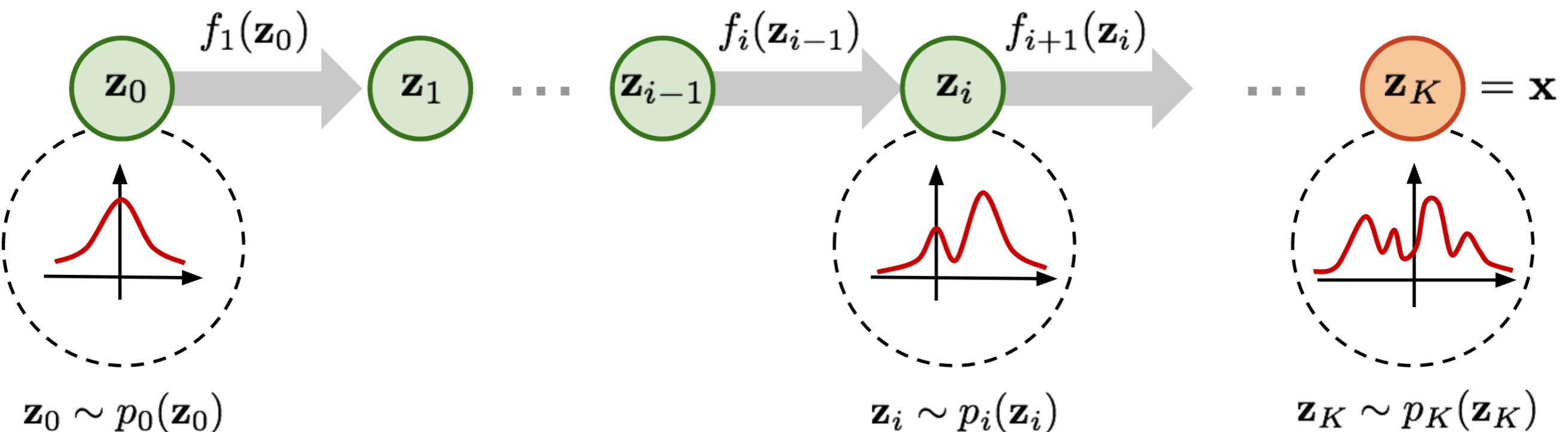
Weakness of GANs

- GANs have not been great for physics
 - While they get the bulk of simulation ok
 - Struggle with the many orders of magnitude needed
 - **Also they are pain to tune**



Normalizing Flows

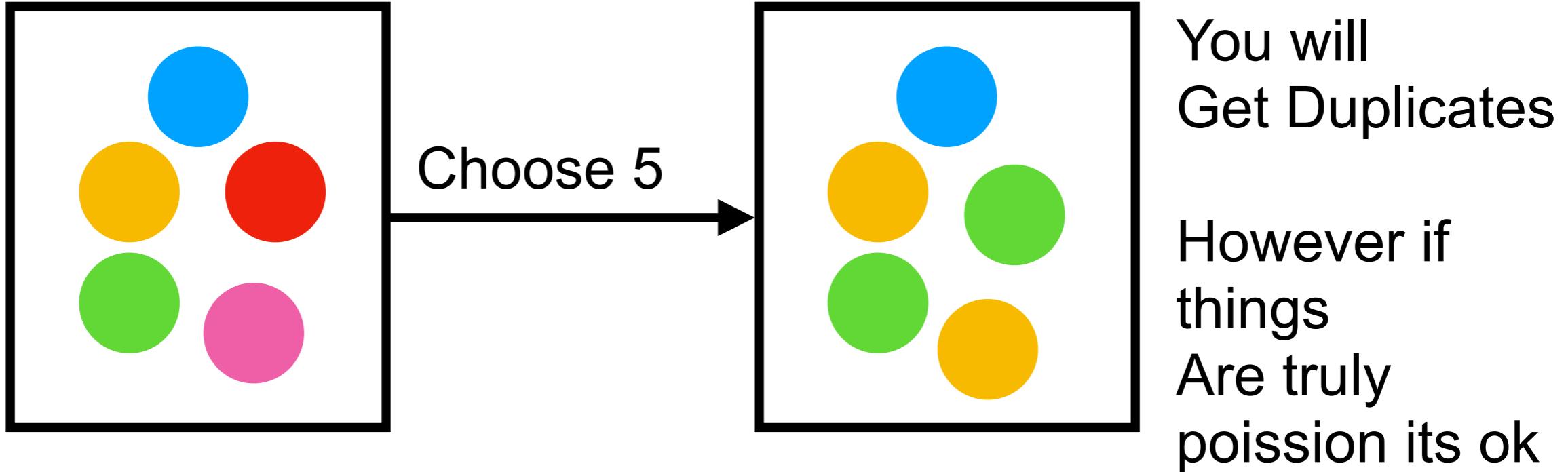
- While I don't show it here
 - The normalizing flow is next evolution of VAE
 - Allows for a more expressive latent space
 - Normalizing flow transforms flows to other spaces



Bootstrapping

- Lets say you have a fixed dataset (\vec{x})
 - You cannot generate more
 - You compute something very complicated on it
 - For example $\vec{y} = \text{NN}(\vec{x})$
 - You want to know the uncertainty on \vec{y} given by $\sigma_{\vec{y}}$
 - However your function $\frac{d\text{NN}(\vec{x})}{dx}$ \notin exists

Bootstrapping: Strategy

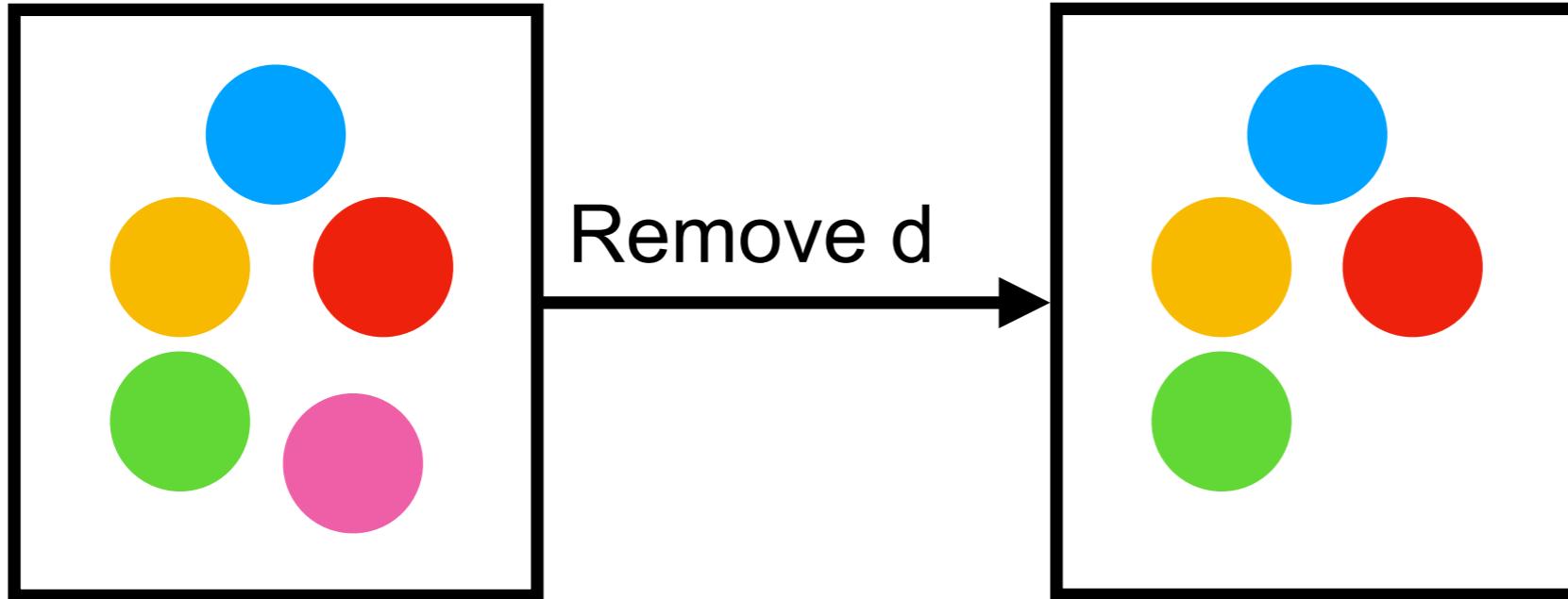


- Randomly sample the pool of events with the same size
 - Now use this sample to compute all our observables
- Variance over our random samples will be the uncertainty

$$\sigma_{\text{boot}} = \frac{1}{N_{\text{samples}} - 1} \sum_i^{N_{\text{samples}}} (\mathcal{O}_i - \bar{\mathcal{O}})^2$$

We can do this
with any
observable

Delete-D Jackknife



- Like what we did with Bootstrapping
 - However, doesn't duplicate events

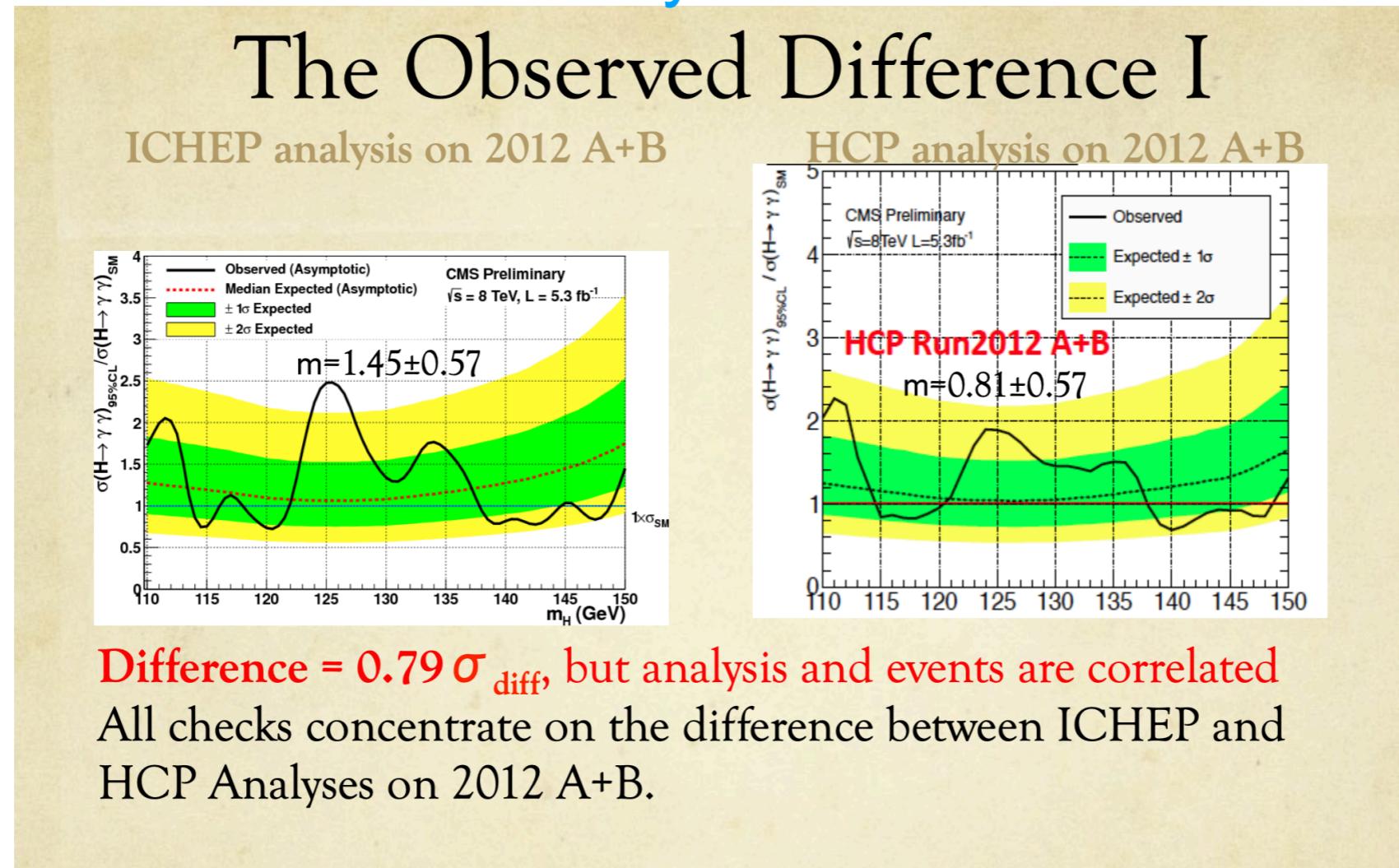
d=removed samples

$$\sigma_{\text{Jackknife}} = \frac{N_s - d}{(N_s C_d) d} \sum_i^{N_s C_d} (\mathcal{O} - \bar{\mathcal{O}})^2$$

Total Unc
From Jackknife

Fun Story

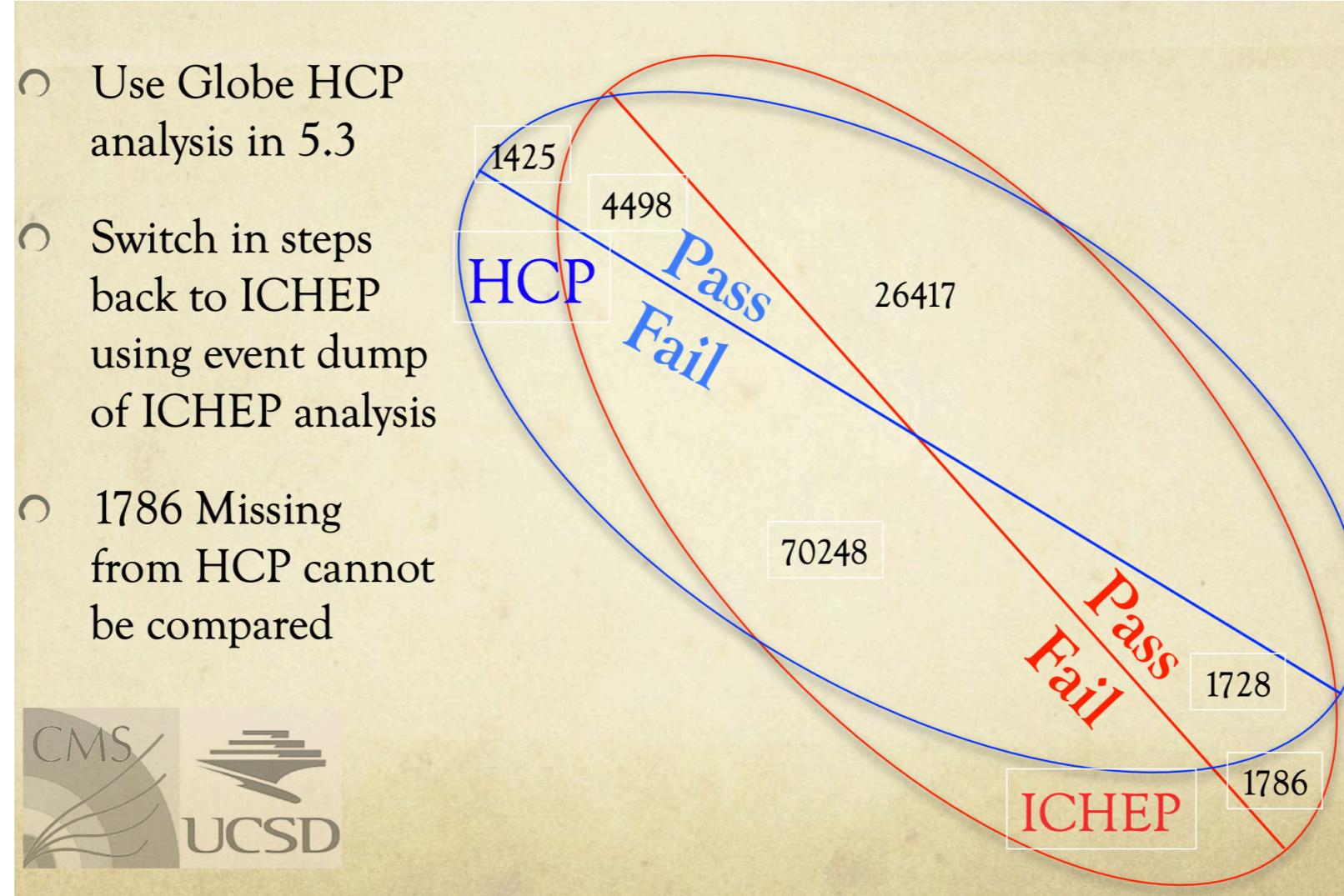
Higgs to diphoton result changed dramatically in 2012
On exactly the same data



Fun Story

Higgs to diphoton result changed dramatically in 2012
On exactly the same data

- Use Globe HCP analysis in 5.3
- Switch in steps back to ICHEP using event dump of ICHEP analysis
- 1786 Missing from HCP cannot be compared



Delete-d Jackknife was the way we were able to analyze the unc.