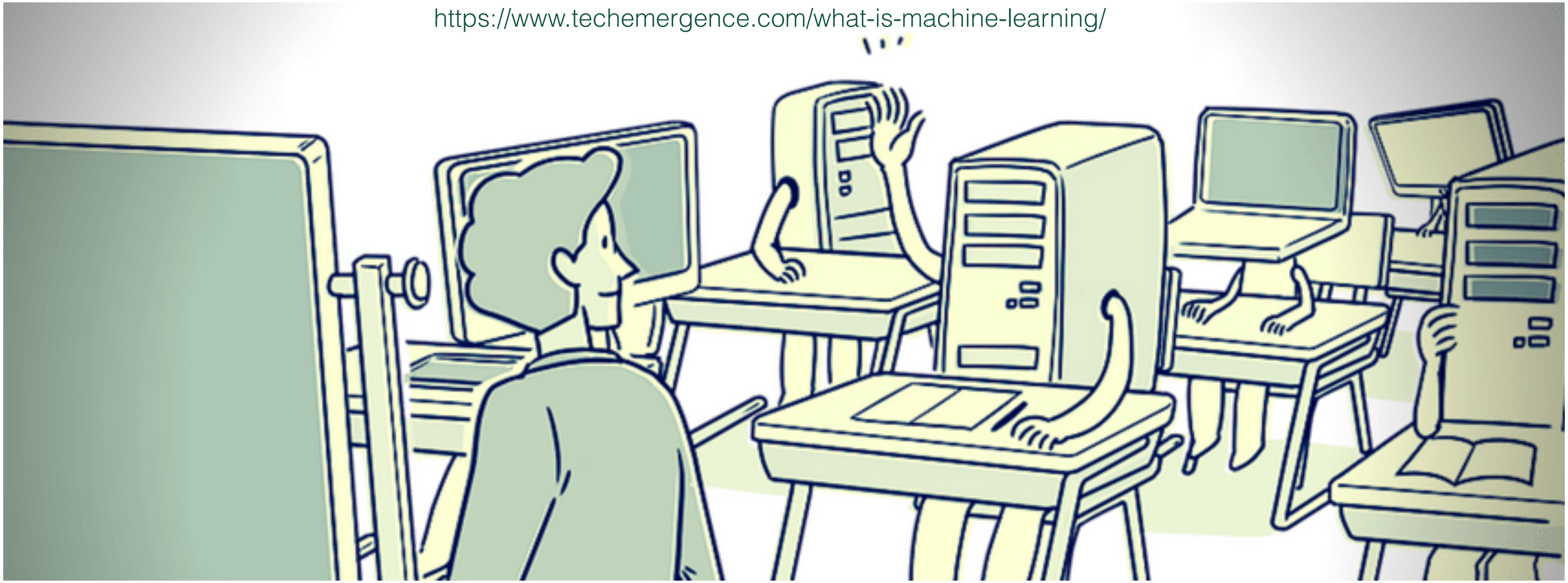
Introduction to Machine Learning

Lecture 3

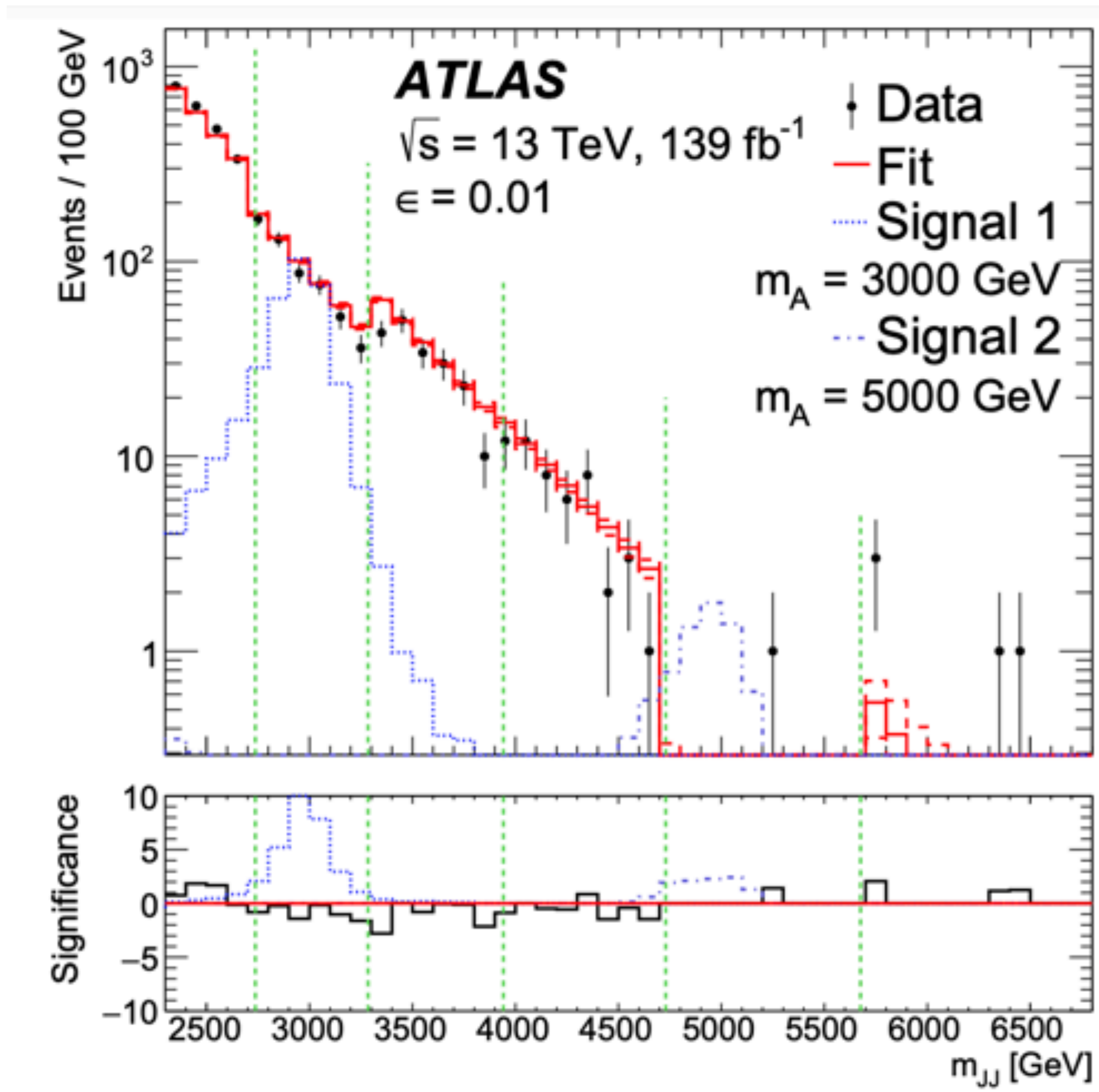


QUC Winter School 2021
KIAS
December 2021

Bryan Ostdiek

Questions from last lecture?

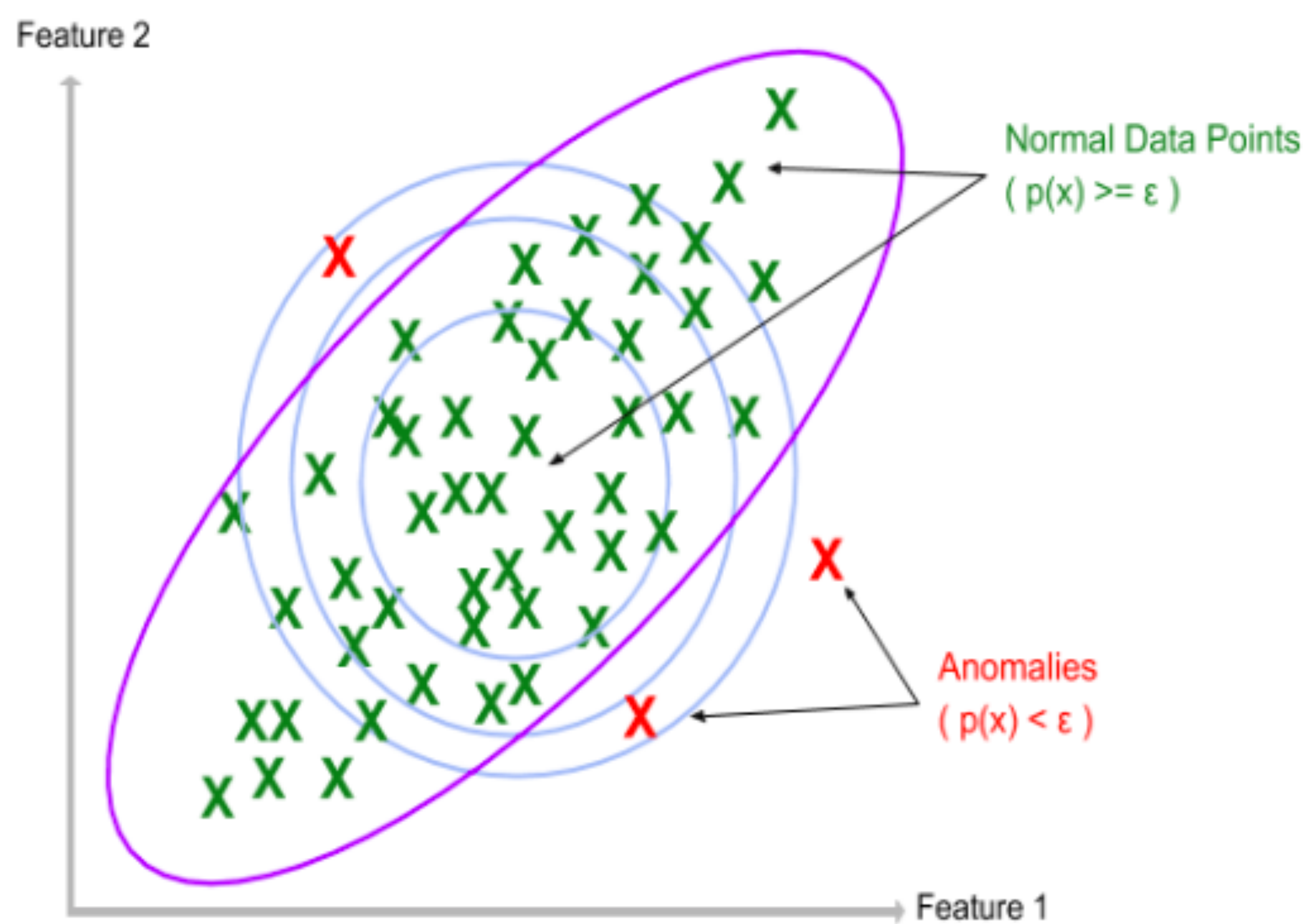
ATLAS search using weak supervision
[2005.02983]



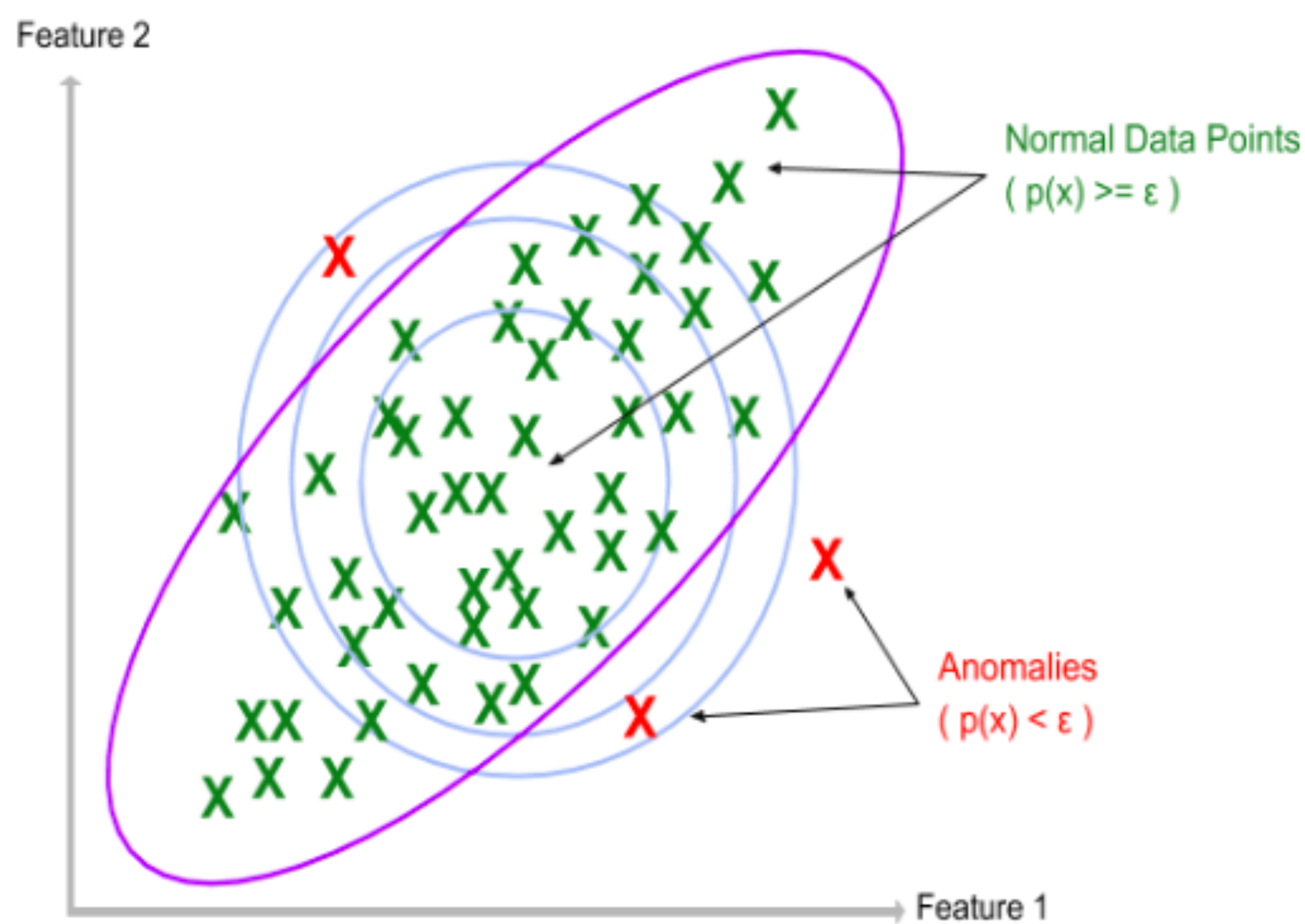
Start of lecture 3

- Anomaly Detection
- Generative Networks
- Flows for sampling/inference

What is Anomaly Detection?

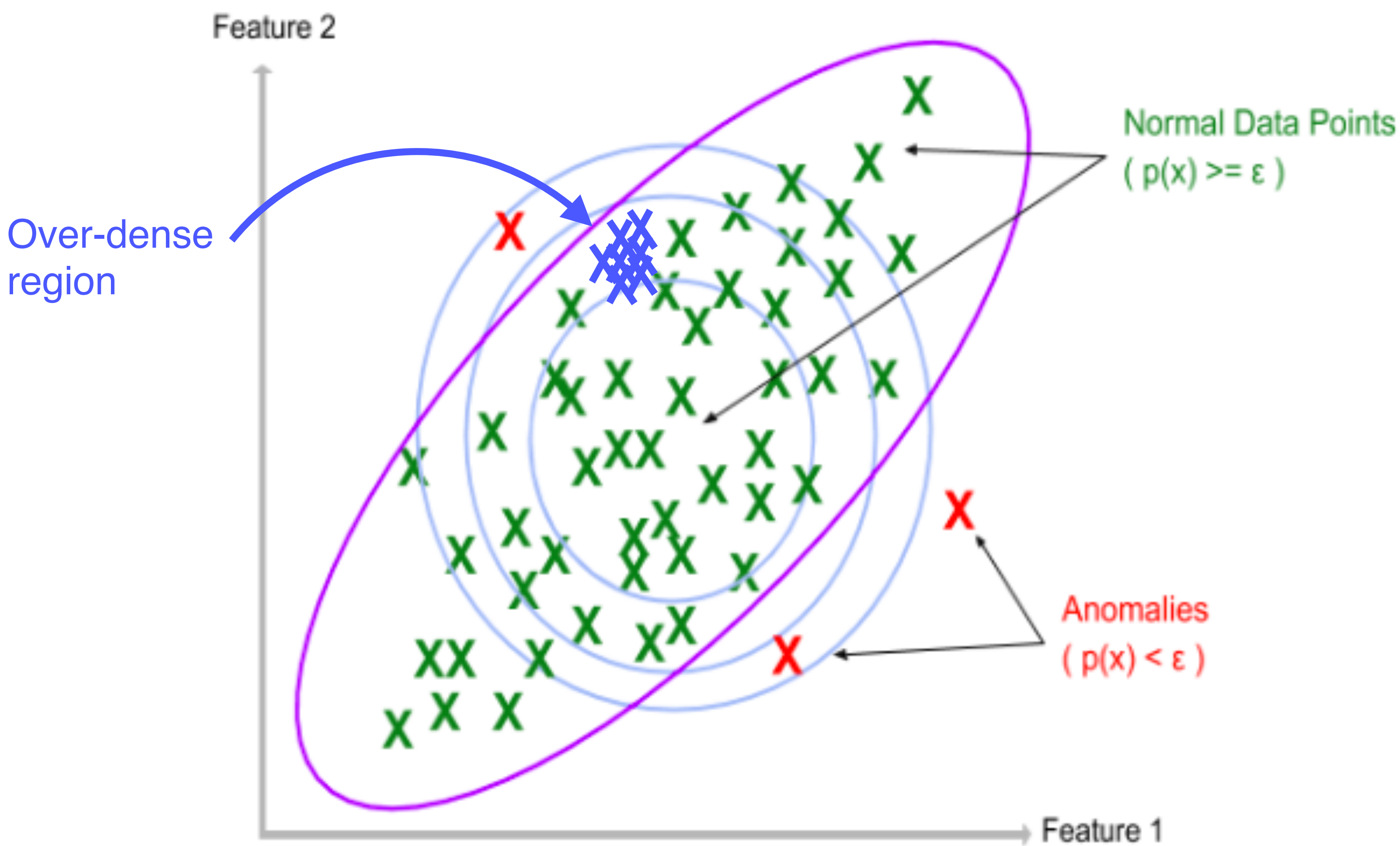


What is Anomaly Detection?



1) Events that occur with low probability (look different in some feature space)

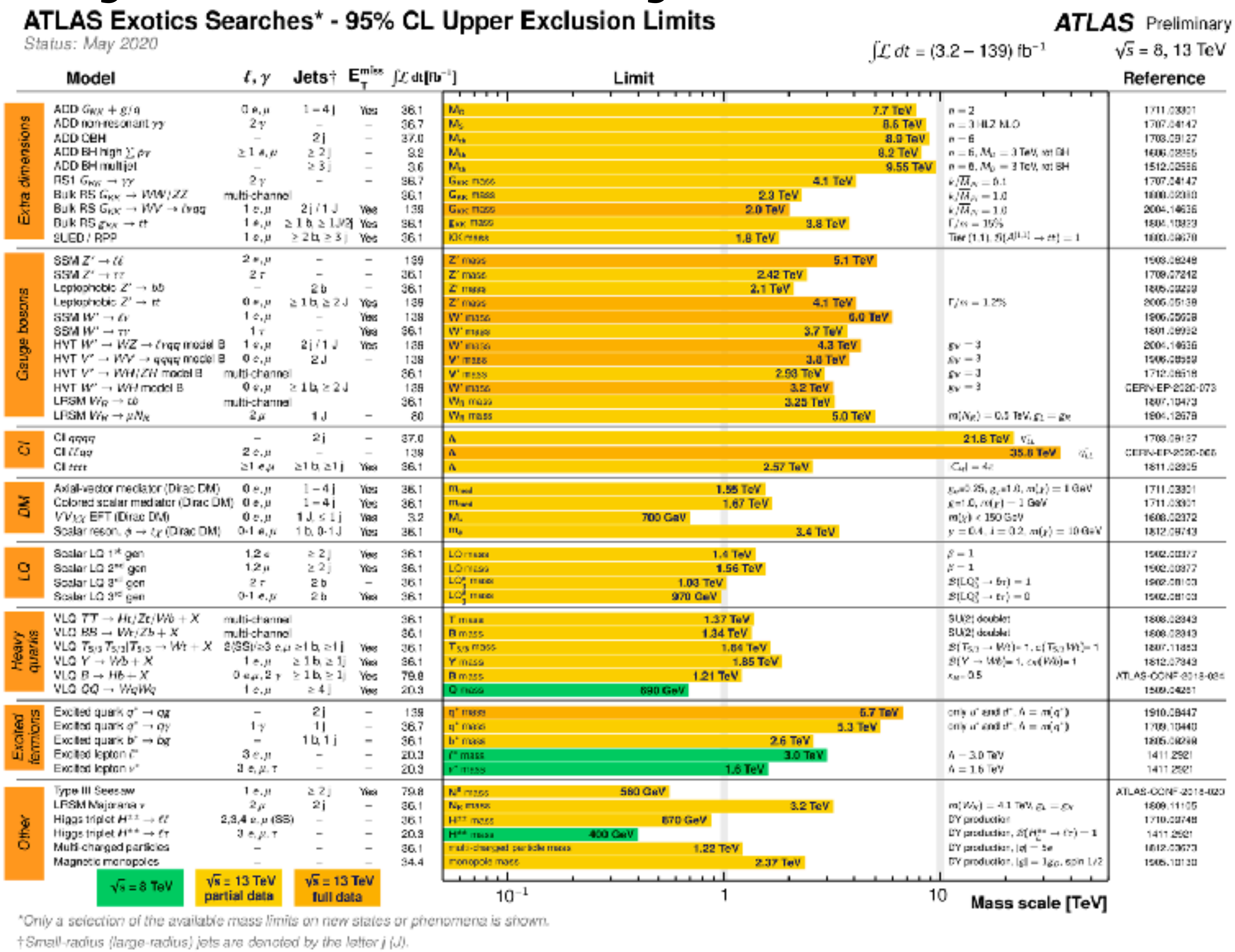
What is Anomaly Detection?



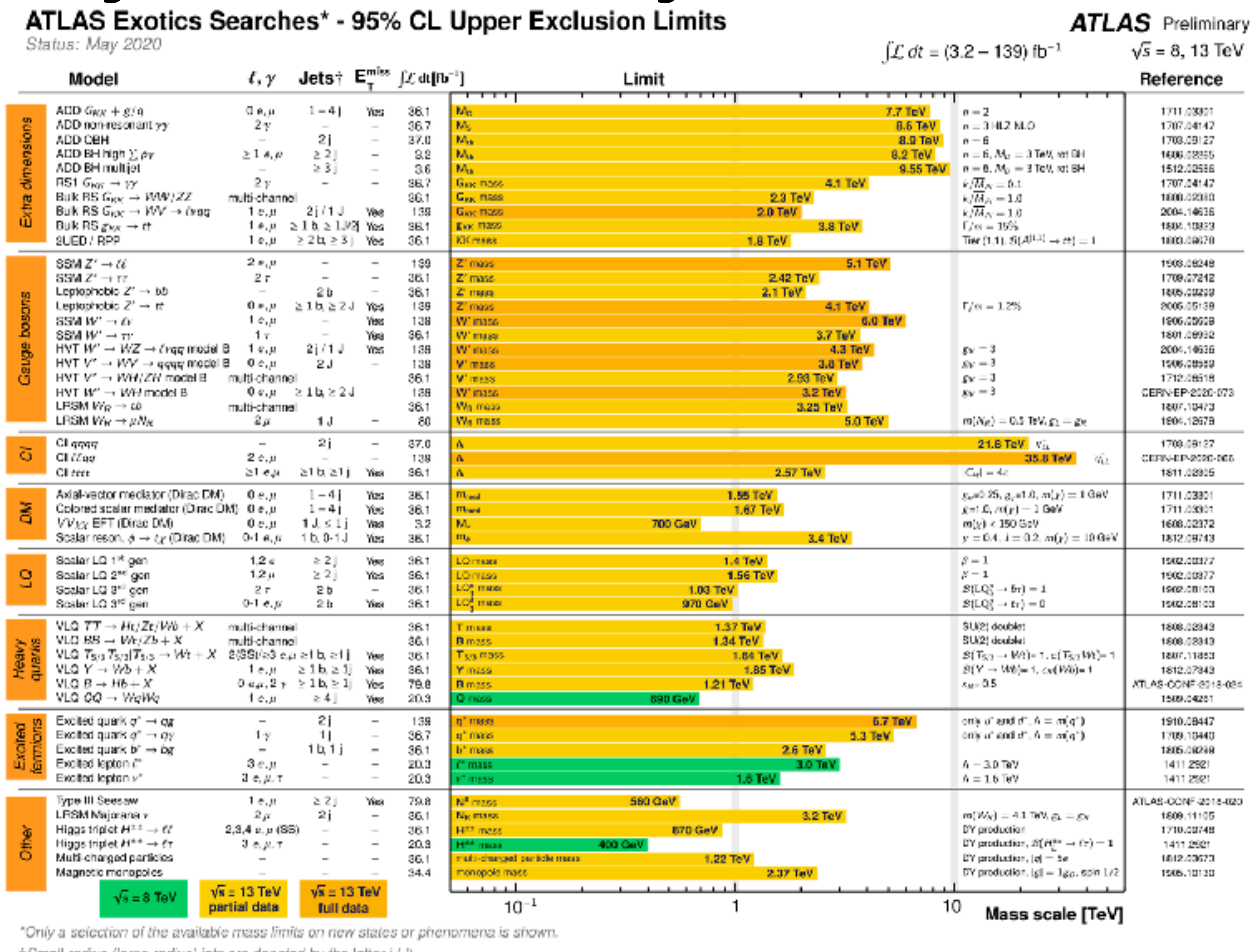
1) Events that occur with low probability (look different in some feature space)

2) Events that don't look very different, but occur at a higher rate than expected

Why Anomaly Detection?



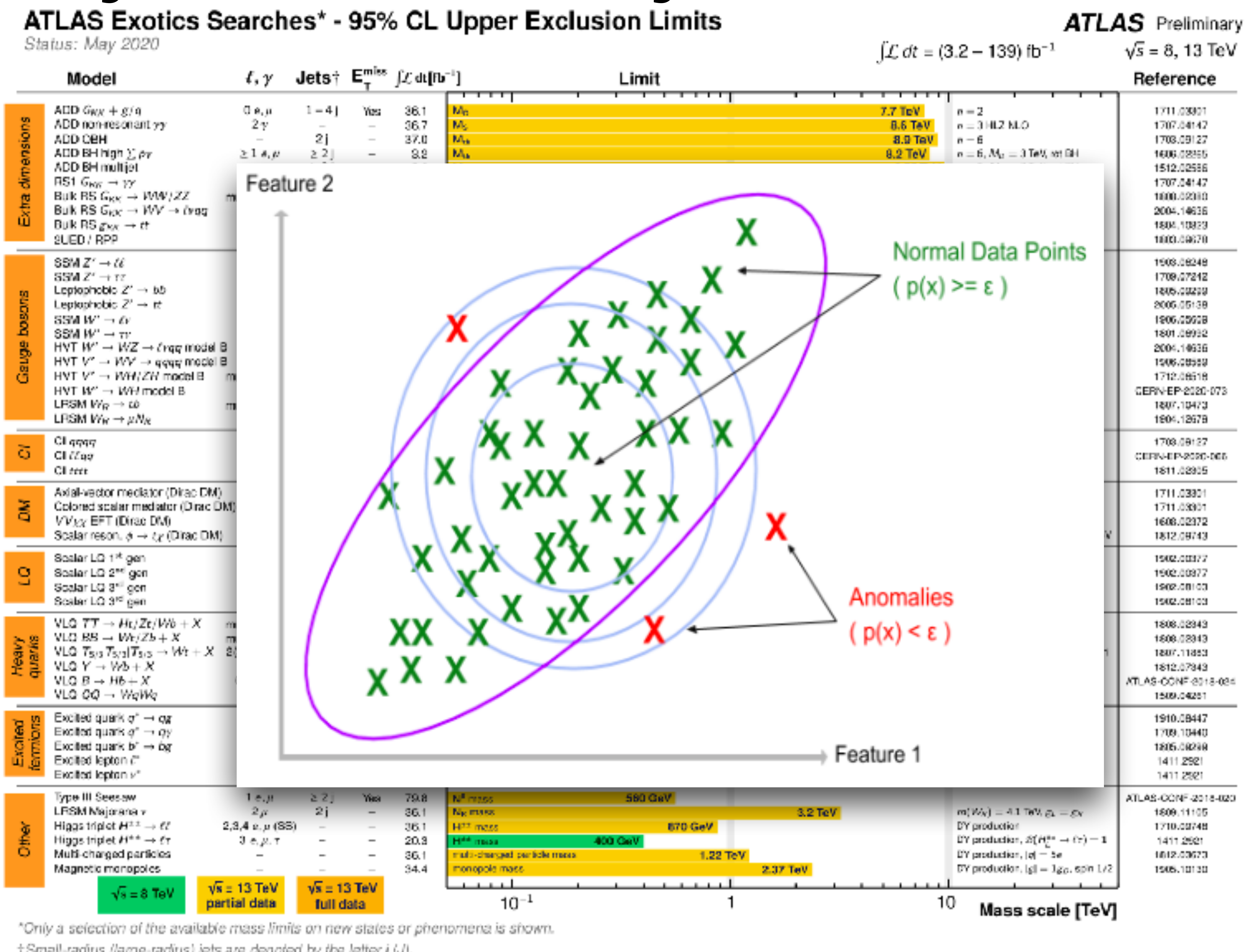
Why Anomaly Detection?



Higher rates, but hard to distinguish from SM

Look more different, but more low rates

Why Anomaly Detection?

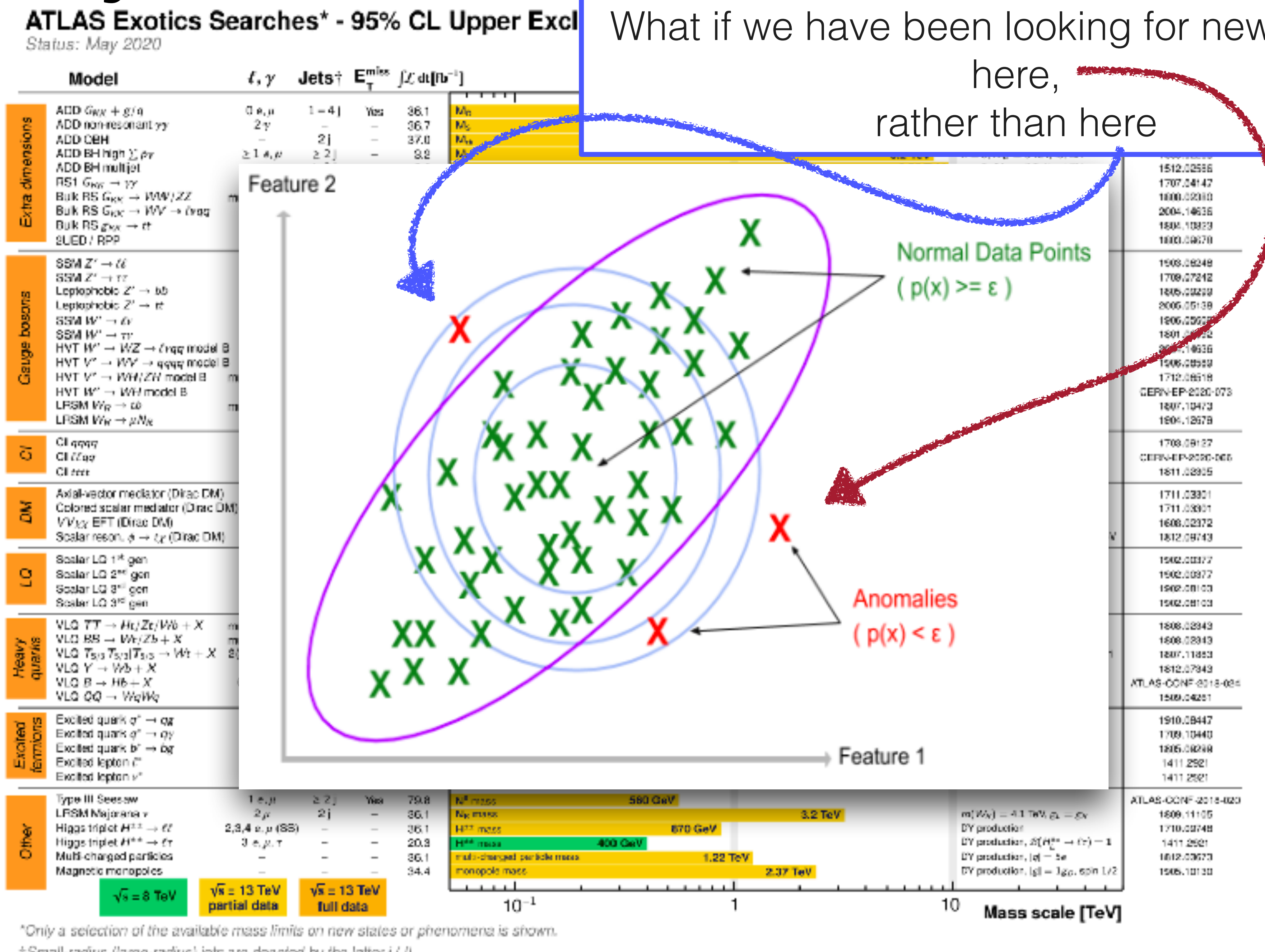


Higher rates, but hard to distinguish from SM

Look more different, but more low rates

Why Anomaly Detection?

What if we have been looking for new physics here,
rather than here



Higher rates, but hard
to distinguish from SM

Look more different,
but more low rates

Why Anomaly Detection?

1) Model Agnostic: We are good at looking for models we know of, but what if we don't know what we should be looking for?

2) Simulation Independent: With no signal model, it is possible to use methods directly on data from the LHC without Monte Carlo simulations.

How to detect anomalies?

Over 50 papers in HEP anomaly detection
https://iml-wg.github.io/HEPML-LivingReview/

LHC Olympics [2101.08320] focuses on finding over densities in all-hadronic events

- Black Box 1: Similar to example data: 4 methods found resonance
- Black Box 2: SM only. 4 methods claimed a resonance, 1 claimed lack-of-resonance
- Black Box 3: Correct resonance not detected by any group

Dark Machines Challenge [2105.14027] focuses on finding individual events which look different

- Train on SM-only events, apply to many different signals
- Find methods which work best for most signals
- No method finds every new physics signal

How to detect anomalies?

PHYSICAL REVIEW D 101, 075021 (2020)

Searching for new physics with deep autoencoders

Marco Farina,[1,2] Yuichiro Nakai,[2] and David Shih[2]
[1]*C.N.Yang Institute for Theoretical Physics, Stony Brook, New York 11794, USA*
[2]*NHETC, Dept. of Physics and Astronomy Rutgers, The State University of New Jersey, Piscataway, New Jersey 08854, USA*

(Received 10 January 2019; accepted 27 March 2020; published 13 April 2020)

We introduce a potentially powerful new method of searching for new physics at the LHC, using autoencoders and unsupervised deep learning. The key idea of the autoencoder is that it learns to map "normal" events back to themselves, but fails to reconstruct "anomalous" events that it has never encountered before. The reconstruction error can then be used as an anomaly threshold. We demonstrate the effectiveness of this idea using QCD jets as background and boosted top jets and R parity violating (RPV) gluino jets as signal. We show that a deep autoencoder can significantly improve signal over background when trained on backgrounds only, or even directly on data which contain a small admixture of signal. Finally, we examine the correlation of the autoencoders with jet mass and show how the jet mass distribution can be stable against cuts in reconstruction loss. This may be important for estimating QCD backgrounds from data. As a test case, we show how one could plausibly discover 400 GeV RPV gluinos using an autoencoder combined with a bump hunt in jet mass. This opens up the exciting possibility of training directly on actual data to discover new physics with no prior expectations or theory prejudice.

DOI: 10.1103/PhysRevD.101.075021

[1808.08992]

SciPost Phys. 6, 030 (2019)

QCD or what?

Theo Heimel[1], Gregor Kasieczka[2], Tilman Plehn[1*] and Jennifer M. Thompson[1]

1 Institut für Theoretische Physik, Universität Heidelberg, Germany
2 Institut für Experimentalphysik, Universität Hamburg, Germany

* plehn@uni-heidelberg.de

Abstract

Autoencoder networks, trained only on QCD jets, can be used to search for anomalies in jet-substructure. We show how, based either on images or on 4-vectors, they identify jets from decays of arbitrary heavy resonances. To control the backgrounds and the underlying systematics we can de-correlate the jet mass using an adversarial network. Such an adversarial autoencoder allows for a general and at the same time easily controllable search for new physics. Ideally, it can be trained and applied to data in the same phase space region, allowing us to efficiently search for new physics using un-supervised learning.

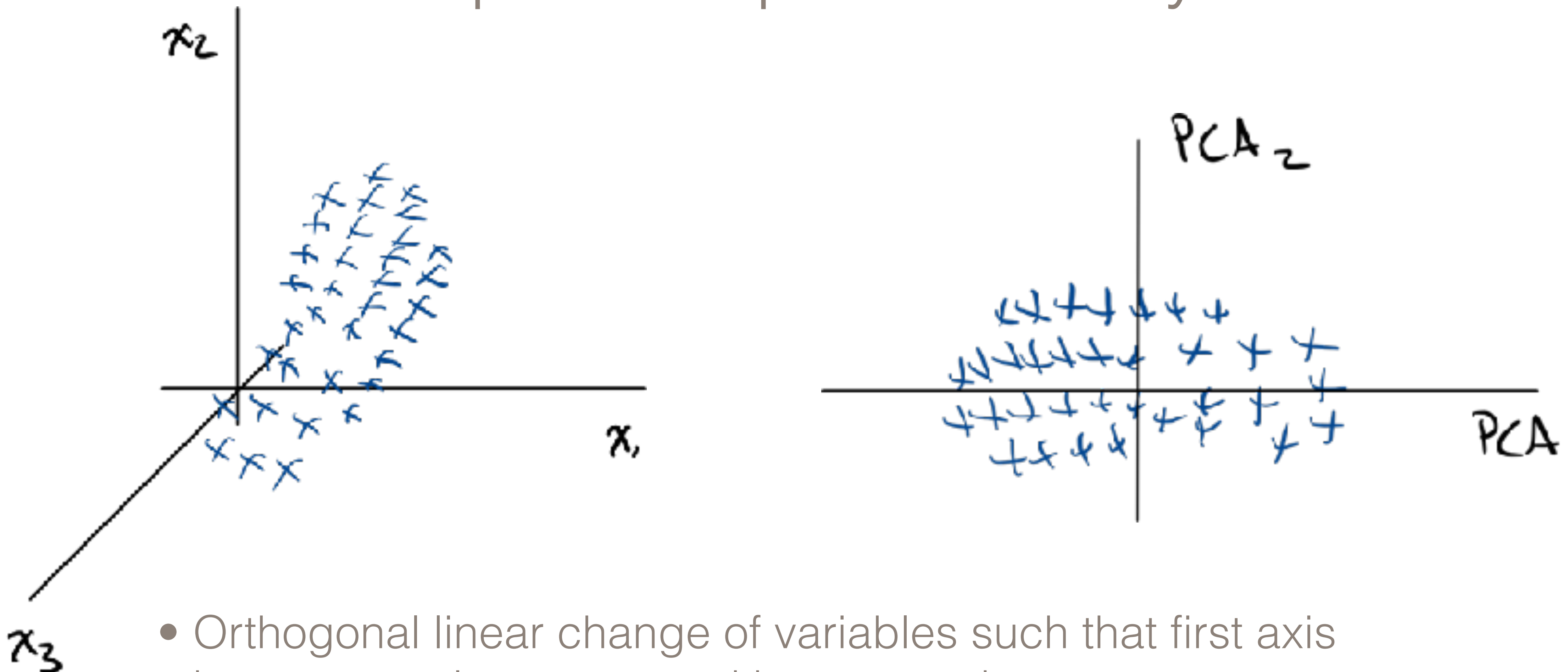
[1808.08979]

1. Use *Autoecoders* for anomaly detection
2. Emphasize training directly on data, how much signal can be in training and still work?

Introduction to Autoencoders

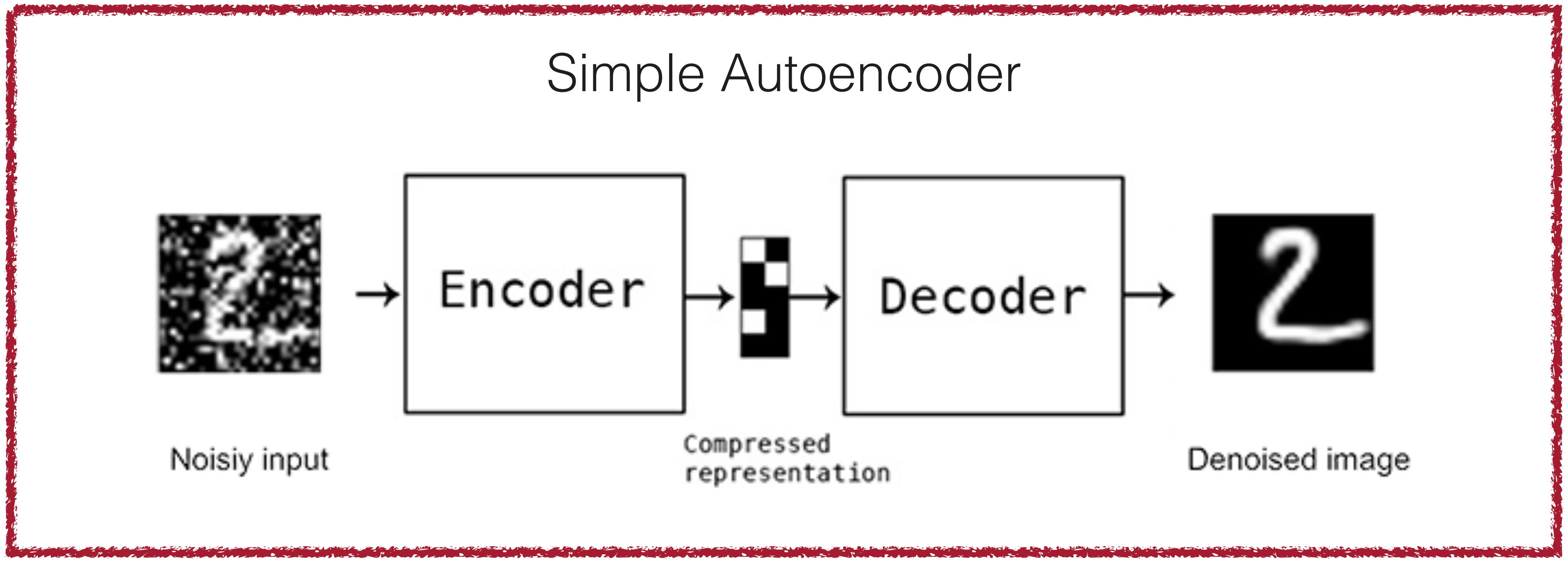
- Method for dimensionality reduction
- Focuses on important pieces of information and ignores noise

Principal Component Analysis



- Orthogonal linear change of variables such that first axis has most variance, second has second most, etc
- Which variables describe most of the variation of the data?
- Invertible (lossy if not using full basis)

Introduction to Autoencoders



- Encoding and Decoding can be non-linear
- Encoder learns what is important in the data and what is not
- Size of compressed representation chosen before training
- Compressed representation changed each training of the networks

Introduction to Autoencoders

Introduction to Autoencoders

- Unlike PCA, autoencoders need to be trained
- Need input data, output predictions, and a target
  - Target data is the same as input data

Introduction to Autoencoders

- Unlike PCA, autoencoders need to be trained
- Need input data, output predictions, and a target
  - Target data is the same as input data
- How to compare output predictions and target?
  - Use Mean Squared Error

$$d_{\mathrm{ME},2} = \mathrm{MSE} \equiv \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left| \mathrm{D}\big(\mathrm{E}(\mathrm{x})\big)_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}} \right|^2$$

Introduction to Autoencoders

- Unlike PCA, autoencoders need to be trained
- Need input data, output predictions, and a target
  - Target data is the same as input data
- How to compare output predictions and target?
  - Use Mean Squared Error

$$d_{\mathrm{ME},2} = \mathrm{MSE} \equiv \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left| \mathrm{D}\big(\mathrm{E}(\mathrm{x})\big)_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}} \right|^2$$

y = target (=x)
(or the input in this case)

Introduction to Autoencoders

- Unlike PCA, autoencoders need to be trained
- Need input data, output predictions, and a target
  - Target data is the same as input data
- How to compare output predictions and target?
  - Use Mean Squared Error

$$d_{\mathrm{ME},2} = \mathrm{MSE} \equiv \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left| \mathrm{D}\big(\mathrm{E}(\mathrm{x})\big)_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}} \right|^2$$

D(E((x)) = output of decoded, encoded data
use f(x) for rest of talk

y = target (=x)
(or the input in this case)

Introduction to Autoencoders

- Unlike PCA, autoencoders need to be trained
- Need input data, output predictions, and a target
  - Target data is the same as input data
- How to compare output predictions and target?
  - Use Mean Squared Error

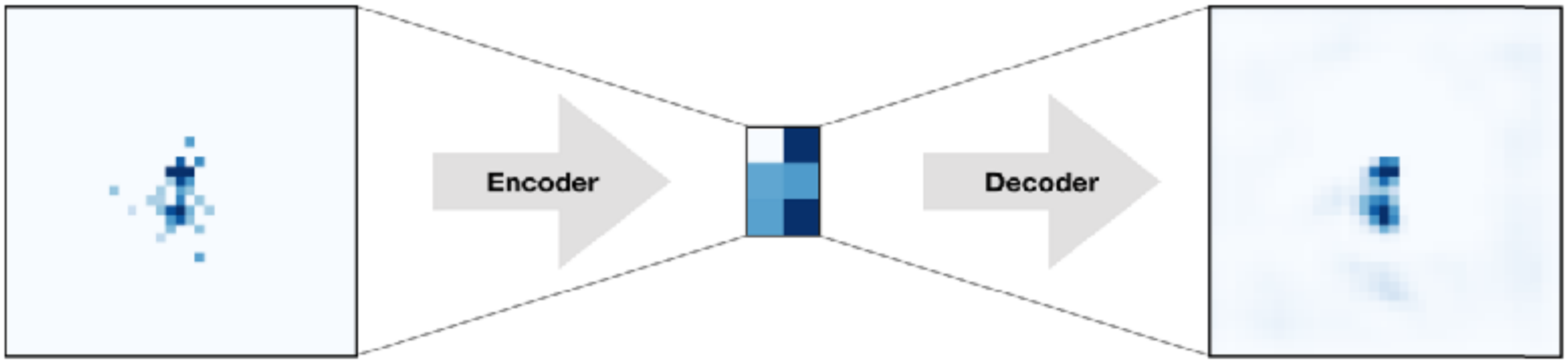
Take the average over all dimensions of the data vector

$$d_{\mathrm{ME},2} = \mathrm{MSE} \equiv \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \left| \mathrm{D}\big(\mathrm{E}(\mathrm{x})\big)_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}} \right|^2$$

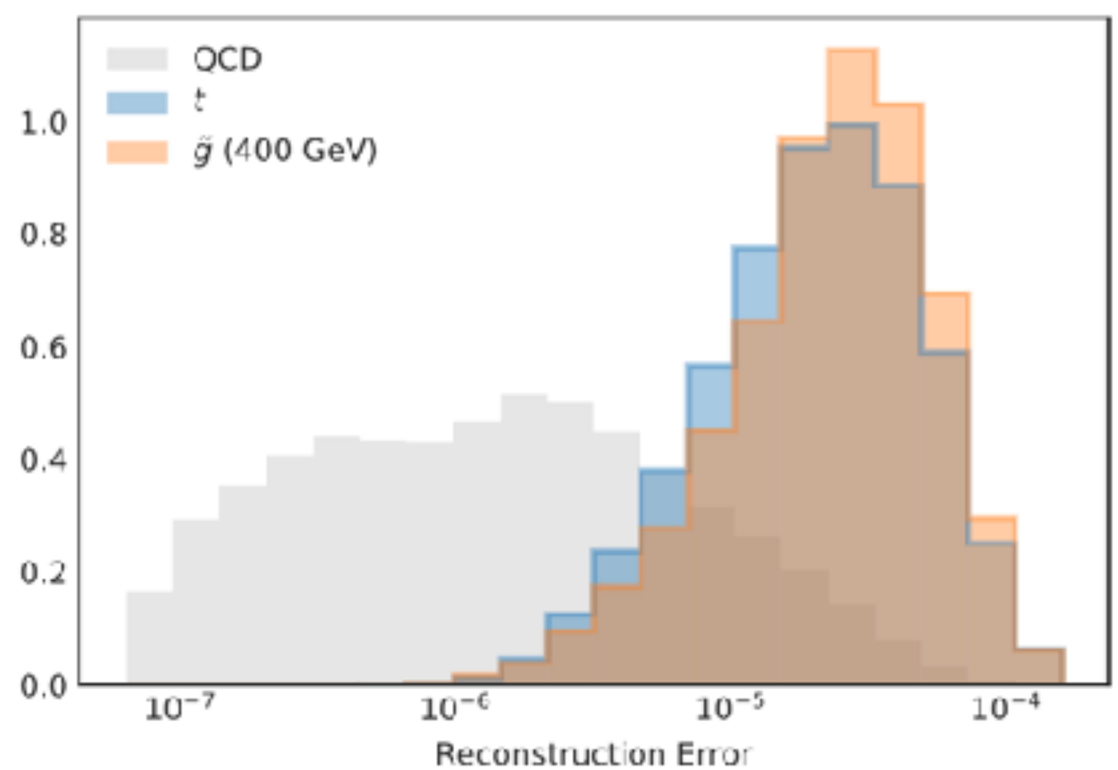
D(E((x)) = output of decoded, encoded data
use f(x) for rest of talk

y = target (=x)
(or the input in this case)

Autoencoders for Anomaly Detection

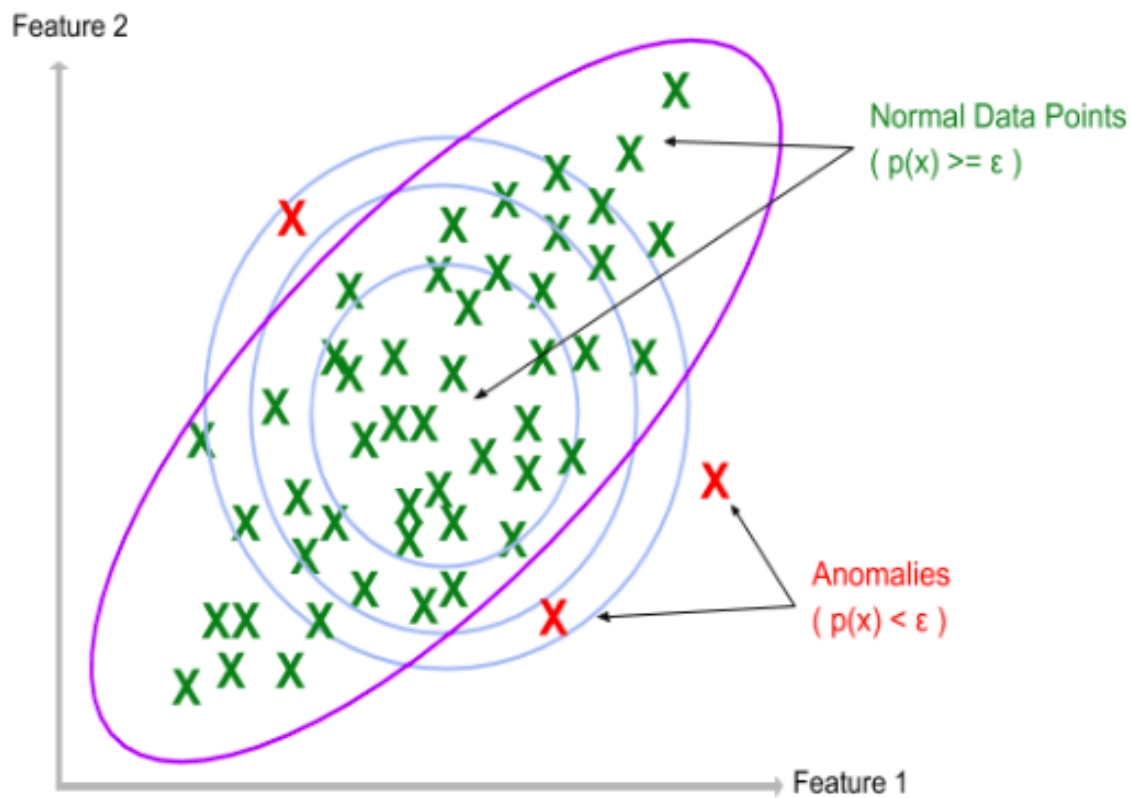


- Networks are trained to minimize the reconstruction error of events from SM background
- The encoding-decoding of BSM events will have larger reconstruction errors



[1808.08992]

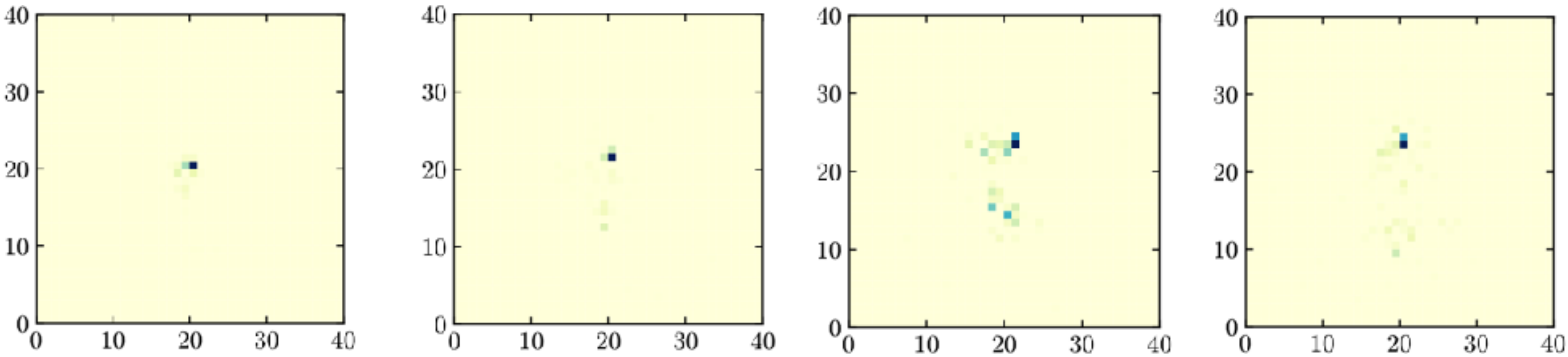
Challenges for Autoencoders

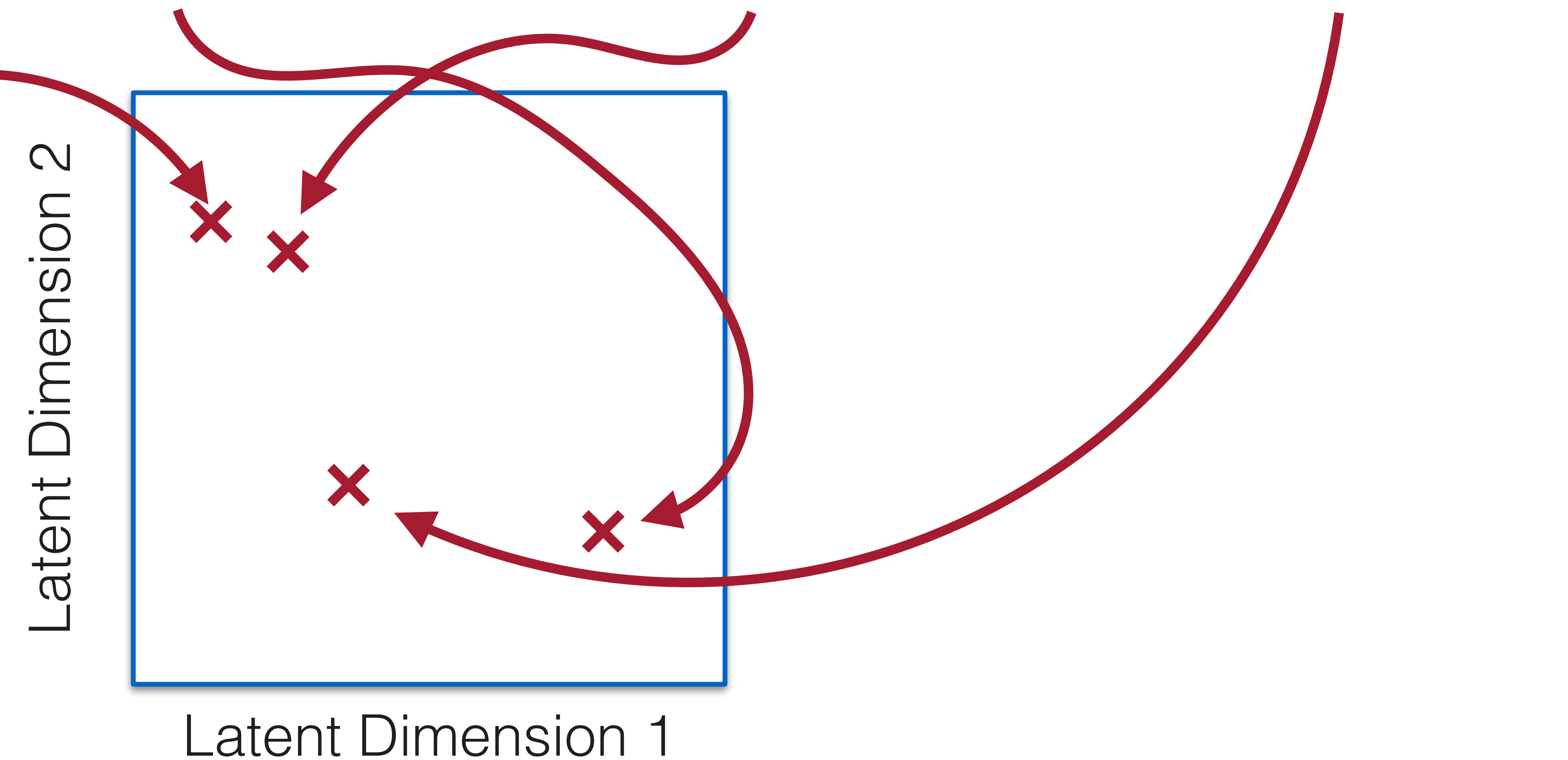


Do autoencoders fit with this picture of anomaly detection?

- Delicate balance between reconstructing SM well, but anomalous data poorly
- Latent space doesn't have structure, can't assign probabilistic interpretation
- Is MSE the right metric to use for reconstruction?

Plain Autoencoders

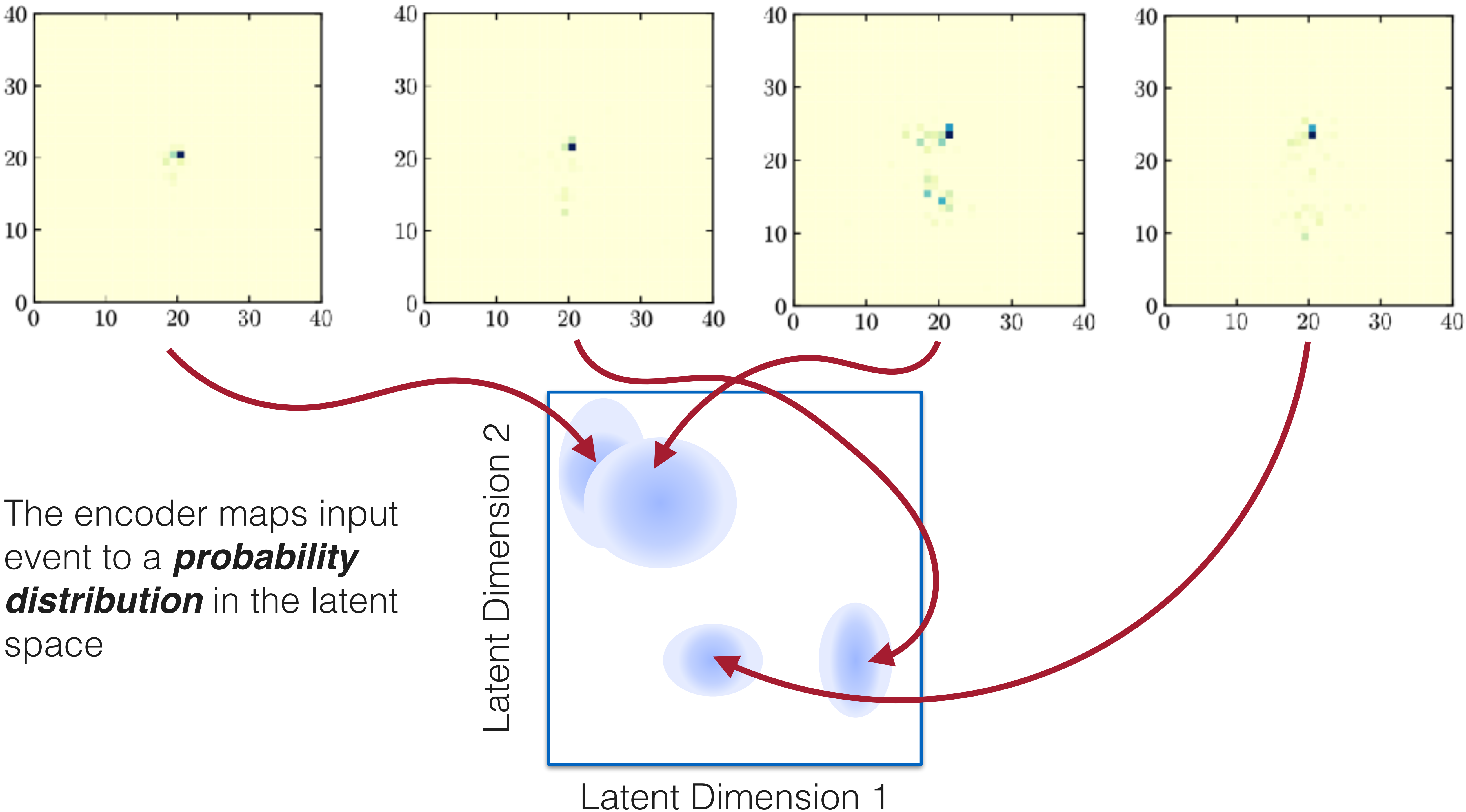




The encoder maps input event to a specific ***point*** in the latent space

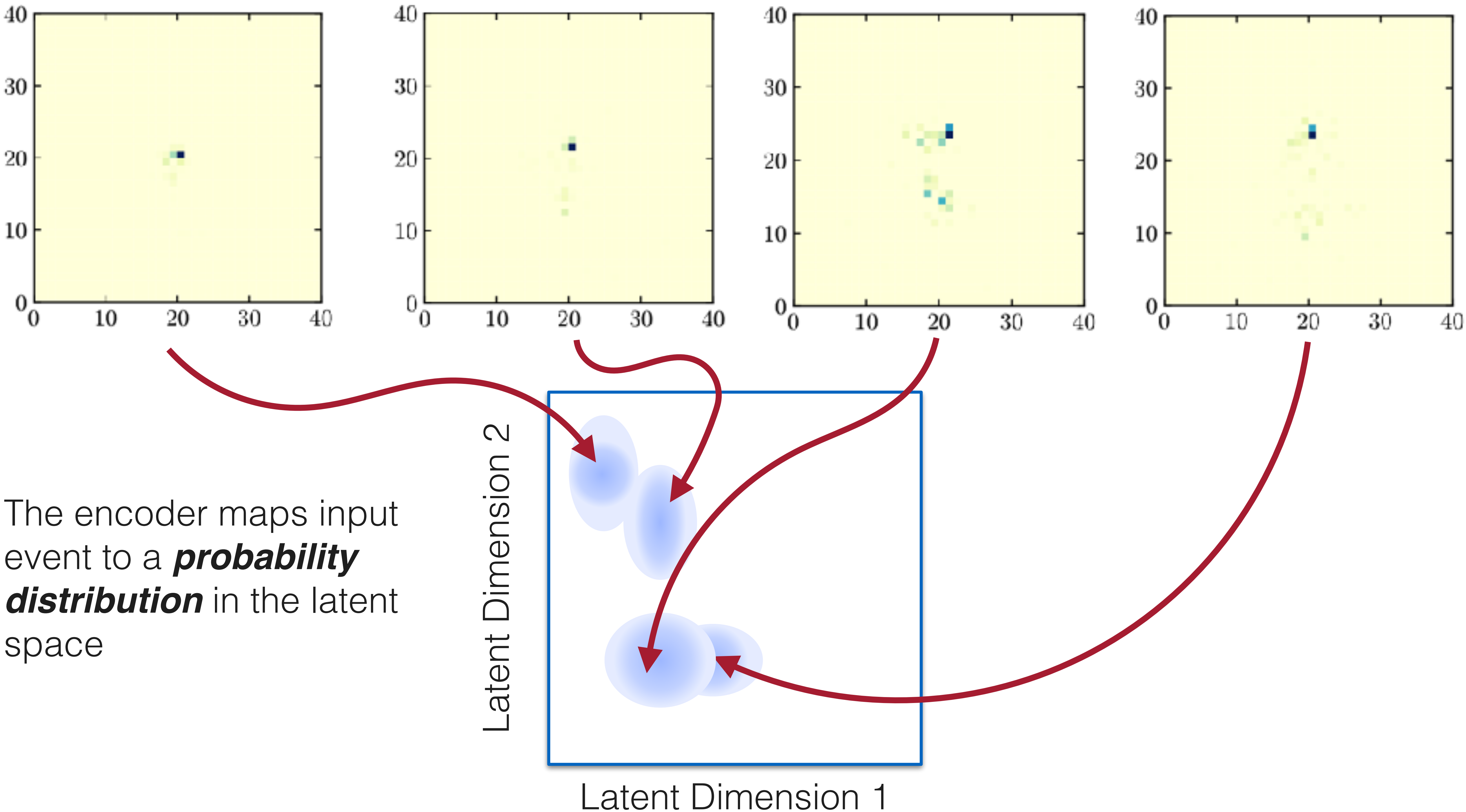
There is no intrinsic structure in the latent space of an Autoencoder

Variational Autoencoders



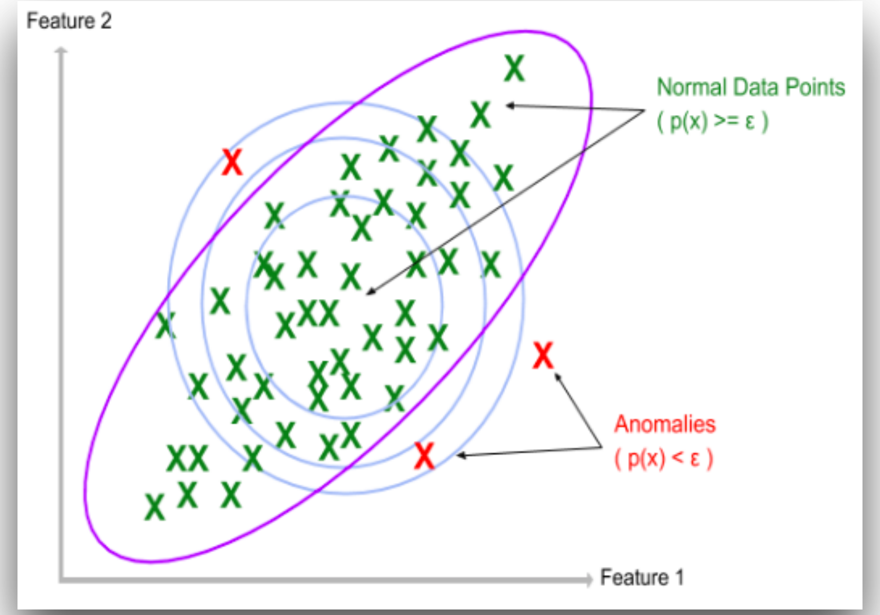
The encoder maps input event to a ***probability distribution*** in the latent space

Variational Autoencoders



The encoder maps input event to a ***probability distribution*** in the latent space

Variational Autoencoders



Add a probabilistic interpretation

How likely is a detector event, given a point in latent space?

Model each detector pixel as a Gaussian:

$$p(E|z) \propto \exp\left(\frac{-\left(E - D(z)\right)^2}{2\sigma^2}\right)$$

$$x = \{E_1, E_2, \cdots, E_N\}$$

$$p(x|z) = \prod_{i=1}^{N} p(E_i|z)$$

Probability of the observing an event

$$p(x) = \int p(x|z)p(z)dz \equiv \mathbb{E}_{p(z)}\left[p(x|z)\right]$$

What is the latent space to integrate over?

Variational Autoencoders

Probability of the observing an event

$$p(x) = \int p(x|z)p(z)dz \equiv \mathbb{E}_{p(z)}\left[p(x|z)\right]$$

What is the latent space to integrate over?

Use the encoder to learn/sample the latent space

$$p(x) = \int p(x|z)p(z)\frac{q(z|x)}{q(z|x)}dz = \mathbb{E}_{q(z|x)}\left[p(x|z)\frac{p(z)}{q(z|x)}\right]$$

Variational Autoencoders

Probability of the observing an event

$$p(x) = \int p(x|z)p(z)dz \equiv \mathbb{E}_{p(z)}\left[p(x|z)\right]$$

What is the latent space to integrate over?

Use the encoder to learn/sample the latent space

$$p(x) = \int p(x|z)p(z)\frac{q(z|x)}{q(z|x)}dz = \mathbb{E}_{q(z|x)}\left[p(x|z)\frac{p(z)}{q(z|x)}\right]$$

Take log likelihood and use Jensen's inequality to move log inside expectation value

$$\log p(x) \geq \mathbb{E}_{q(z|x)}\left[\log\left(p(x|z)\right) + \log\left(\frac{p(z)}{q(z|x)}\right)\right]$$

Variational Autoencoders

Probability of the observing an event

$$p(x) = \int p(x|z)p(z)dz \equiv \mathbb{E}_{p(z)}\left[p(x|z)\right]$$

What is the latent space to integrate over?

Use the encoder to learn/sample the latent space

$$p(x) = \int p(x|z)p(z)\frac{q(z|x)}{q(z|x)}dz = \mathbb{E}_{q(z|x)}\left[p(x|z)\frac{p(z)}{q(z|x)}\right]$$

Take log likelihood and use Jensen's inequality to move log inside expectation value

$$\log p(x) \geq \mathbb{E}_{q(z|x)}\left[\log\left(p(x|z)\right) + \log\left(\frac{p(z)}{q(z|x)}\right)\right]$$

~ MSE of encoded-decoded event

Variational Autoencoders

Probability of the observing an event

$$p(x) = \int p(x|z)p(z)dz \equiv \mathbb{E}_{p(z)}\left[p(x|z)\right]$$

What is the latent space to integrate over?

Use the encoder to learn/sample the latent space

$$p(x) = \int p(x|z)p(z)\frac{q(z|x)}{q(z|x)}dz = \mathbb{E}_{q(z|x)}\left[p(x|z)\frac{p(z)}{q(z|x)}\right]$$

Take log likelihood and use Jensen's inequality to move log inside expectation value

$$\log p(x) \geq \mathbb{E}_{q(z|x)}\left[\log\left(p(x|z)\right) + \log\left(\frac{p(z)}{q(z|x)}\right)\right]$$

~ MSE of encoded-decoded event

Kullback-Leibler Divergence (KLD) between encoded representation and latent prior

Summary of Autoencoders

Plain autoencoders: no sampling, $L = \mathrm{MSE}$

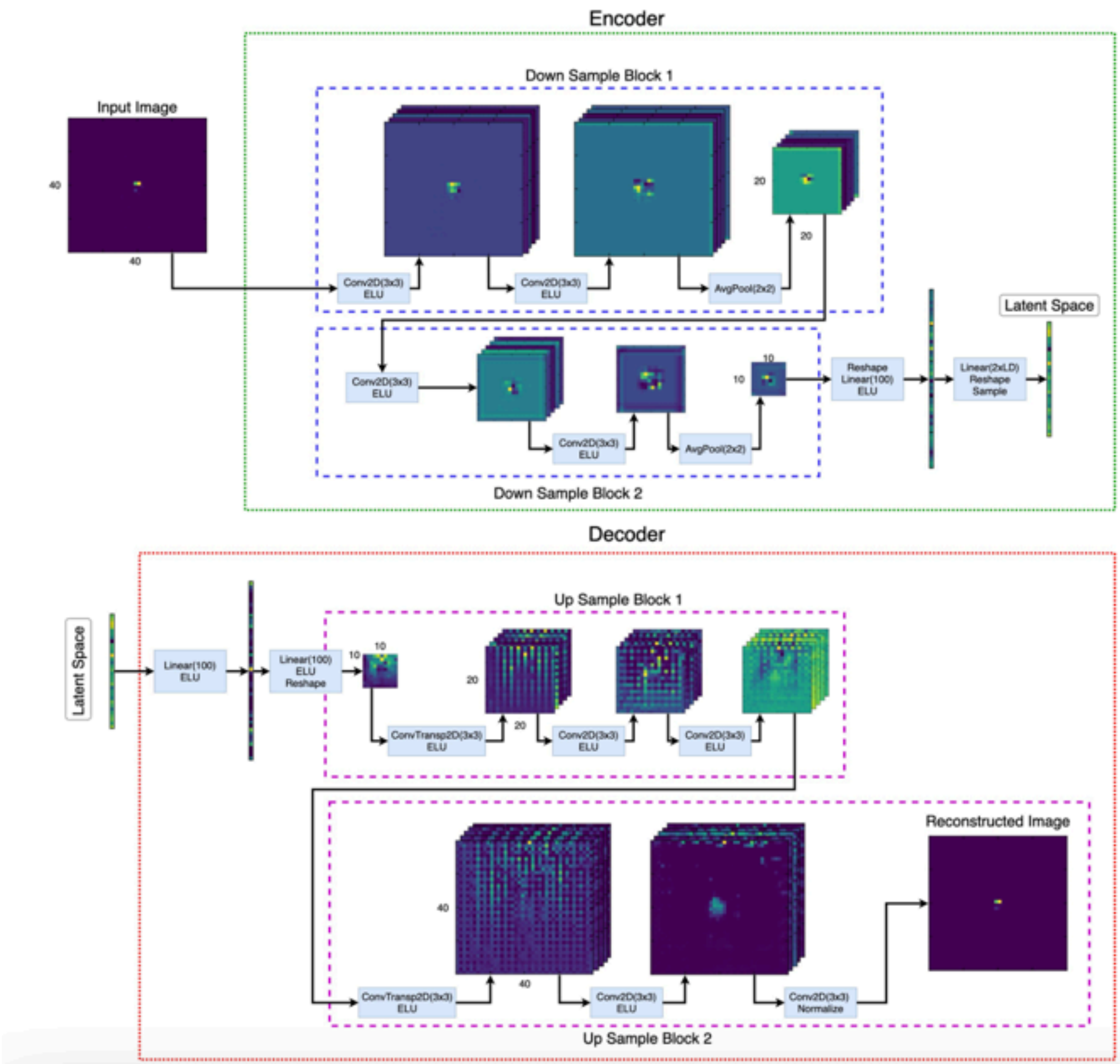
Variation autoencoders: sampling in latent space, $L = \mathrm{MSE} + \mathrm{KLD}$

Sampling adds/forces structure on the latent space.
KLD term helps to regularize and adds to probabilistic interpretation

In practice, these terms may be too far apart, introduce a scaling between them

$$L = (1-\beta)\,\mathrm{MSE} + \beta\,\mathrm{KLD}$$

Generative Networks



Can randomly sample in the latent space to get “new” data

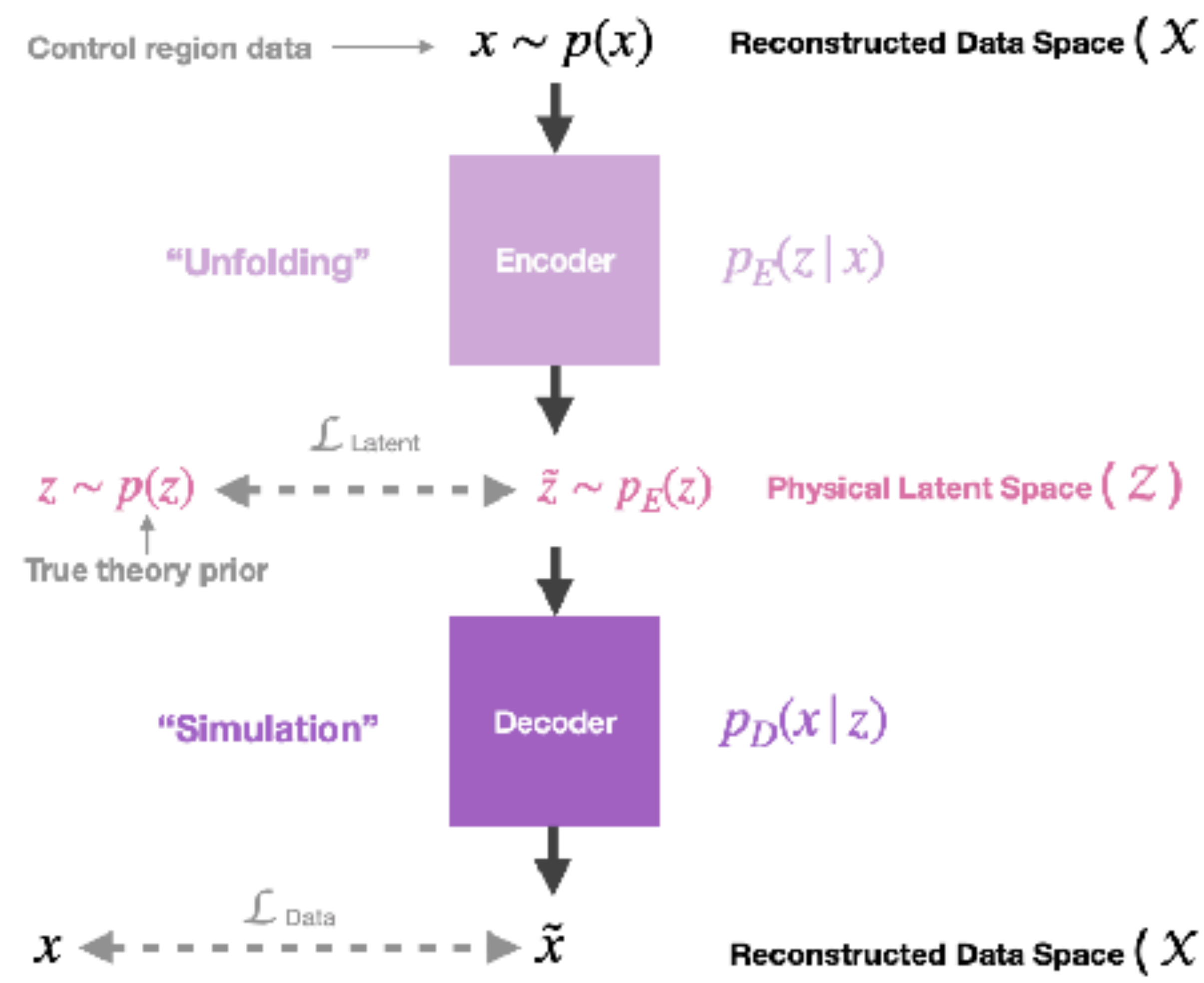
Alter the latent space

- Latent loss ($\mathcal{L}_{\text{Latent}}$)
  - SW distance for finite samples
- Data loss ($\mathcal{L}_{\text{Data}}$):
  - Mean Squared Error (MSE)
- Total SWAE loss function:

  $$\mathcal{L}_{\text{SWAE}} = \mathcal{L}_{\text{Data}} + \lambda \mathcal{L}_{\text{Latent}}$$

- Easy to add additional physically-motivated constraints

  $$\mathcal{L} = \mathcal{L}_{\text{SWAE}} + \lambda_i' \mathcal{L}_i$$



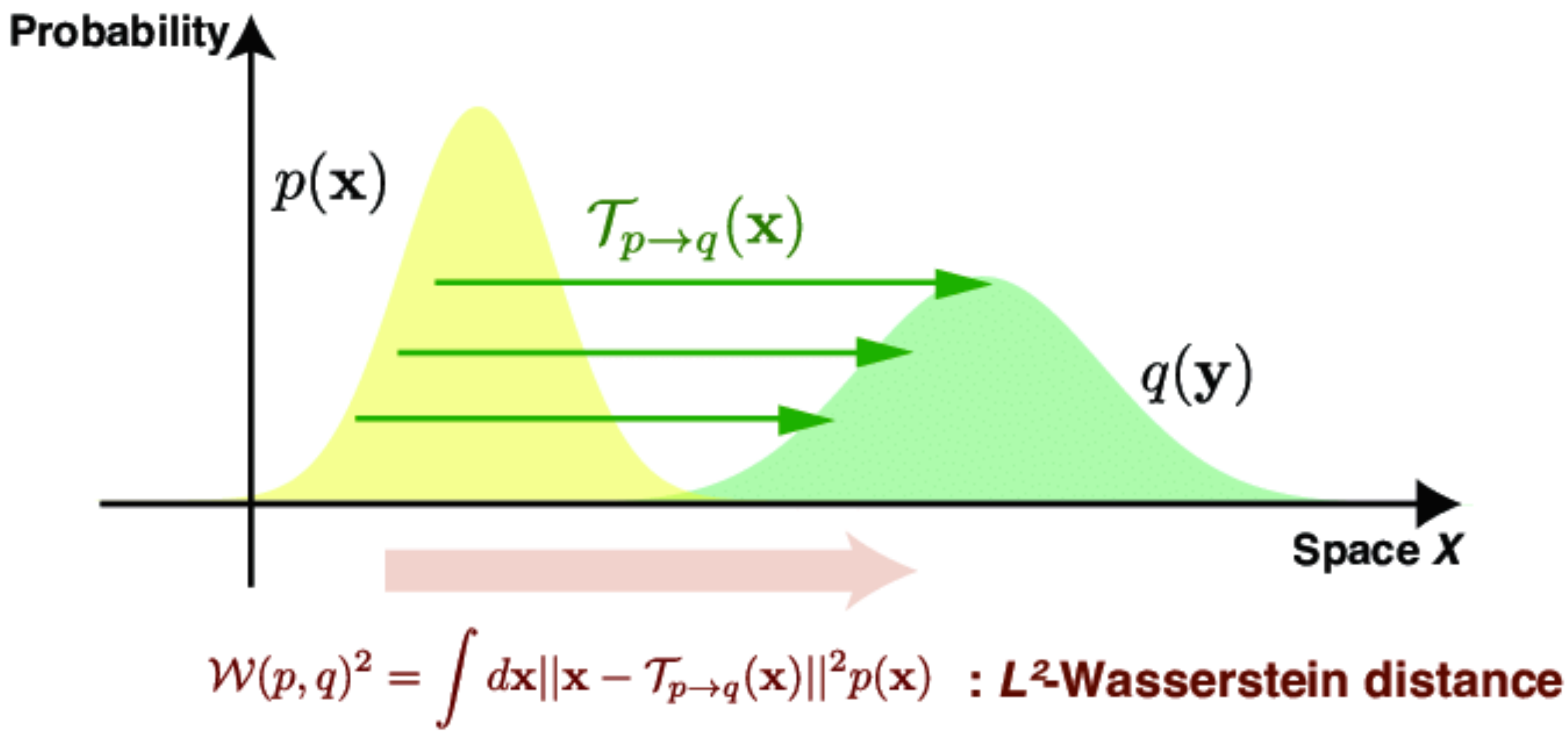
SWAE Structure

In the previous discussion of VAEs, we used a Gaussian for the prior of the latent space. However, it is also possible to use different/more motivated priors. For instance, use the parton-level distributions.

[2101.08944]

Alter the latent space

Use the Earth mover's distance (Wasserstein distance) to compare the latent prior with the learned latent distributions



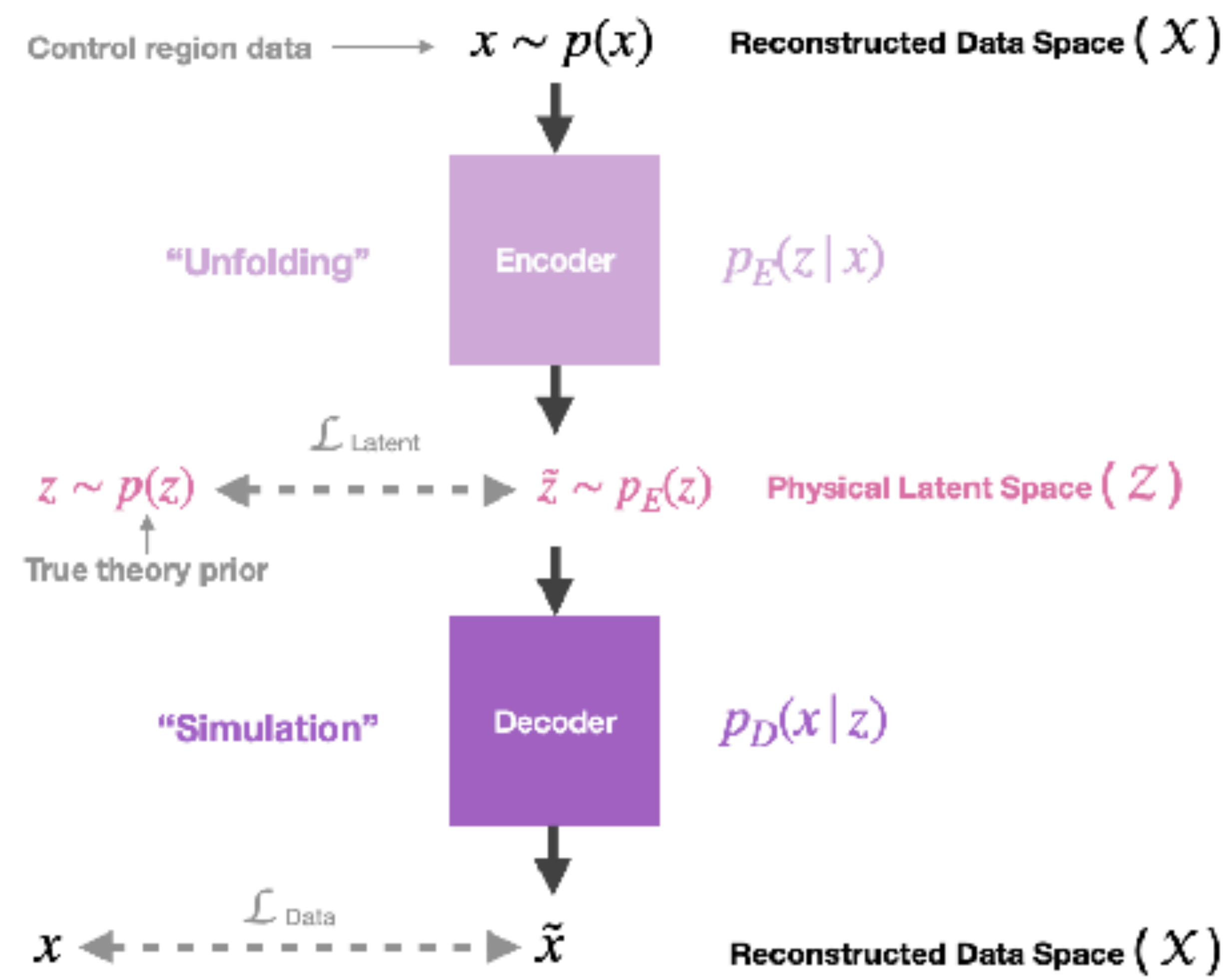
Side note: Wasserstein distances are also an active area of research for the other areas of machine learning in HEP, such as classification and anomaly detection.

Alter the latent space

- Latent loss ($\mathcal{L}_{\text{Latent}}$)
  - SW distance for finite samples
- Data loss ($\mathcal{L}_{\text{Data}}$):
  - Mean Squared Error (MSE)
- Total SWAE loss function:

  $\mathcal{L}_{\text{SWAE}} = \mathcal{L}_{\text{Data}} + \lambda \mathcal{L}_{\text{Latent}}$
- Easy to add additional physically-motivated constraints

  $\mathcal{L} = \mathcal{L}_{\text{SWAE}} + \lambda_i' \mathcal{L}_i$



SWAE Structure

Example in paper: Examine p p > t t~ events with semi-leptonic decays,

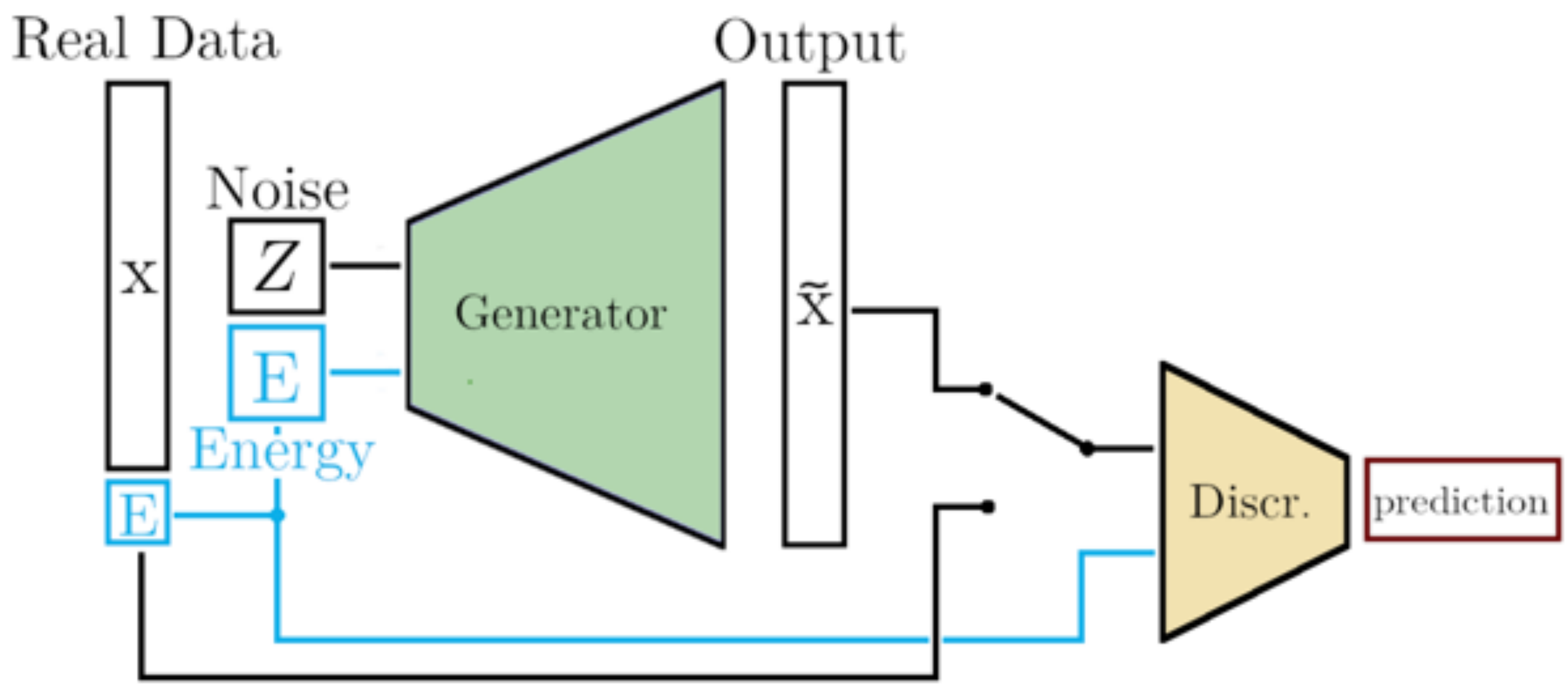
X is the detector level information (lepton 4-vector, MET, and jet four vectors)

Z is the parton level information (lepton, neutrino, and quark four vectors)

Many challenges, active research

[2101.08944]

Other generative models



Generative Adversarial Networks (GAN) work by taking in random noise and generating an event (image, observables, etc). There are two components to this, first is the network which generates the event, second is the discriminator network. This network is trained to tell the difference between real and generated data. The generator is trained to trick the discriminator.

Over 55 papers in the HEP literature using GANs

Other generative models

Ideas seen for generating events:
GANS -> Noise to data
VAES -> Latent representation to data

In both of these, we are trying to capture something about the underlying distributions of the true data, why not try to learn that directly?

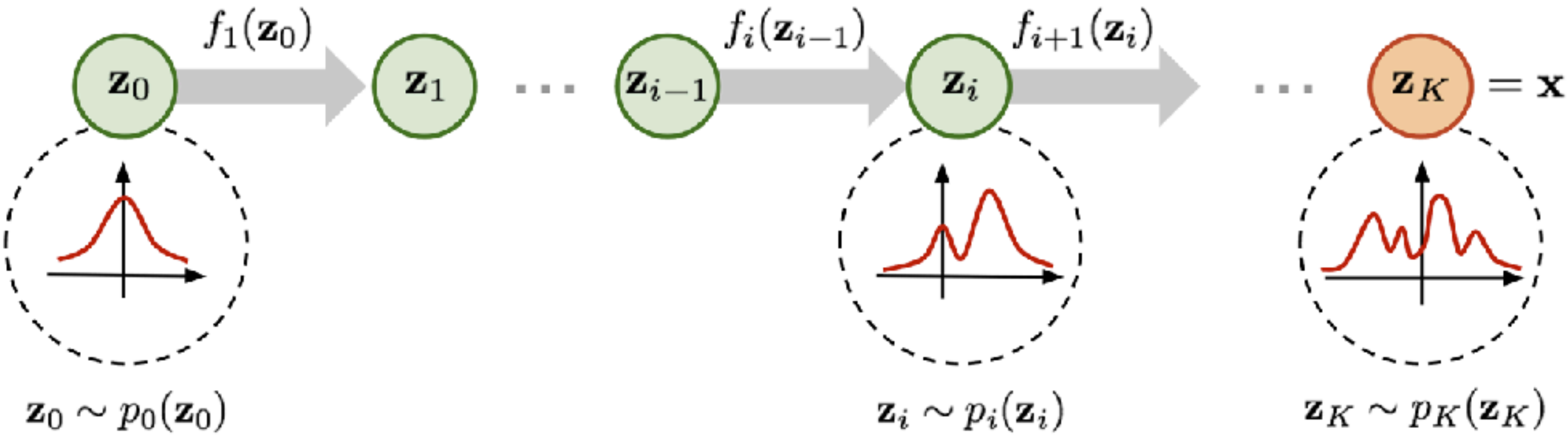
Normalizing Flows

Normalizing Flows

- Normalizing flows implement a change of variables from a simple distribution to the real distribution.
- Often use special layers which are easily invertible and have a simple Jacobian
- Once trained, can draw a random sample from the easy distribution and obtain a sample from a complex distribution
- Invertibility then also allows for taking a sample from the complex distribution and computing the likelihood

Nice review of the methods [1912.02762.pdf] (not HEP)

Normalizing Flows



These models allow for easy density estimation (learn the PDF of the data)

Allows for: Generating events, Generating field configurations for Lattice Gauge Theory, or Anomaly Detection

Conclusions

- There are many types of machine learning which do not simply use labeled data for supervision
- We saw how AEs for anomaly detection do not build in a probabilistic interpretations, but a VAE does
- VAES can also be used to generate data (and the latent space can be made more intuitive)
- There are many other ways to make generative networks
- We can use ML to estimate probabilities which allows for inference as well

Ideas for Project (Tutorial 3)

1) Compare raw-4vectors, n-subjettiness basis, and jet-images for top-tagging dataset
    a) Which does best?
    b) Are the number of parameters the similar?
    c) Speed of training?
2) Use **anomaly detection** techniques to train only on the background data. Then apply to the test set and see how the performance compares to supervised classification.

References

https://iml-wg.github.io/HEPML-LivingReview/

Particle Data Group has a new review chapter on ML which covers all of these techniques and more https://pdg.lbl.gov/