RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

Searching for New Physics with Deep Autoencoders

Yuichiro Nakai (Rutgers)

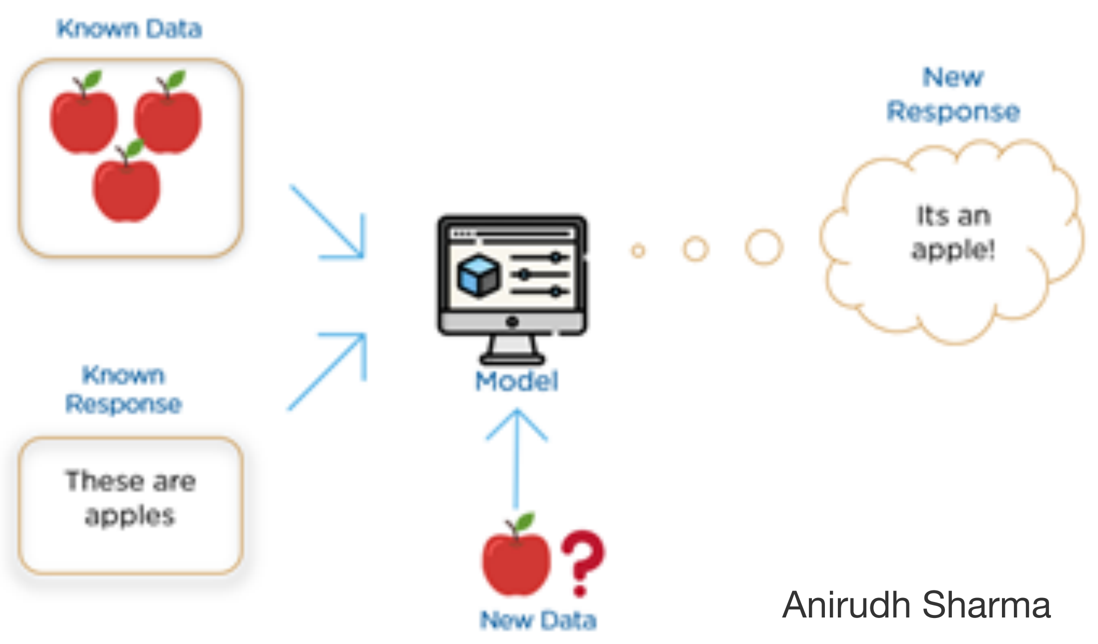
Based on M. Farina, YN and D. Shih, arXiv:1808.08992 [hep-ph].

Supervised or Unsupervised

Machine learning algorithms can be classified into:

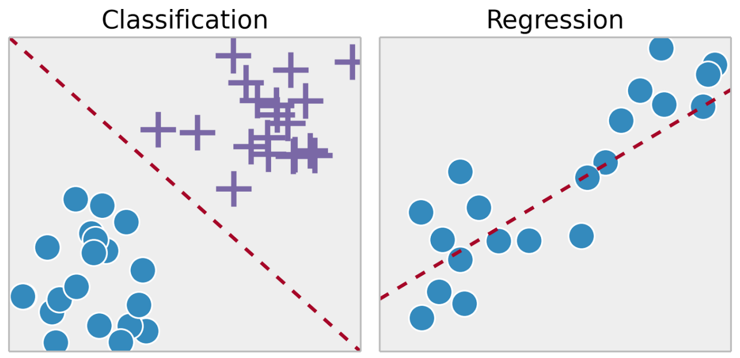
Supervised learning

Learn from labeled data



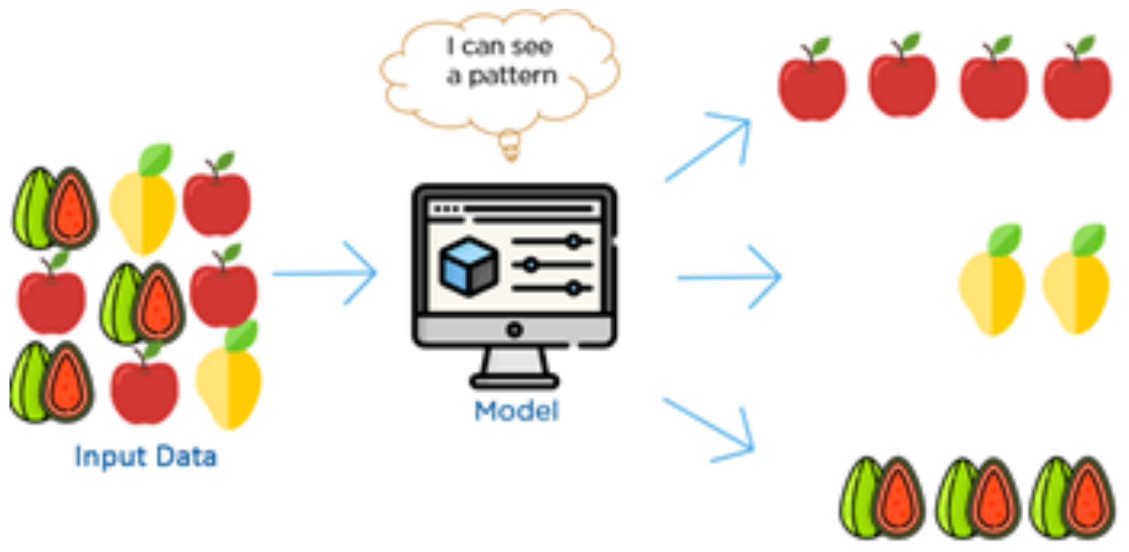
Anirudh Sharma

Applications)



Unsupervised learning

Learn from unlabeled data



The system looks for patterns and extracts features in data.

Applications) Clustering

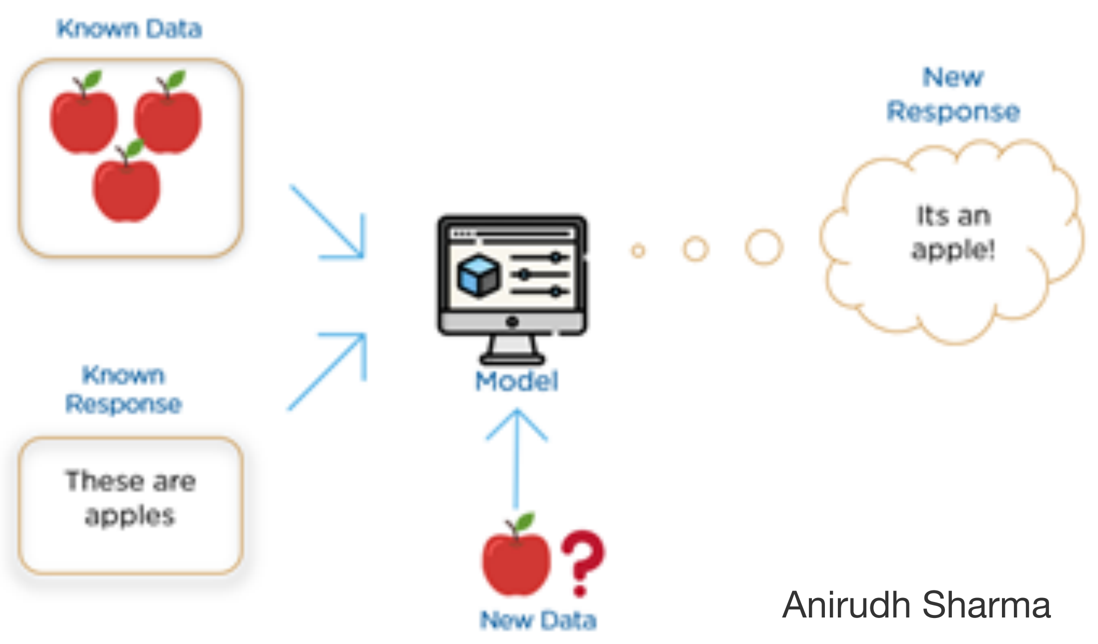
Anomaly detection

Supervised or Unsupervised

Machine learning algorithms can be classified into:

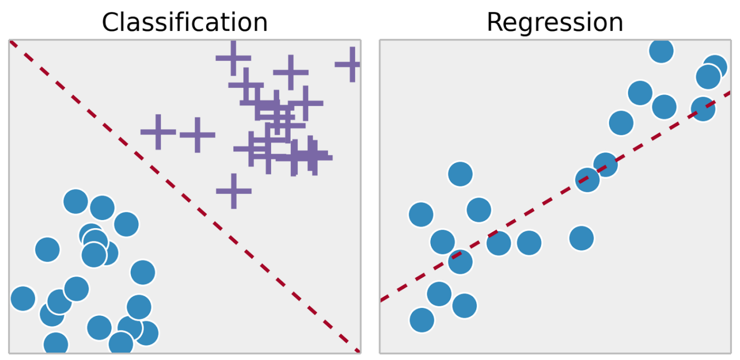
Supervised learning

Learn from labeled data



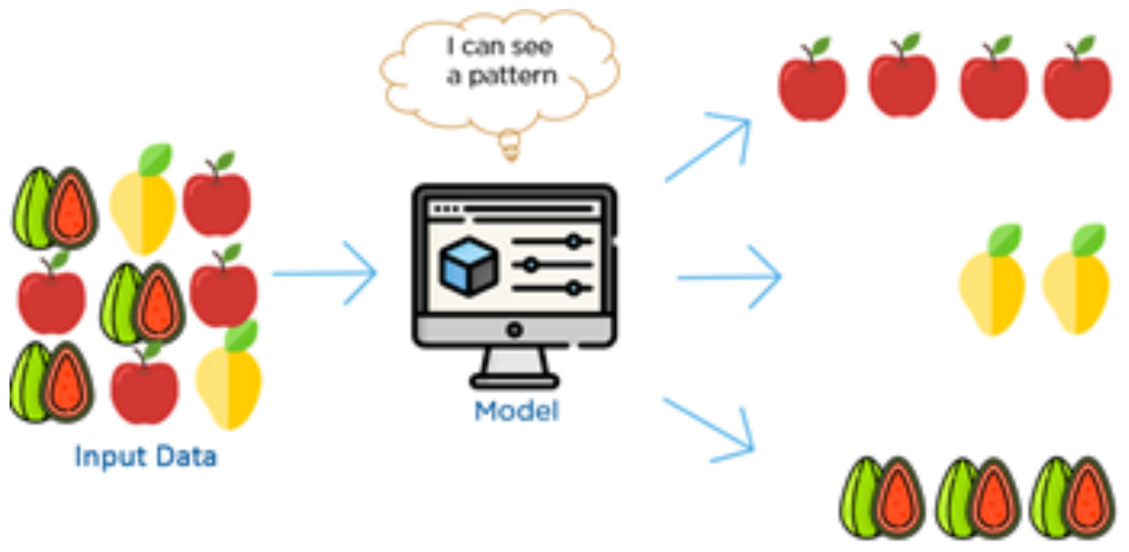
Anirudh Sharma

Applications)



Unsupervised learning

Learn from unlabeled data



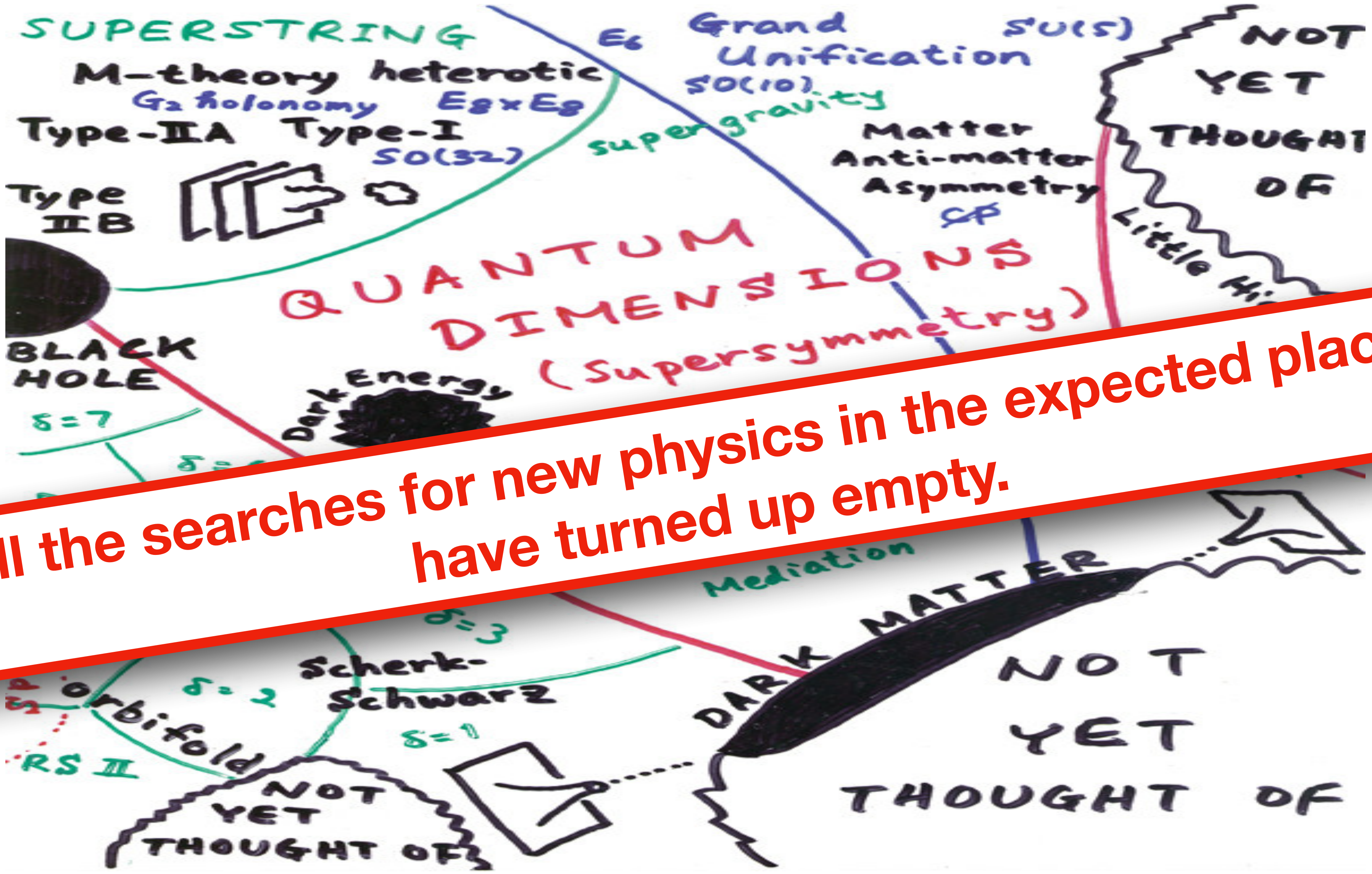
The system looks for patterns and extracts features in data.

Applications) Clustering

Anomaly detection

Anomaly Detection

We have considered many possibilities of BSM physics with top-down theory prejudice (supersymmetry, extra dimension, …)



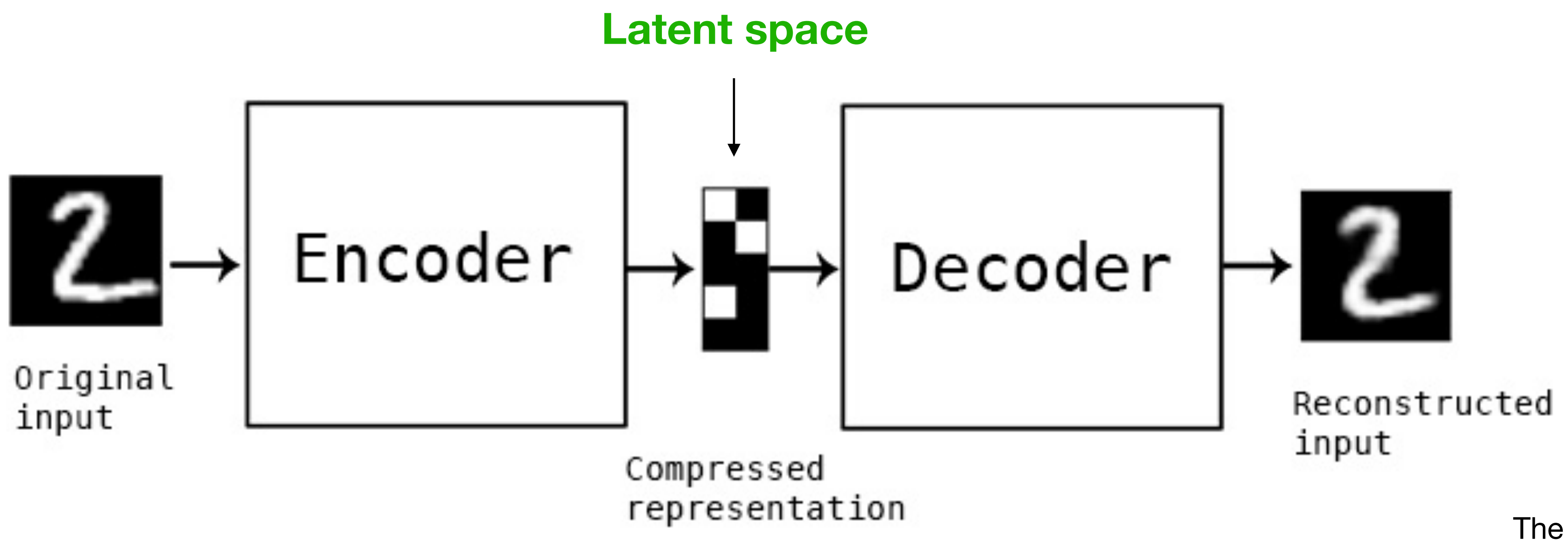
All the searches for new physics in the expected places have turned up empty.

Hitoshi Murayama

We need more ways to discover <u>the unexpected</u> at the LHC, and here is where unsupervised machine learning comes into play.

Autoencoder

Autoencoder is an unsupervised learning algorithm that maps an input to a latent compressed representation and then back to itself.



The Keras Blog

Anomaly detection with autoencoder

- Autoencoder learns to map background events back to themselves.
- It fails to reconstruct anomalous events that it has never encountered.

➡ Signal the existence of anomaly !

Sample Generation

The idea is general, but concentrate on detection of anomalous jets.

Generate jet samples by using PYTHIA for hadronization and Delphes for detector simulation.

Background : <u>QCD jets</u> $p_T \in [800,\ 900]\ \mathrm{GeV}$ $|\eta| < 1$

Signal jets: <u>top jets</u>, <u>RPV gluino jets</u> $m_{\tilde{g}} = 400\ \mathrm{GeV}$

(decay to 3 light quark jets)

Match requirement : heavy resonance is within the fat jet, $\Delta R < 0.6$

Merge requirement : the partonic daughters of heavy resonance is within the fat jet, $\Delta R < 0.6$

We use sample sizes of 100k events for training and testing.
(The performance seems to saturate.)

Jet Images

Concentrate on jet images (2D of eta and phi) whose pixel intensities correspond to total pT.

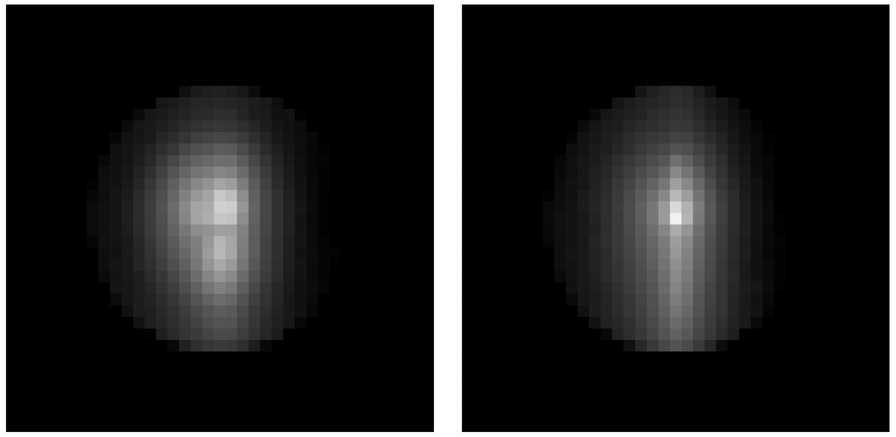
Image pre-processing

1. Shift an image so that the centroid is at the origin
2. Rotate the image so that the major principal axis is vertical
3. Flip the image so that the maximum intensity is in the upper right region
4. Normalize the image to unit total intensity
5. Pixelate the image (37 x 37 pixels)

Average images

Left : top jets

Right : QCD jets



Macaluso, Shih (2018)

Autoencoder Architectures

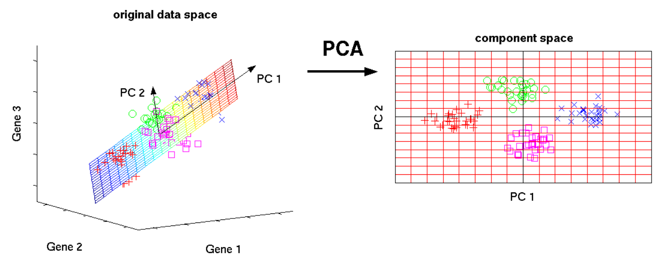
Reconstruction error : a measure for how well autoencoder performs.

$$L(x,\hat{x}) = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - \hat{x}_i\right|^2$$

x : inputs

$\hat{x}$: outputs

Train autoencoder to minimize the reconstruction error on background events.

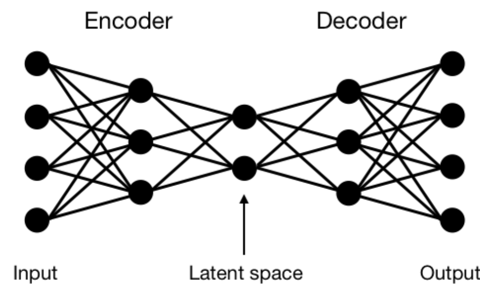


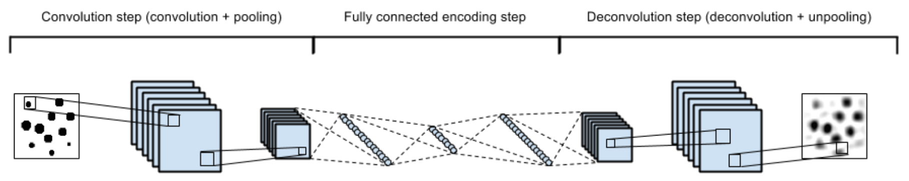
Architectures we consider :

✓ Principal Component Analysis (PCA)

✓ Simple (dense) autoencoder

✓ Convolutional (CNN) autoencoder



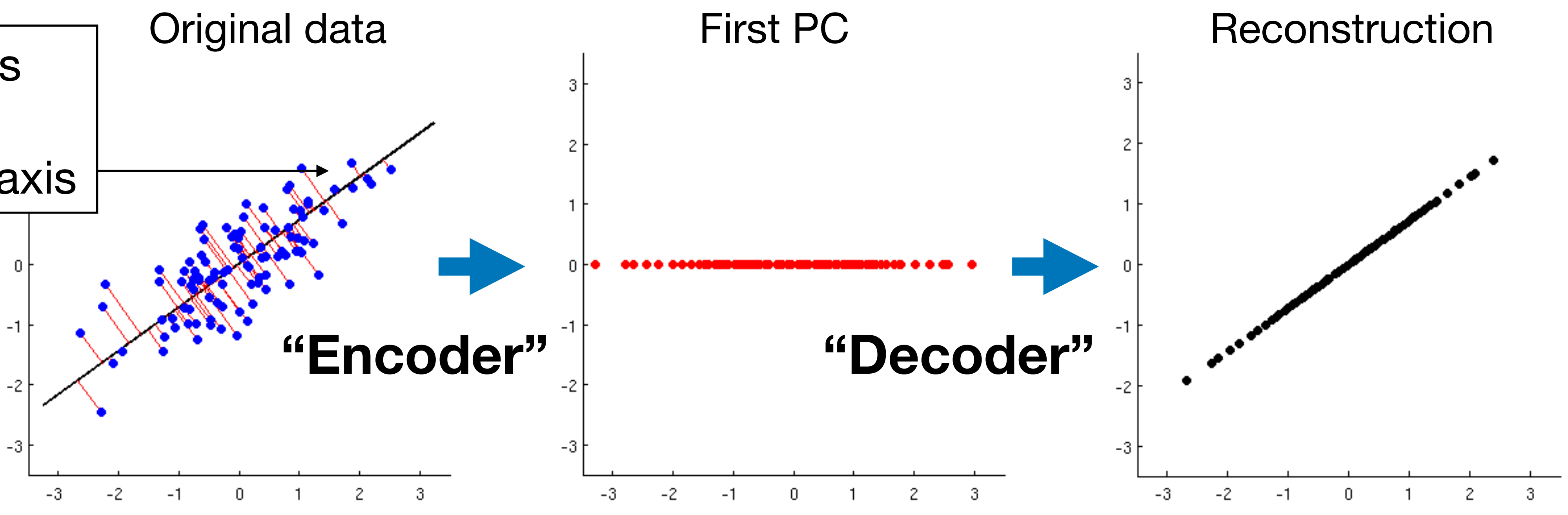


Convolution step (convolution + pooling)
Fully connected encoding step
Deconvolution step (deconvolution + unpooling)

Principal Component Analysis

PCA is a technique to drop the least important variables by focusing on variance of data.

Find the axis and project data to the axis



Eigenvectors of covariance matrix of $\boldsymbol{x}_n - \boldsymbol{c}_0 \;\; (\boldsymbol{c}_0 = \sum_n \boldsymbol{x}_n / N)$ give desired axes.

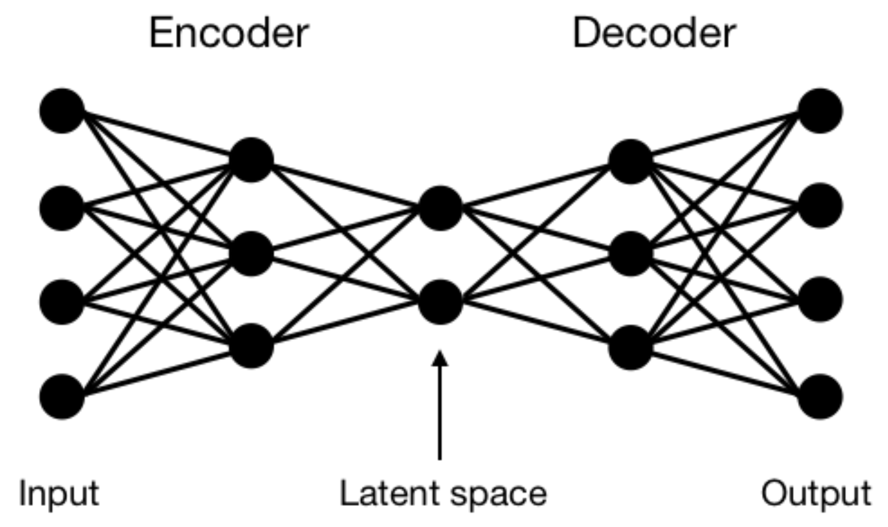
$\Gamma = (\boldsymbol{e}_1 \; \boldsymbol{e}_2 \; \dots \; \boldsymbol{e}_d)$ *d* : the number of principal components ($d < D$)

"PCA autoencoder"

"Encoder" : $\tilde{\boldsymbol{x}}_n = (\boldsymbol{x}_n - \boldsymbol{c}_0)\Gamma$ "Decoder" : $\boldsymbol{x}'_n = \tilde{\boldsymbol{x}}_n \Gamma^T + \boldsymbol{c}_0$

Simple Autoencoder

Autoencoder with a single dense (fully-connected) layer as encoder and as decoder.



✓ Encoder and decoder are symmetric.

✓ The number of neurons in a hidden layer = 32.

✓ Flatten a jet image into a single column vector.

✓ We use Keras with Tensorflow backend for implementation.

Training details

✦ The default Adam algorithm for optimizer.

✦ Minibatch size of 1024 ⟵ The number of images fed into the network at one time
~100 iterations of optimization in one epoch

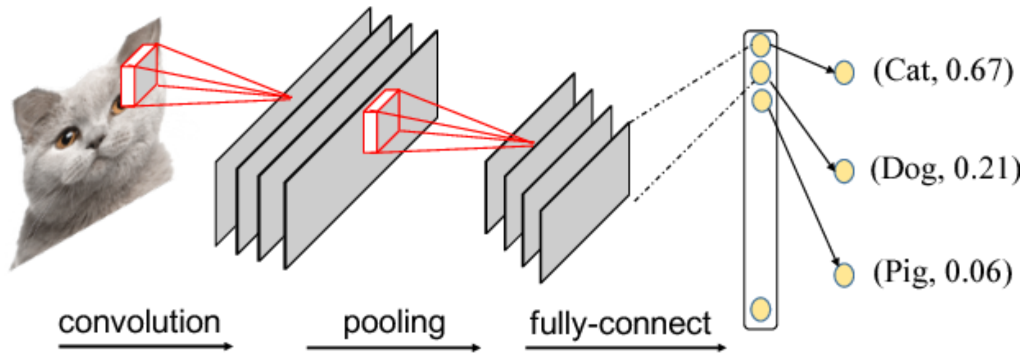
✦ Early stopping : threshold = 0 and patience = 5 ⟵ To avoid overtraining

Convolutional Autoencoder

Convolutional Neural Network (CNN)

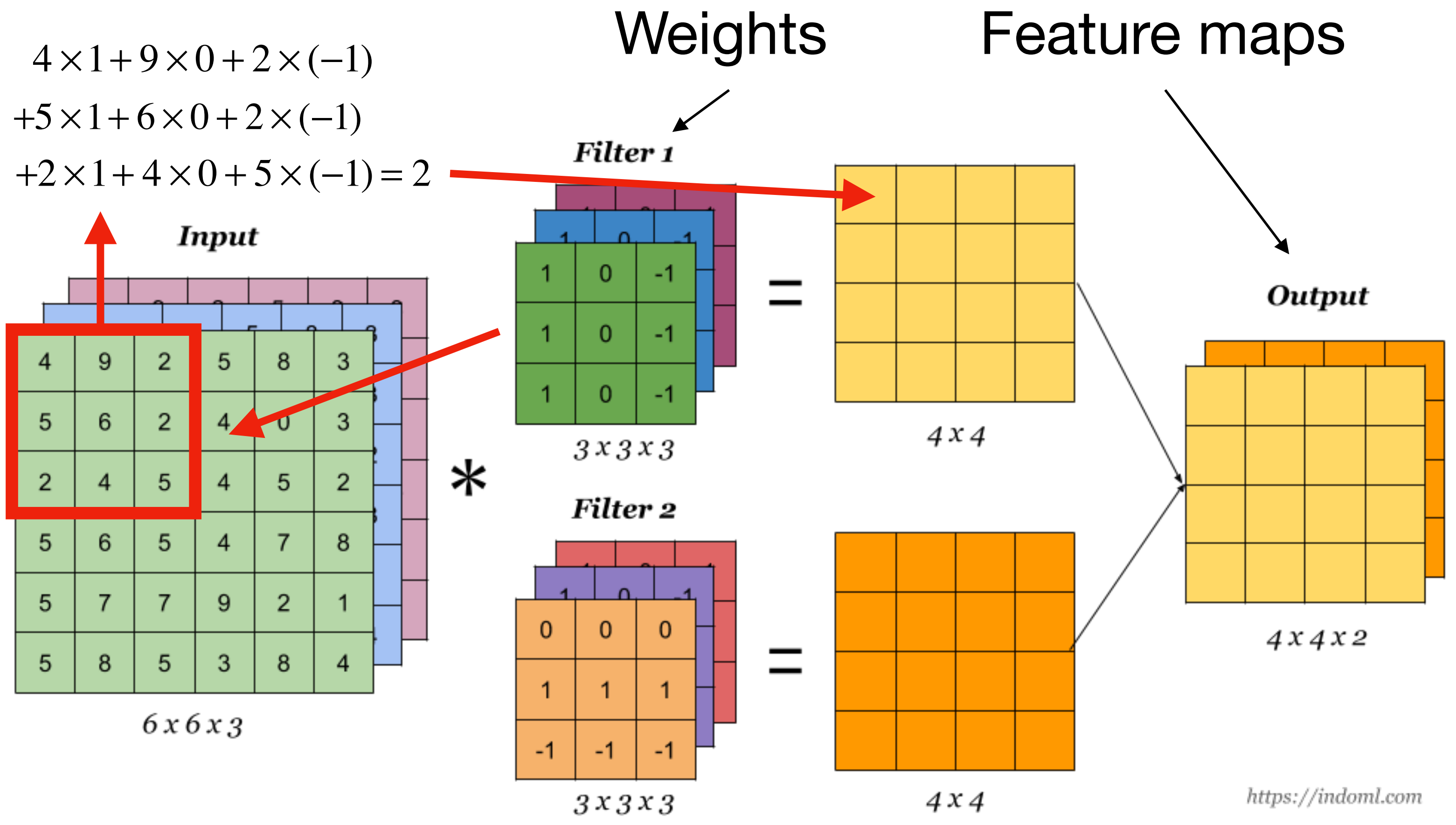
✓ Show high performance for image recognitions

✓ Maintain the spacial information of images



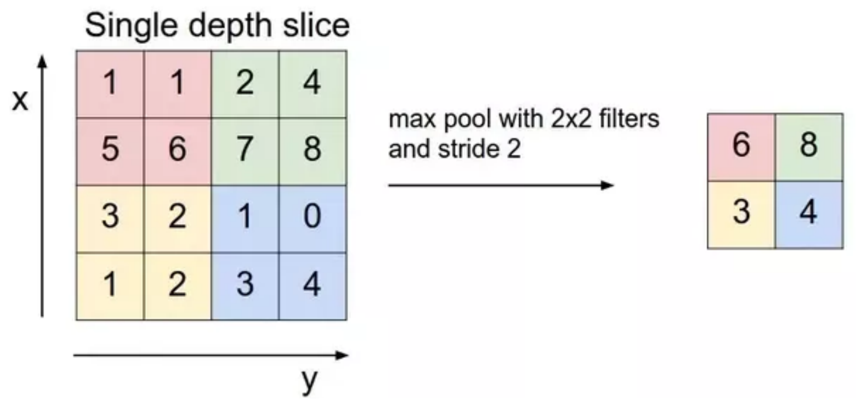
arXiv:1712.01670

Convolutional layer



Max pooling

Reduce the image size



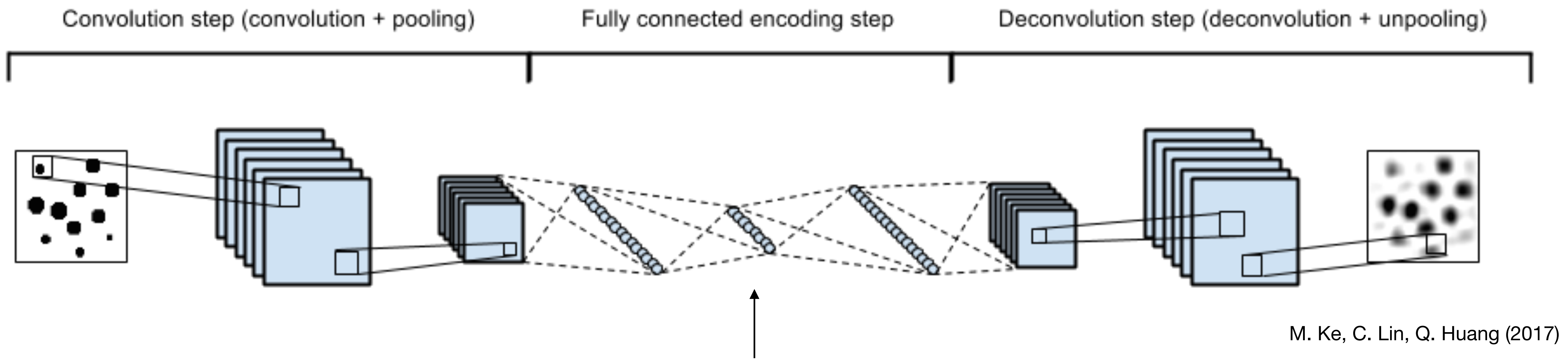
1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Up sampling (pooling) also exists in autoencoder.

Convolutional Autoencoder

Autoencoder architecture :


Convolution step (convolution + pooling)
Fully connected encoding step
Deconvolution step (deconvolution + unpooling)
M. Ke, C. Lin, Q. Huang (2017)

Encoder **Latent space** **Decoder**

128C3-MP2-128C3-MP2-128C3-32N-6N-32N-12800N-128C3-US2-128C3-US2-1C3

128C3 : 128 filters with a 3x3 kernel

MP2 : max pooling with a 2x2 reduction factor

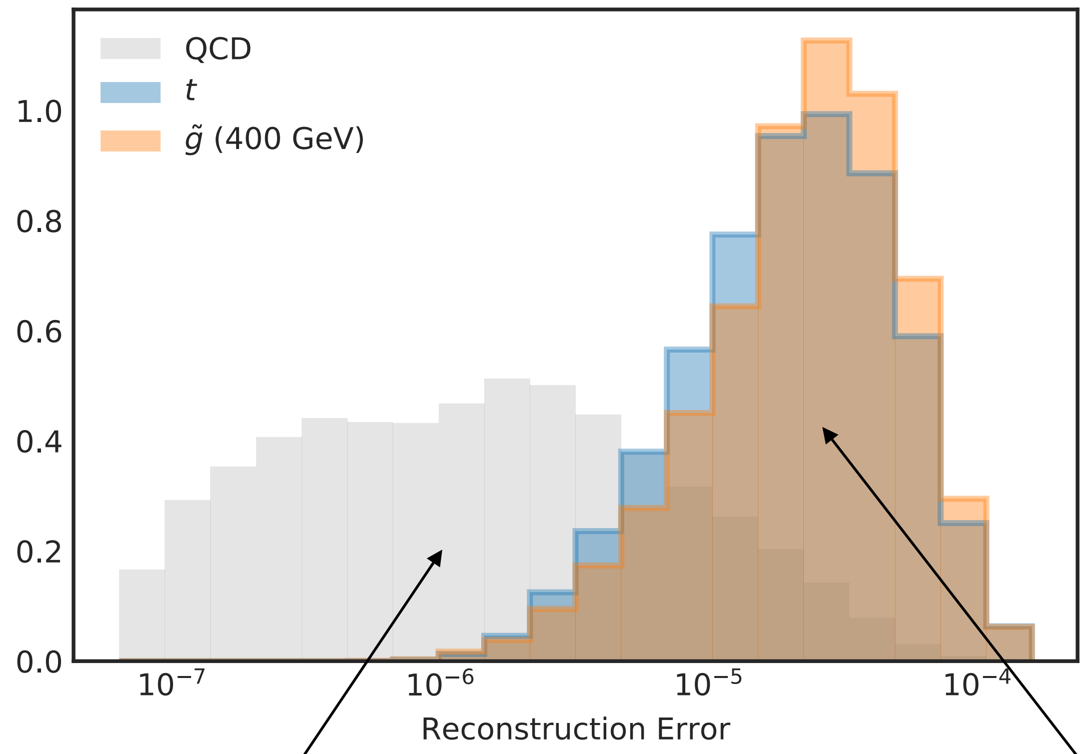
32N : a fully-connected layer with 32 neurons

US2 : up sampling with a 2x2 expansion factor

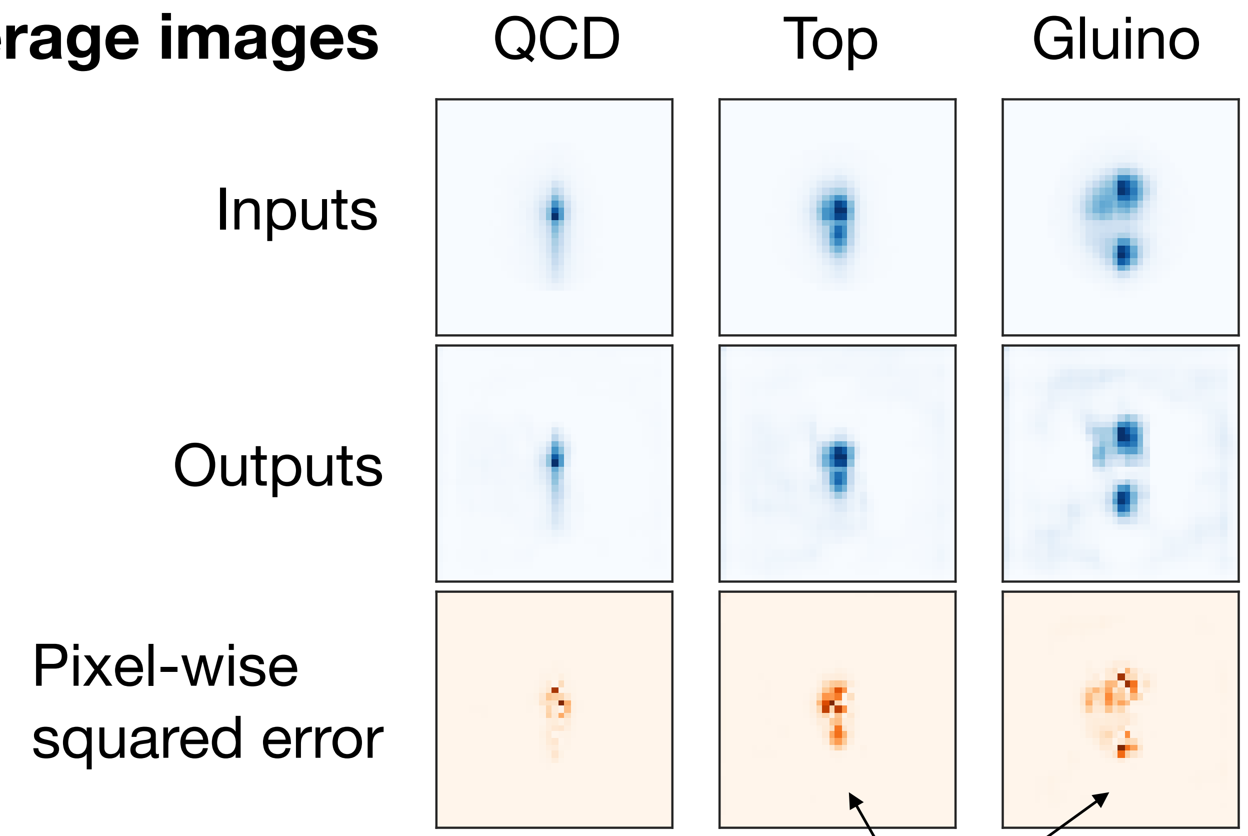
Weakly-supervised mode

Weakly-supervised case with pure background events for training.

Convolutional autoencoder



Average images



More error

Autoencoder learns to reconstruct the QCD background.

Autoencoder fails to reconstruct the signals.

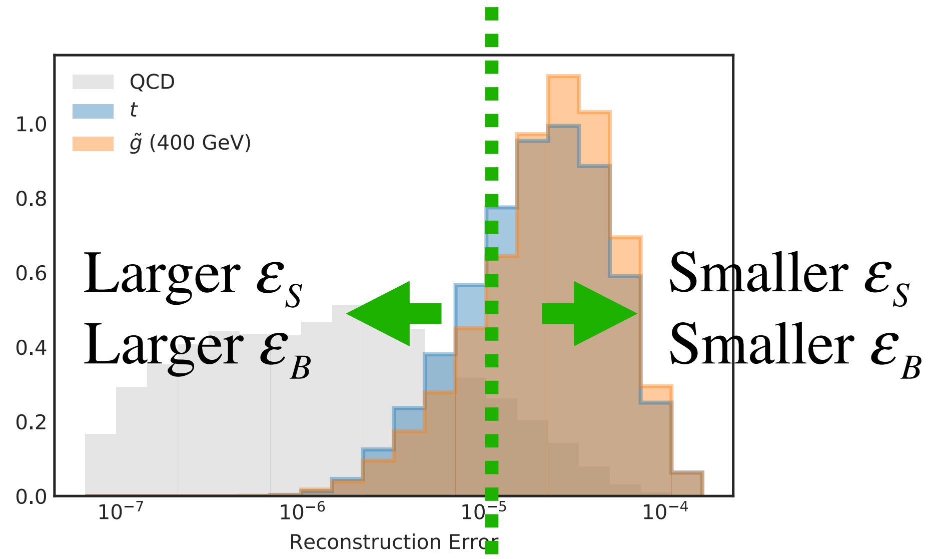
Reconstruction error is used as an anomaly threshold.

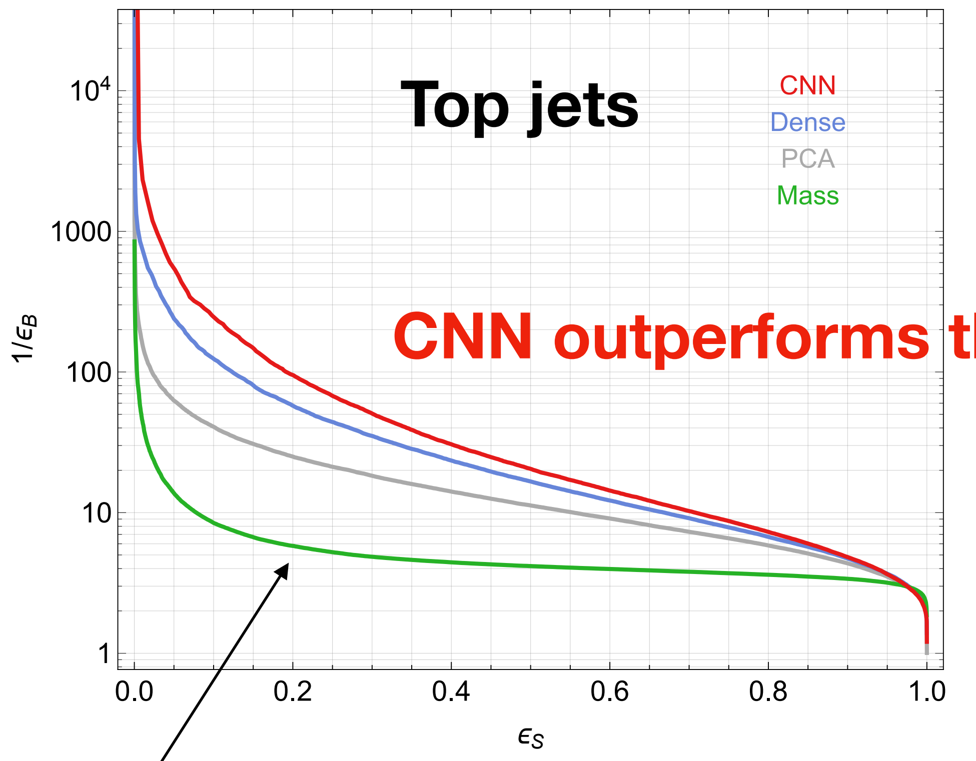
Autoencoder Performance

Performance measure :

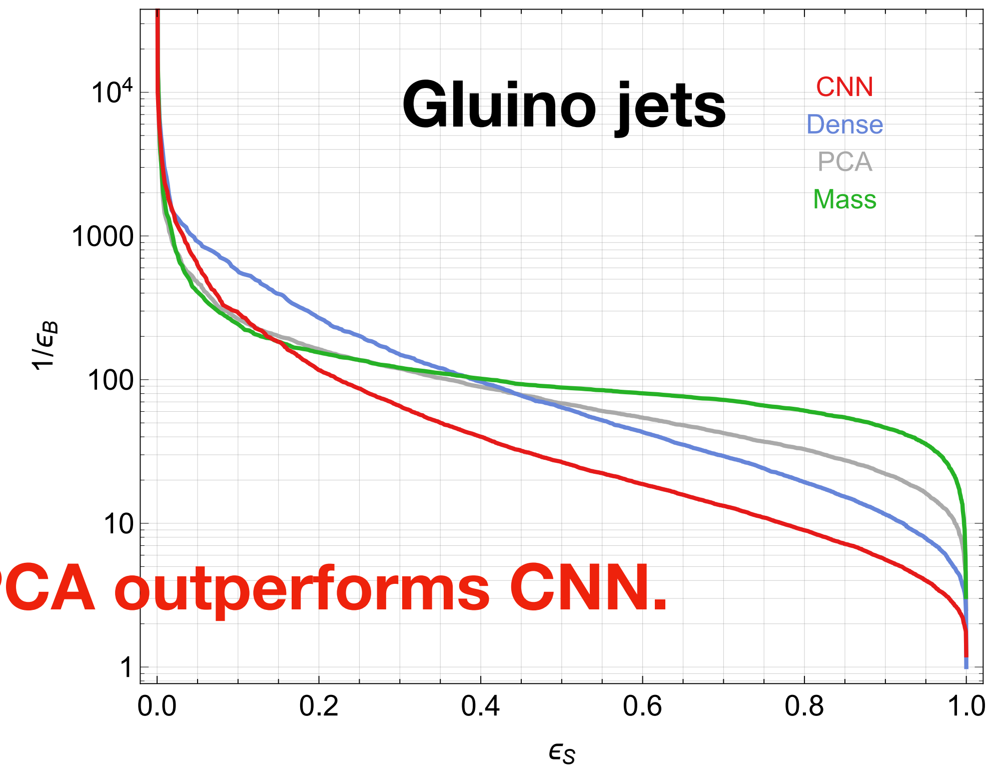
$$\varepsilon_S = \frac{\text{(Correctly classified into signals)}}{\text{(Total number of signal jets)}}$$

$$\varepsilon_B = \frac{\text{(Misclassified into signals)}}{\text{(Total number of backgrounds)}}$$





CNN outperforms the others.



PCA outperforms CNN.

Jet mass as anomaly threshold

For gluino jets, PCA ROC curve approaches jet mass ROC curve, suggesting PCA reconstruction error is highly correlated with jet mass.

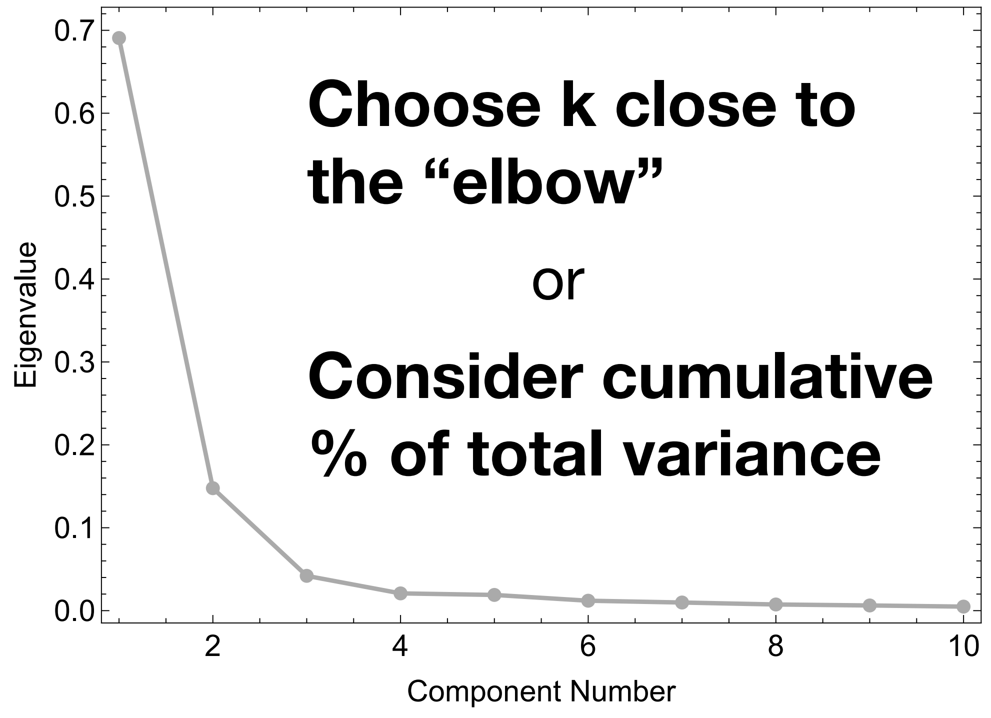
Choosing the Latent Dimension k

Too small k	➡	Autoencoder cannot capture all the features.
Too large k	➡	Autoencoder approaches trivial representation.

Optimizing the latent dimension using various signals is **NOT** a good idea.

Instead, we use the number of principal components in PCA and reconstruction error.

Amount of variance ("scree plot") :

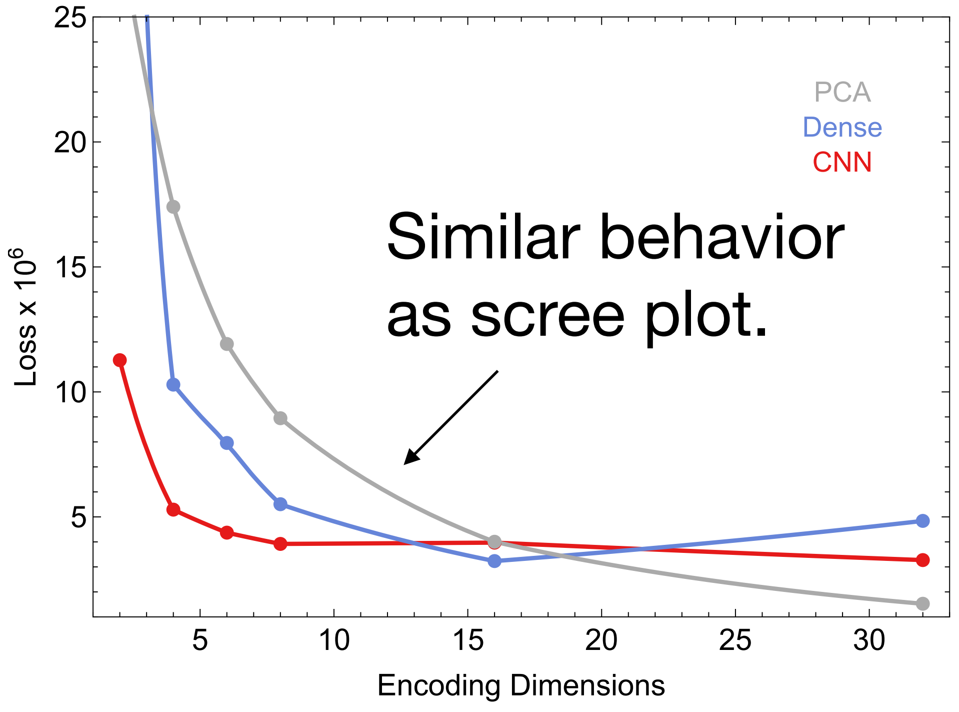


Choose k close to the "elbow"

or

Consider cumulative % of total variance

Reconstruction error :



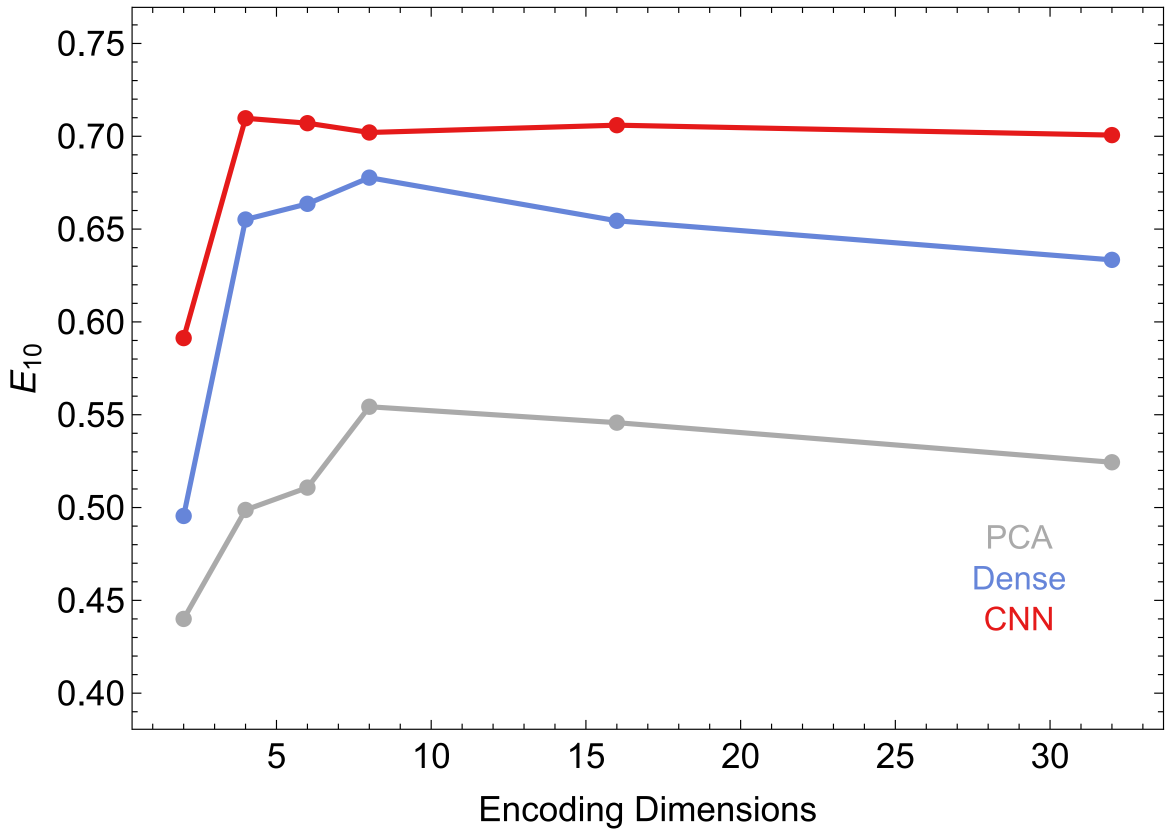
Similar behavior as scree plot.

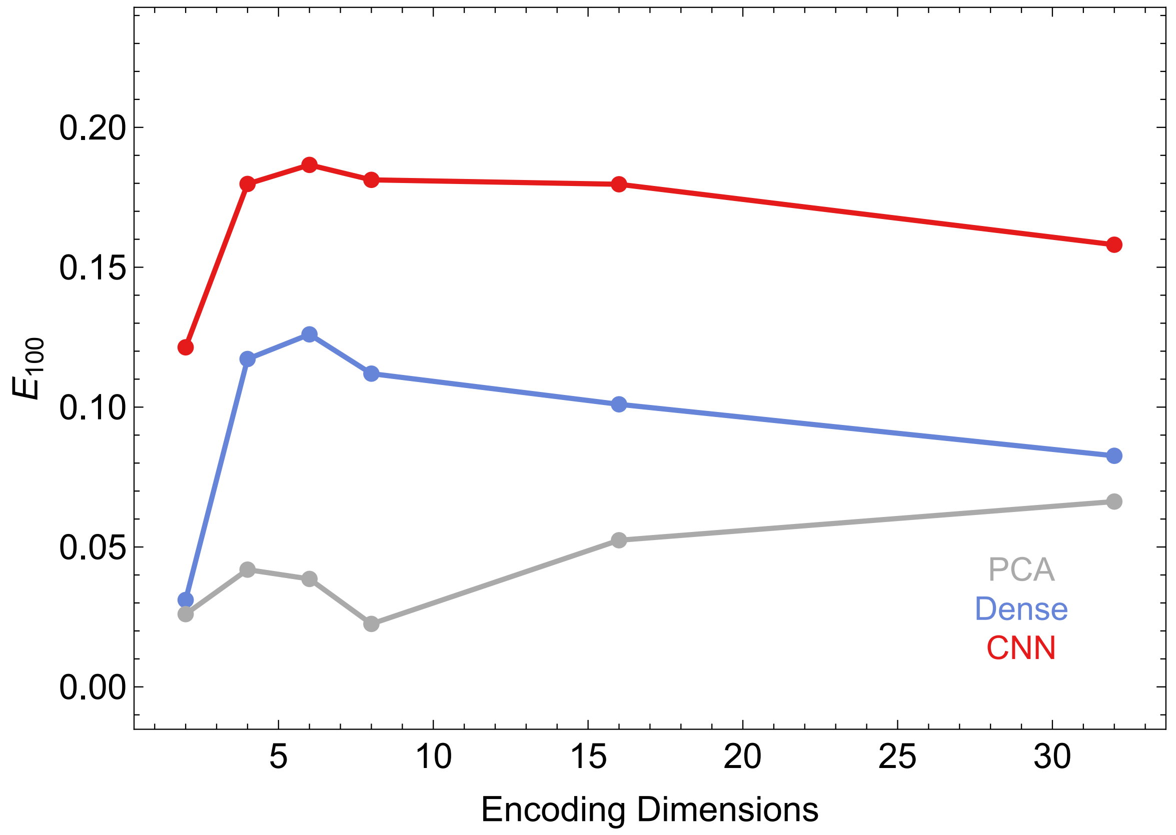
We choose k = 6.

Choosing the Latent Dimension k

Let’s examine our choice by looking at the top signal.

$E_{10,\,100}$: the signal efficiency at 90% and 99% background rejection





Each dot corresponds to the average of 5 independent training runs.

Autoencoder performance plateaus around k = 6.

Robustness with Other Monte Carlo

Autoencoder really does not learn artifacts special to a Monte Carlo?

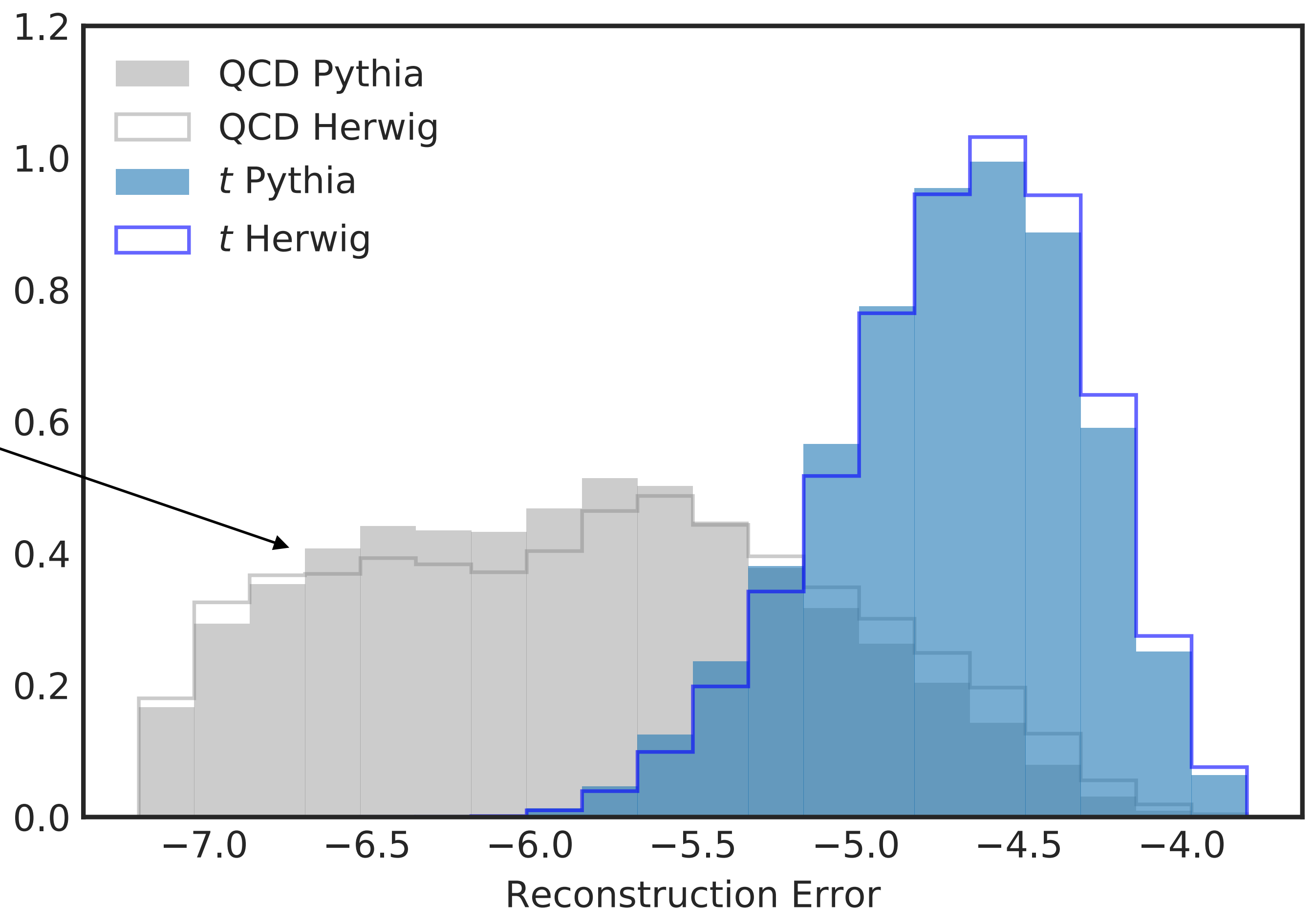
One possible check :

Evaluate autoencoder (trained on PYTHIA samples) on jet samples produced with HERWIG.

Comparison of reconstruction error (top jets, CNN)

The differences are small.

Separation between background and anomaly is preserved.



Autoencoder probably learns fundamental jet features.

Unsupervised mode

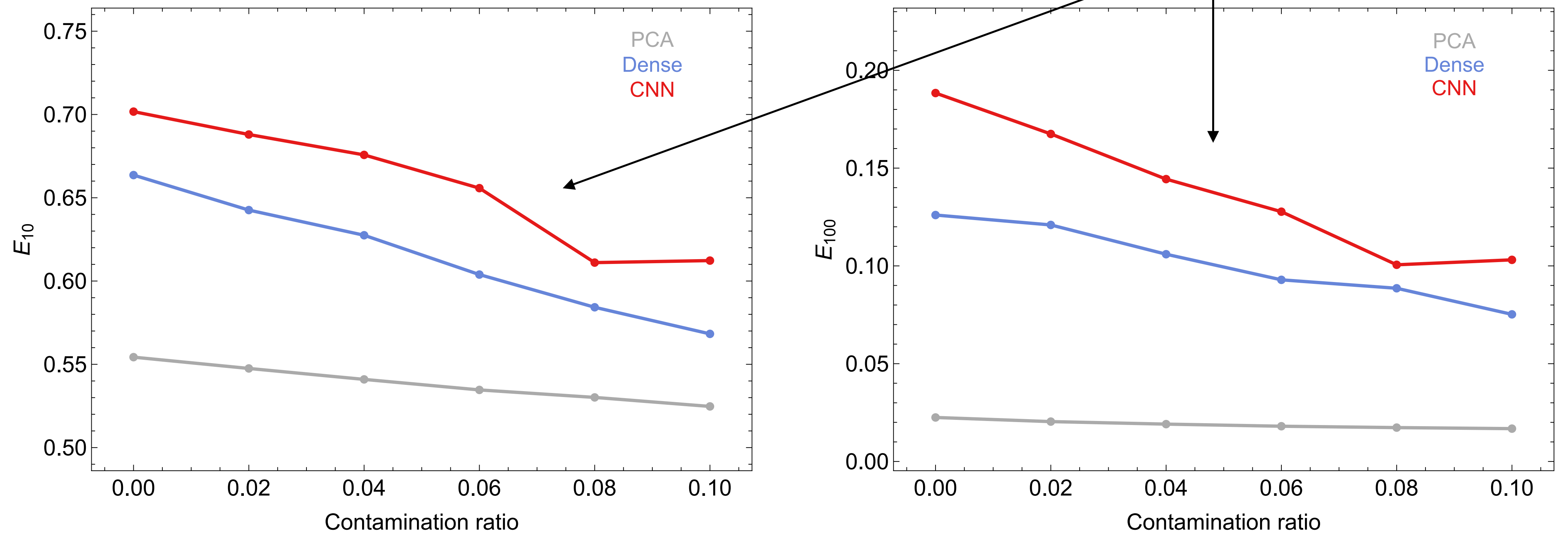
A much more exciting possibility is to train autoencoder on actual data (which may contain some amount of signals).

Train autoencoder on a sample of backgrounds contaminated by a small fraction of signal events.

➡ **Autoencoder performance is remarkably stable against signal contamination.**

Reduction is not dramatic !

Top jets for anomalous events



Correlation with Jet Mass

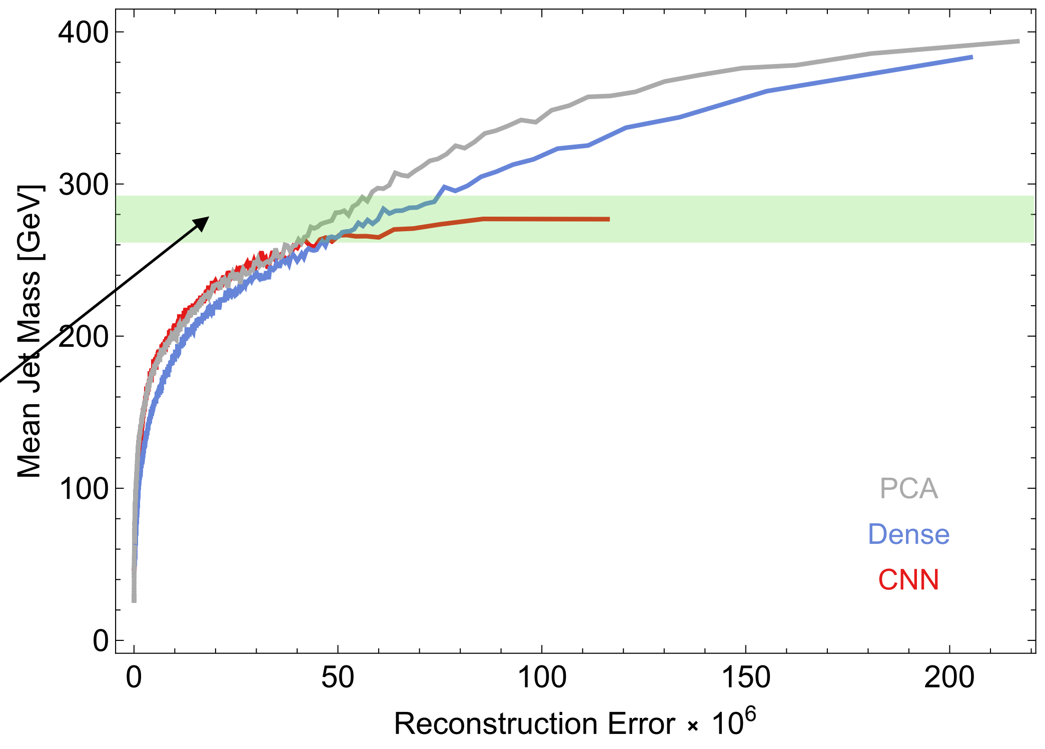
In actual new physics searches, we look for subtle signals …

It's more powerful to combine autoencoder with another variable such as jet mass.

Cut hard on reconstruction error to clean out the QCD background and look for a bump in jet mass distribution.

Reconstruction error should not be correlated with jet mass.

Mean jet mass in bins of reco error for the QCD background

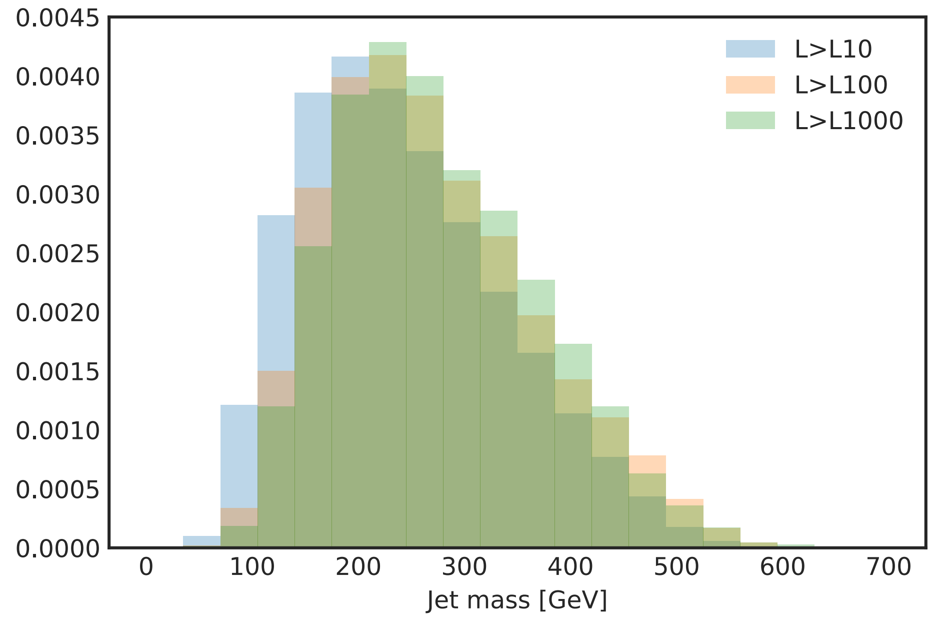


For **PCA and dense**, reco error is correlated with jet mass.

Jet mass distribution is stable against cutting on **CNN loss.**

Correlation with Jet Mass

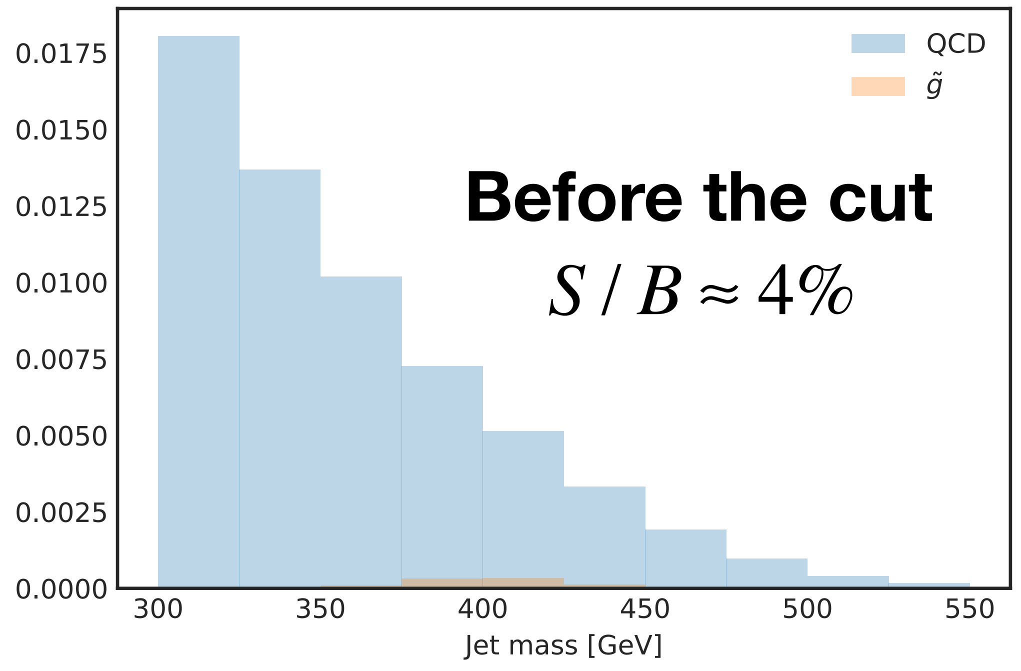
Jet mass distributions after cuts on CNN loss

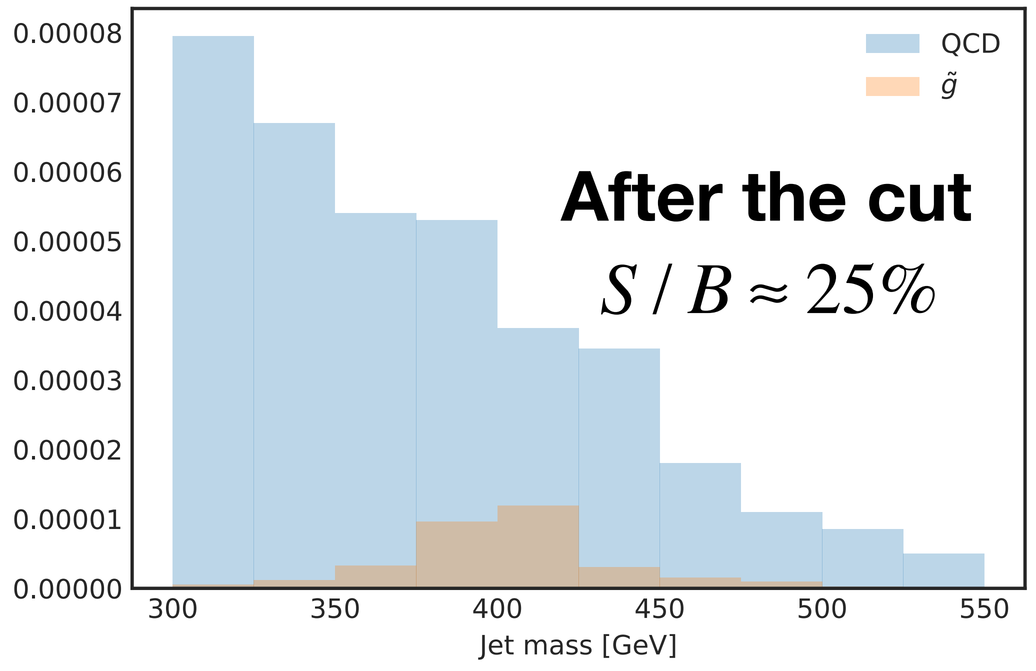


Reduce the QCD background by a factor of 10, 100 and 1000.

Convolutional autoencoder is useful for a bump hunt in jet mass above 300 GeV.

Jet mass histograms normalized to LO gluino and QCD cross sections





Comments on "QCD or What?"

T. Heimel, G. Kasieczka, T. Plehn, J. Thompson, arXiv:1808.08979 [hep-ph].

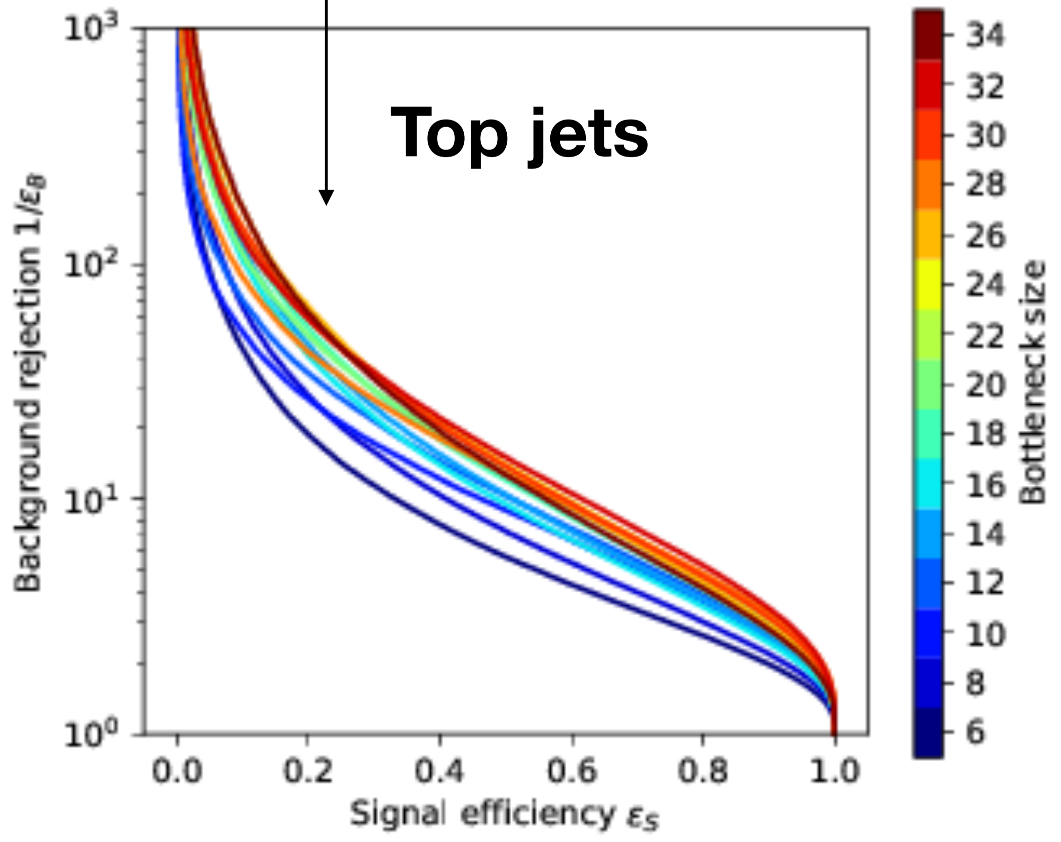
They also consider anomaly detection through autoencoder.

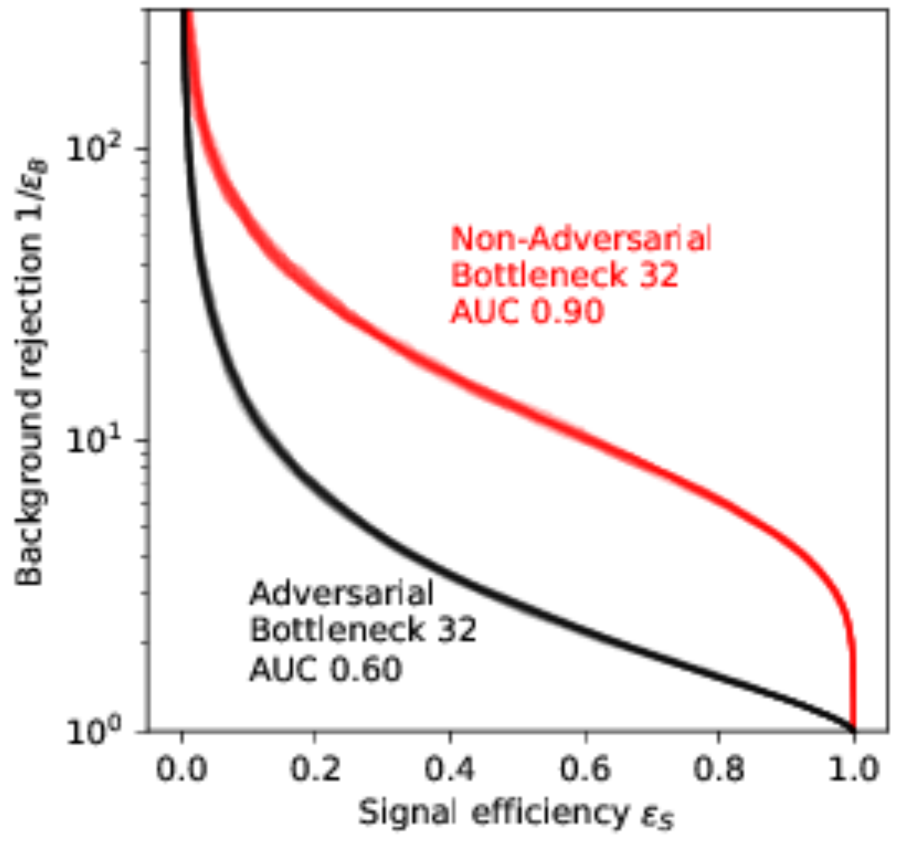
Signal jets : top jets, scalar decay to jets, dark showers

Performance is comparable.

$$pp \to (\phi \to aa \to c\bar{c}\ c\bar{c}) + \text{jets}$$

$$m_a = 4\ \text{GeV} \qquad m_\phi = m_t = 175\ \text{GeV}$$





Comments on "QCD or What?"

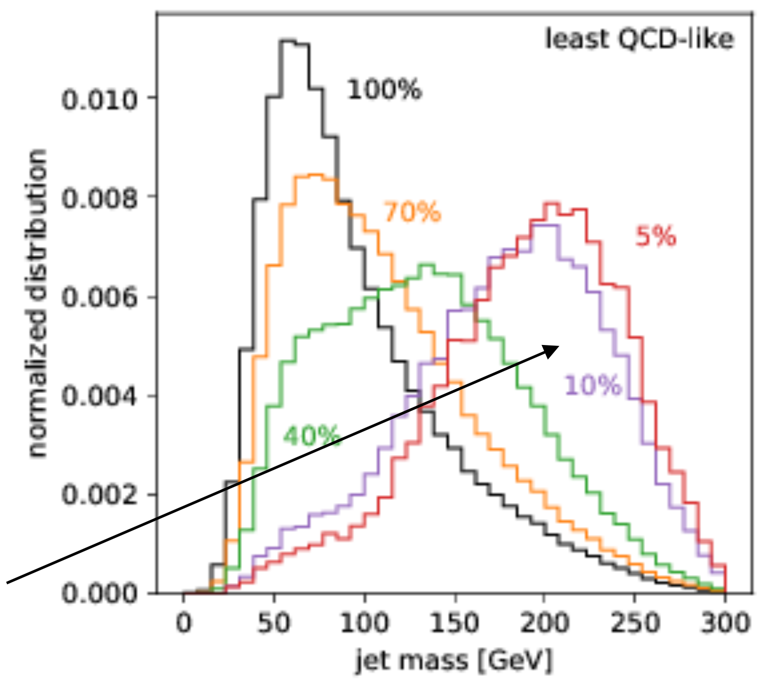
Correlation with jet mass

They take an alternative approach using adversarial networks.

Additional adversary tries to extract jet mass from autoencoder output. $\leftrightarrow$ Autoencoder wants the adversary to be as unsuccessful as possible.

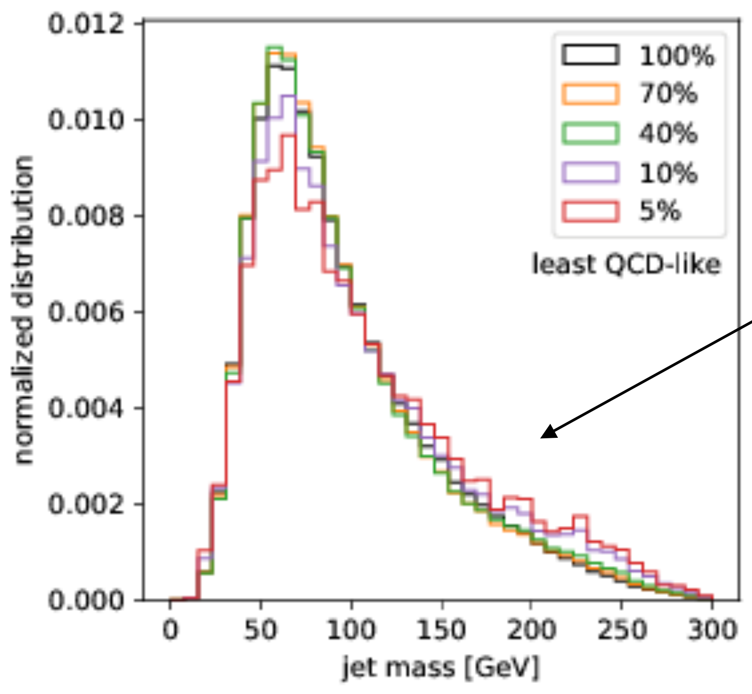
Autoencoder will avoid all information on jet mass.

Non-adversarial



Fake peak

Adversarial



Flatten

Summary

✓ Autoencoder learns to map background events back to themselves but fails to reconstruct signals that it has never encountered before.

✓ Reconstruction error is used as an anomaly threshold.

✓ Autoencoder performance is stable against signal contamination which enables us to train autoencoder on actual data.

✓ Jet mass distribution is stable against cutting on CNN loss and convolutional autoencoder is useful for a bump hunt in jet mass.

✓ Thresholding on reco error gives a significant improvement of S/B.

Future directions

✓ Testing out autoencoder on other signals.
(Other numbers of subjets, non-resonant particles, …)

✓ Training autoencoder to flag entire events as anomalous, instead of just individual fat jets.

✓ Trying other autoencoder architectures on the market to improve the performance.

✓ Understanding what the latent space actually learns.
(Jet mass? N-subjettiness?) …

Autoencoder is a powerful new method to search for any signal of new physics without prejudice !

Thank you.

Backup Material

What is Machine Learning?

Machine learning : technique to give computer systems the ability to learn with data without being explicitly programmed.

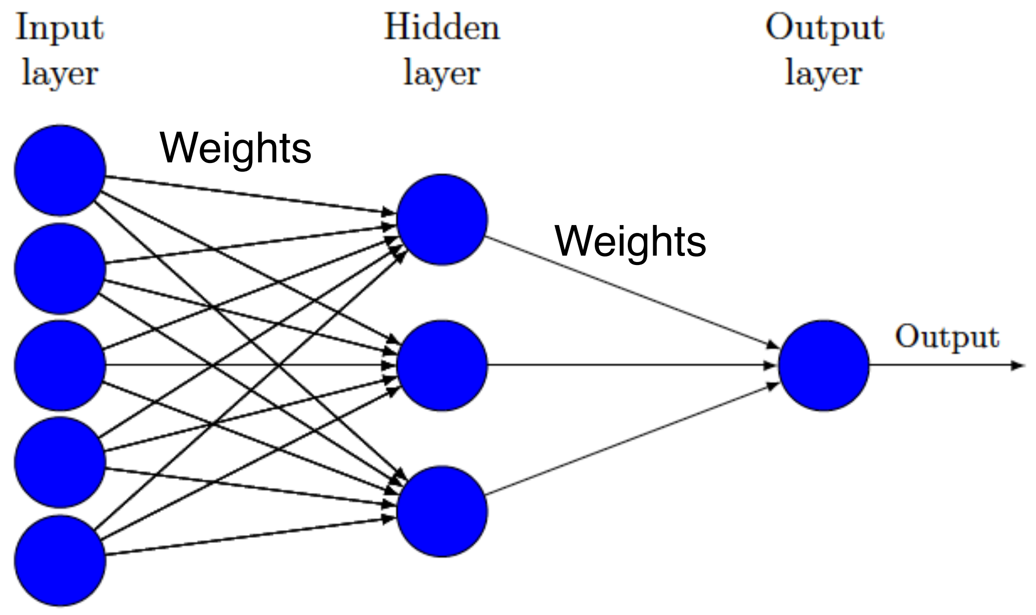
Machine can learn the feature of data which human has not realized !

Neural Networks

✓ Powerful machine learning-based techniques used to solve many real-world problems

✓ Modeled loosely after the human brain

✓ Containing weights between neurons that are tuned by learning from data



Networks contain multiple hidden layers → **Deep learning**

What is Machine Learning?

The goal of training is to minimize loss function :

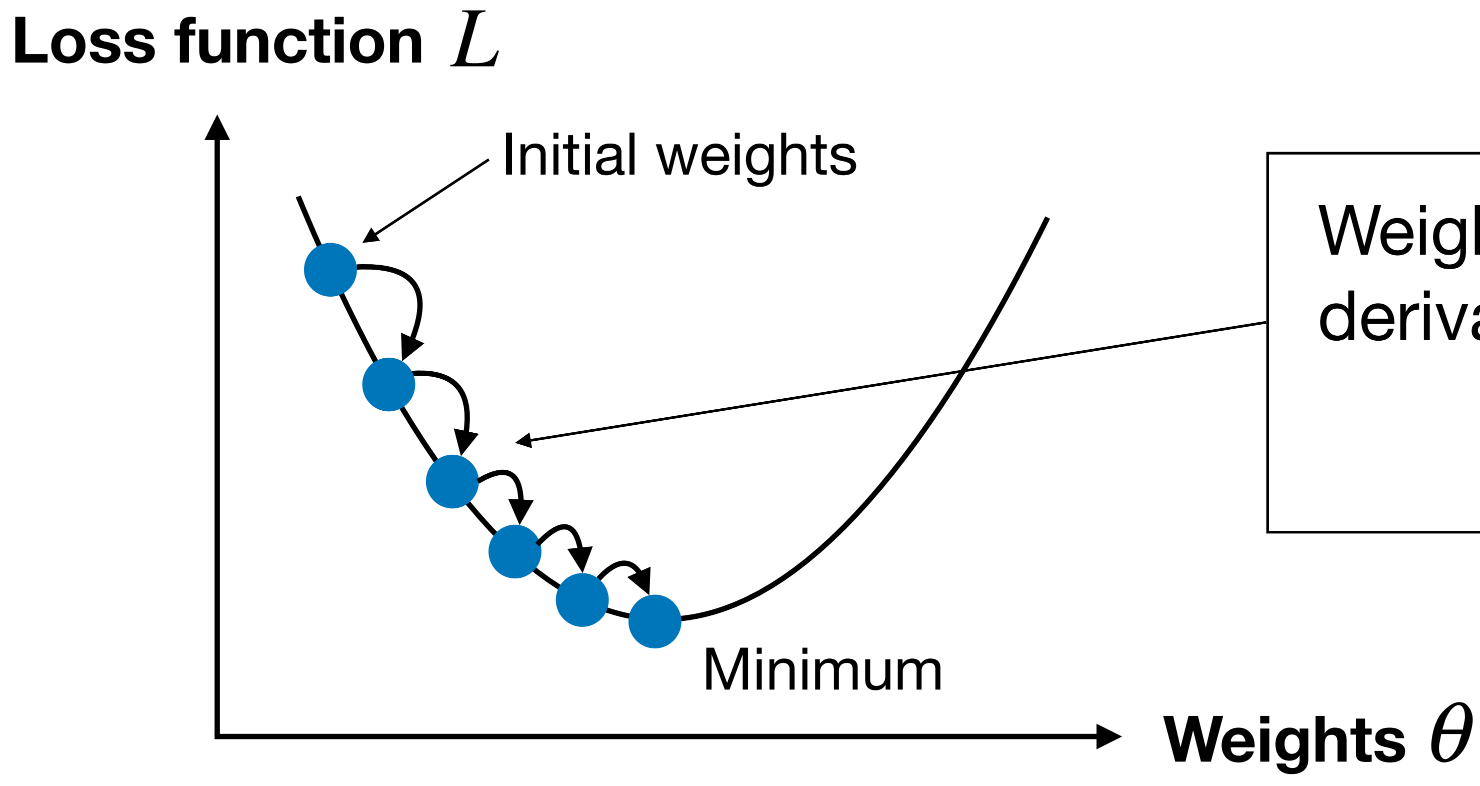
$$L = \sum_i f(p(\theta, x_i), y_i)$$

$p(\theta, x_i)$: Prediction θ : Weights

x_i : Input y_i : Target value of example i

Mean squared error (MSE) : $f(p, y) = (p - y)^2$

Cross entropy : $f(p, y) = -(y \log p + (1 - y)\log(1 - p))$



Weights are updated according to derivative of loss function :

$$\Delta\theta = -\eta \nabla L$$

Learning rate

Keras Codes

- **Simple autoencoder**

```
input_img = Input(shape=(37*37,))
layer = Dense(32, activation='relu')(input_img)
encoded = Dense(6, activation='relu')(layer)

layer = Dense(32, activation='relu')(encoded)
layer = Dense(37*37, activation='relu')(layer)
decoded=Activation('softmax')(layer)

autoencoder=Model(input_img,decoded)
autoencoder.compile(loss=keras.losses.mean_squared_error,
              optimizer=keras.optimizers.Adam())
```

Keras Codes

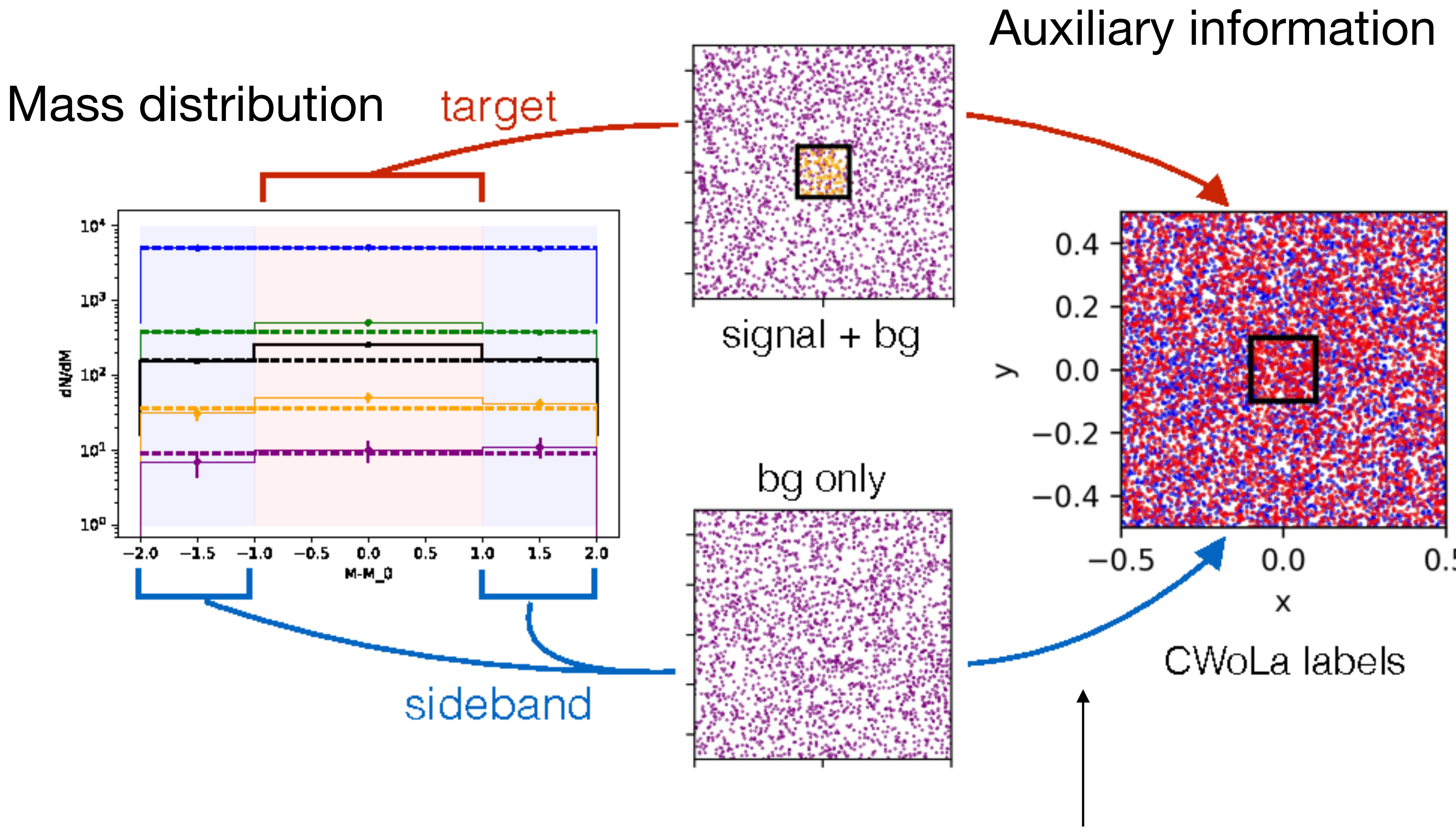
- **Convolutional autoencoder**

```
input_img=Input(shape= (40, 40, 1))

layer=input_img
layer=Conv2D(128, kernel_size=(3, 3),
             activation='relu',padding='same')(layer)
layer=MaxPooling2D(pool_size=(2, 2),padding='same')(layer)
layer=Conv2D(128, kernel_size=(3, 3),
             activation='relu',padding='same')(layer)
layer=MaxPooling2D(pool_size=(2, 2),padding='same')(layer)
layer=Conv2D(128, kernel_size=(3, 3),
             activation='relu',padding='same')(layer)
layer=Flatten()(layer)
layer=Dense(32, activation='relu')(layer)
layer=Dense(6)(layer)
encoded=layer

layer=Dense(32, activation='relu')(encoded)
layer=Dense(12800, activation='relu')(layer)
layer=Reshape((10,10,128))(layer)
layer=Conv2D(128, kernel_size=(3, 3),
             activation='relu',padding='same')(layer)
layer=UpSampling2D((2,2))(layer)
layer=Conv2D(128, kernel_size=(3, 3),
             activation='relu',padding='same')(layer)
layer=UpSampling2D((2,2))(layer)
layer=Conv2D(1, kernel_size=(3, 3),padding='same')(layer)
layer=Reshape((1,1600))(layer)
```

CWoLa Hunting

J. Collins, K. Howe, B. Nachman, arXiv:1805.02664 [hep-ph].

Another approach to anomaly detection to extend bump hunt with machine learning.

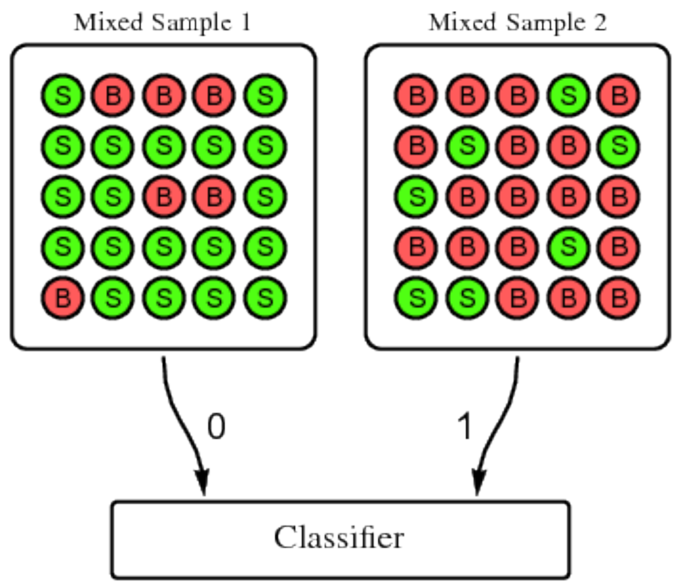


Toy model

$$Y = (x, y)$$

Background : $-0.5 < x < 0.5$, $-0.5 < y < 0.5$

Signal : $-w/2 < x < w/2$, $-w/2 < y < w/2$



Classification without labels (CWoLa)

A classifier is trained to distinguish statistical mixtures of classes.

Metodiev, Nachman, Thaler