Politecnico
di Bari

DIPARTIMENTO
INTERATENEO
DI FISICA

Dottorato in Fisica – XXXIX ciclo - 2024

Machine Learning techniques for particle physics

Federica Maria Simone - federica.simone@poliba.it

Unsupervised training
Anomaly Detection
Autoencoders

Anomaly Detection: introduction

What are anomalies/outliers?

- The set of data points that are considerably different than the remainder of the data

Variants of Anomaly/Outlier Detection Problems

- Given a database D, find all the data points x ∈ D with anomaly scores greater than some threshold t

- Given a database D, find all the data points x ∈ D having the top-n largest anomaly scores f(x)

- Given a database D, containing mostly normal (but unlabeled) data points, and a test point x, compute the anomaly score of x with respect to D

Applications:

- Credit card fraud detection, telecommunication fraud detection, network intrusion detection, fault detection

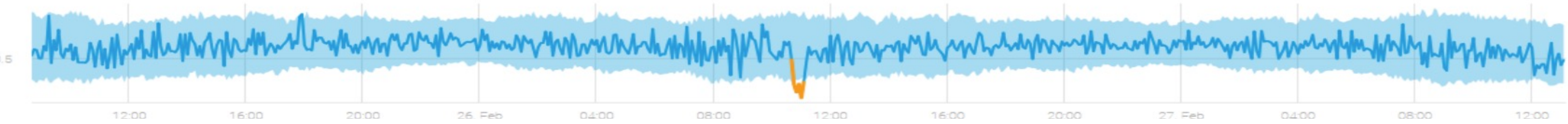
Anomaly Detection: challenges

Challenges

- How many outliers are there in the data?
- Method is unsupervised: validation can be quite challenging (just like for clustering)
- Finding needle in a haystack

Working assumption:

- There are considerably more “normal” observations than “abnormal” observations (outliers/anomalies) in the data



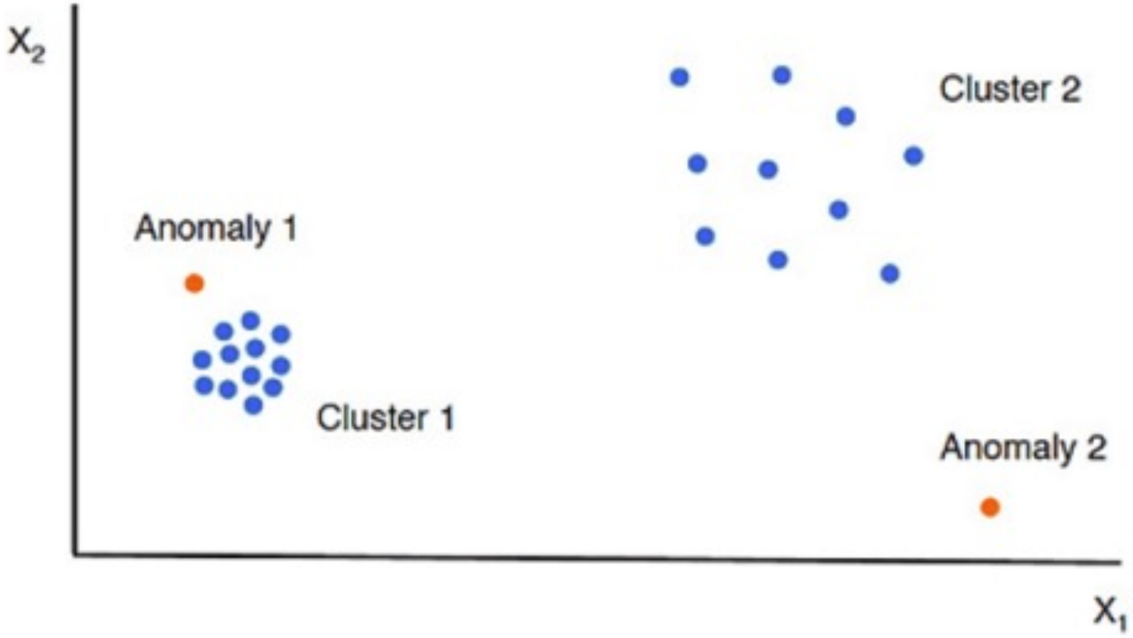
Anomaly Detection workflows

General Steps

- Build a profile of the “normal” behavior
  - Profile can be patterns or summary statistics for the overall population
- Use the “normal” profile to detect anomalies
  - Anomalies are observations whose characteristics differ significantly from the normal profile

Types of anomaly detection schemes

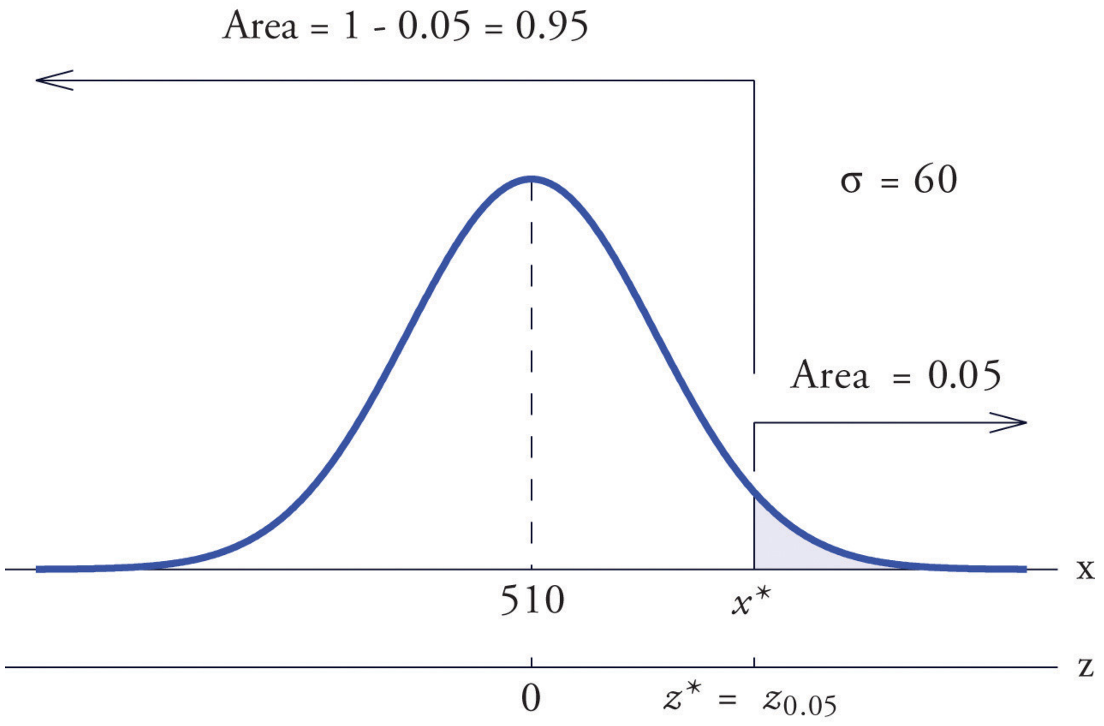
- Statistical-based
- Distance-based
- Model-based



Statistical approach: statistical test

Assume a parametric model describing the distribution of the data (e.g., normal distribution)

Apply a statistical test that depends on
- Data distribution
- Parameter of distribution (e.g., mean, variance)
- Number of expected outliers (confidence limit)



Statistical approach: likelihood

Assume the **dataset D** contains samples from a mixture of two **probability distributions**:

- **M** (majority distribution)
- **A** (anomalous distribution)

General Approach:

- Initially, assume all the data points belong to M
- Let $L_t(D)$ be the log likelihood of D at time t
- For each point x_t that belongs to M, move it to A
  - Let $L_{t+1}(D)$ be the new log likelihood.
  - Compute the difference, $\Delta = L_t(D) - L_{t+1}(D)$
  - If $\Delta > c$ (some threshold), then x_t is declared as an anomaly and moved permanently from M to A

Statistical approach: likelihood

Data distribution → D = (1 – λ) M + λ A

M is a probability distribution estimated from data
- Can be based on any modeling method (naïve Bayes, maximum entropy, etc)

A is initially assumed to be uniform distribution

Likelihood at time t:

$$L_t(D) = \prod_{i=1}^{N} P_D(x_i) = \left((1-\lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right) \left(\lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right)$$

$$LL_t(D) = |M_t| \log(1-\lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

Limitations of statistical approach:
- Most of the tests are for a single attribute
- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution

Distance-based approaches

Data is represented as a **vector of features**

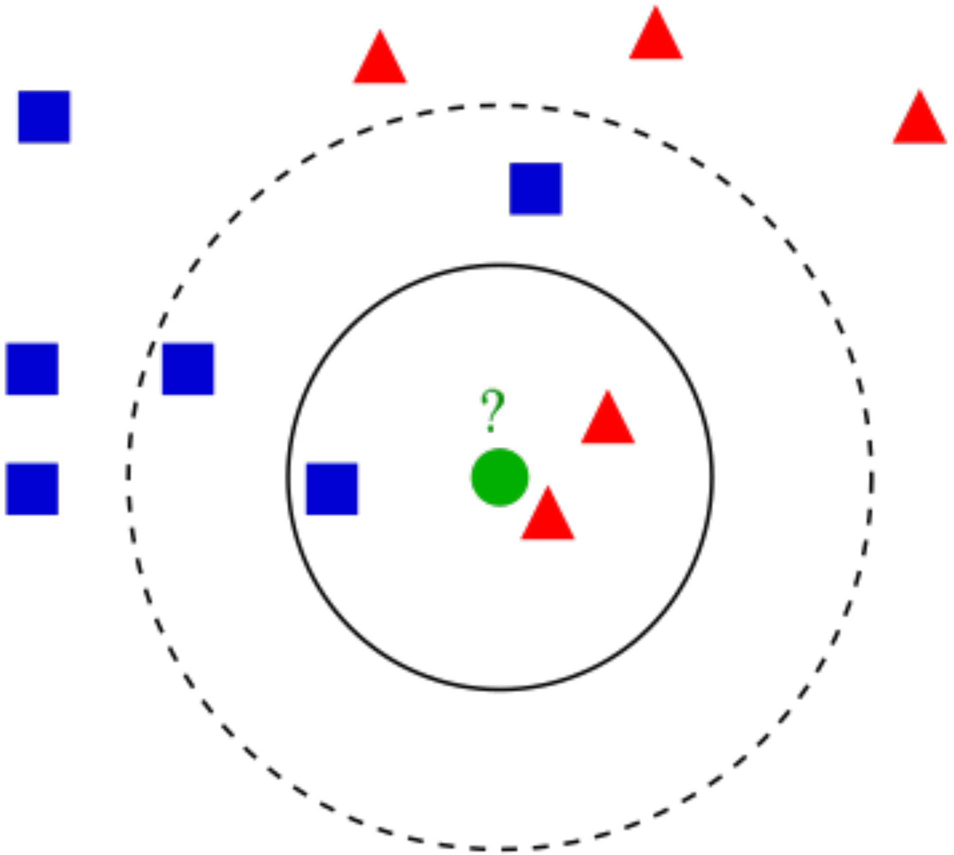
Three major approaches
– Nearest-neighbor based
– Density based
– Clustering based

Nearest-neighbor

Compute the distance between every pair of data points

There are various ways to define outliers:

- Data points for which there are fewer than p neighboring points within a distance D
- The top n data points whose distance to the kth nearest neighbor is greatest
- The top n data points whose average distance to the k nearest neighbors is greatest

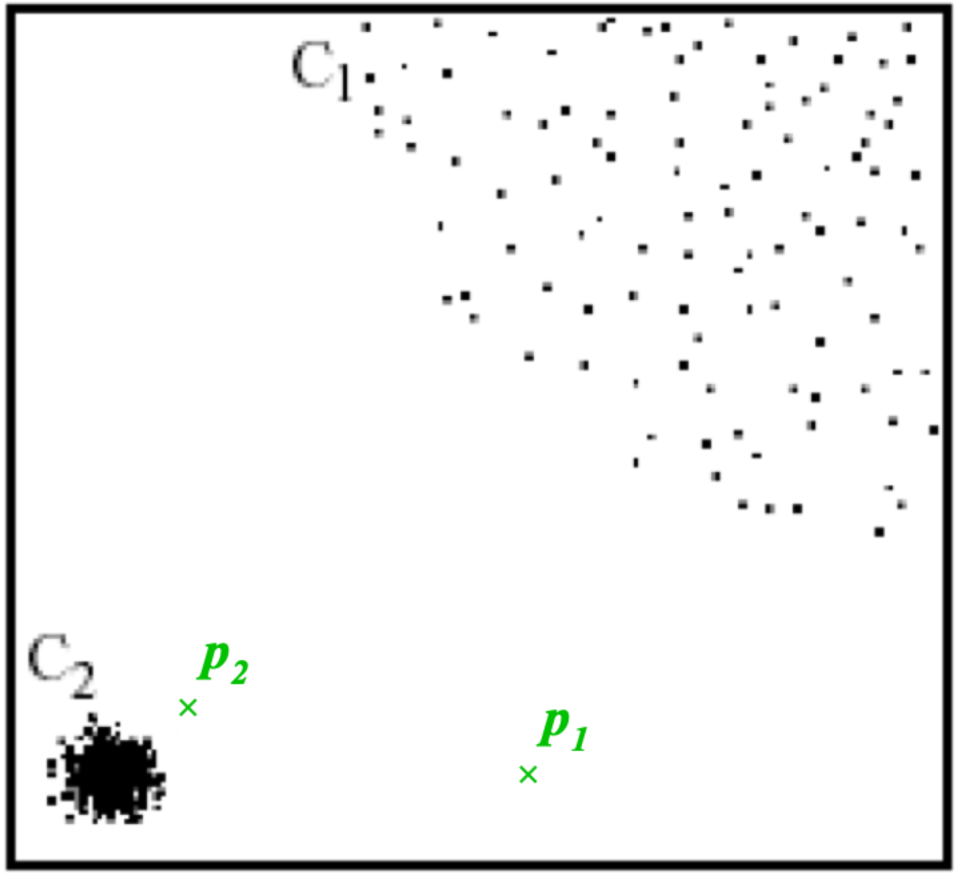


Density

For each point, compute the density of its local neighborhood

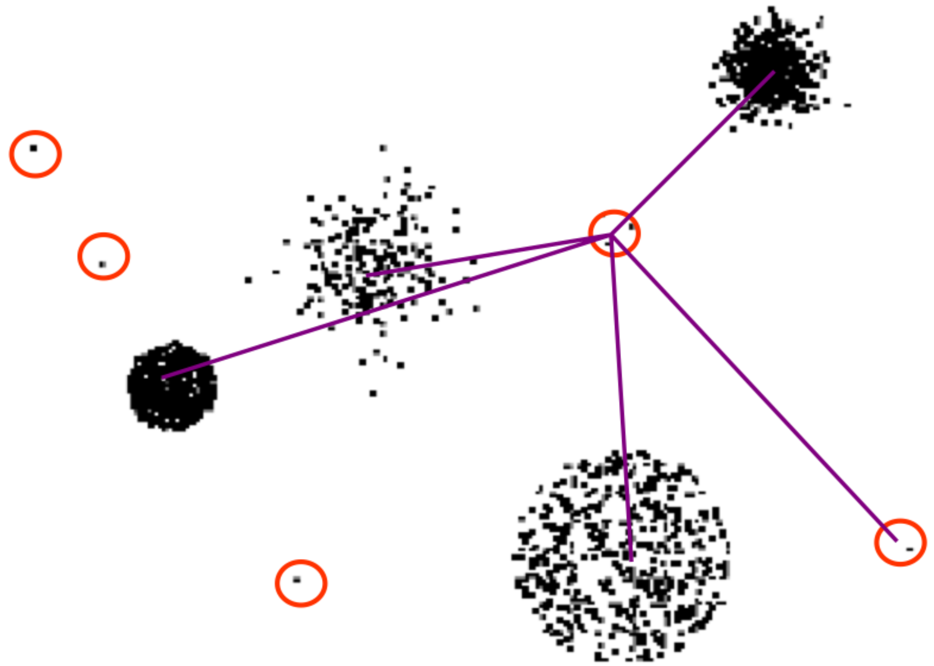
Compute local outlier factor (LOF) of a sample p as the average of the ratios of the density of sample p and the density of its nearest neighbors

Outliers are points with largest LOF value



Clustering

Basic idea:
- Cluster the data into groups of different density
- Choose points in small cluster as candidate outliers
- Compute the distance between candidate points and non-candidate clusters.
    - If candidate points are far from all other non-candidate points, they are outliers



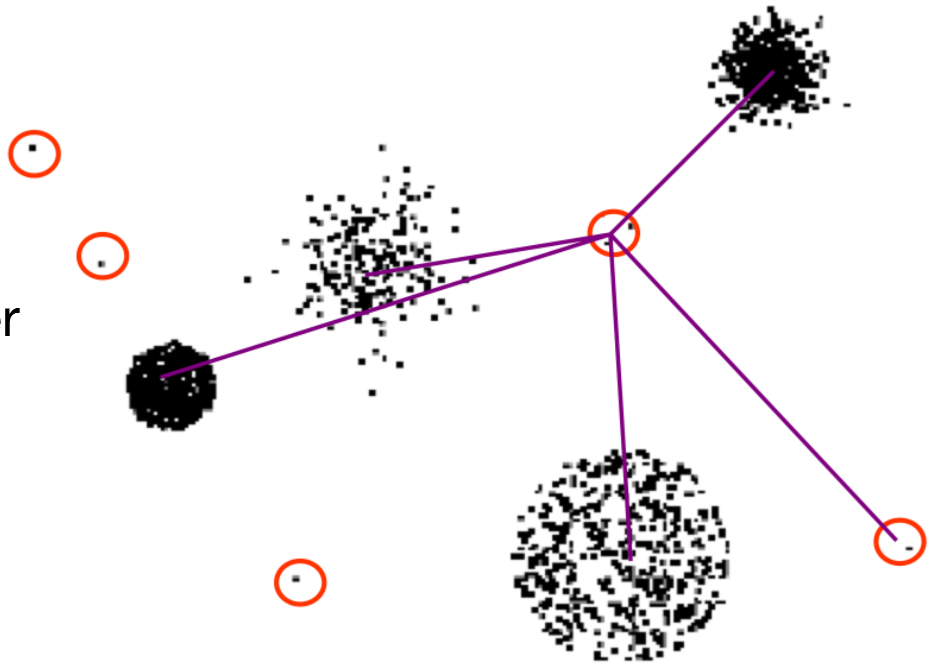
© Tan,Steinbach, Kumar

Unsupervised learning: the k-means clustering algorithm

Data: Input: $\{x^{(1)} \dots x^{(n)}\}$, where $x^{(i)} \in \mathbb{R}^d$
No labels $y^{(i)}$!

Problem: group data into cohesive clusters

"k" is a parameter of the algorithms and indicates the number of clusters



1. Initialize **cluster centroids** $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^d$ randomly.
2. Repeat until convergence: {

   For every i, set
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set
   $$\mu_j := \frac{\sum_{i=1}^n 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^n 1\{c^{(i)} = j\}}.$$

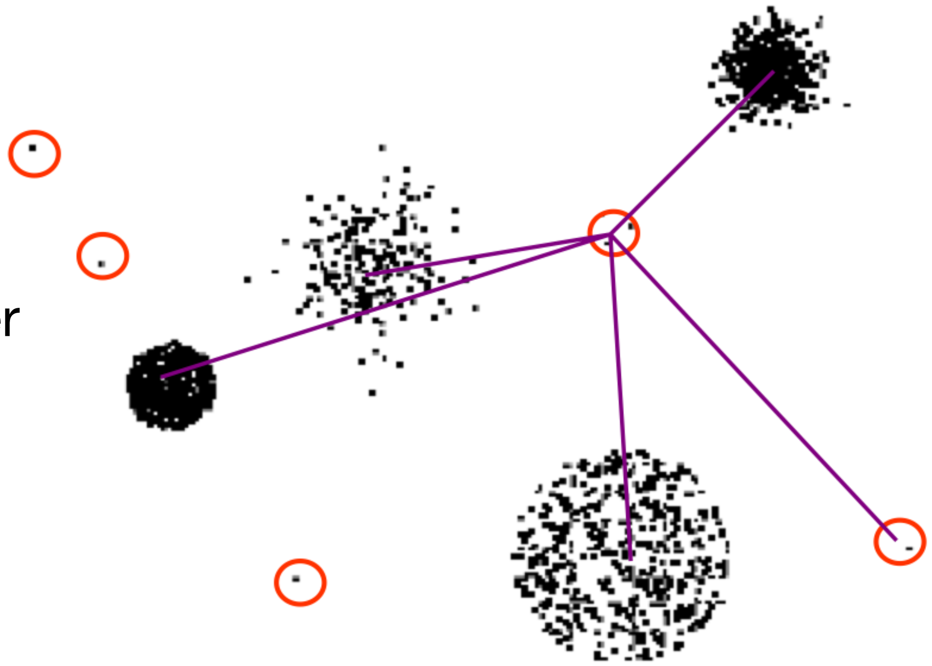
   }

Unsupervised learning: the k-means clustering algorithm

Data: Input: $\{x^{(1)} \dots x^{(n)}\}$, where $x^{(i)} \in \mathbb{R}^d$
No labels $y^{(i)}$!

Problem: group data into cohesive clusters

"k" is a parameter of the algorithms and indicates the number of clusters



1. Initialize **cluster centroids** $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^d$ randomly.
2. Repeat until convergence: {

   For every i, set
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

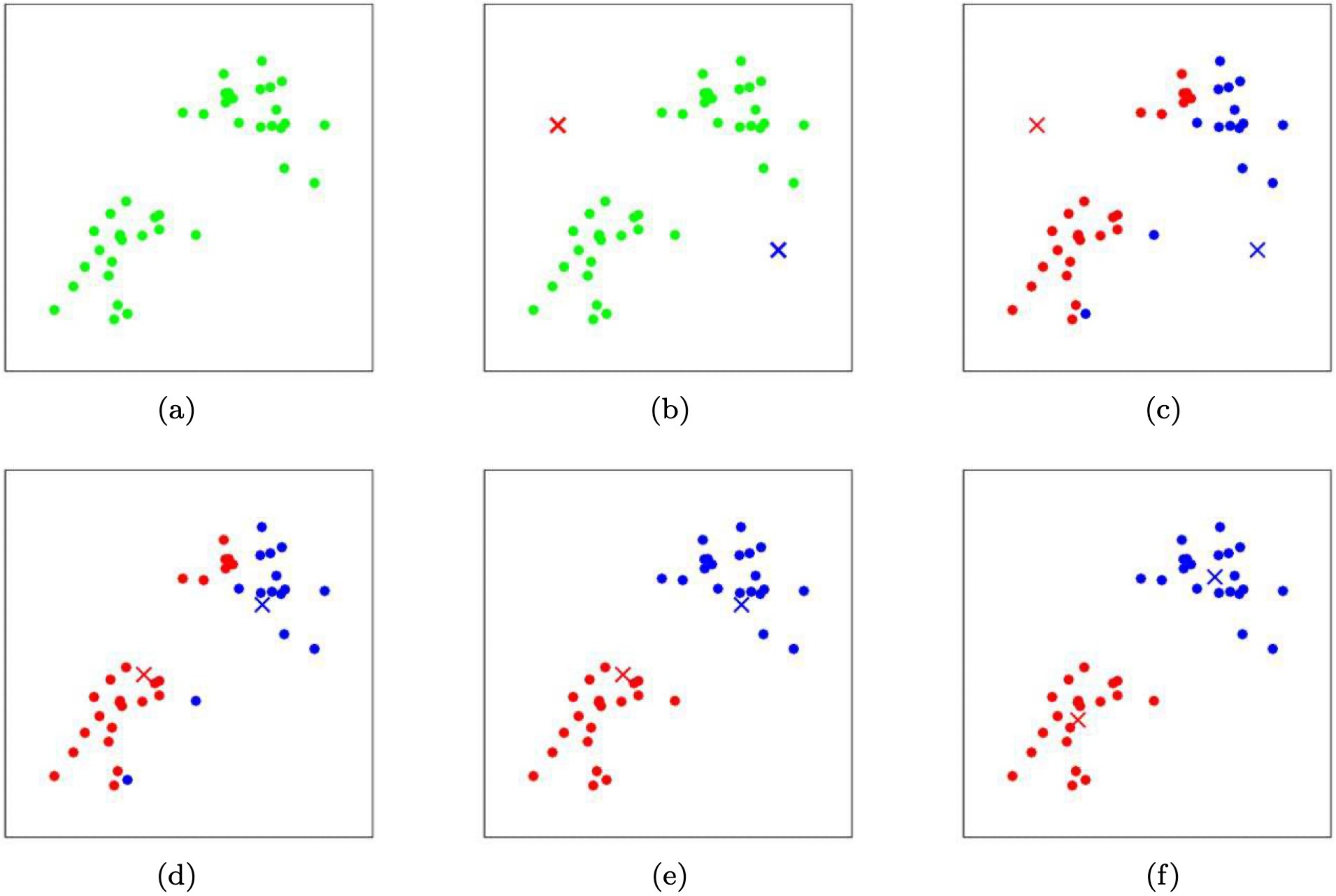
   For each j, set
   $$\mu_j := \frac{\sum_{i=1}^n 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^n 1\{c^{(i)} = j\}}.$$

   }

"Assigning" each training example $x^{(i)}$ to the closest cluster centroid μ_j

Moving each cluster centroid μ_j to the mean of the points assigned to it

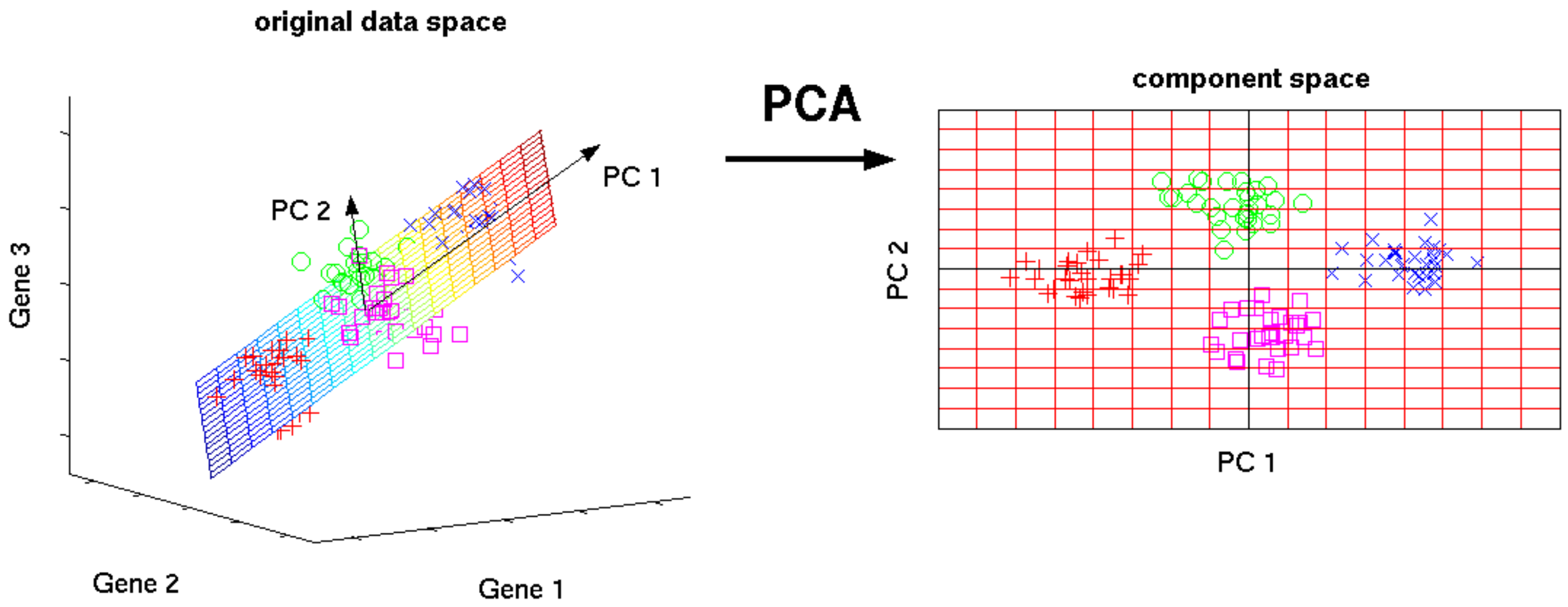
Unsupervised learning: the k-means clustering algorithm



Unsupervised learning: Principal Component Analysis

Data: Input: $\{x^{(1)} \dots x^{(n)}\}$, where $x^{(i)} \in \mathbb{R}^d$
No labels $y^{(i)}$!

Problem: identify the k-dimension subspace (k < d) in which the data approximately lies.
→ Detect correlations between variables and reduce dataset dimentionality!



http://www.nlpca.org/pca_principal_component_analysis.html

Unsupervised learning: Principal Component Analysis

Procedure:

- Normalise features to have mean 0 and variance 1

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \qquad \text{where } \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(x_j^{(i)} - \mu_j)^2$$

- Compute the covariance matrix of the data

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} x^{(i)} {x^{(i)}}^T$$

- Compute the top k eigenvectors of the matrix Σ $(\widehat{u_1}, \widehat{u_2} \ldots \widehat{u_k})$

- $(\widehat{u_1}, \widehat{u_2} \ldots \widehat{u_k})$ is the base of our k-dimensional subspace with respect to which the variance of the datapoints is maximum

- Output: new k-dimentional representation of the input data

$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k.$$

Unsupervised learning: Principal Component Analysis

Example:

- d = 2, k = 1

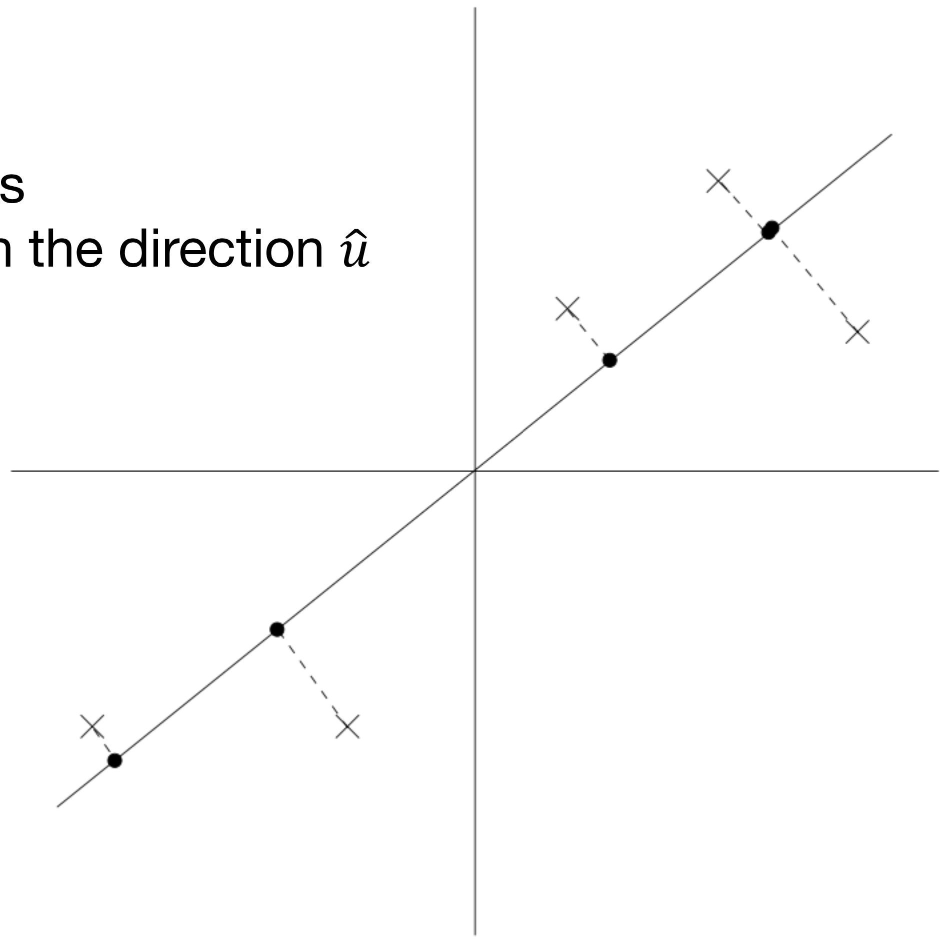
Crosses indicate data points

Unsupervised learning: Principal Component Analysis

Example:

- d = 2, k = 1

Crosses indicate data points
Dots indicate projections on the direction $\hat{u}$



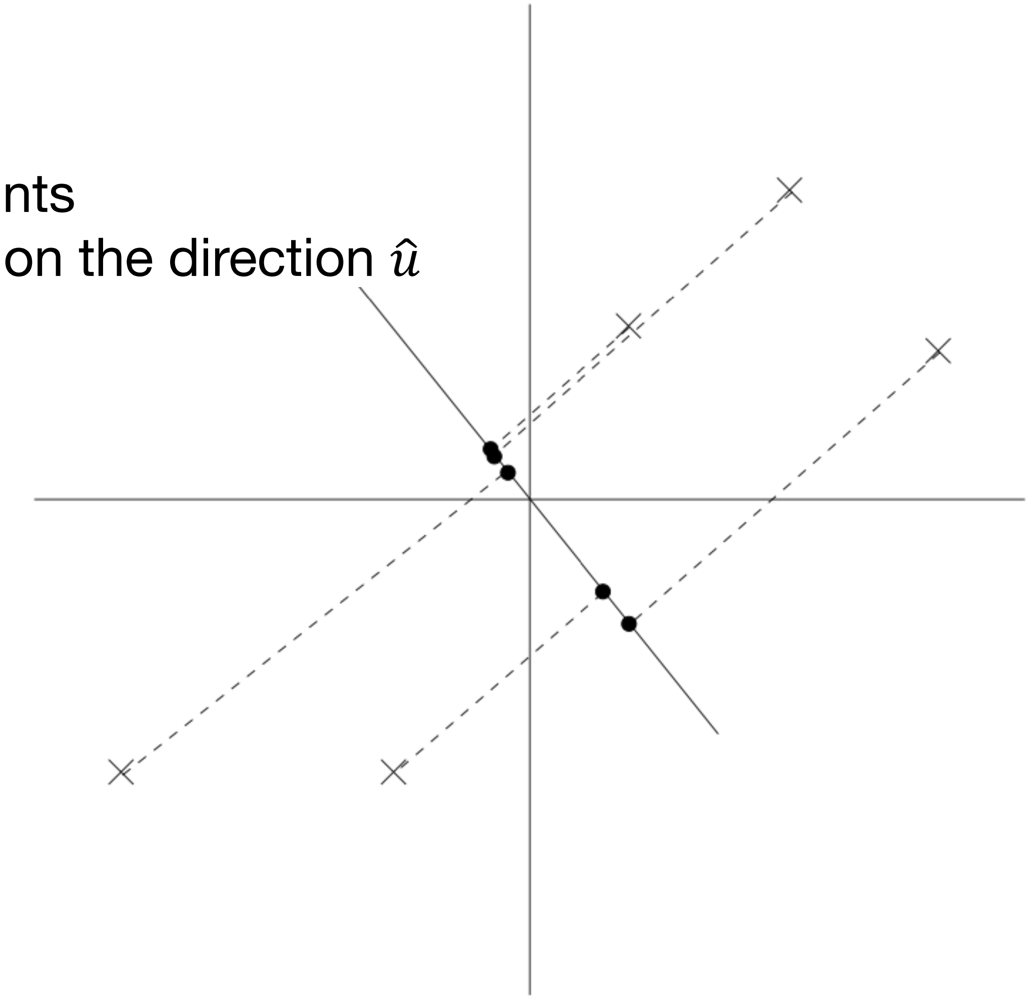
Dots have large variance

Unsupervised learning: Principal Component Analysis

Example:

- d = 2, k = 1

Crosses indicate data points
Dots indicate projections on the direction $\hat{u}$

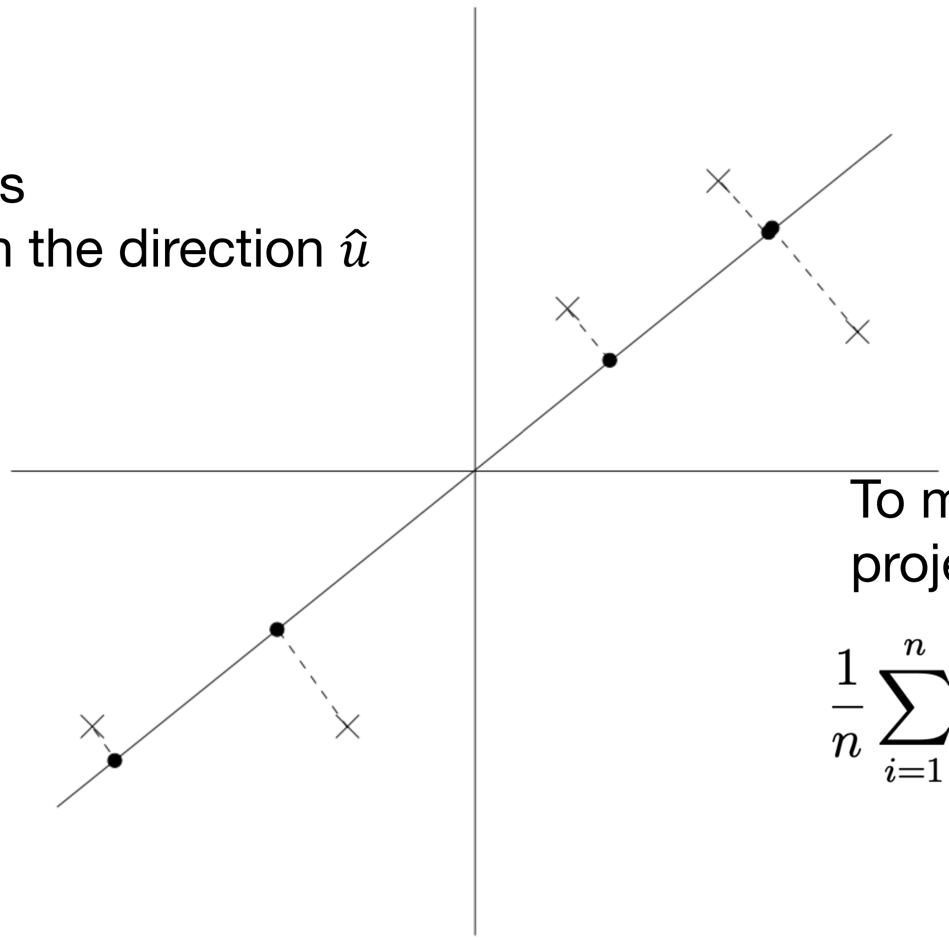


Dots have small variance

Unsupervised learning: Principal Component Analysis

Example:
- d = 2, k = 1

Crosses indicate data points
Dots indicate projections on the direction $\hat{u}$



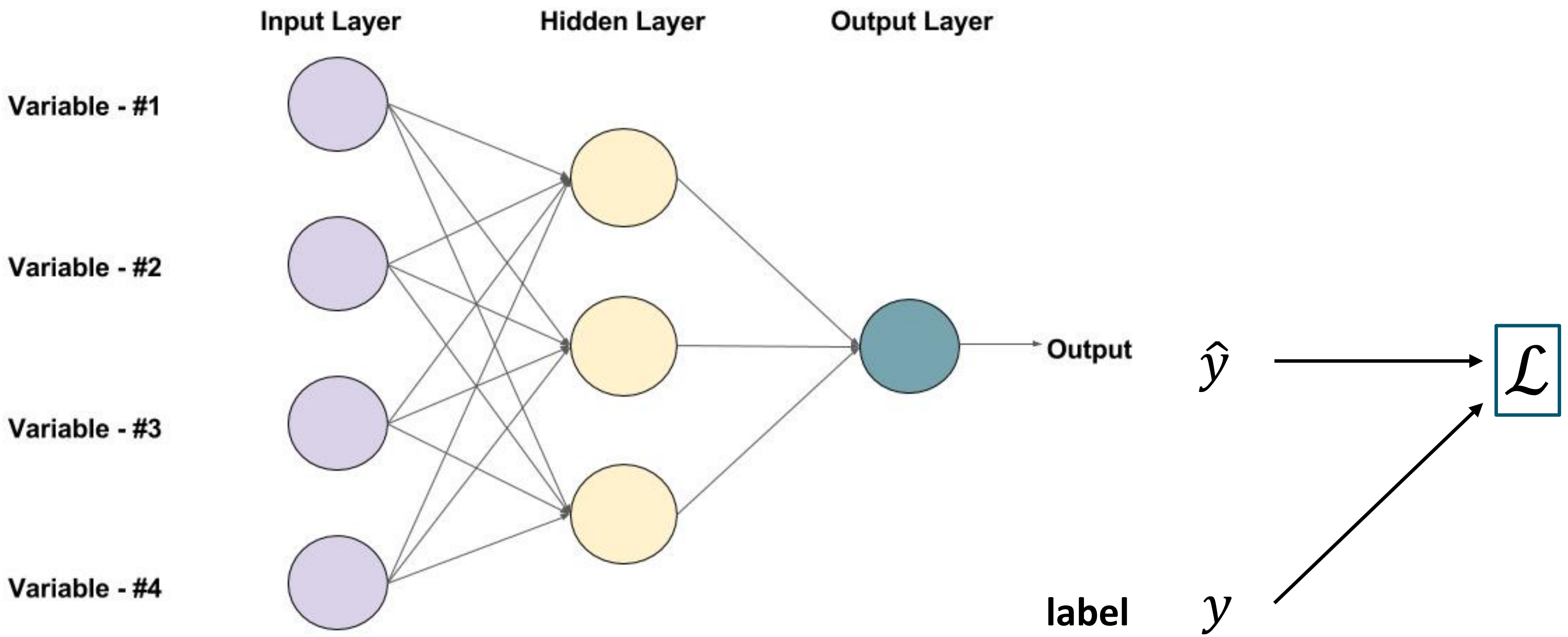
To maximise the variance of the projections, we need to maximise

$$\frac{1}{n}\sum_{i=1}^{n}({x^{(i)}}^T u)^2 = \frac{1}{n}\sum_{i=1}^{n} u^T x^{(i)} {x^{(i)}}^T u$$
$$= u^T \left(\frac{1}{n}\sum_{i=1}^{n} x^{(i)} {x^{(i)}}^T\right) u.$$

That is equivalent to find the eigenvectors of Σ

Unsupervised learning: Autoencoder

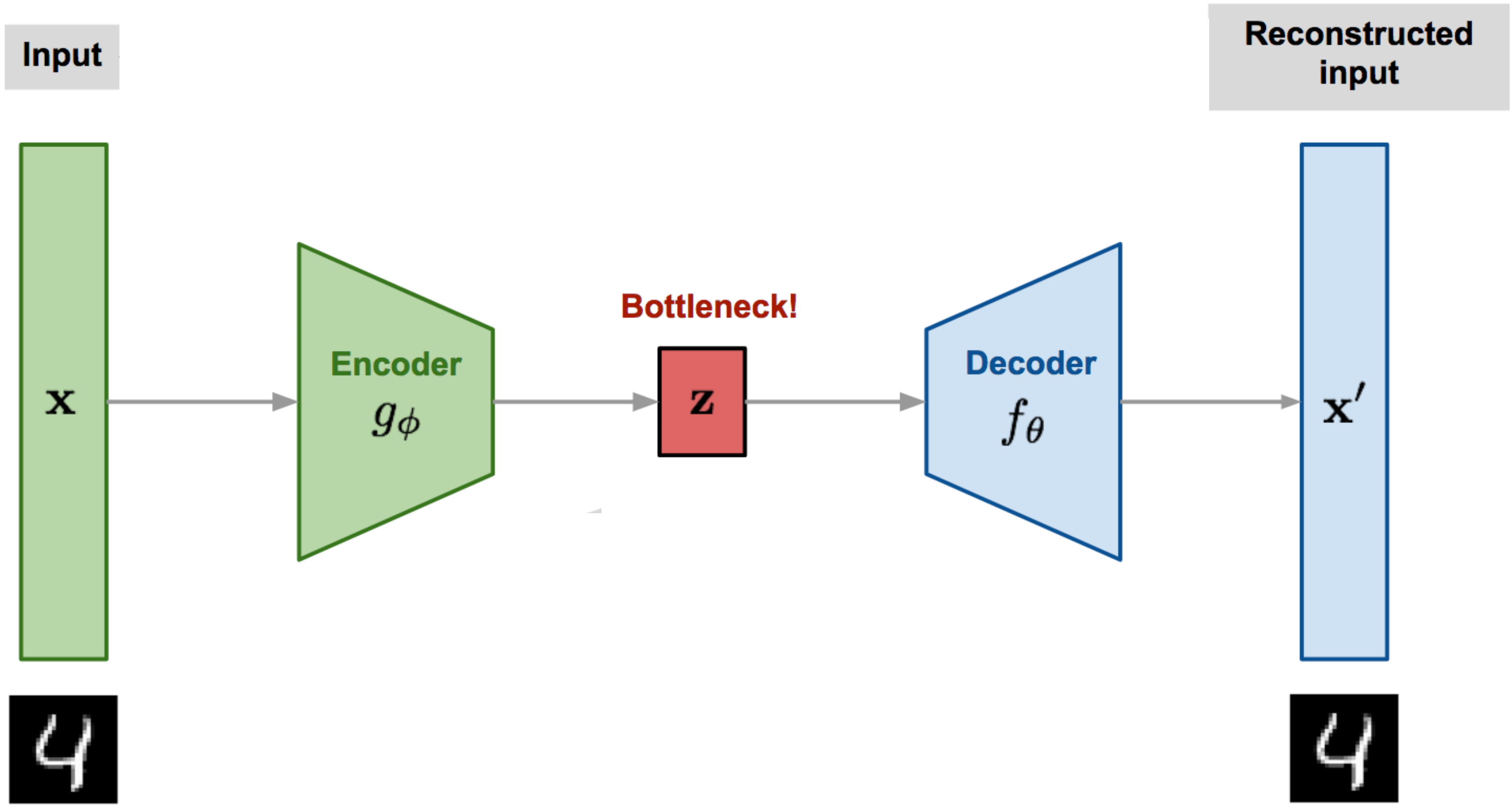
Recap: in feed-forward NN, the output $\hat{y}$ are compared with labels y through the loss function, which is a (scalar) function of the weights and biases



An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

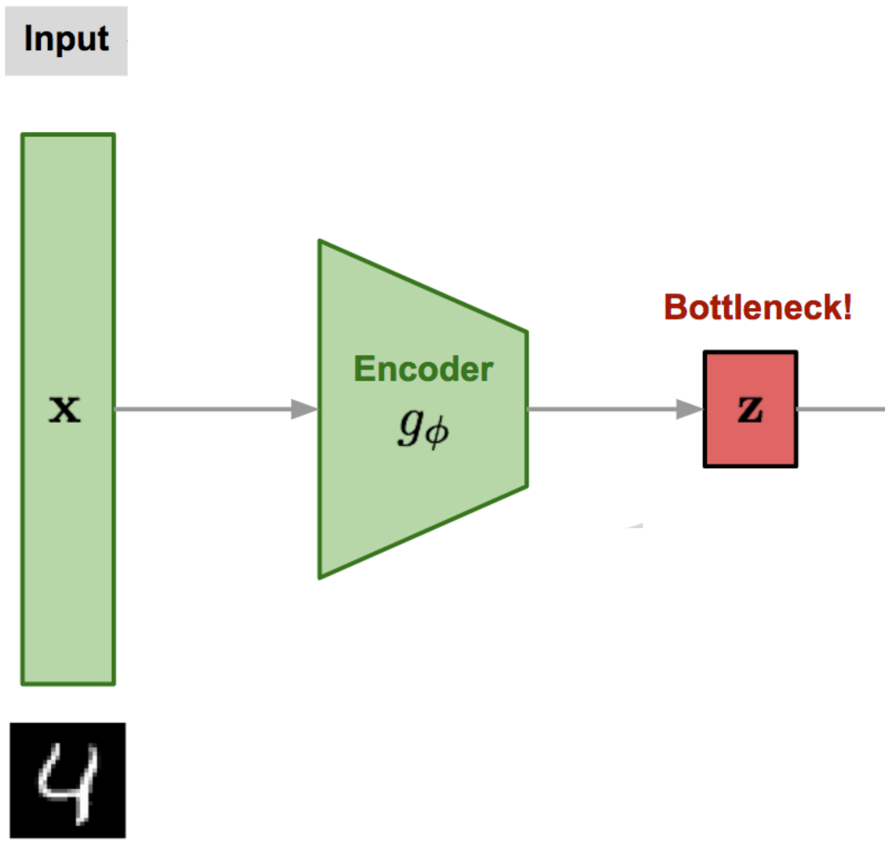
Unsupervised learning: Autoencoder

Autoencoders are an example of deep unsupervised learning



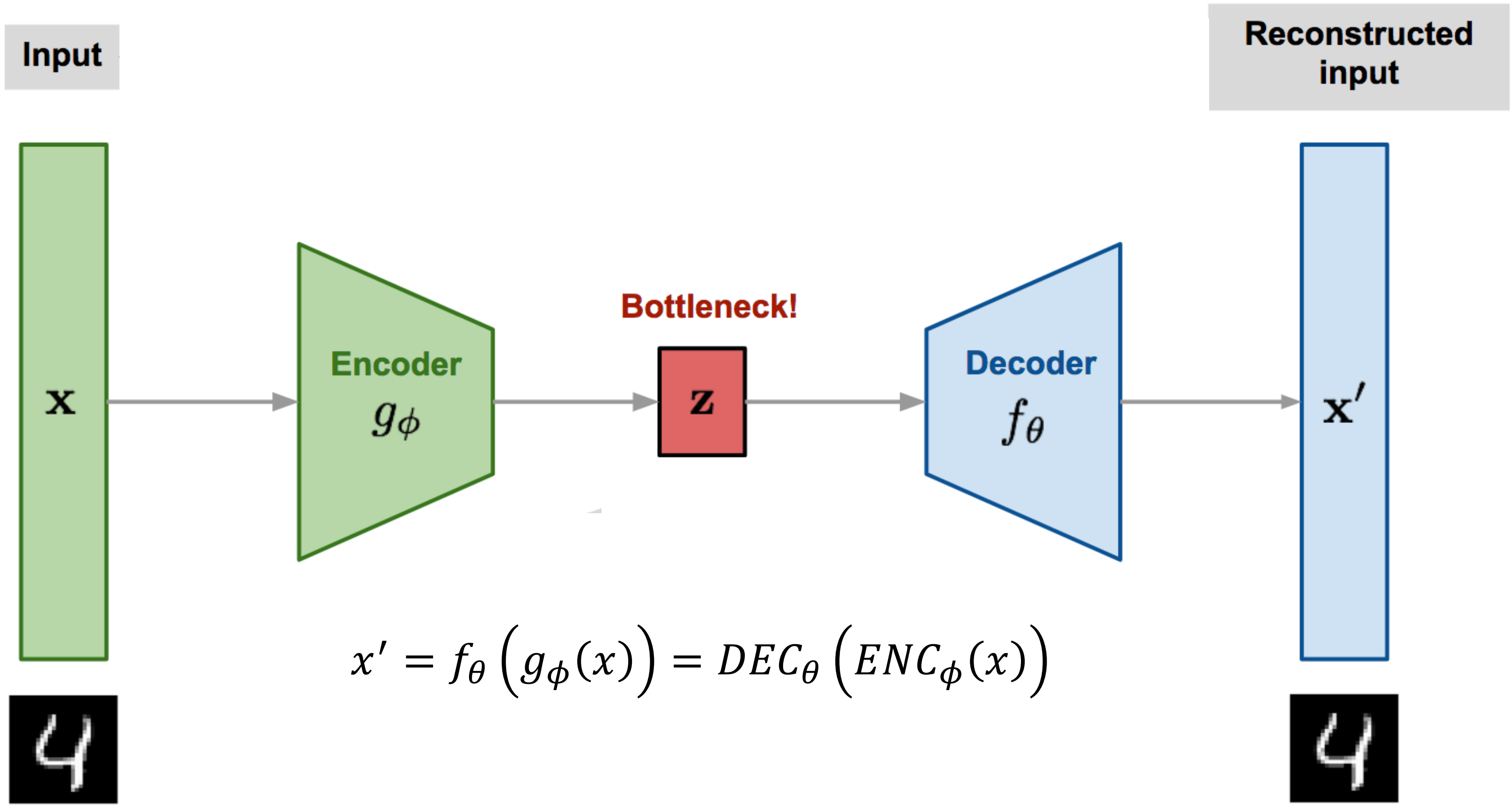
Unsupervised learning: Autoencoder

Dimensionality reduction → ENCODER $z = g_\phi(x)$



Unsupervised learning: Autoencoder

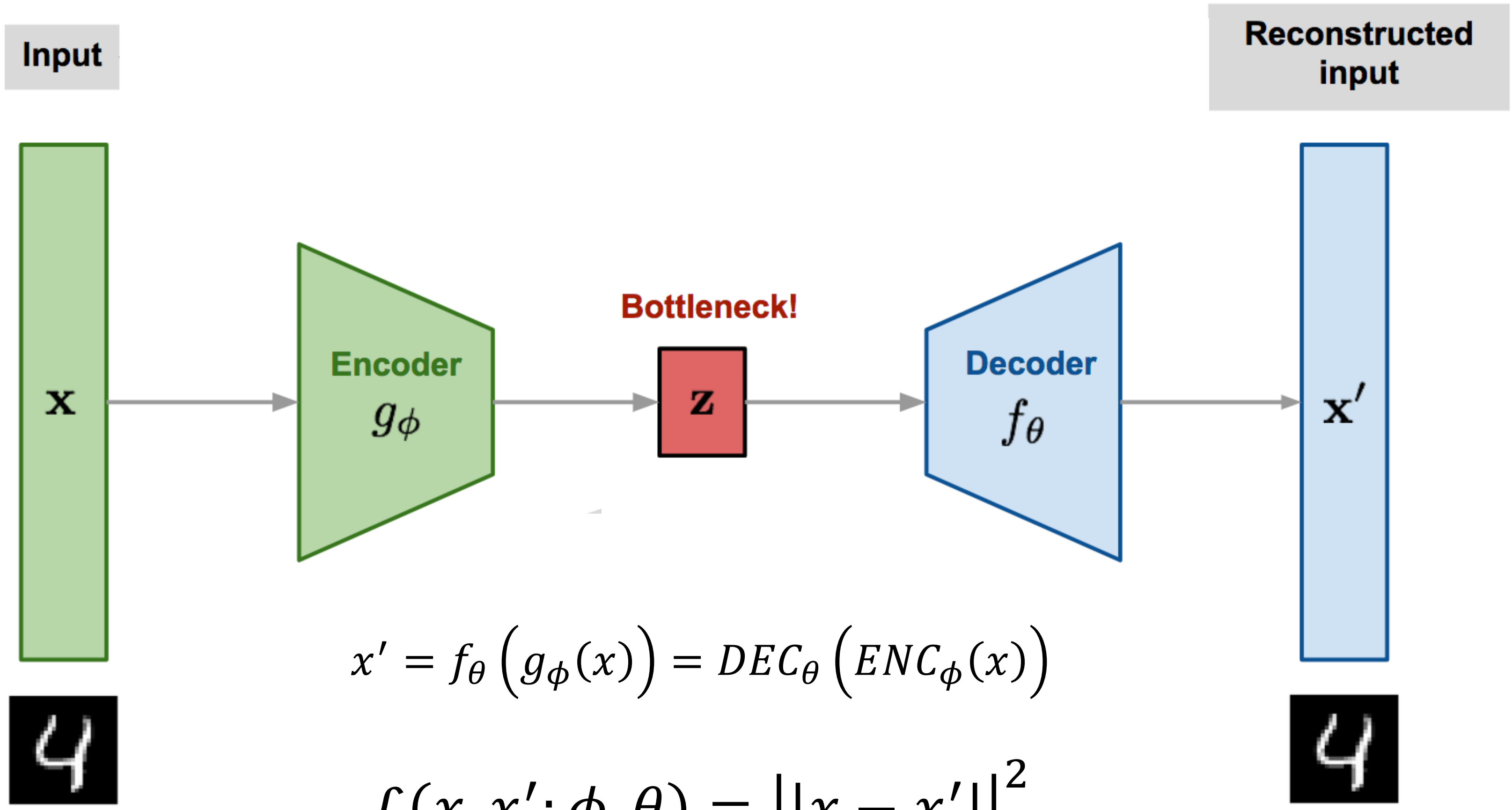
Dimensionality reduction → ENCODER $z = g_\phi(x)$ → Original dimentionality restored → DECODER



$$x' = f_\theta\left(g_\phi(x)\right) = DEC_\theta\left(ENC_\phi(x)\right)$$

Unsupervised learning: Autoencoder

Which loss function? Remember: no labels here!

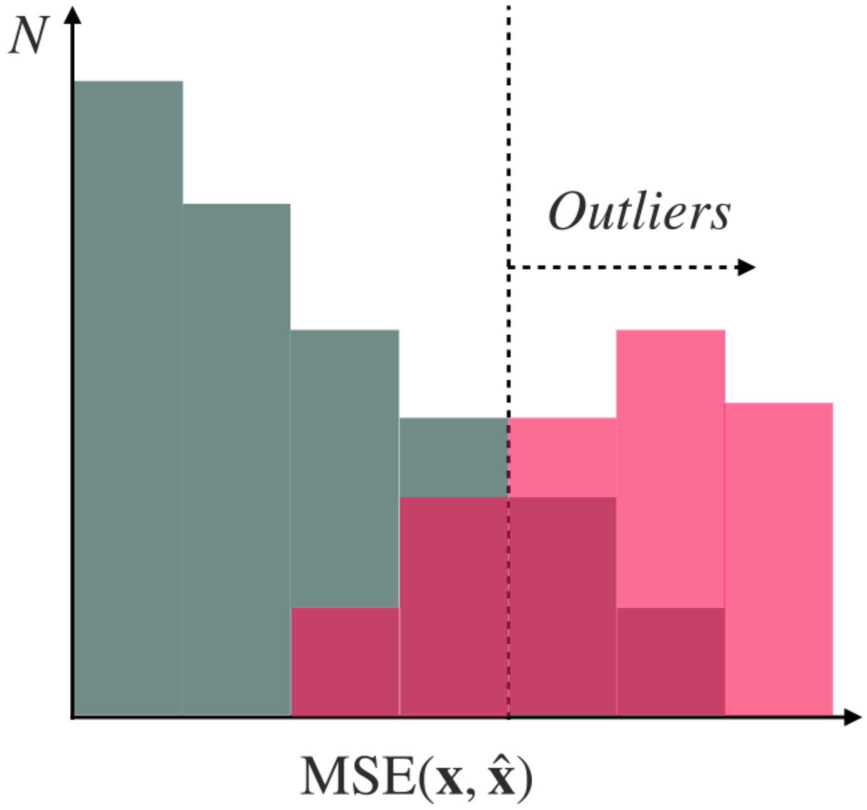


$$x' = f_\theta\left(g_\phi(x)\right) = DEC_\theta\left(ENC_\phi(x)\right)$$

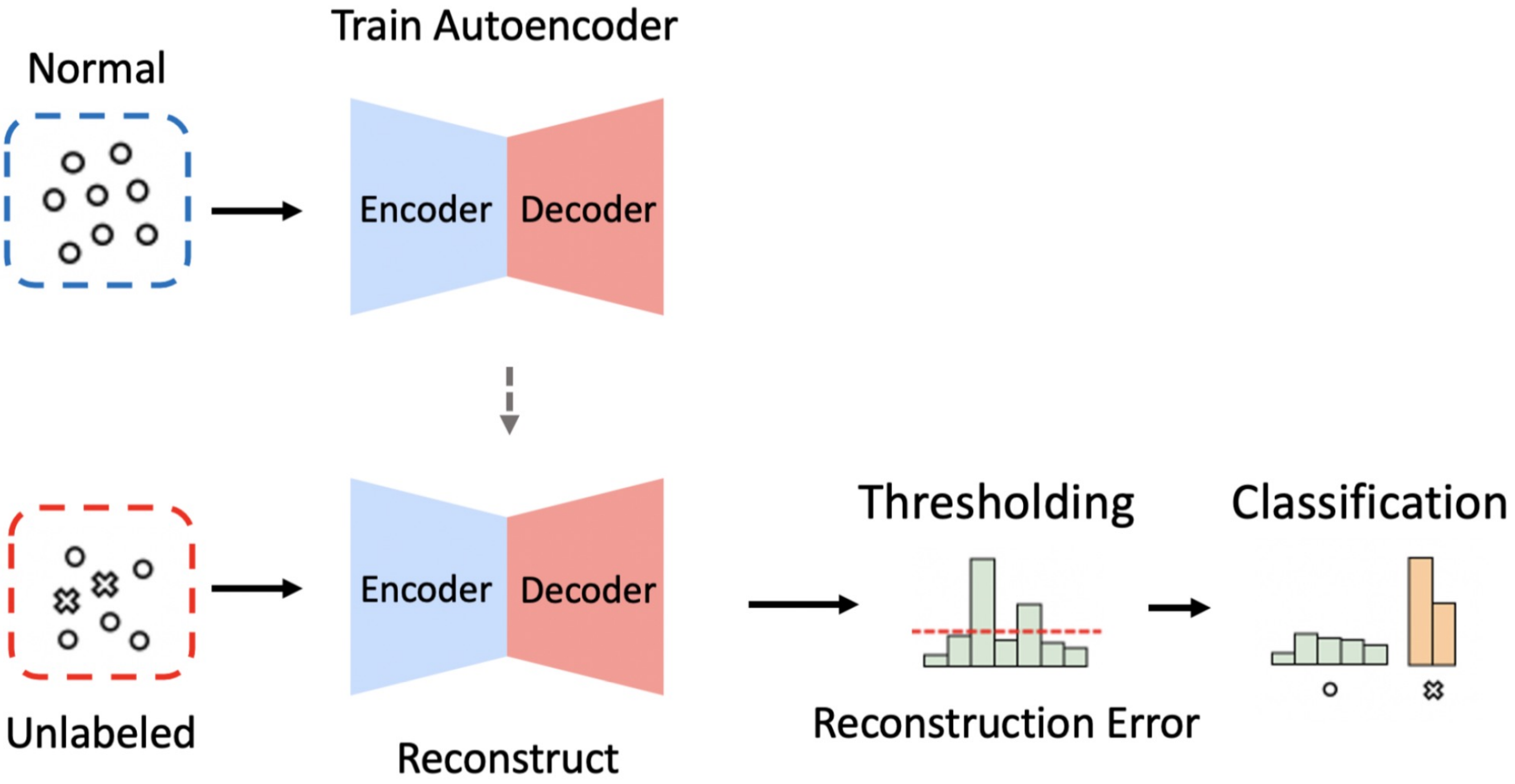
$$\mathcal{L}(x, x'; \phi, \theta) = ||x - x'||^2$$

Unsupervised learning: Autoencoder

Which loss function? Remember: no labels here!

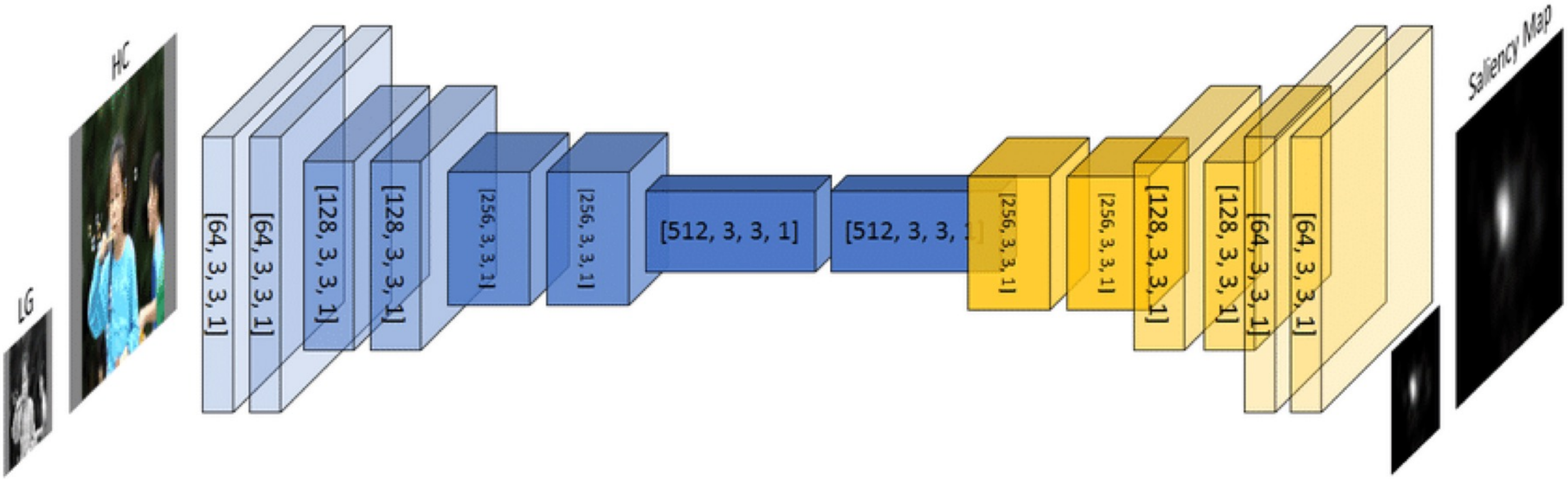


Unsupervised learning: Autoencoder

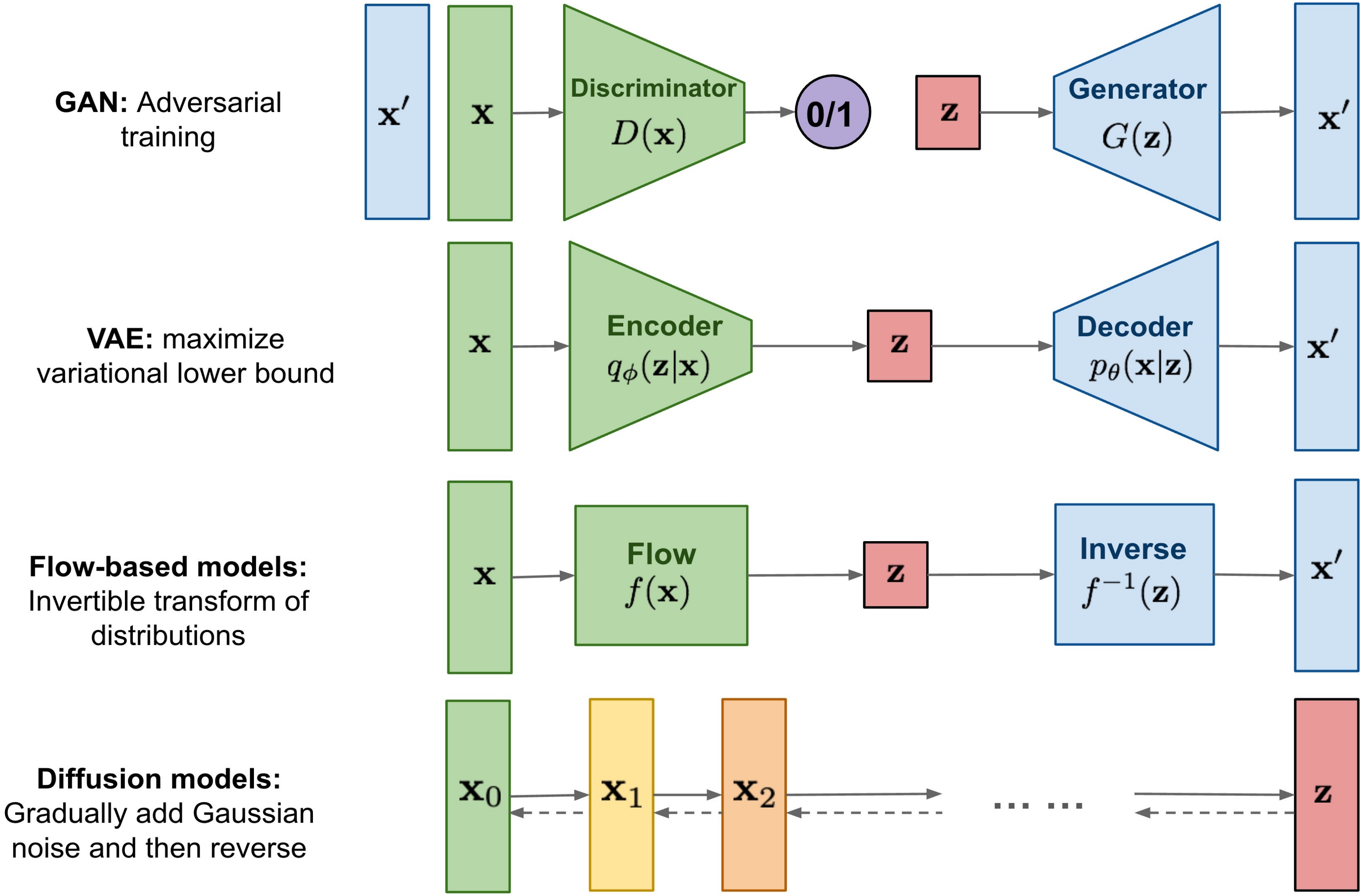


Unsupervised learning: Autoencoder

Can be extended to image reconstruction



Generative models



https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

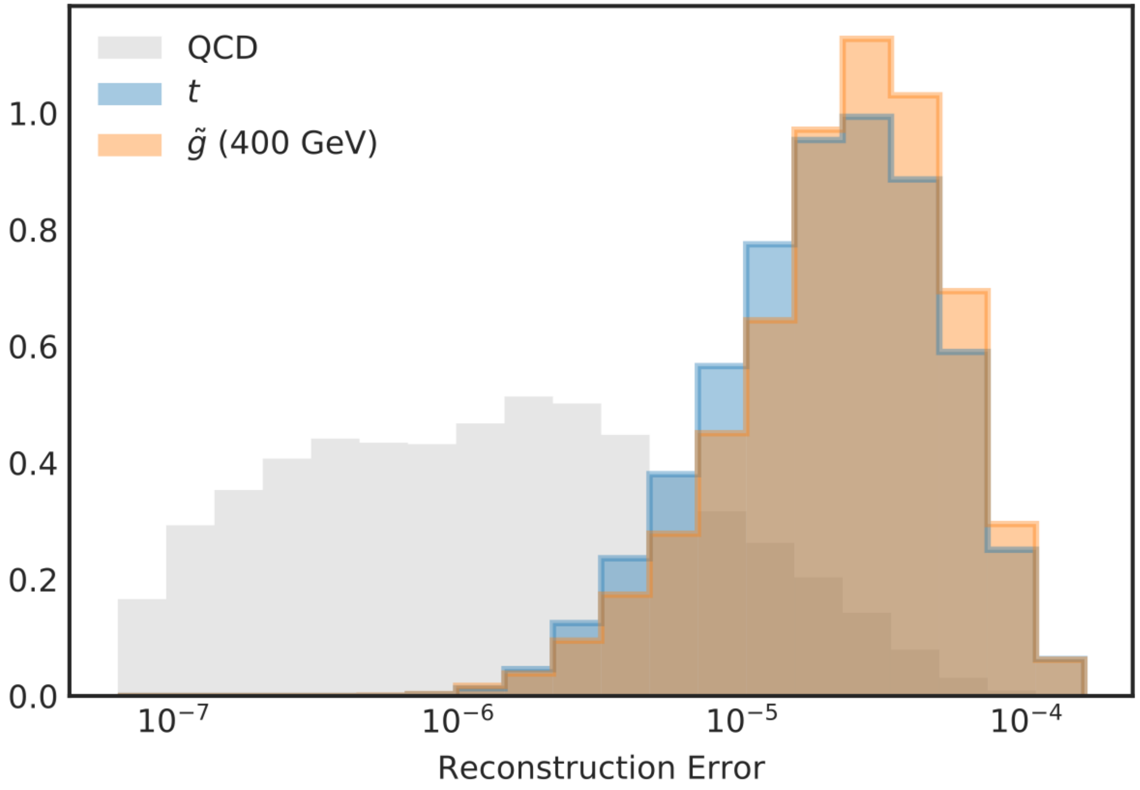
Applications in HEP

https://indico.slac.stanford.edu/event/7540/contributions/5437/attachments/3225/8919/SLAC%20Summer%20Institute%202023%20Lecture%201%20Anomaly%20Detection.pdf

Example: searching for NP with autoencoders

Farina, Nakai & **DS** 1808.08992; Heimel, Kasieczka, Plehn & Thompson 1808.08979; Cerri et al 1811.10276; and many more…

from 1808.08992

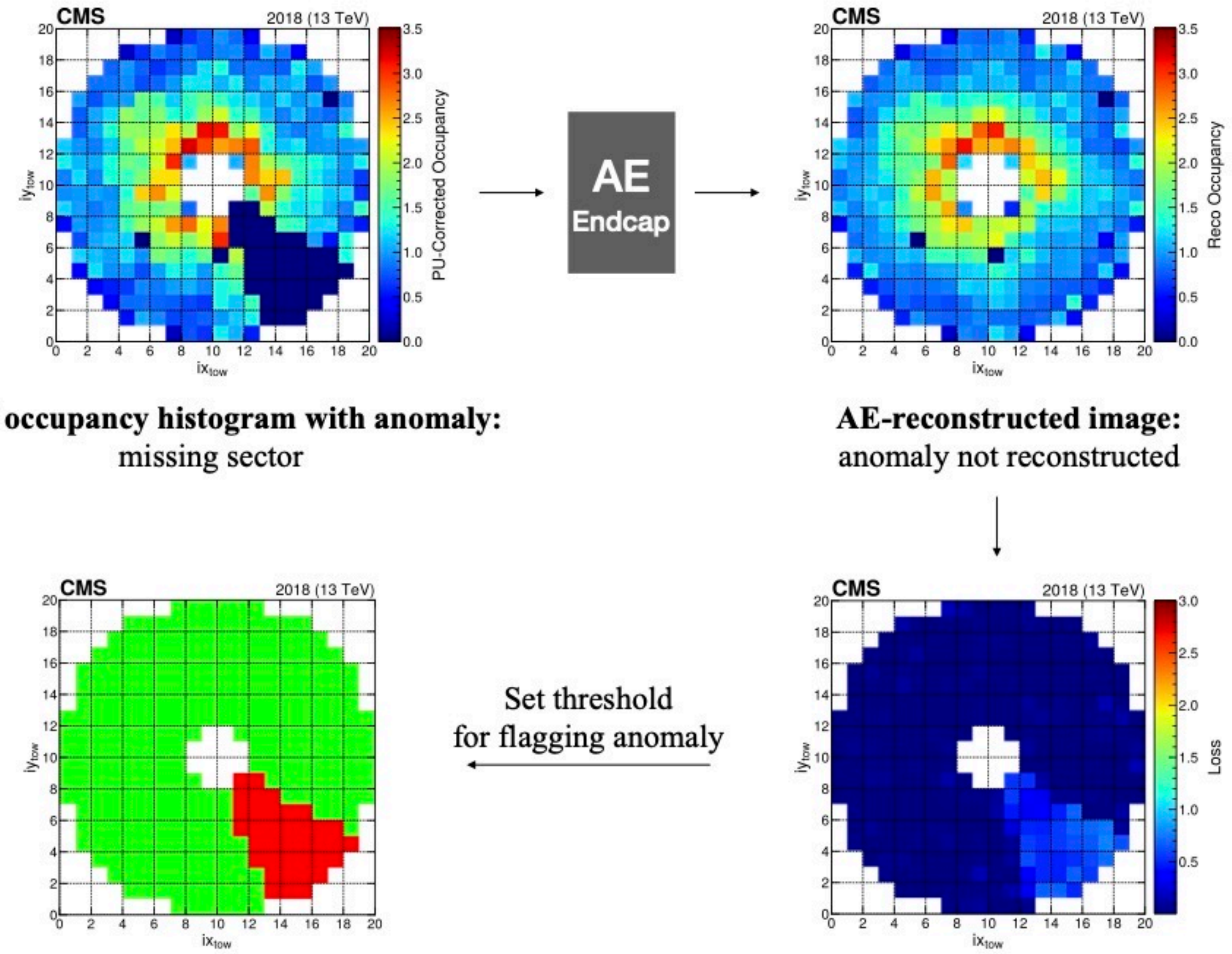


background (QCD) signal (tops and 400 GeV RPV gluinos)

We showed that the AE could detect interesting physics anomalies.

Train the AE on QCD jets only.
Can detect top and gluino jets as anomalous!

Data Quality Monitoring



Input occupancy histogram with anomaly:
missing sector

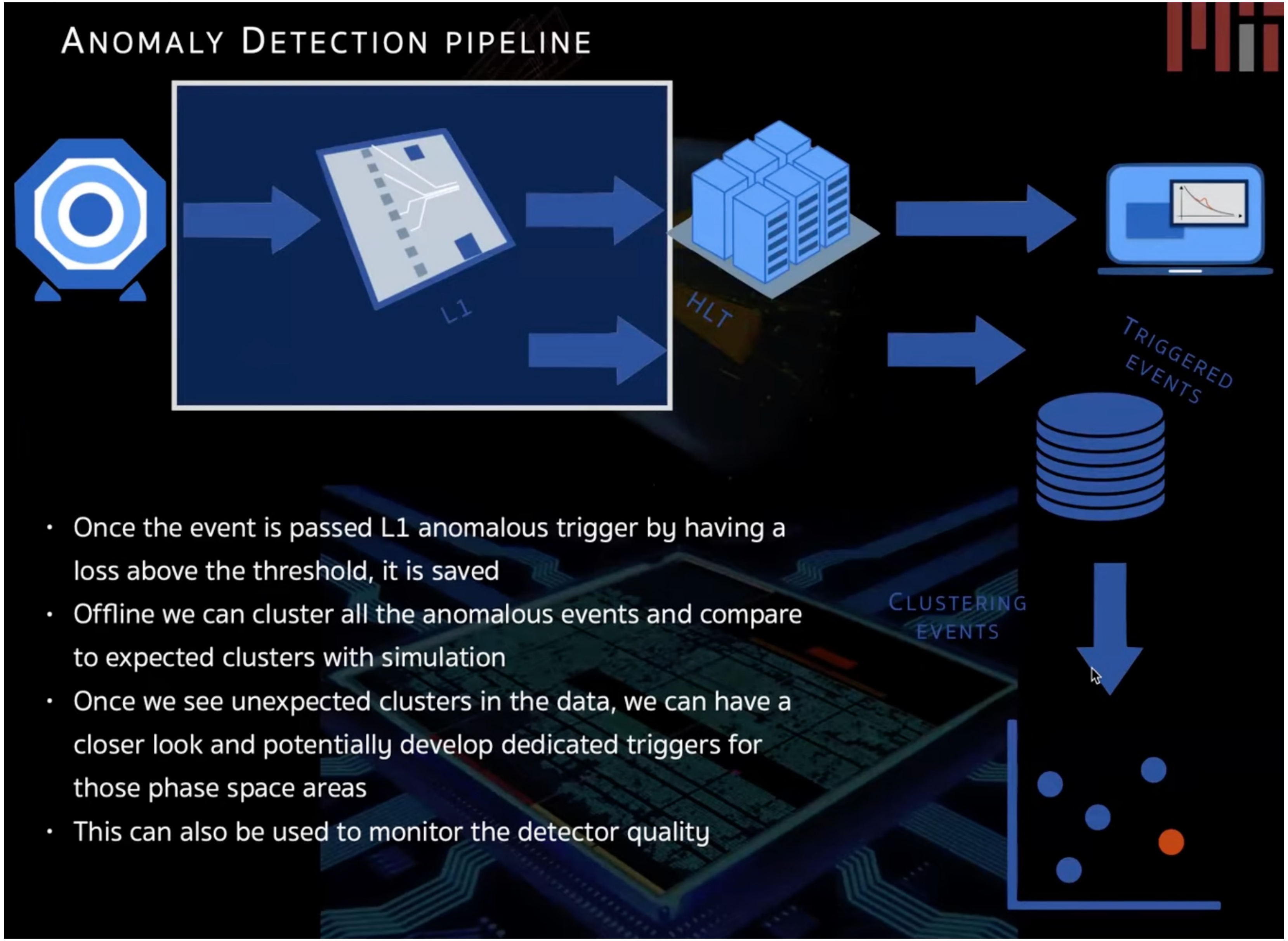
AE-reconstructed image:
anomaly not reconstructed

Loss map:
anomalous showing high loss

Final quality output:
anomalous towers: red
good towers: green

https://arxiv.org/pdf/2308.16659

Hardware implementation: trigger!



Hands-on!

Reconstructing images with autoencoders

https://github.com/fsimone91/course_ml4hep/tree/2024/notebooks/6_convolutional_autoencoder.ipynb

Hands-on!

Comparing different unsupervised methods for anomaly detection

https://github.com/fsimone91/course_ml4hep/tree/2024/notebooks/2024/6_unsupervised_methods.ipynb