Research Project Proposal:
Multiple Source Feedback Clustering

Francesco Fulco Gonzales
francesco.gonzales@mail.polimi.it
CSE

POLITECNICO MILANO 1863

HONOURS PROGRAMME HP-SR in Information Technology

Outline

- Introduction
- State of the art
- Research idea
- Research plan

Feedback Clustering

- Goal: perform an online learning task on a set of objects with sparse interaction data by compensating the data scarcity with a clustering approach
- Unstructured input data
- Three-stage architecture:

1. Initial clustering on objects data (task-independent)
2. Refine clusters by observing user interaction data (task-specific)
3. Online update of clusters as new interactions with objects are collected

Pricing case study

- Pricing e-commerce products
- Objects are the products on sale
- Interactions
- Goal: select the optimal price for each product in order to maximize profits
- Solution to the long-tail problem

Contribution

- Scientific contribution
  - Algorithmic framework to incorporate an online feedback as supervisory signal to clustering
- Application
  - Solution to the long-tail problem in pricing
  - Significant impact on bottom line of e-commerce businesses

Assumptions

- Three main assumptions:

  1. Object images contain a significant amount of information that can be leveraged for clustering

  2. The initial similarity between objects carries over into the similarity for the chosen task

  3. The proposed framework is general enough to provide a feasible solution to several different tasks

State of the art

- Deep Clustering
- Multi-Armed Bandit

Deep Clustering

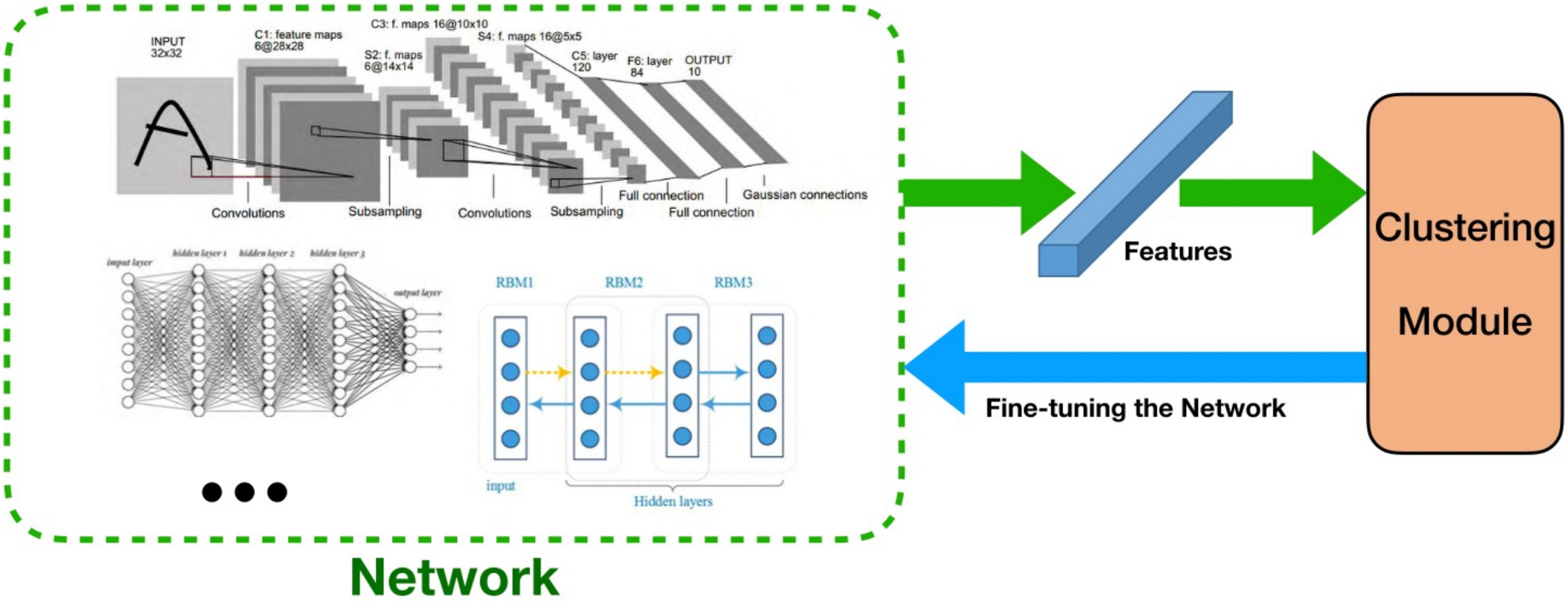
- Clustering methods with deep learning
- State-of-the-art performance
- NNs are capable of learning complex non-linear representations of data
- Embeddings are much easier to cluster

Deep Clustering architectures

- Clustering DNN
- Autoencoder
- Variational Autoencoder
- Generative Adversarial Network

Clustering DNN

Neural network architecture for clustering characterized by the absence of a network loss



Clustering DNN

- Risk of learning a corrupted feature space:
  tight clusters, hence small clustering loss but meaningless
- The network can be initialized in three ways:
  - Supervised: extract features from deep CNNs trained on large and diverse labeled datasets
  - Unsupervised: train a RBM or an autoencoder in an unsupervised manner, then fine-tune the network by the clustering loss
  - No pretraining: needs well-designed clustering

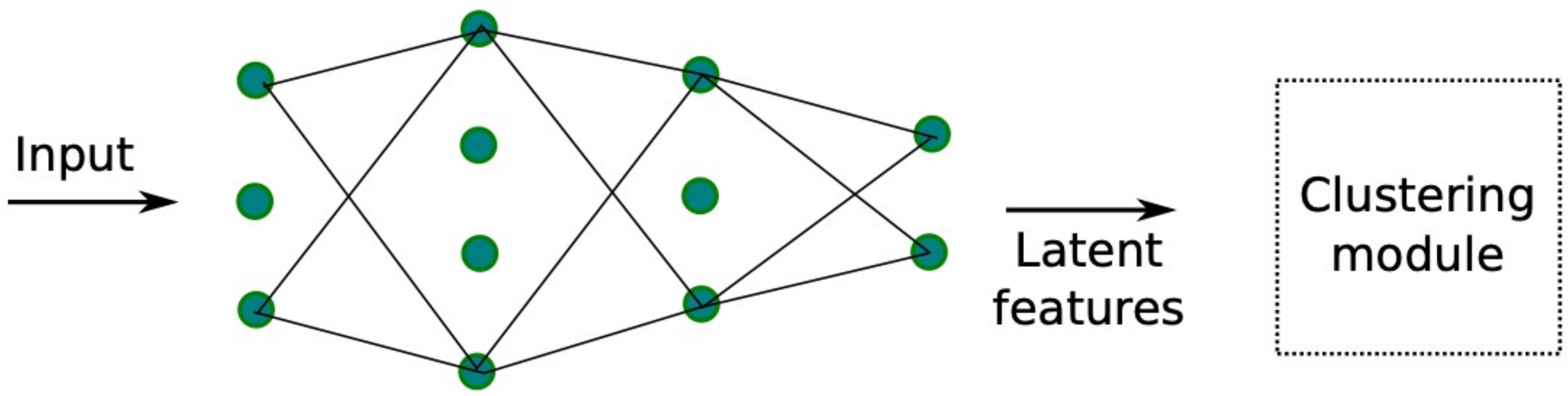
Deep Embedded Clustering

- **Joint optimization** of DNN and clustering **only with clustering loss**

Algorithm 1 DEC

Parameter initialization with AE
until convergence:
  Soft assignment of points to centroids
  Update parameters with current high confidence assignments

- Learn representation that brings data distribution closer to high confidence samples distribution

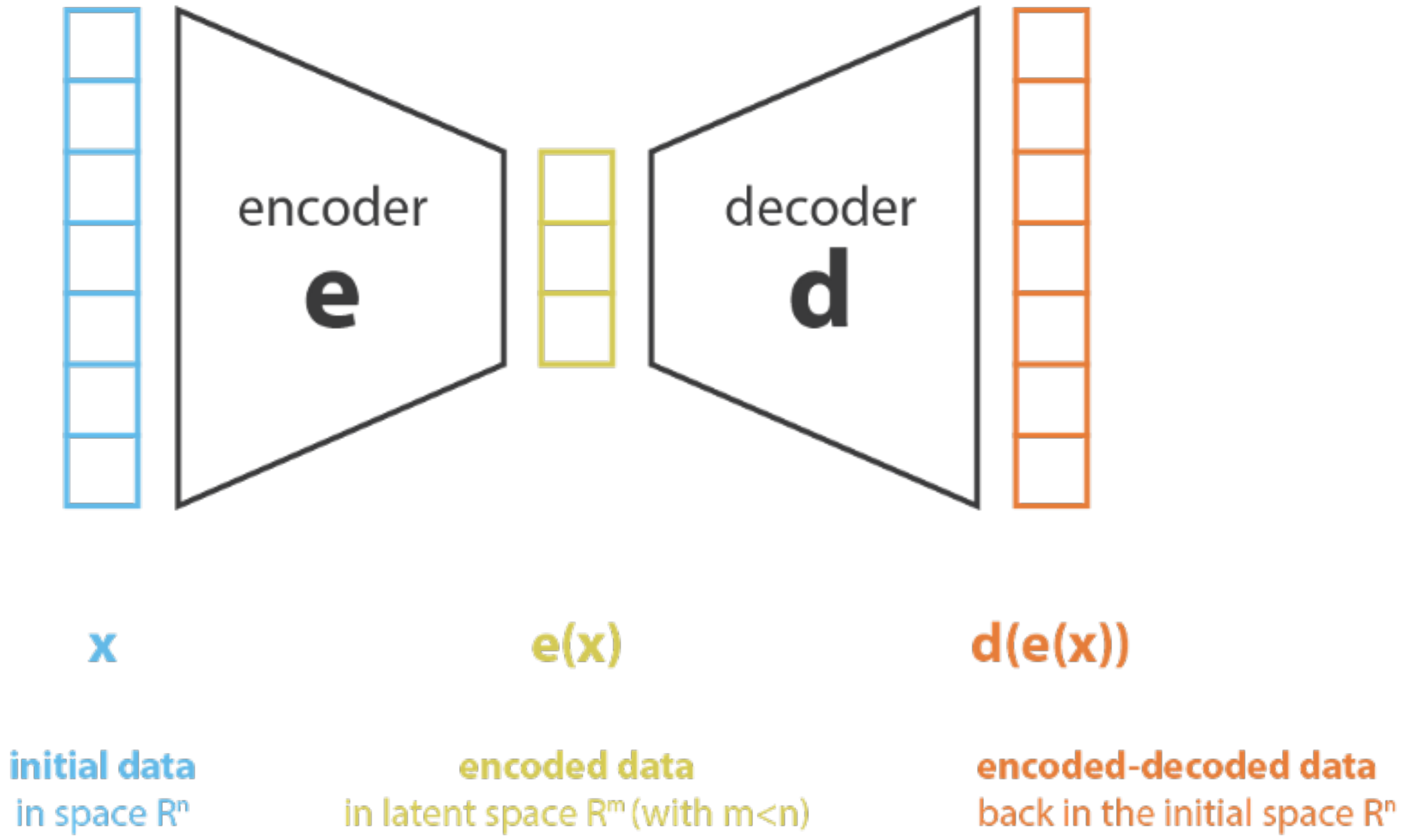


Autoencoder

An unsupervised network architecture that consists of two parts:

1. an encoder function $h = f_{\varphi}(x)$ that maps original data x into a latent representation h

2. a decoder function $r = f_{\theta}(h)$ that produces the reconstruction



Autoencoder

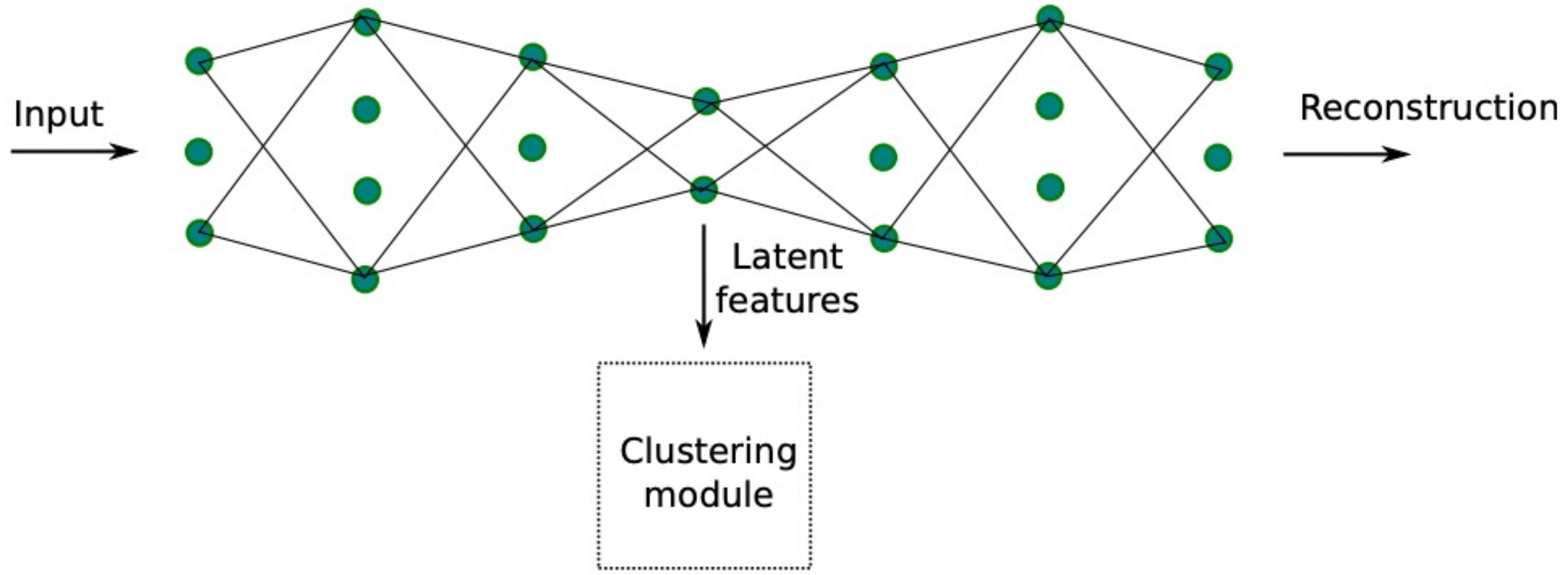
The reconstructed representation r is required to be as similar as possible to x, which is achieved by minimizing the reconstruction loss:

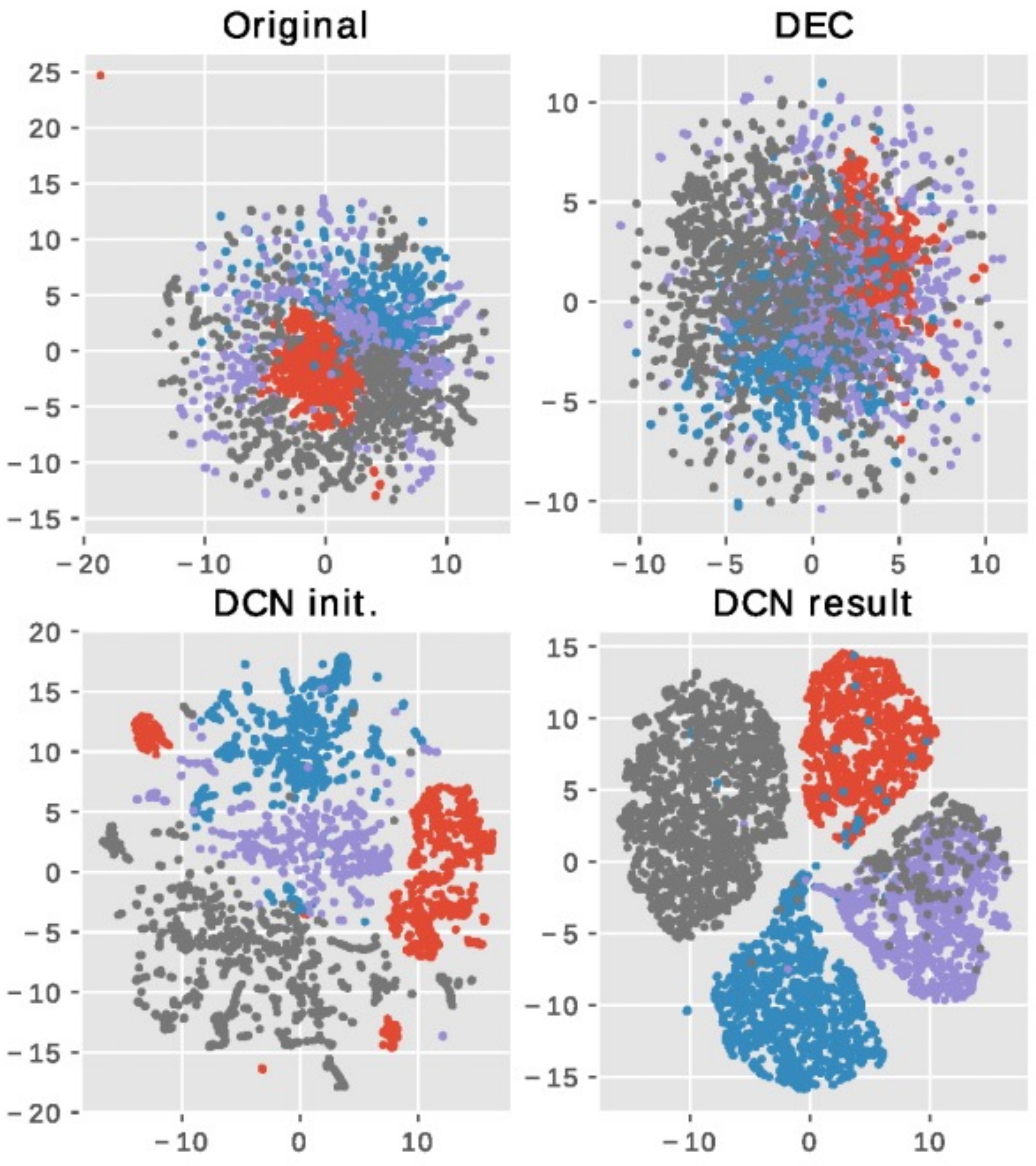
$$L_{rec}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} ||x_i - g_\theta(f_\phi(x_i))||^2$$

It manages to learn a lower dimensional embedding of the input data in a fully unsupervised manner.

Deep Clustering Network

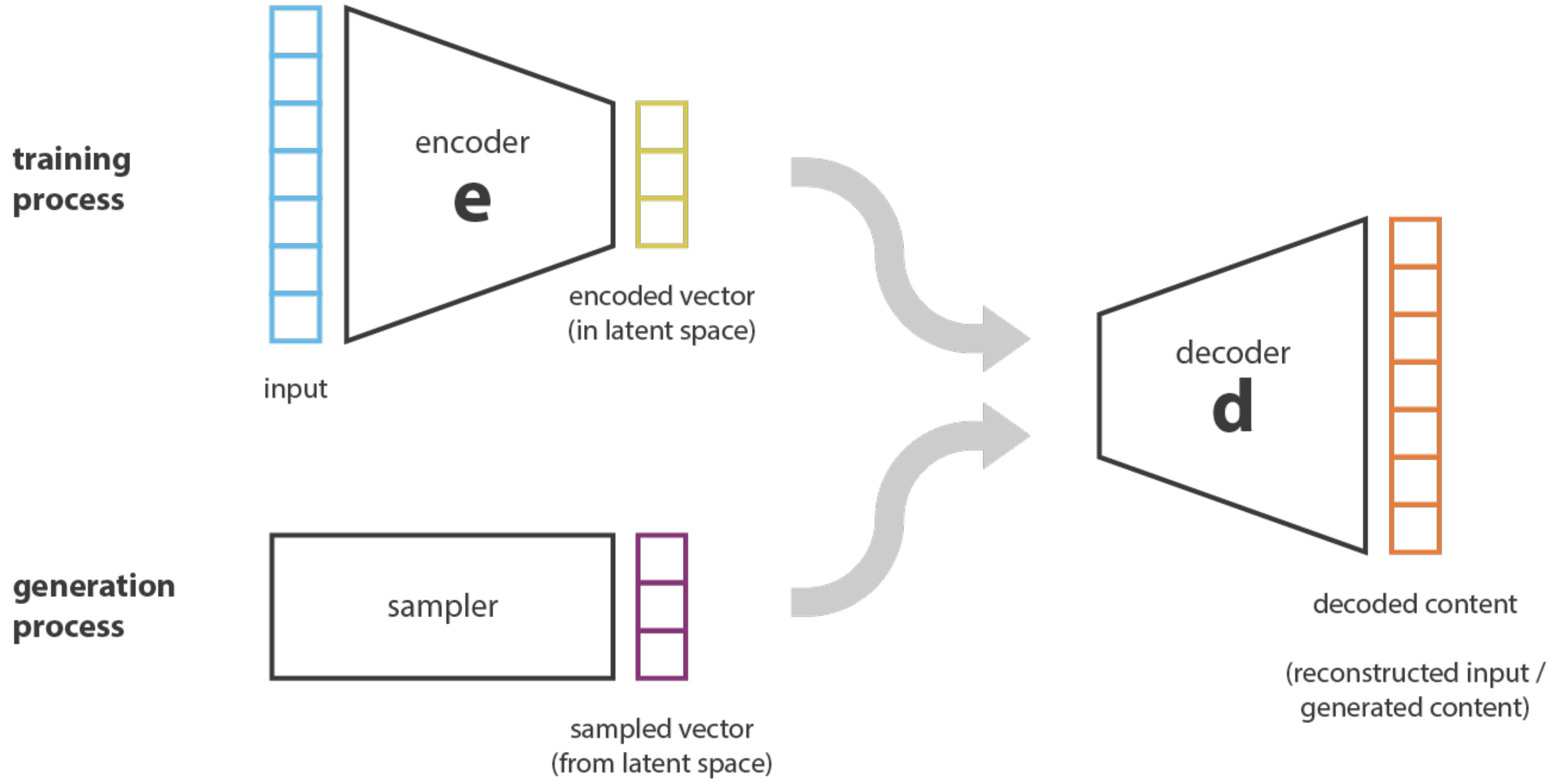
- Initial autoencoder pre-training to learn a feature representation
- Joint optimization of dimensionality reduction and clustering
- **Reconstruction loss** to avoid trivial solutions
- Clustering with k-means





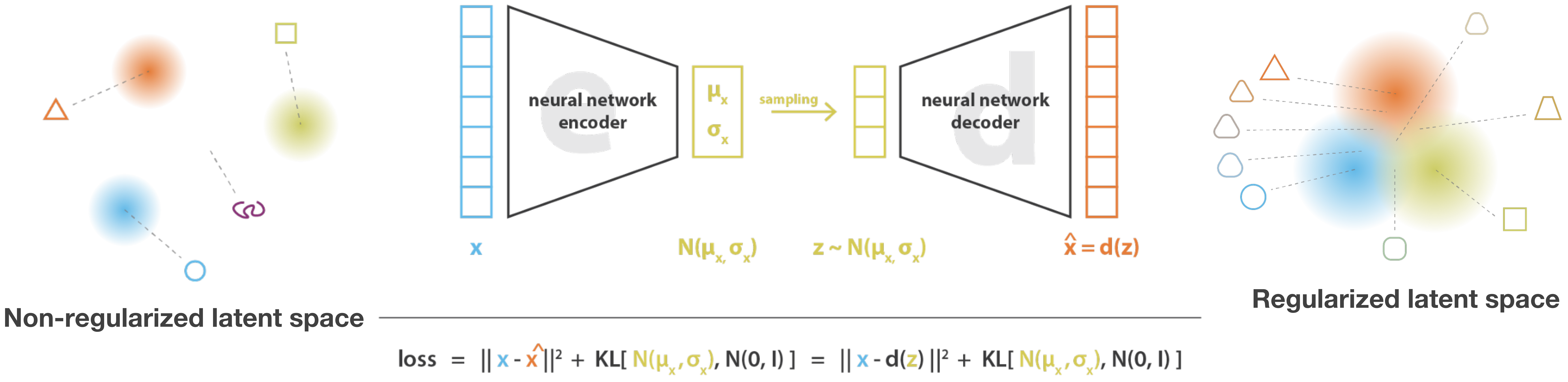
Variational Autoencoder

- Generative variant of the Autoencoder (AE)
- An AE learns the input data distribution and the latent representation, but not the latent representation distribution
- To generate data with an AE, the encodings are **sampled randomly** and passed to the decoder to generate a new sample ⇒ unrealistic output



Variational Autoencoder

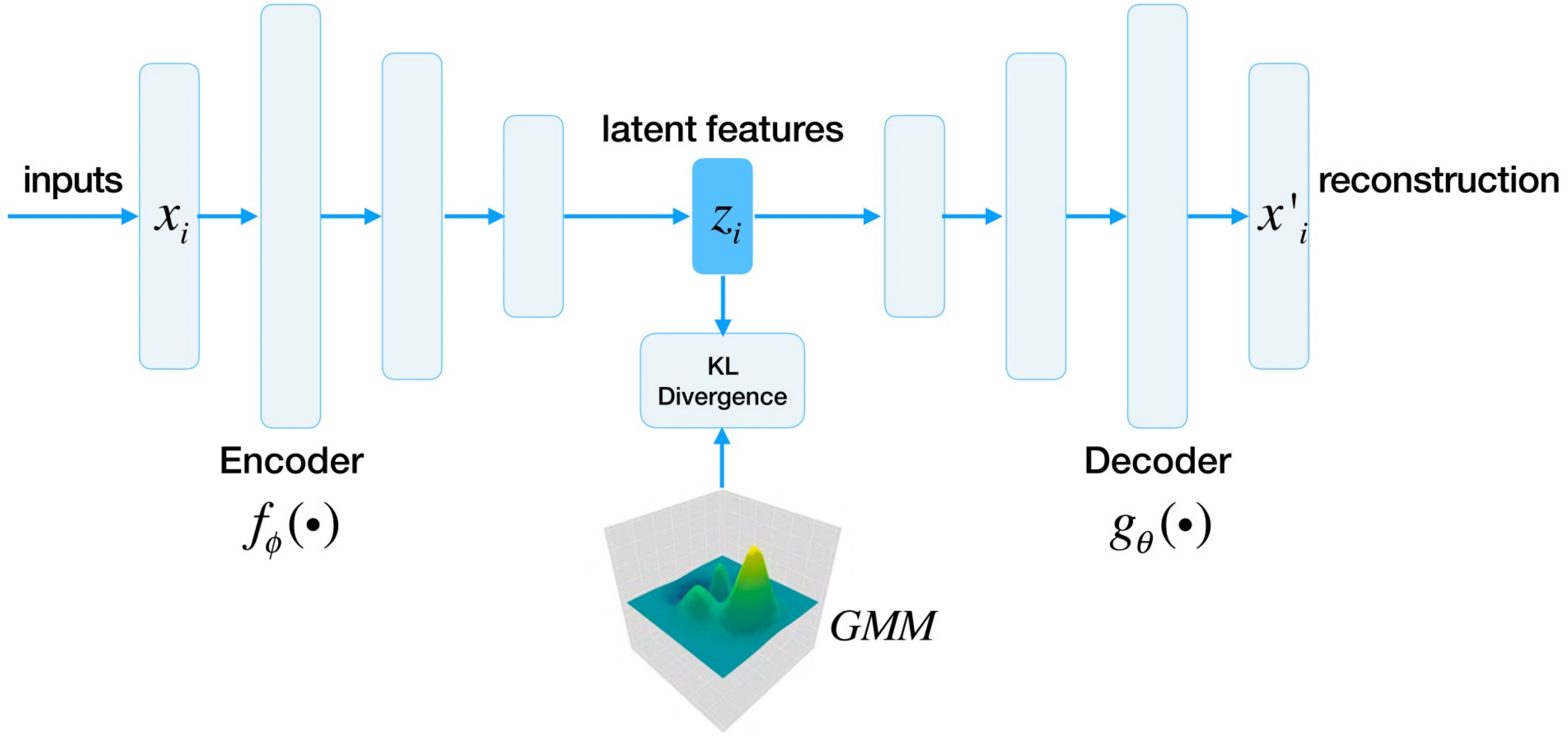
- VAE instead enforces the latent code of AE to follow a predefined distribution
- Allows sampling noisy data from the learned latent distribution, which is forced to be close to a standard normal for regularization purposes
- Backprop cannot be run through a sampling node: reparameterization trick

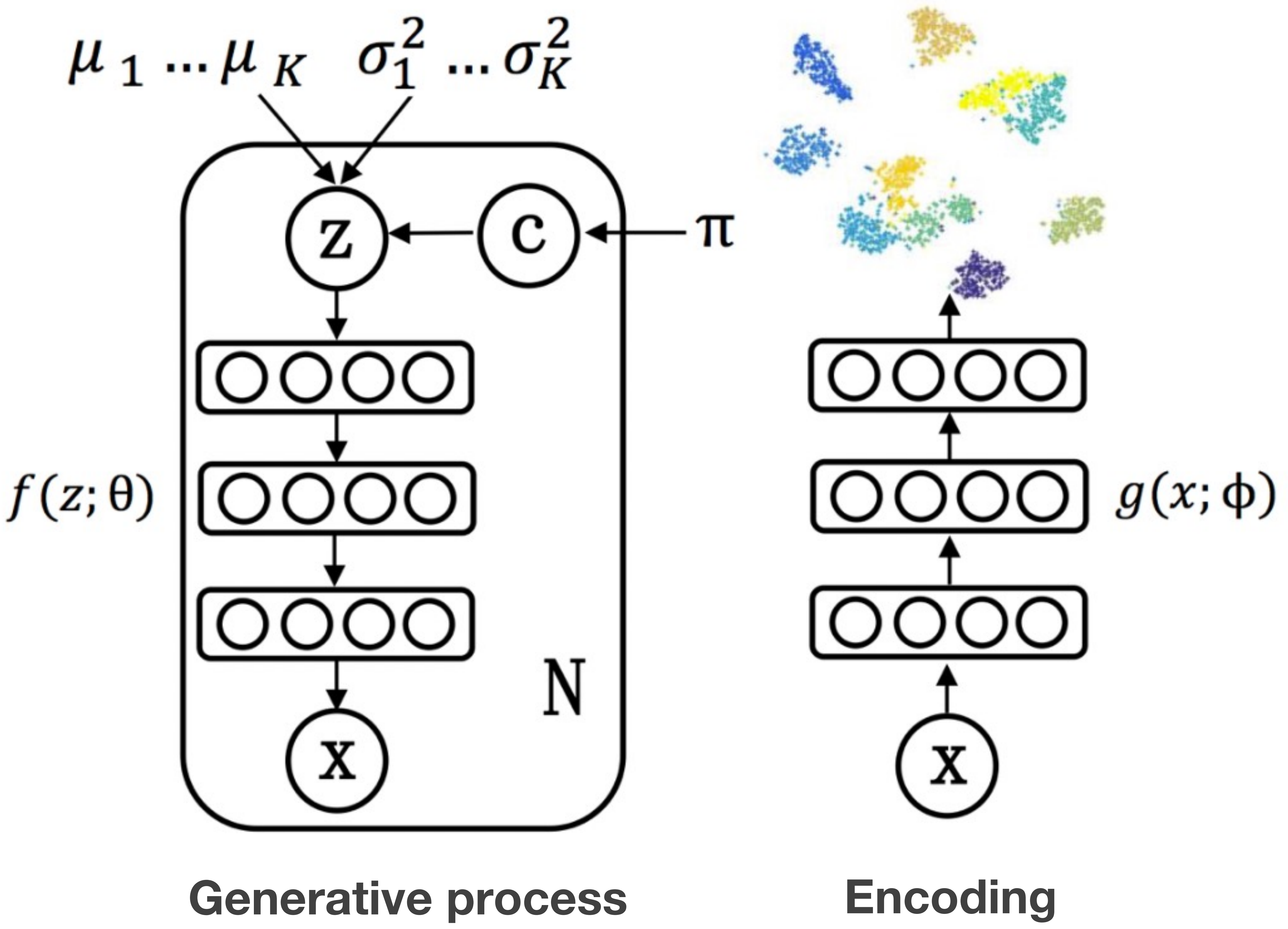


Variational Deep Embedding

- VaDE is probabilistic clustering method that combines VAE with a Gaussian Mixture Model

- Mixture-of-Gaussians prior replaces the single Gaussian prior



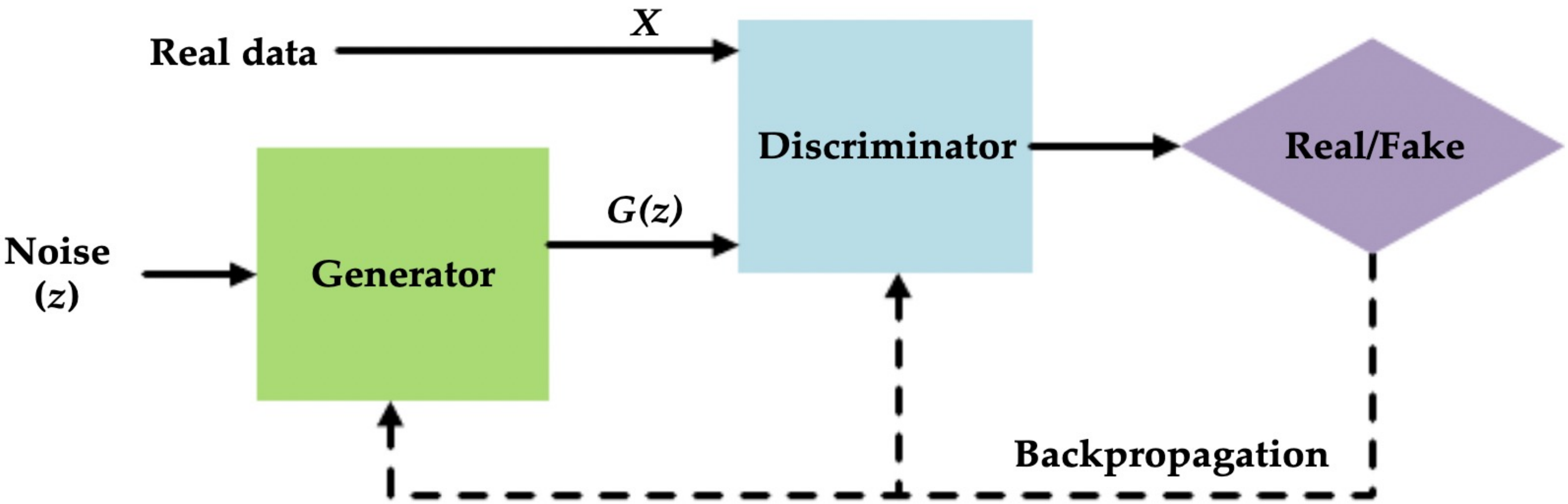


Generative Adversarial Network

Min-max adversarial game between two networks:

1. Generator that tries to transform a sample z (noise) from a prior distribution $p(z)$ to the input data space

2. Discriminator that tries to predict weather a sample is real or generated by the generative network

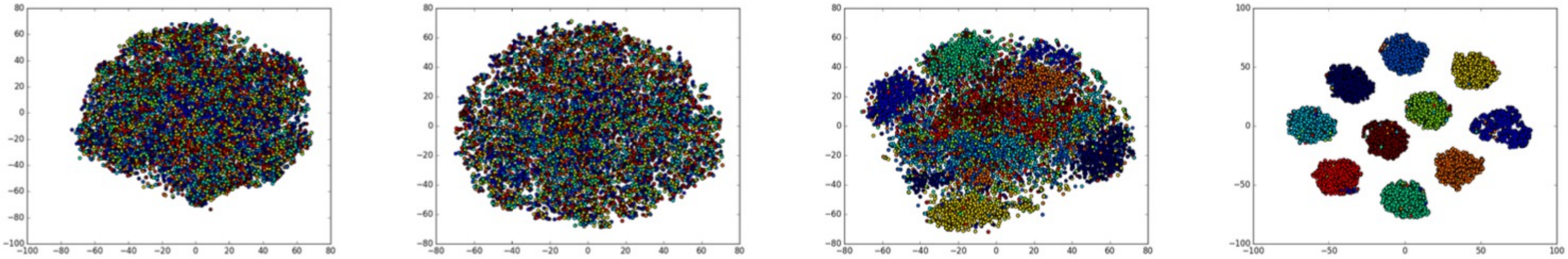


ClusterGAN

- Vanilla GAN does not cluster well in the latent space: unfavorable data distribution in latent space

- Sampling from a **discrete-continuous mixture** prior helps, a lot

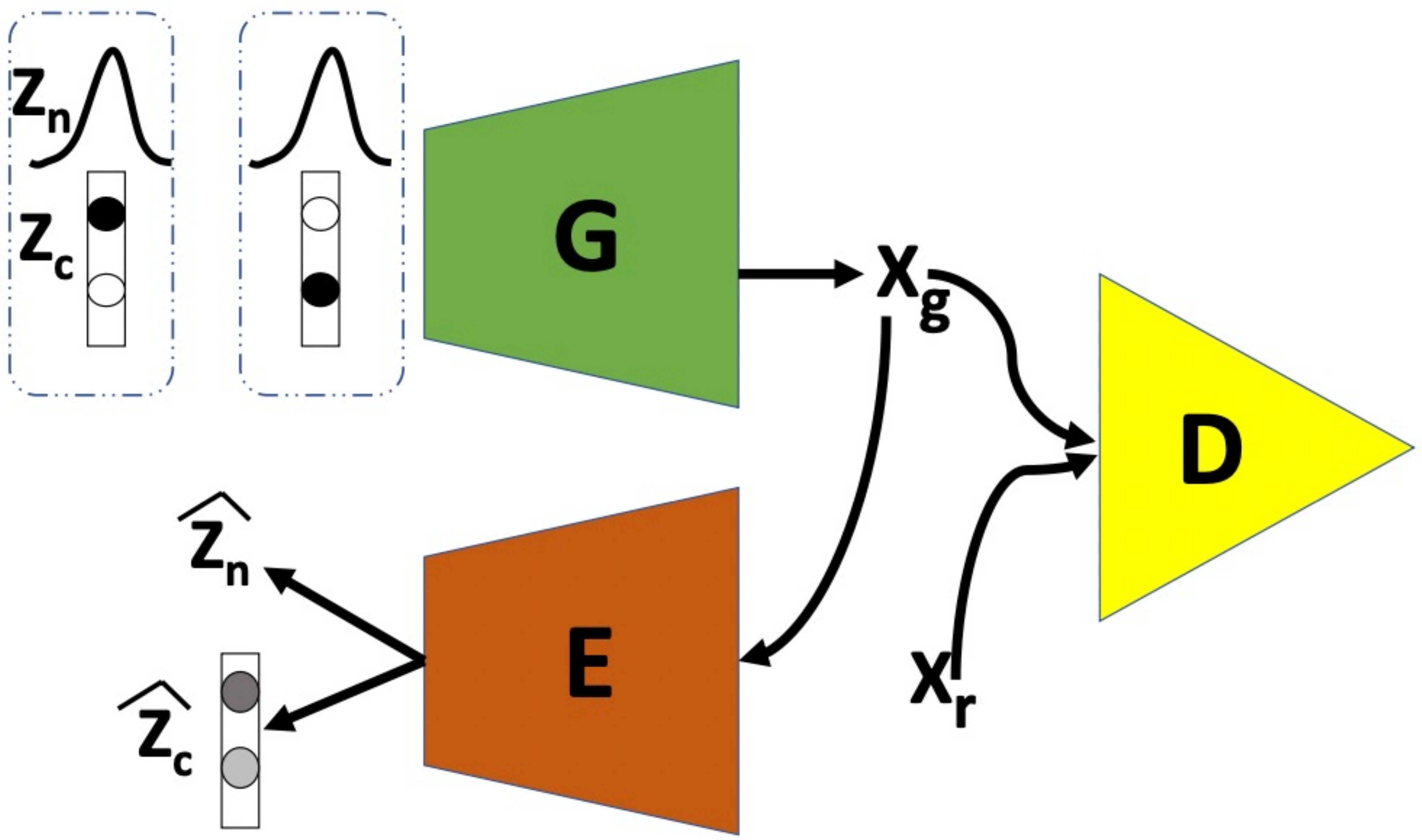
- Adapted backpropagation algorithm to accommodate the discrete-continuous mixture



(a) $z \sim$ Uniform (b) $z \sim$ Normal (c) $z \sim$ Gaussian Mix (d) $z \sim (z_n, z_c)$

ClusterGAN

- Explicit inverse-mapping network (encoder) to obtain the latent variables given a sample
- Interpolation in latent space is preserved
- Joint training of the GAN and the encoder with a clustering-specific loss to enforce structure in the GAN training

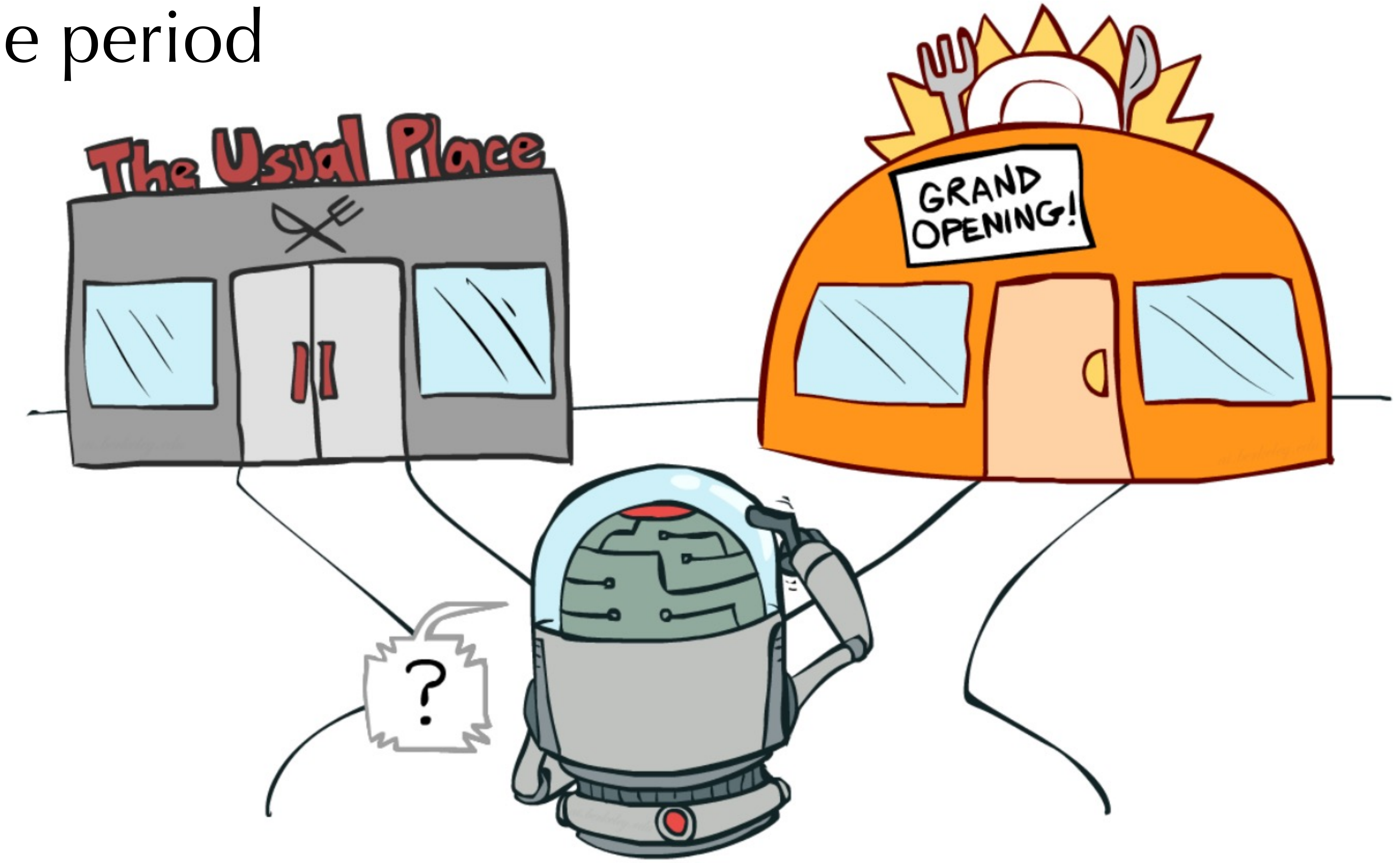


State of the art

- Deep Clustering
- Multi-Armed Bandit

Multi-Armed Bandit

- An agent is faced repeatedly with a choice among multiple actions
- After each choice the agent receives a scalar reward sampled from a probability distribution that depends on the selected action
- The objective of the agent is to maximize the expected total reward over some finite time period



Contextual MAB

- At each round the agent also is given a context vector that is used together with the past observations to choose the arm to play

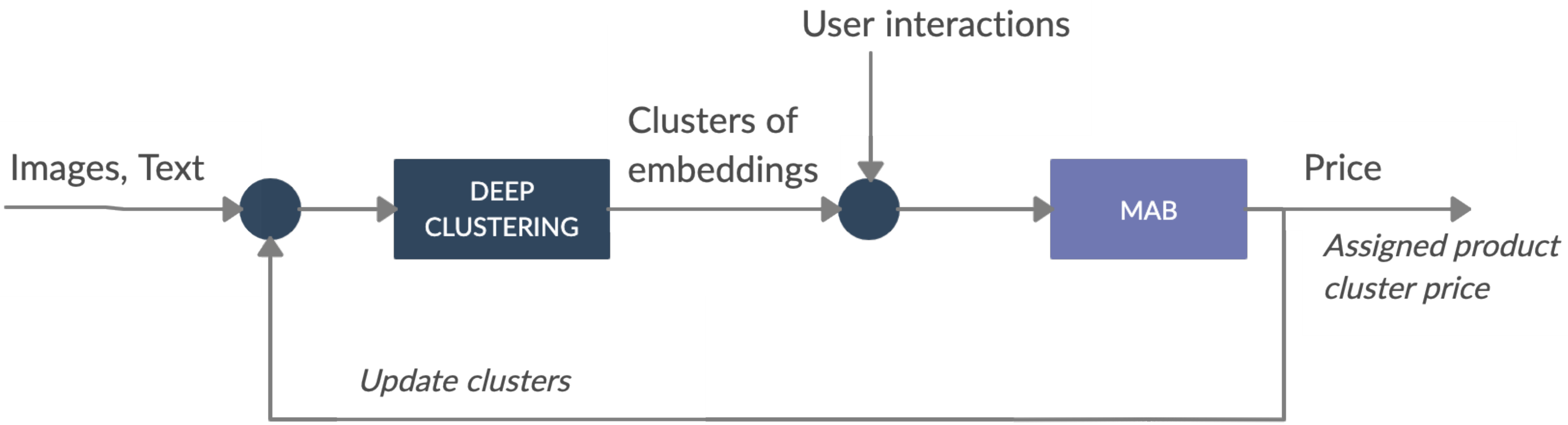
- Over time the agent learn how context vectors and rewards relate to each other and will be able to predict the next best action to perform by using the context vectors

“Life, like lunch, is full of difficult choices.”

Contextual MAB: Pricing

- Product cluster represents the context vector given as input to the learner
- The action is the price that the learner must choose for the given cluster
- Reward will be given according to the user response to the chosen price for the products belonging to that cluster and the margin on the sale

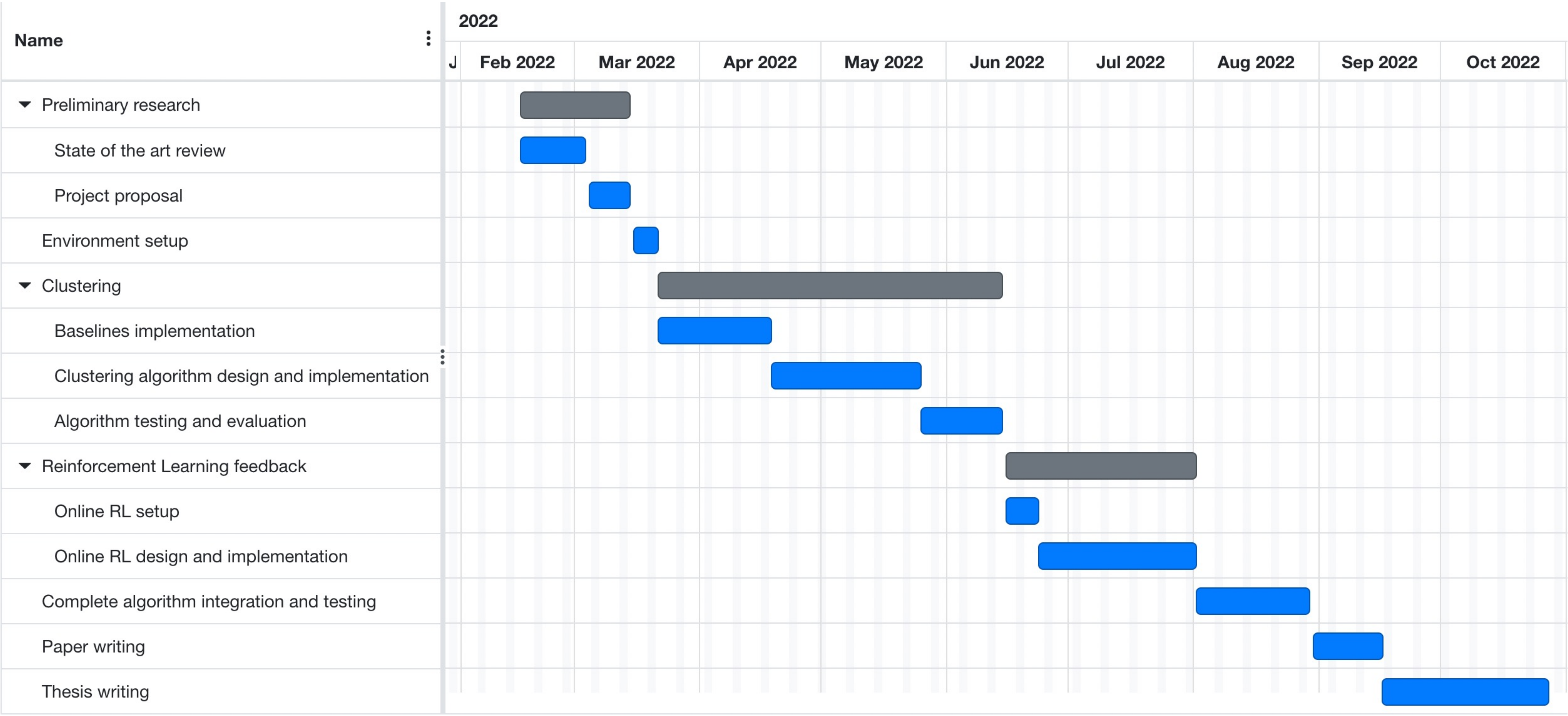
The Feedback



- The reward will serve as supervisory signal to adjust the chosen price (explicit goal of the MAB)

- Clustering of objects by changing the latent representation of input data by having the new price acting as feedback control to adjust the clustering

Research Plan



Thank you for your attention!