

PART V: A VERY BRIEF INTRODUCTION TO DEEP UNSUPERVISED LEARNING

*elements based on D. Kirkby lectures at KSPA19

WHAT DOES MACHINE LEARNING DO?

the machine is told what to look for

SUPERVISED

LEARNS A MAP FROM
X [FEATURES] TO Y
[LABELS]

$$P(X|Y)$$

the machine is NOT told what to look for

UN-SUPERVISED

NO LABELS - DISCOVER
PATTERNS

$$P(X)$$

IN THIS LAST PART WE ARE GOING TO BRIEFLY INTRODUCE CURRENT TECHNIQUES OF UNSUPERVISED LEARNING WITH NEURAL NETWORKS

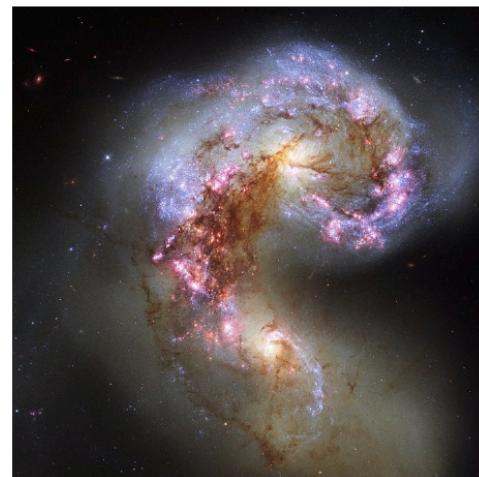
THERE ARE 3 MAJOR APPLICATIONS TO ASTRONOMY (THAT I CAN THINK OF):

- **DIMENSIONALITY REDUCTION:** HOW CAN I REPRESENT MY COMPLEX DATA MORE EFFICIENTLY TO GET NEW INSIGHTS INTO ITS STRUCTURE?
- **GENERATIVE MODELING:** HOW CAN I INTERPOLATE / EXTRAPOLATE A (SMALL, SPARSE) DATASET TO GENERATE NEW DATA SAMPLED FROM THE SAME (UNKNOWN) DISTRIBUTION?
- **PROBABILISTIC MODELING:** WHAT IS THE PROBABILITY THAT A NEW OBSERVATION IS DRAWN FROM THE SAME (UNKNOWN) DISTRIBUTION AS SOME REFERENCE (SMALL, SPARSE) DATASET?

DIMENSIONALITY REDUCTION

NEED OF REPRESENTING THE DATA IN A SMALLER
DIMENSIONALITY SPACE

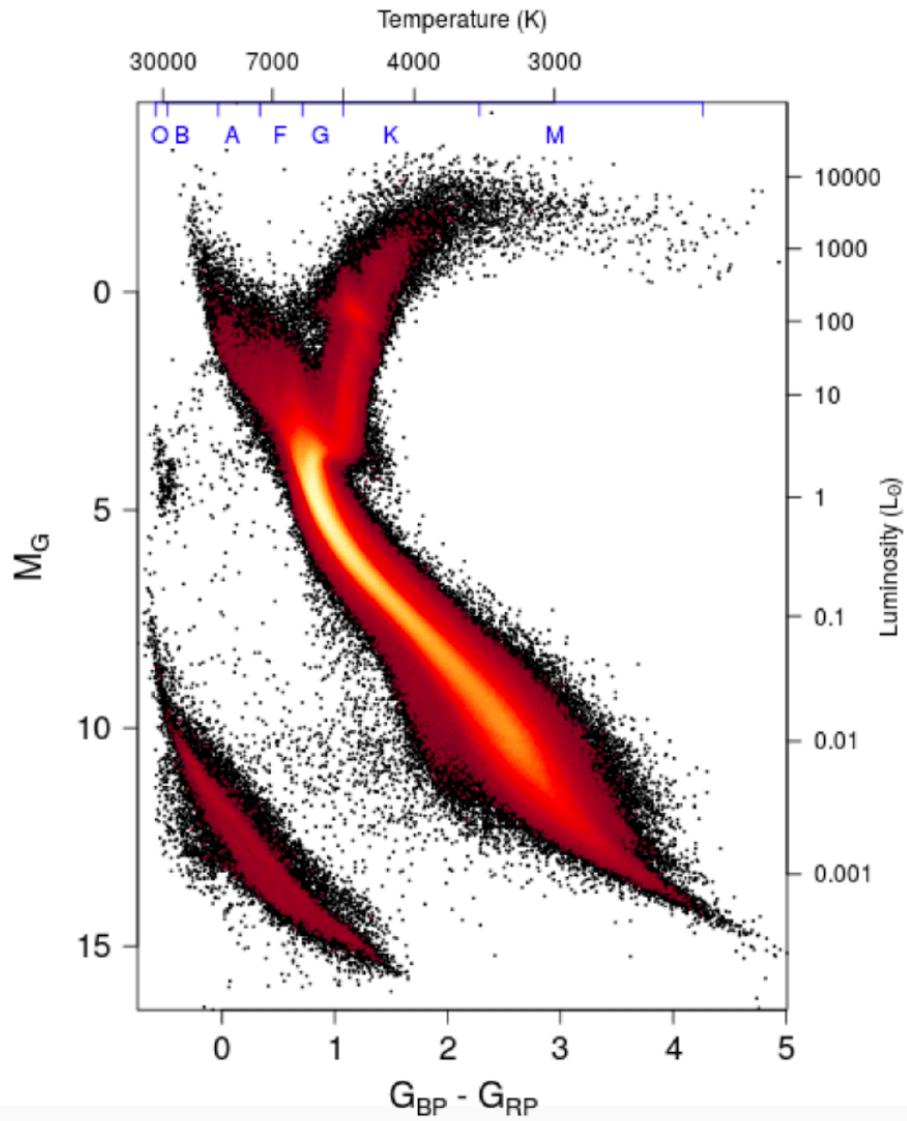
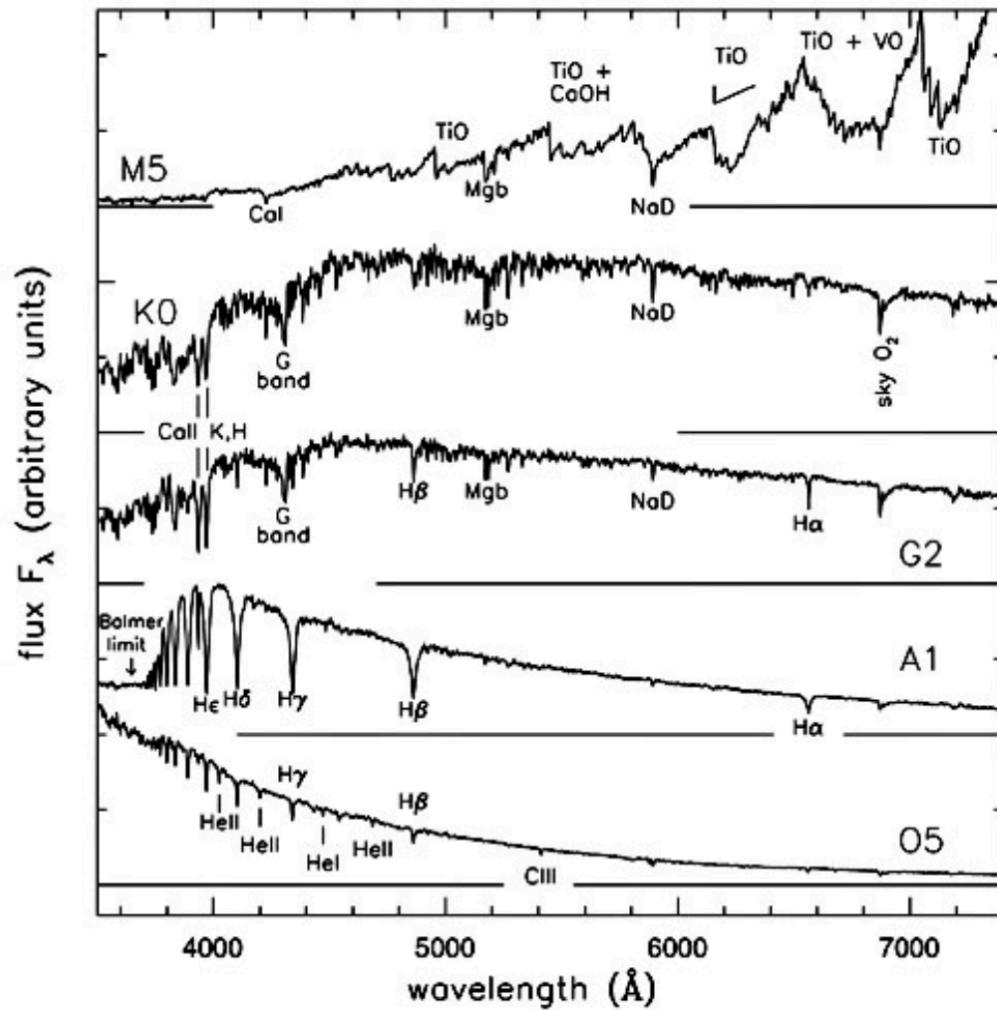
$$G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$



@D. BARON

MOST OF THE SCIENTIFIC WORK IS ABOUT COMPRESSION OF
NATURE IN SOME EXPLANATORY EQUATIONS

From: Gaia Collaboration et al. 2018



@D. BARON

**THE STELLAR MAIN SEQUENCE IS A DIMENSIONALITY
REDUCTION**

$$\begin{matrix} D \text{ dimensions (columns = features)} \\ N \text{ samples (rows)} \end{matrix} \boxed{\mathbf{X} - \boldsymbol{\mu}} = \boxed{\mathbf{Y}} \quad \boxed{\mathbf{M}} \quad d \text{ latent variables}$$

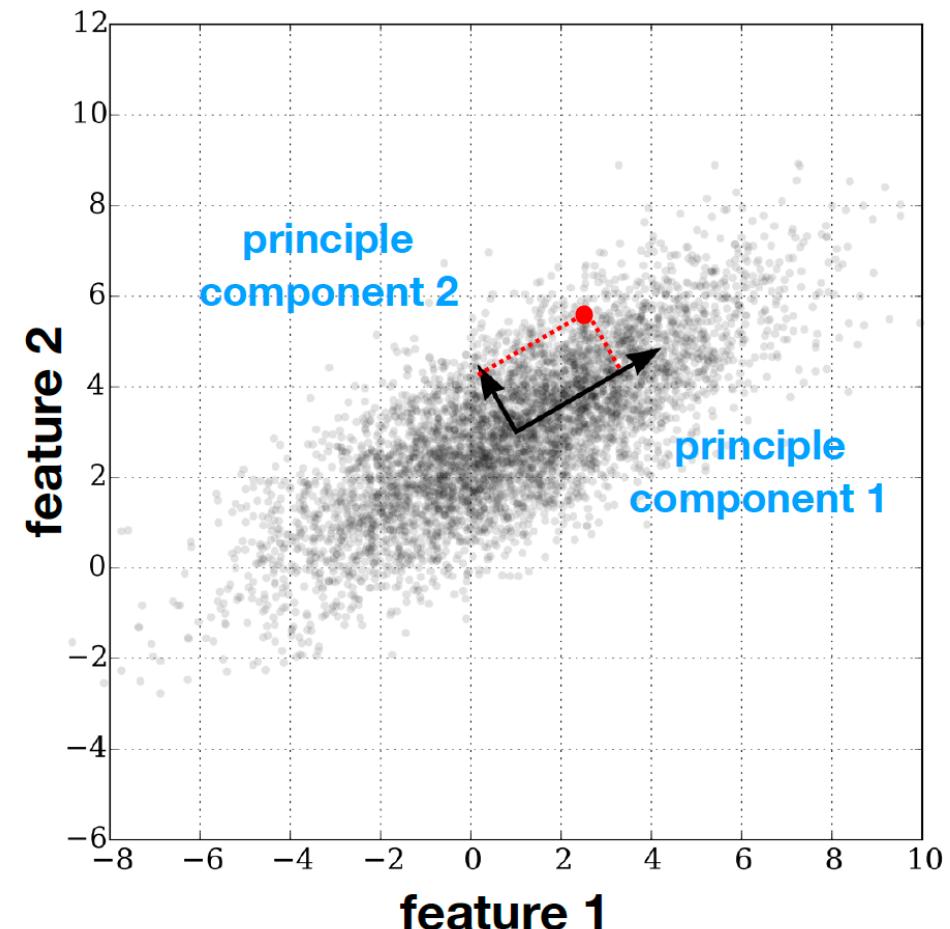
CLASSICAL METHODS FOR DIMENSIONALITY REDUCTION
SEEK A LINEAR DECOMPOSITION THAT BEST EXPLAINS
THE OBSERVATIONS \mathbf{X} IN TERMS OF **LATENT VARIABLES \mathbf{Y}**

PRINCIPAL COMPONENT ANALYSIS

PCA CONVERT A SET OF
(CORRELATED) VARIABLES INTO A
SET
OF VALUES LINEARLY
UNCORRELATED

1. THE FIRST PRINCIPLE COMPONENT (“PROTOTYPE”), HAS THE LARGEST POSSIBLE VARIANCE

2. THE FOLLOWING COMPONENTS HAVE THE LARGEST VARIANCES WITH THE ADDITIONAL CONSTRAINT THAT THEY ARE ORTHOGONAL TO THE PRECEDING COMPONENTS

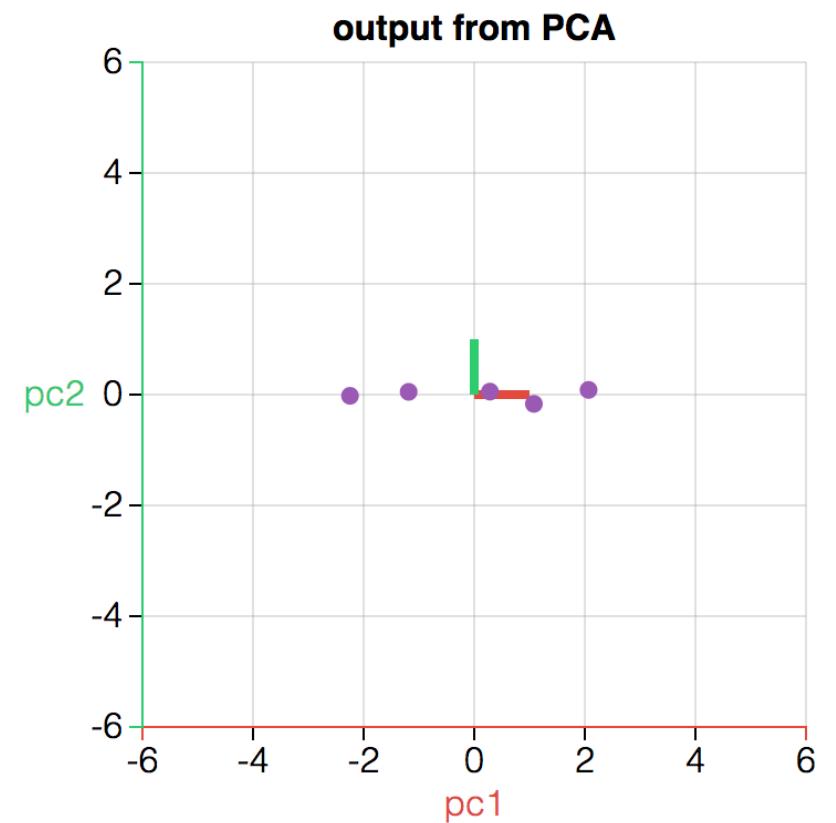
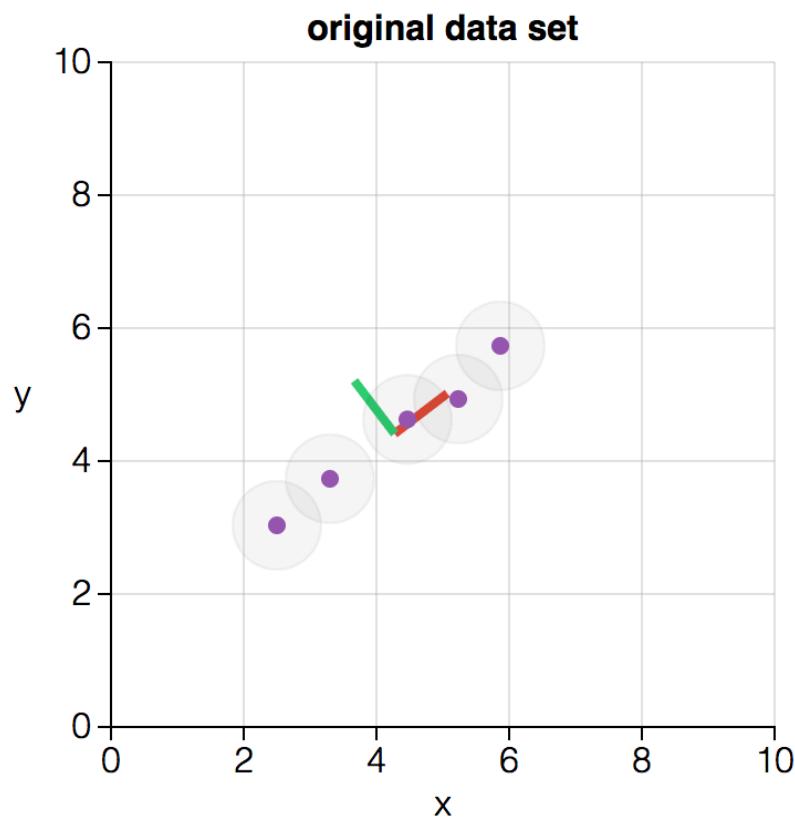


PRINCIPAL COMPONENT ANALYSIS

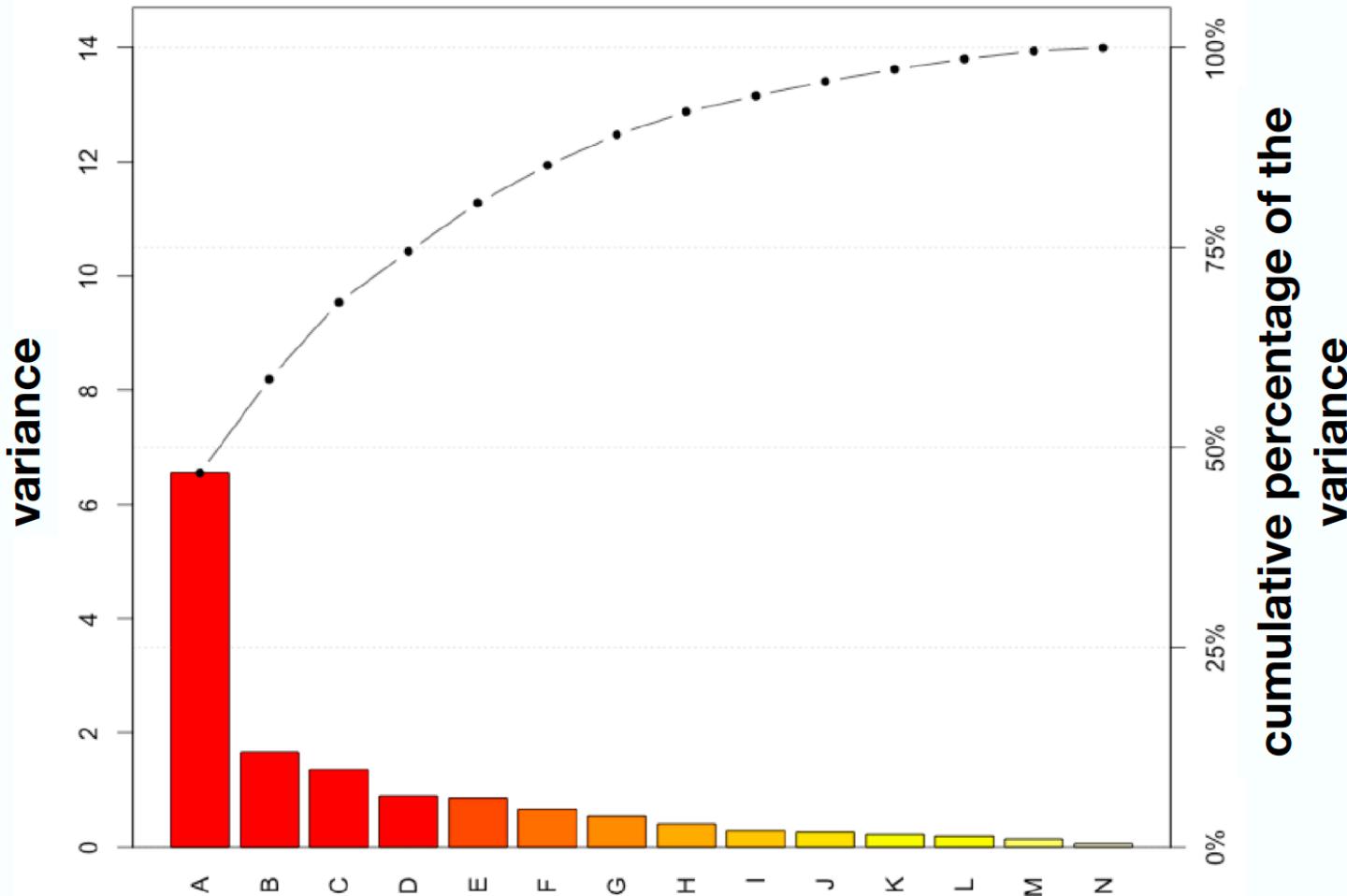
1. COMPUTE THE **COVARIANCE MATRIX** OF YOUR DATA
2. COMPUTE THE **EIGENVALUES AND EIGENVECTORS** OF THE **COVARAINCE MATRIX**
3. THE **EIGENVECTORS** PROVIDE THE DIRECTION OF MAXIMUM VARIANCE AND THE **EIGENVALUES** THE 'IMPORTANCE' OF THAT PARTICULAR FEATURE

REMEMBER, THE COVARIANCE MATRIX IS NOTHING ELSE THAN THAT:

$$K_{XX} = \begin{bmatrix} E[(X_1 - E[X_1])(X_1 - E[X_1])] & E[(X_1 - E[X_1])(X_2 - E[X_2])] & \cdots & E[(X_1 - E[X_1])(X_n - E[X_n])] \\ E[(X_2 - E[X_2])(X_1 - E[X_1])] & E[(X_2 - E[X_2])(X_2 - E[X_2])] & \cdots & E[(X_2 - E[X_2])(X_n - E[X_n])] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n - E[X_n])(X_1 - E[X_1])] & E[(X_n - E[X_n])(X_2 - E[X_2])] & \cdots & E[(X_n - E[X_n])(X_n - E[X_n])] \end{bmatrix}$$

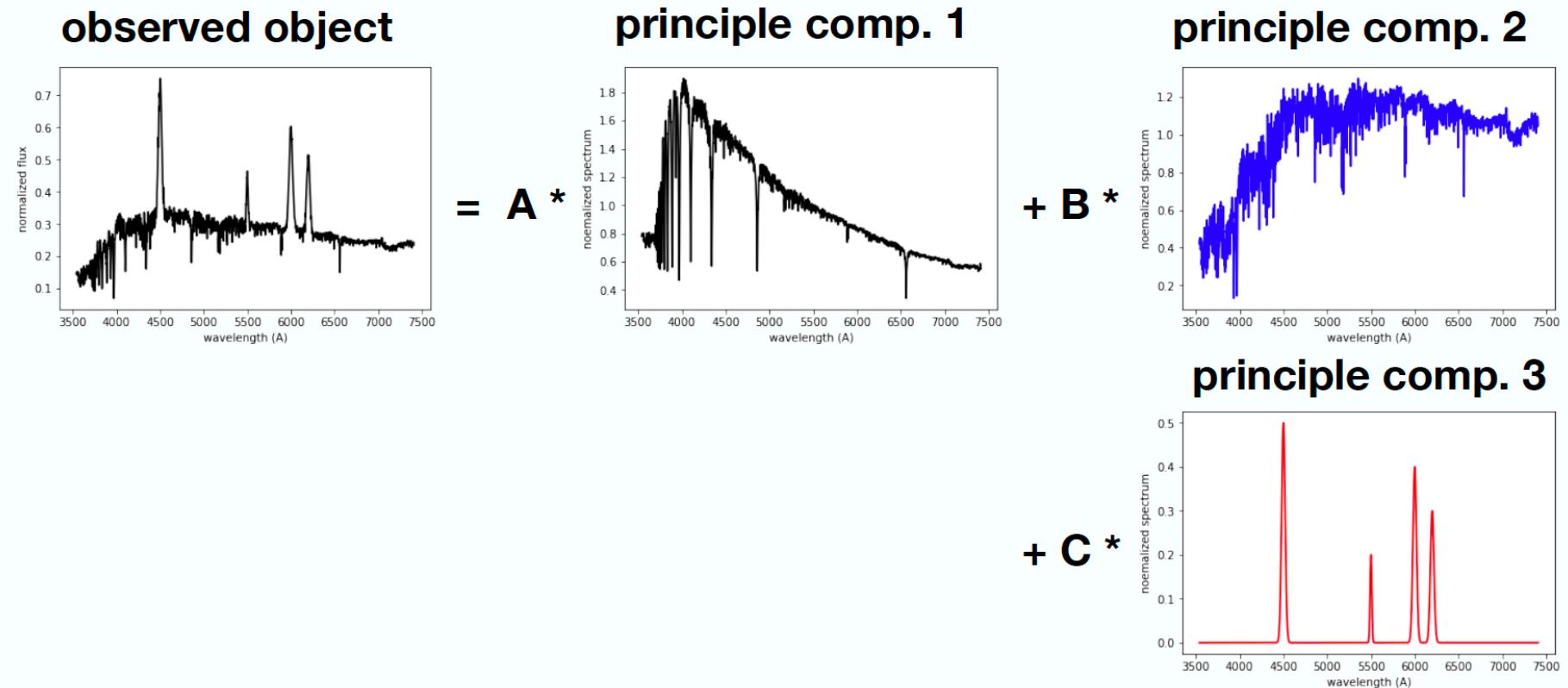


PRINCIPAL COMPONENT ANALYSIS



IT RESULTS IN DATA COMPRESSION,
BY REPRESENTING EACH OBJECT AS A PROJECTION OF THE FIRST PRINCIPLE COMPONENTS

PRINCIPAL COMPONENT ANALYSIS



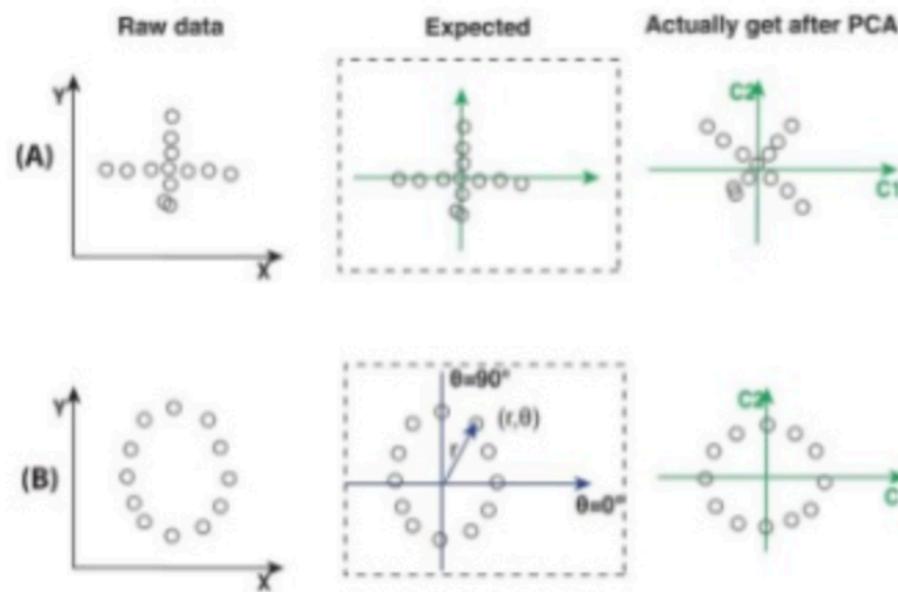
OBJECTS ARE DIVIDED INTO MAJOR PROTOTYPES AND ALL OBJECTS CAN BE OBTAINED AS LINEAR COMBINATIONS

@D. BARON

LIMITATIONS OF PCA

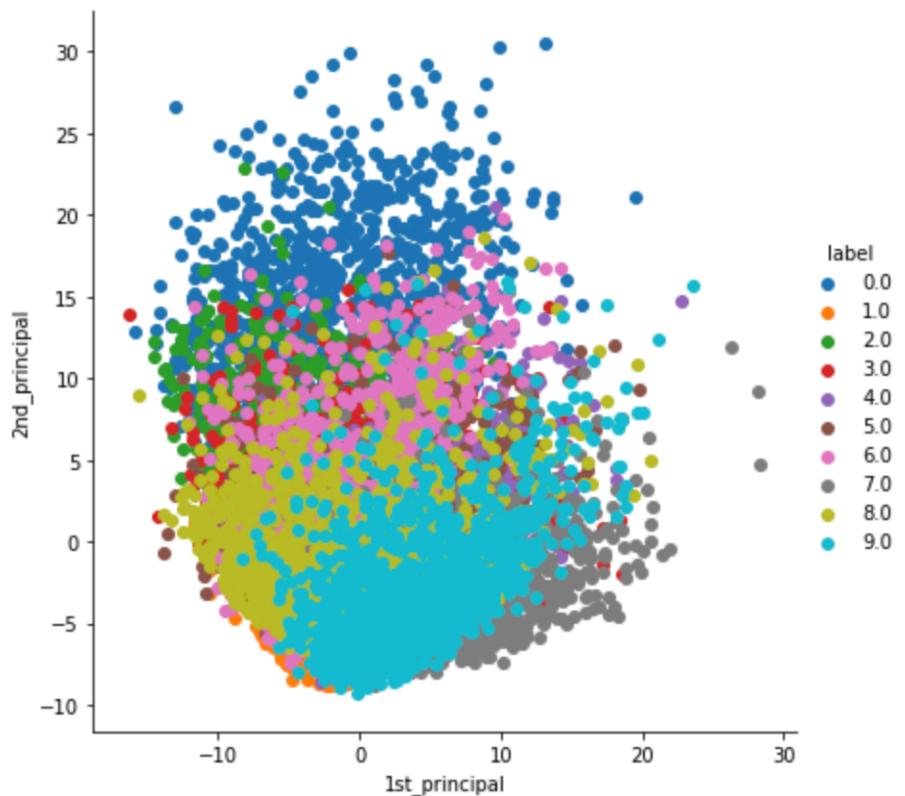
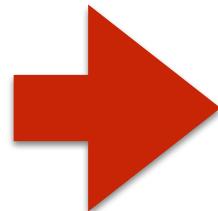
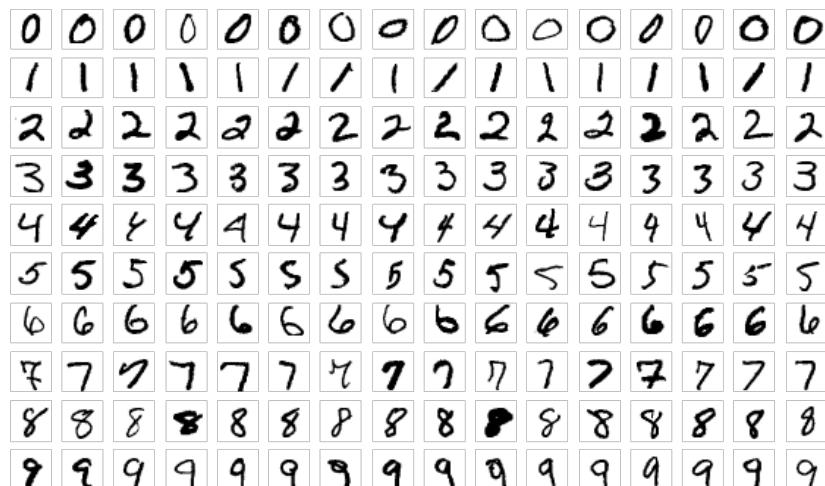
PCA APPLY LINEAR TRANSFORMATIONS

SINCE WE USE THE COVARIANCE MATRIX, IT ASSUMES THAT THE DATA FOLLOWS A **MULTIDIMENSIONAL NORMAL DISTRIBUTION**



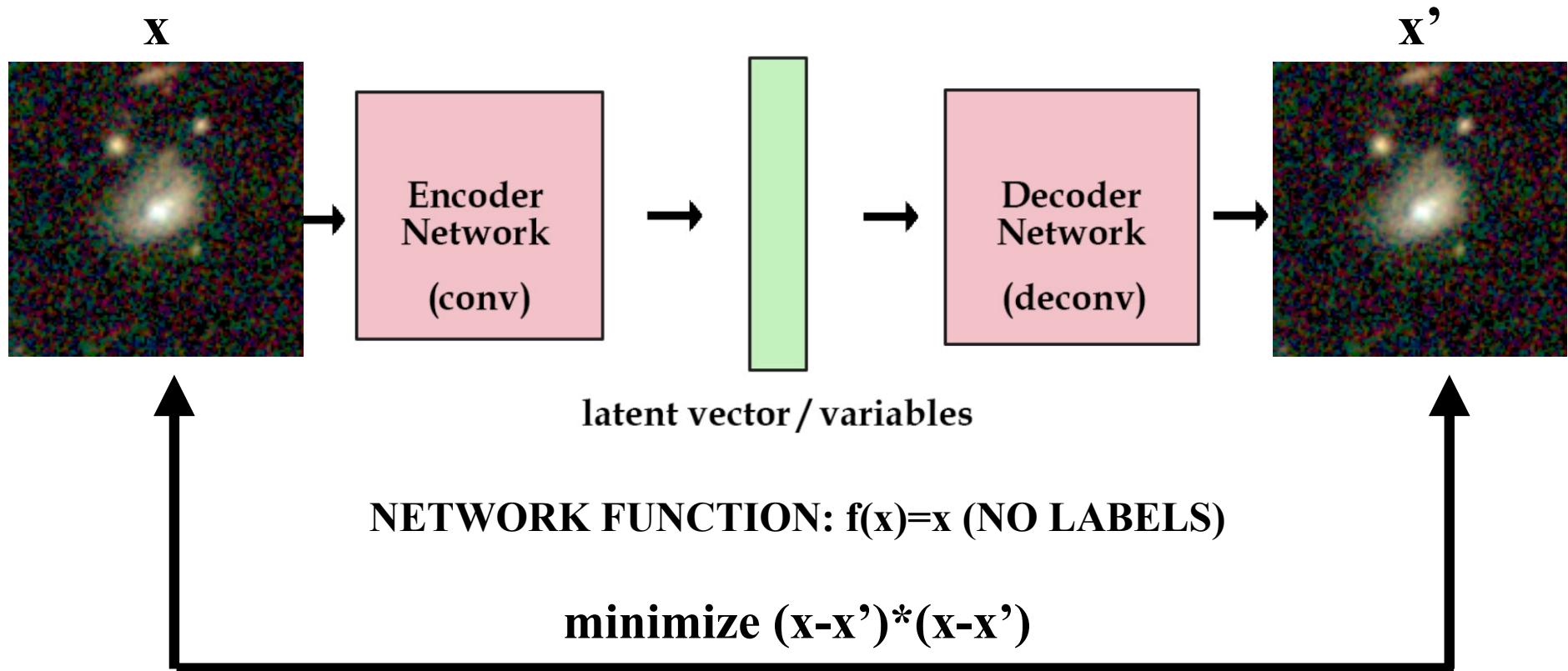
LIMITATIONS OF PCA

AND DATA IS NOT ALWAYS GAUSSIAN....



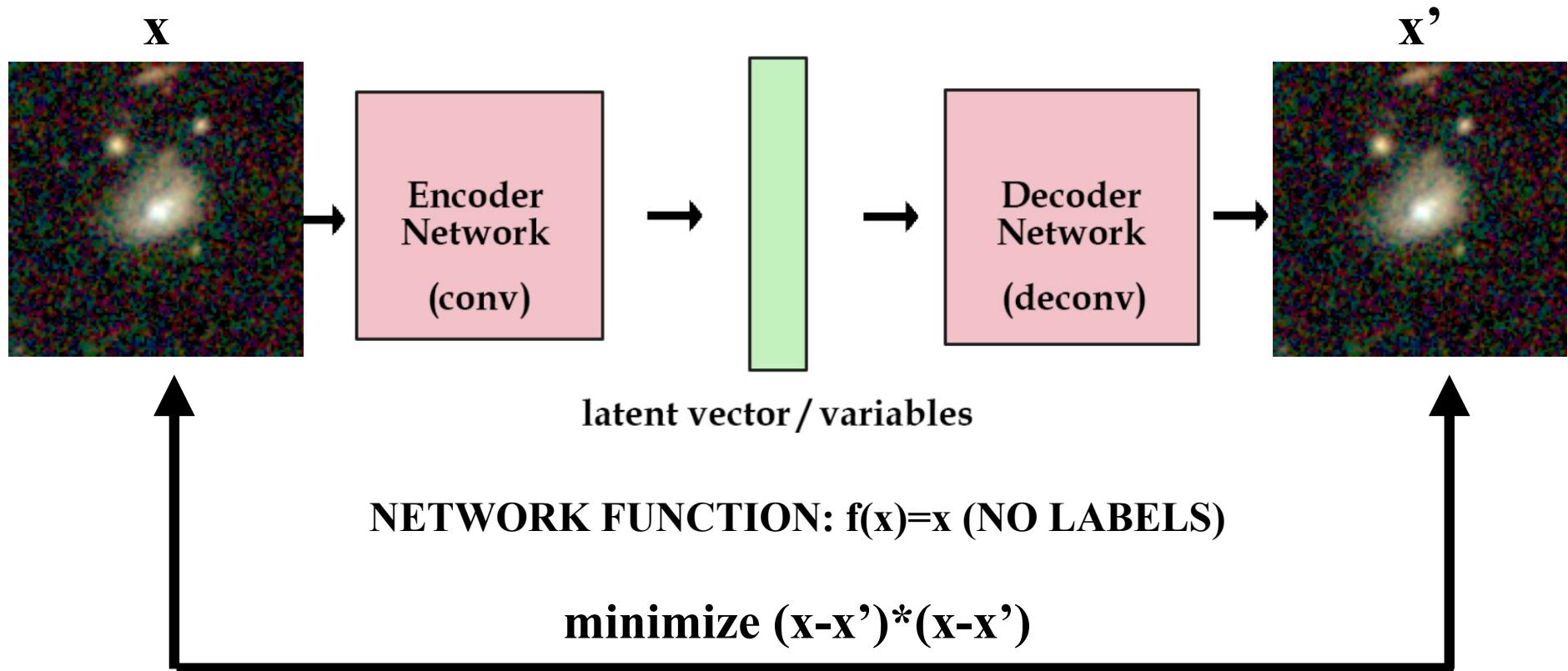
CAN WE GENERALIZE THAT?

CONVOLUTIONAL AUTO-ENCODER



AN AUTO-ENCODER IS ANY NETWORK WITH IDENTICAL INPUT AND OUTPUT

CONVOLUTIONAL AUTO-ENCODER



BY REDUCING THE DIMENSIONALITY IN THE LATENT SPACE WE FORCE THE NETWORK TO LEARN A REPRESENTATION OF THE INPUT DATA IN A LOWER DIMENSIONALITY SPACE

* NO NEED TO BE CONVOLUTIONAL - ANY NEURAL NETWORK WITH A BOTTLENECK WILL DO THE JOB

* **QUESTION:** WHAT WOULD HAPPEN IF WE SET AN AUTOENCODER WITH NO ACTIVATION FUNCTIONS?

SEE EXAMPLE FROM TUTORIALS FOR STAR-GALAXY SEPARATION

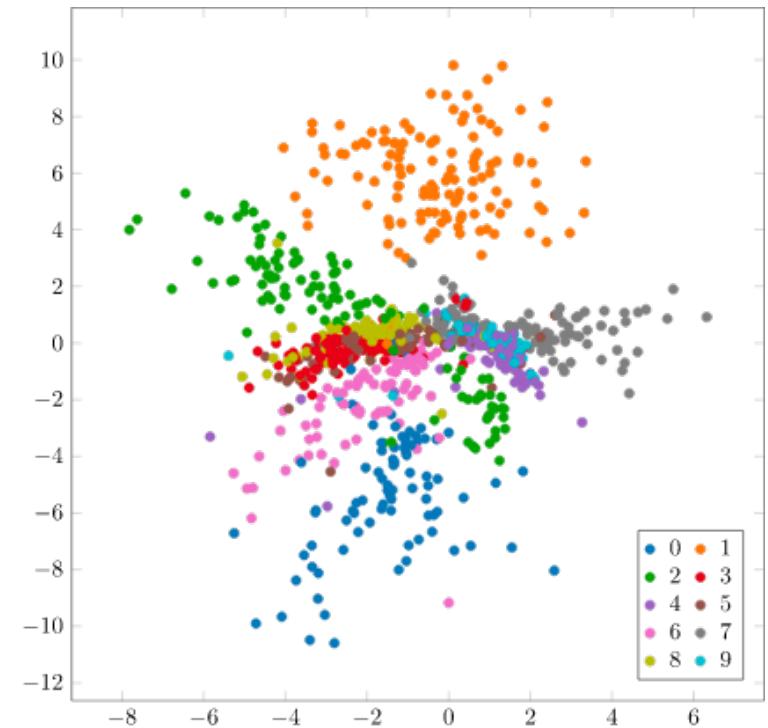
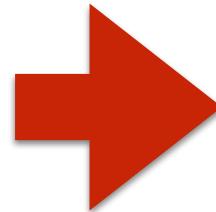
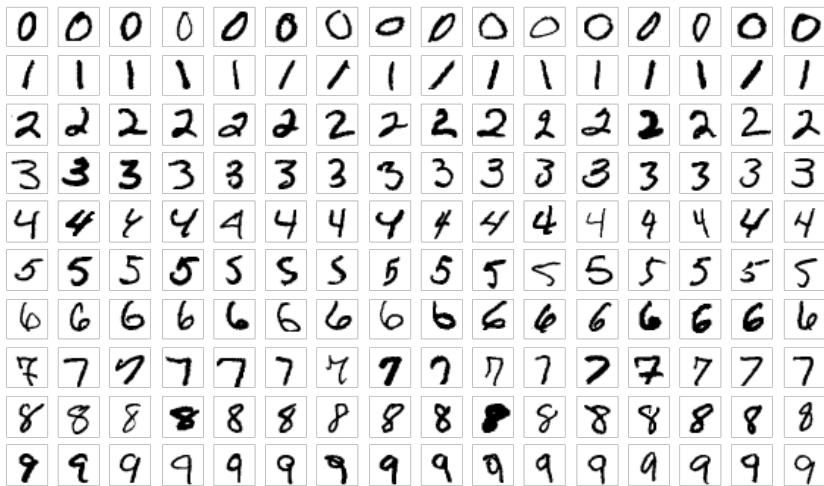
When poll is active, respond at **pollev.com/marchuertasc257**

Text **MARCHUERTASC257** to **22333** once to join

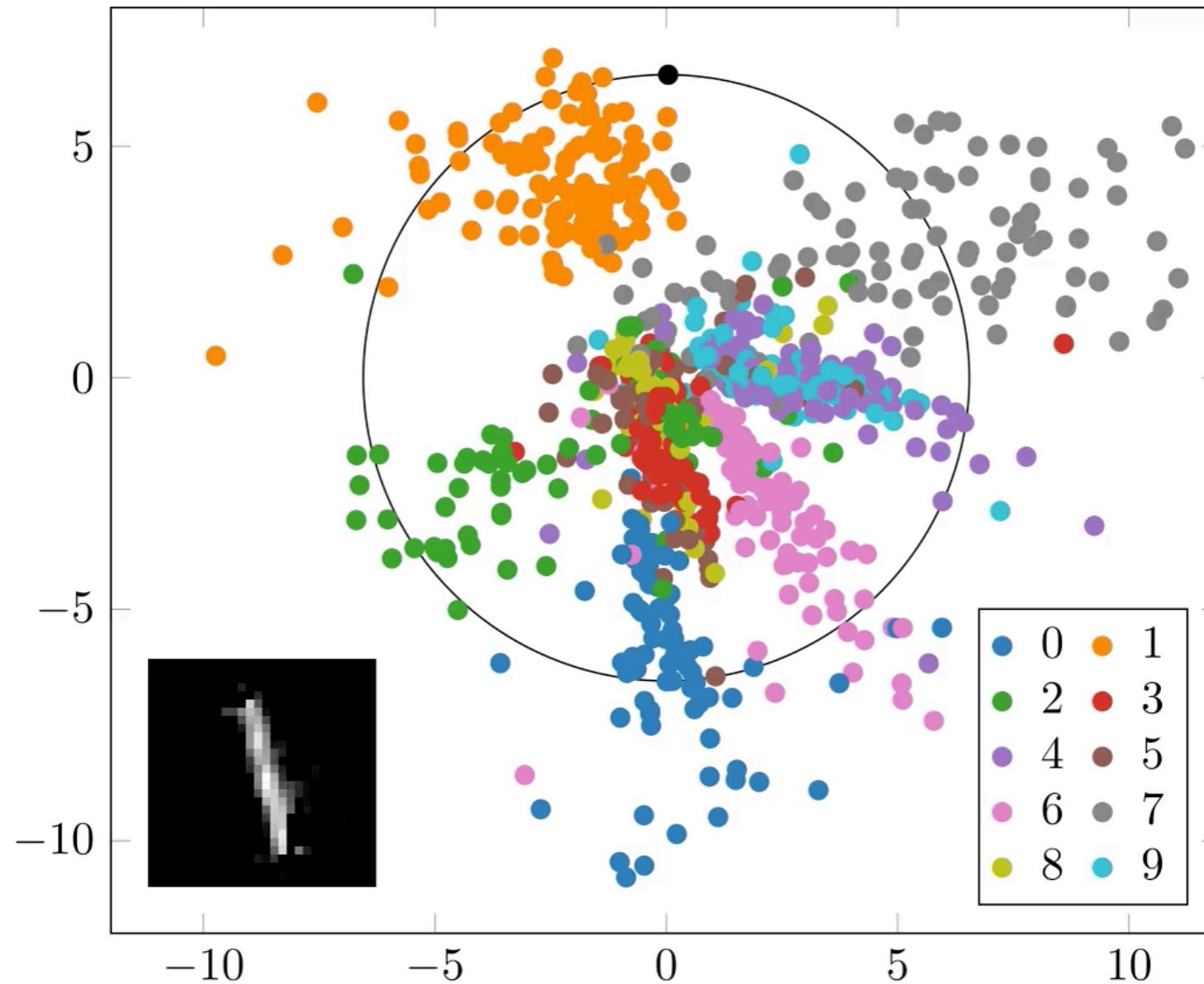
What would happen if we set an auto-encoder with no activation functions?

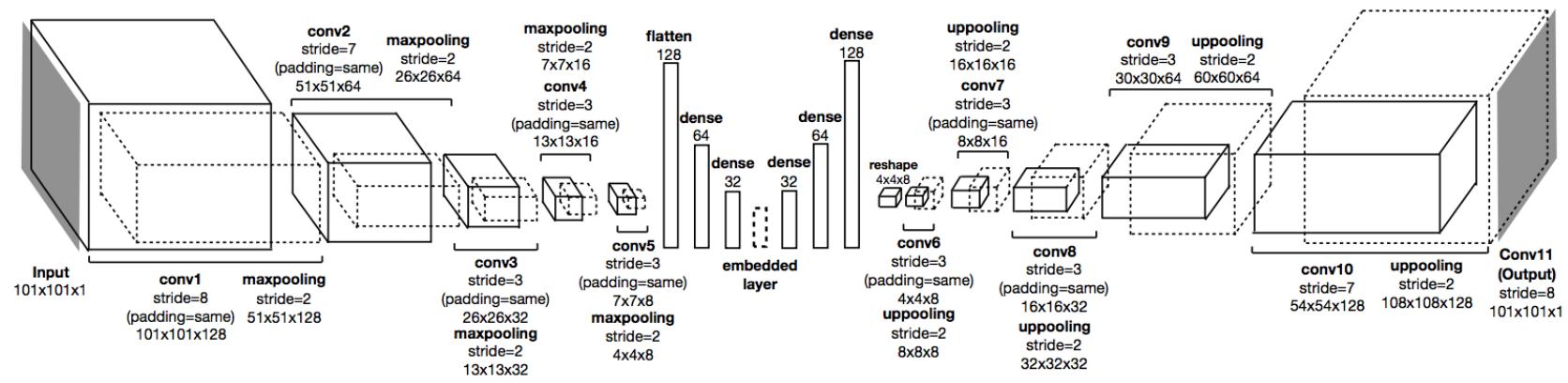
- nothing special - it would be a linear model
- PCA
- Depends on the input data
- Don't know

AUTOENCODER REPRESENTATION OF MNIST

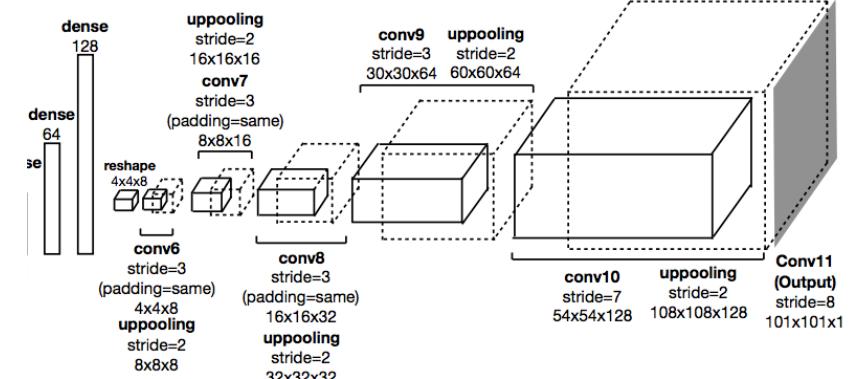
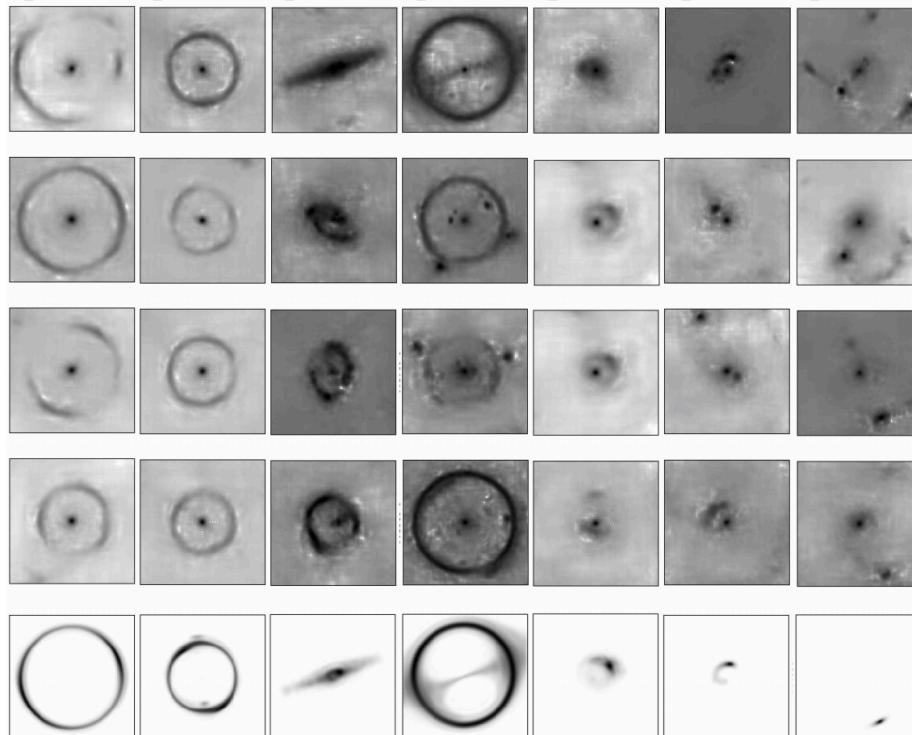


AUTOENCODER REPRESENTATION OF MNIST

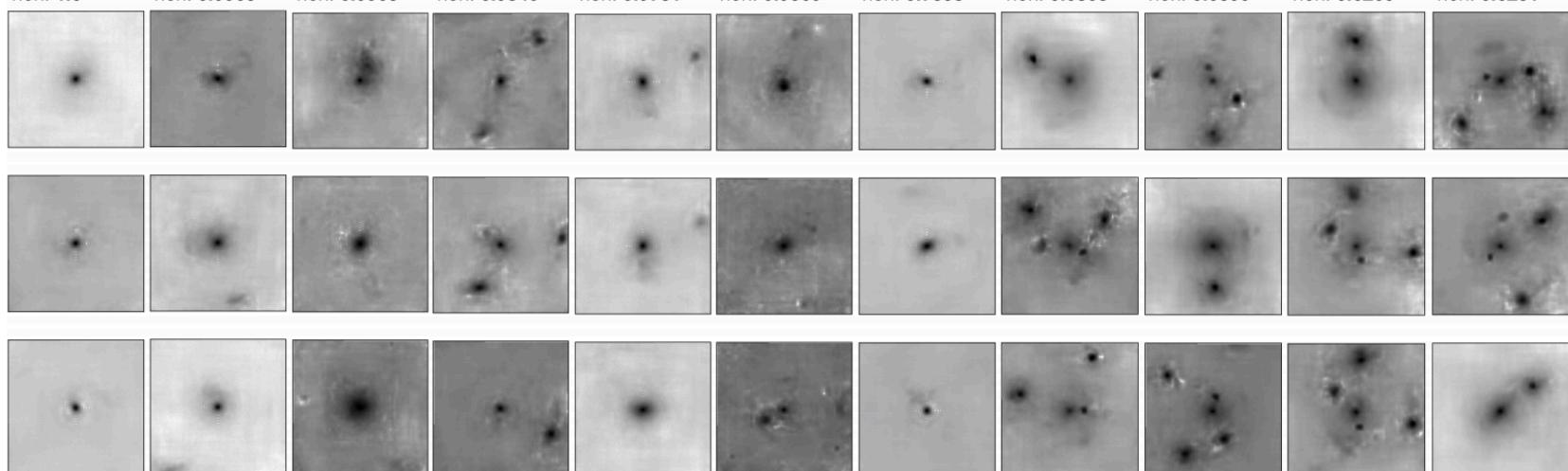




Cluster 17: lensing: 0.9873 non: 0.0127 F_len: 0.0914	Cluster 21: lensing: 0.9448 non: 0.0552 F_len: 0.0695	Cluster 1: lensing: 0.9159 non: 0.0841 F_len: 0.0731	Cluster 6: lensing: 0.8997 non: 0.1003 F_len: 0.0729	Cluster 2: lensing: 0.803 non: 0.197 F_len: 0.1945	Cluster 20: lensing: 0.6206 non: 0.3794 F_len: 0.0575	Cluster 5: lensing: 0.6170 non: 0.3830 F_len: 0.0734
---	---	--	--	--	---	--



Cluster 14: lensing: 0.0 non: 1.0	Cluster 3: lensing: 0.0037 non: 0.9963	Cluster 8: lensing: 0.0037 non: 0.9963	Cluster 22: lensing: 0.0154 non: 0.9846	Cluster 16: lensing: 0.0219 non: 0.9781	Cluster 4: lensing: 0.0431 non: 0.9569	Cluster 0: lensing: 0.2642 non: 0.7358	Cluster 18: lensing: 0.3132 non: 0.6868	Cluster 19: lensing: 0.3601 non: 0.6399	Cluster 13: lensing: 0.3731 non: 0.6269	Cluster 9: lensing: 0.3769 non: 0.6231
--	---	---	--	--	---	---	--	--	--	---



Cheng+20

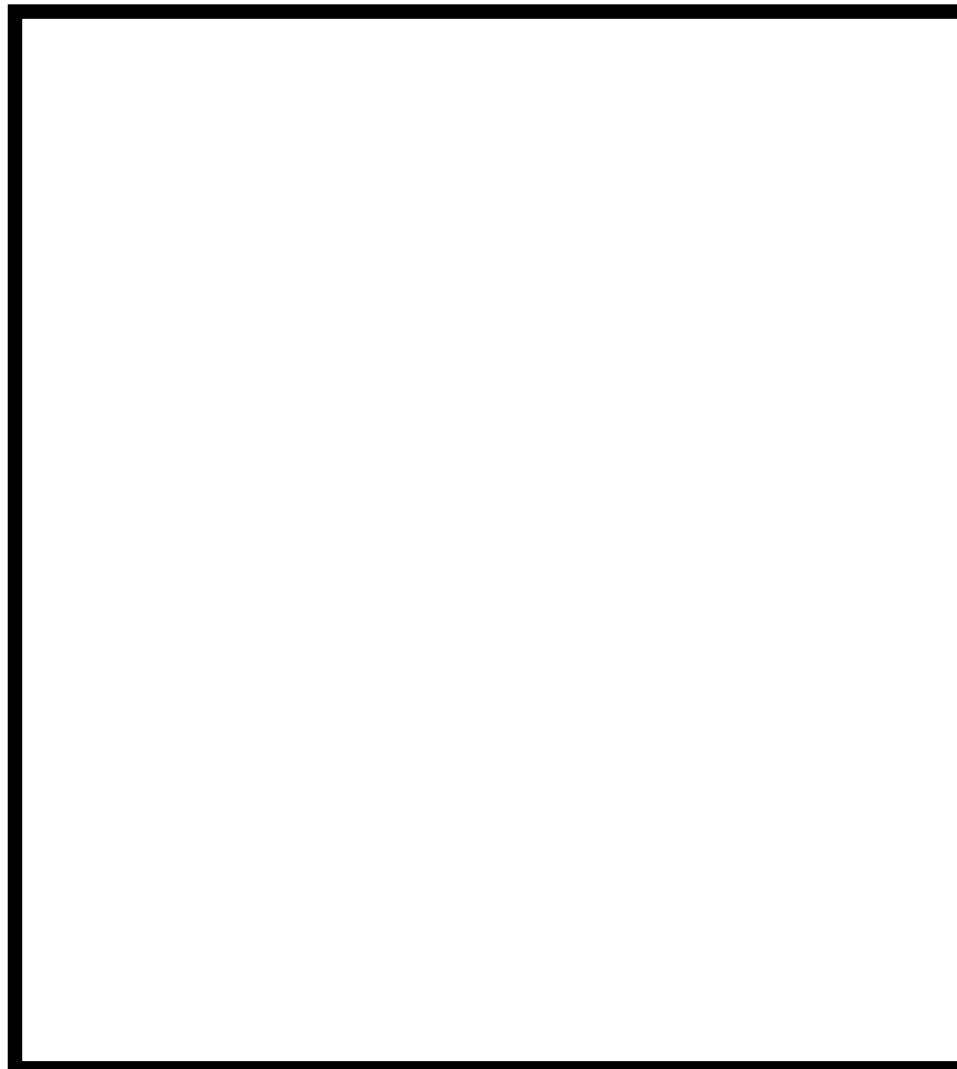
THIS IS OF COURSE NOT THE ONLY WAY TO OBTAIN USEFUL REPRESENTATIONS OF YOUR DATA....

IT IS AN ACTIVE FIELD OF RESEARCH BECAUSE LABELLING IS ONE OF THE MAIN BOTTLENECKS OF MACHINE LEARNING

- ▶ “Pure” Reinforcement Learning (**cherry**)
 - ▶ The machine predicts a scalar reward given once in a while.
 - ▶ A few bits for some samples
- ▶ Supervised Learning (**icing**)
 - ▶ The machine predicts a category or a few numbers for each input
 - ▶ Predicting human-supplied data
 - ▶ 10→10,000 bits per sample
- ▶ Self-Supervised Learning (**cake génoise**)
 - ▶ The machine predicts any part of its input for any observed part.
 - ▶ Predicts future frames in videos
 - ▶ Millions of bits per sample

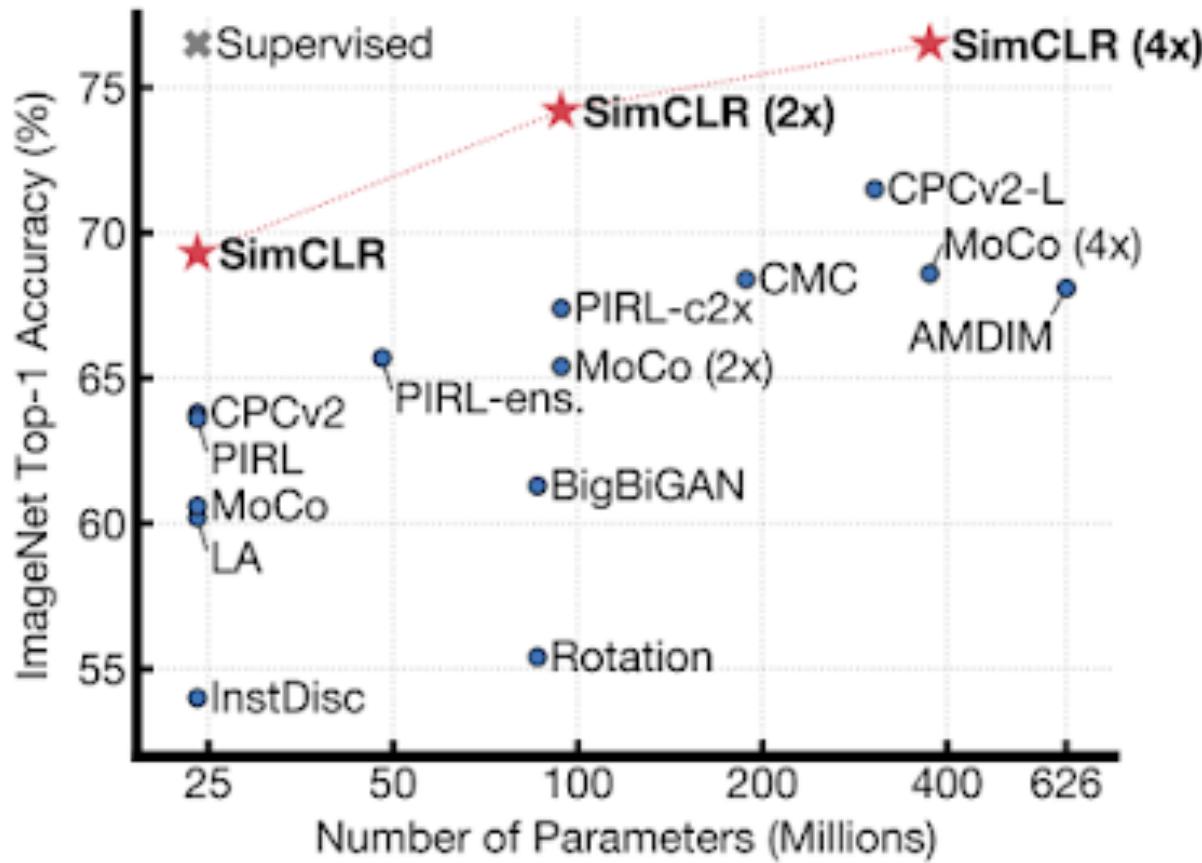


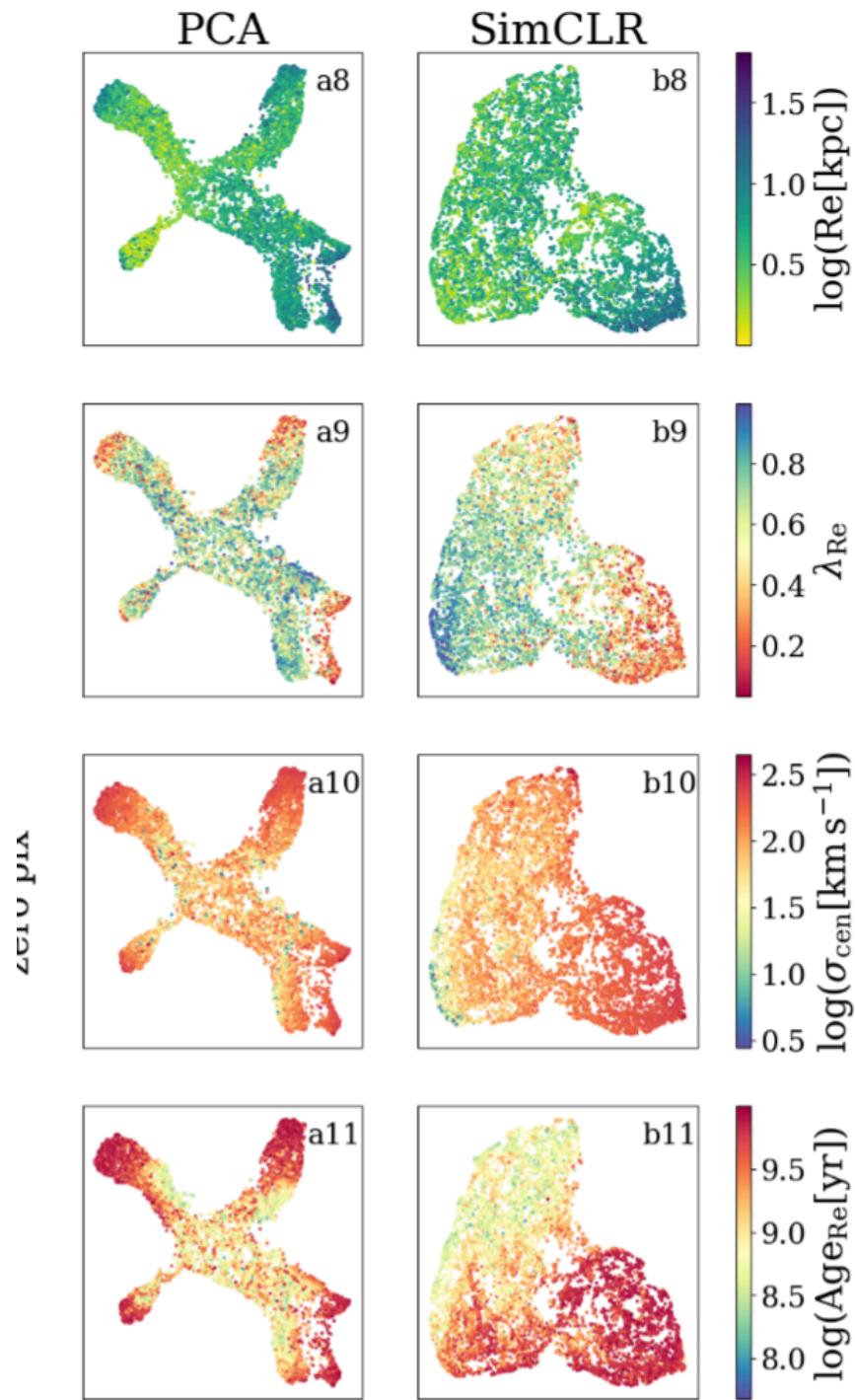
LEARNING REPRESENTATIONS THROUGH SELF-SUPERVISED LEARNING



Chen+20

SELF-SUPERVISED LEARNING REACHES COMPARABLE ACCURACY TO FULLY SUPERVISED APPROACHES...





Representation of Manga Galaxies with Self-Supervised Learning

Sarmiento+21

UMAP AND T-SNE ARE ALSO TWO USEFUL TECHNIQUES TO VISUALIZE LARGE DIMENSIONAL SPACES

INSTEAD OF MATRIX FACTORIZATION SUCH AS PCA, THEY
ARE BASED ON GRAPH LAYOUT

1. WE BUILD A GRAPH OF OUR HIGH-DIMENSIONALITY DATA
2. WE OPTIMIZE TO GET THE MOST SIMILAR GRAPH IN TWO DIMENSIONS

[https://
lvdmaaten.github.io/tsne/](https://lvdmaaten.github.io/tsne/)

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

UMAP EXAMPLES AND EXPLANATIONS;

<https://pair-code.github.io/understanding-umap/>
#:~:text=In%20the%20simplest%20sense%2C%20UMAP,behind%20them%
20is%20remarkably%20simple.

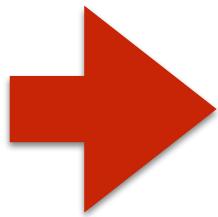
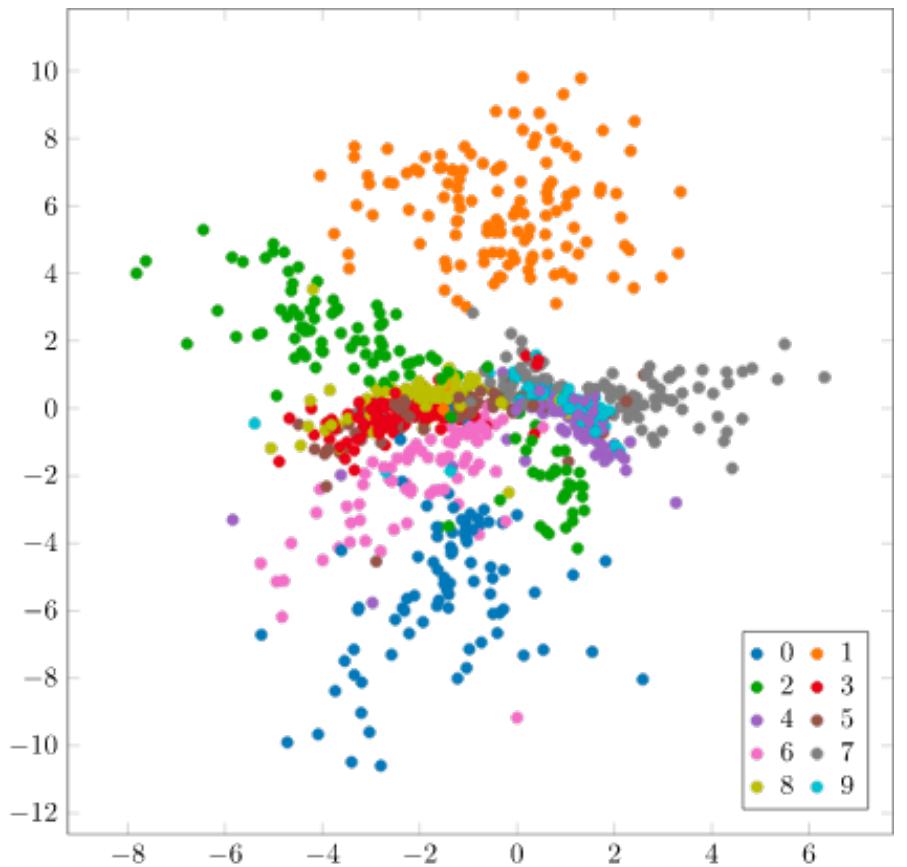
GO HERE:

https://github.com/mhuertascompany/Saas-Fee/blob/main/hands-on/session3/CAE_Star_Galaxy_Saas_Fee.ipynb

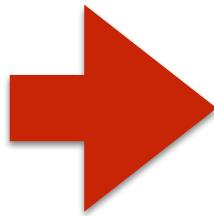
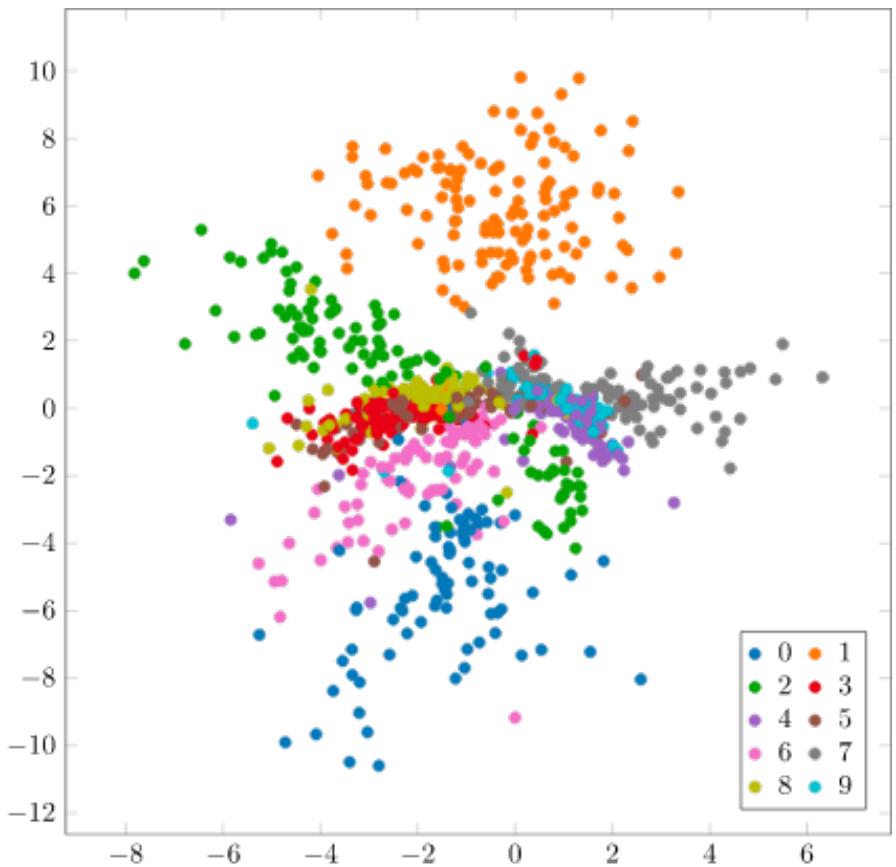
HOW MIGHT YOU SOLVE THESE
RELATED PROBLEMS ONCE YOU HAVE
THE LATENT SPACE COORDINATES FOR
YOUR TRAINING SAMPLE?

GENERATE A RANDOM SAMPLE DRAWN FROM THE INPUT
DISTRIBUTION ("**GENERATIVE MODEL**")

ESTIMATE THE PROBABILITY DENSITY OF AN ARBITRARY
INPUT, RELATIVE TO THE INPUT DISTRIBUTION
 ("**PROBABILISTIC MODEL**")



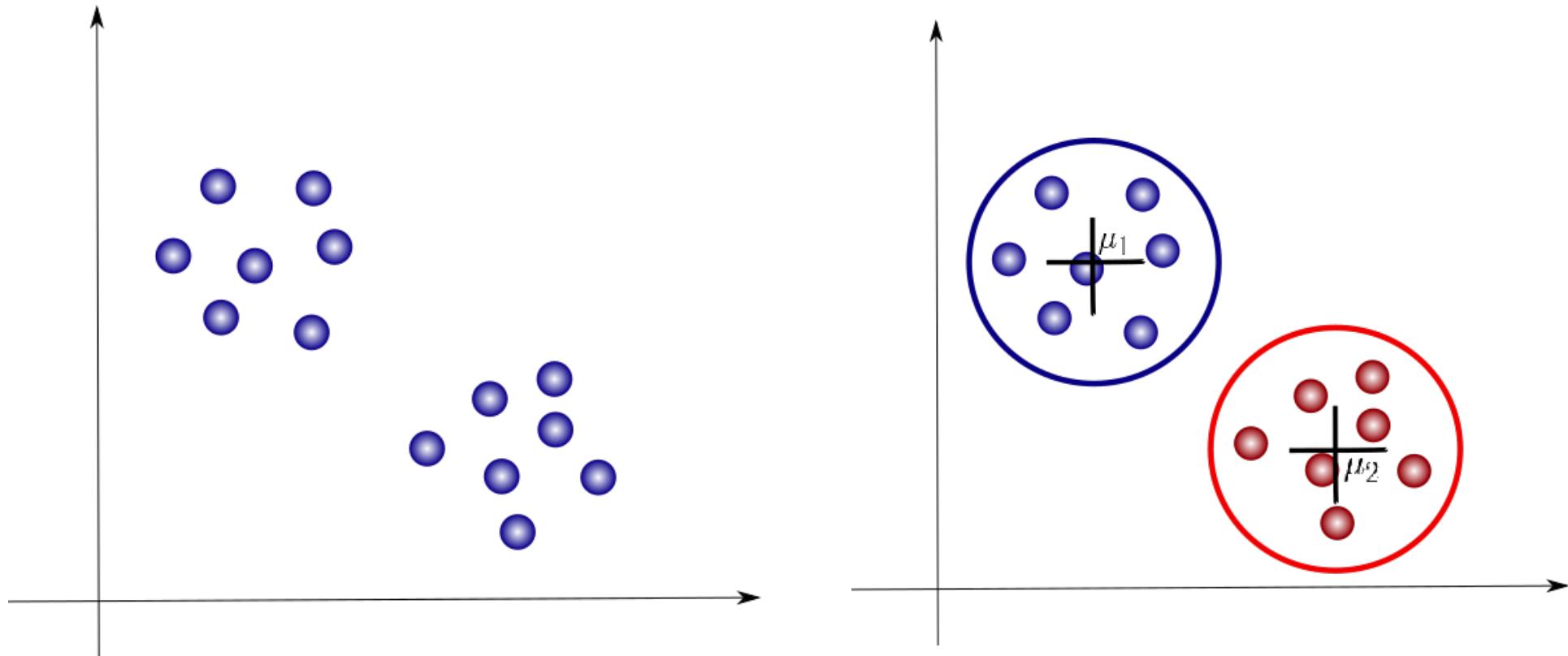
**HOW CAN I ESTIMATE
 $P(X)$?**



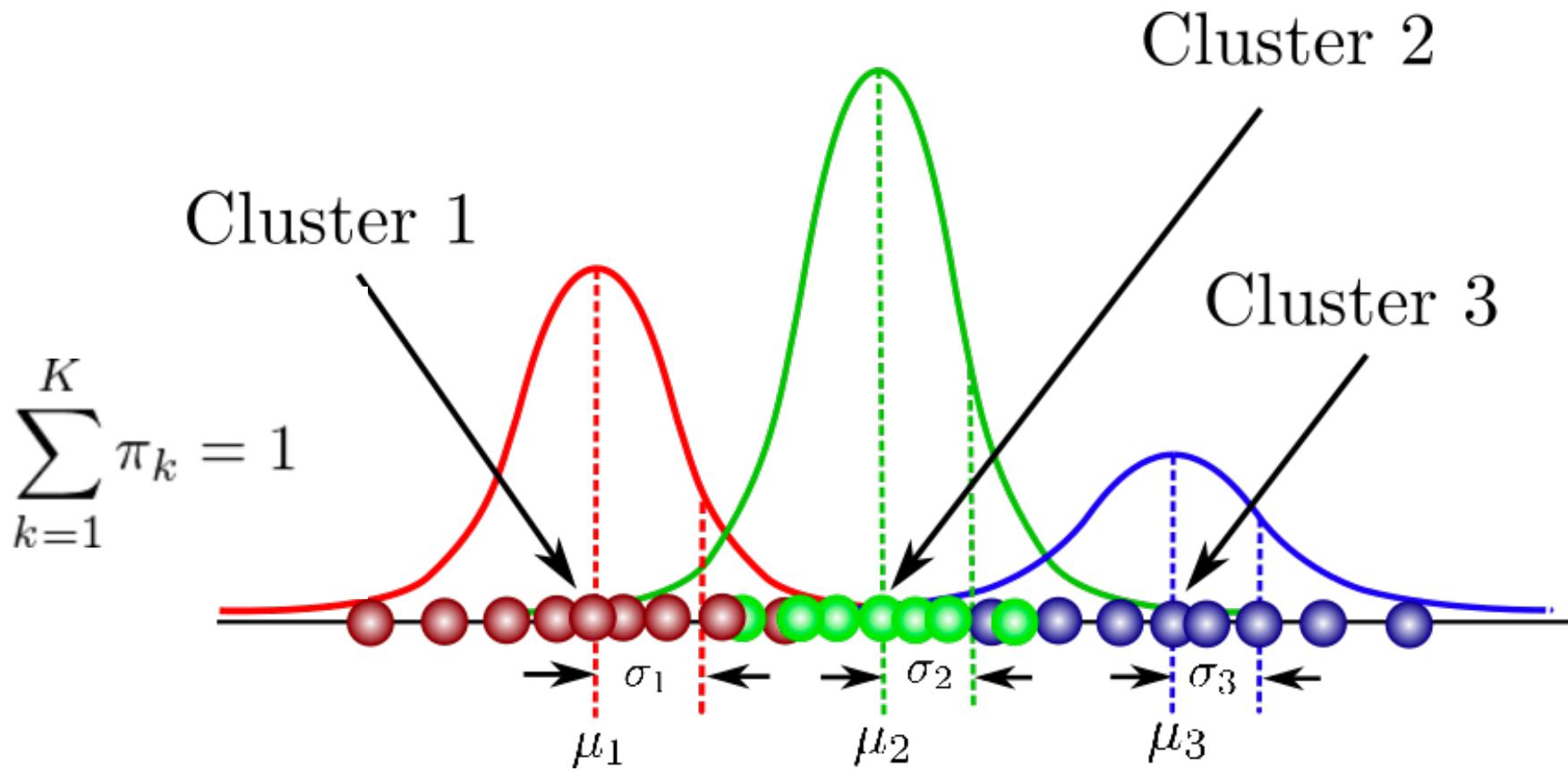
**HOW CAN I ESTIMATE
 $P(X)$?**

WHEN YOU DON'T KNOW, ASSUME IT IS GAUSSIAN....

GAUSSIAN MIXTURE MODELS (GMMs) ARE DENSITY ESTIMATOR METHODS THAT FIT MULTIPLE GAUSSIANS TO THE REPRESENTATION

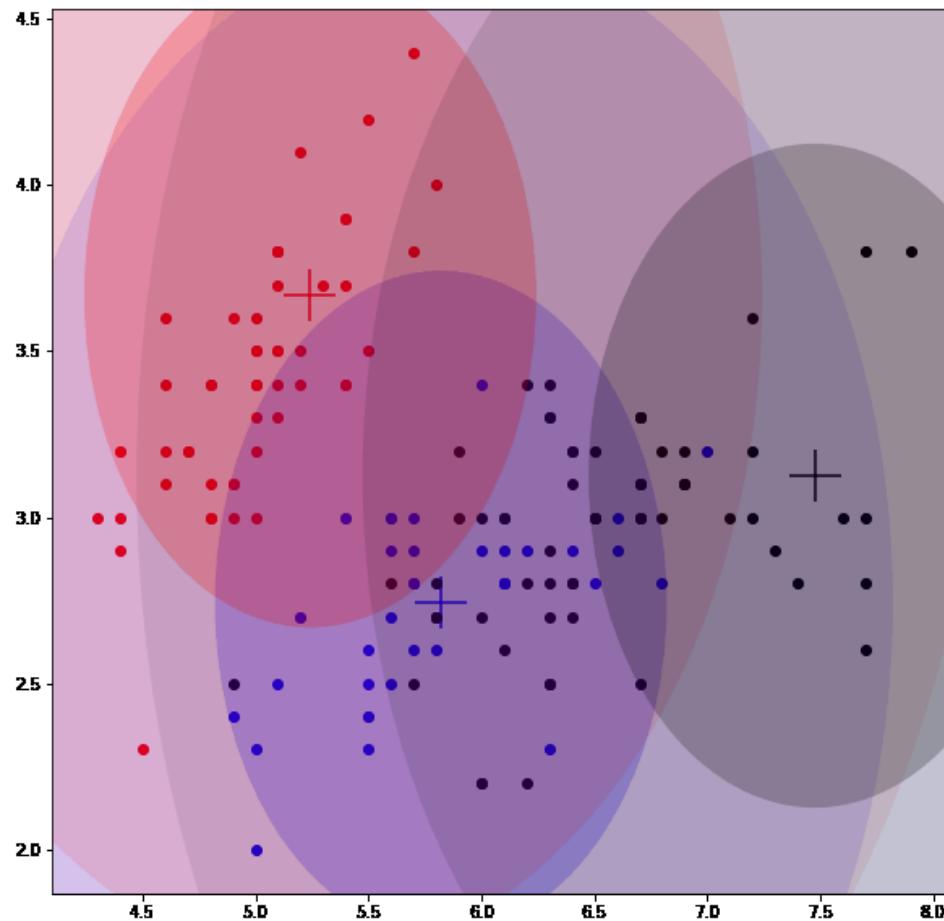


GAUSSIAN MIXTURE MODELS (GMMs) ARE DENSITY ESTIMATOR METHODS THAT FIT MULTIPLE GAUSSIANS TO THE REPRESENTATION

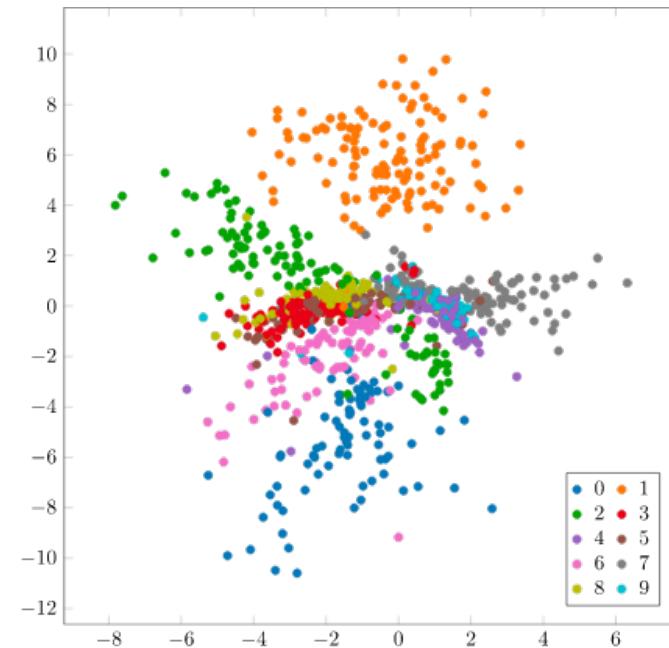
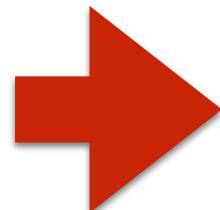
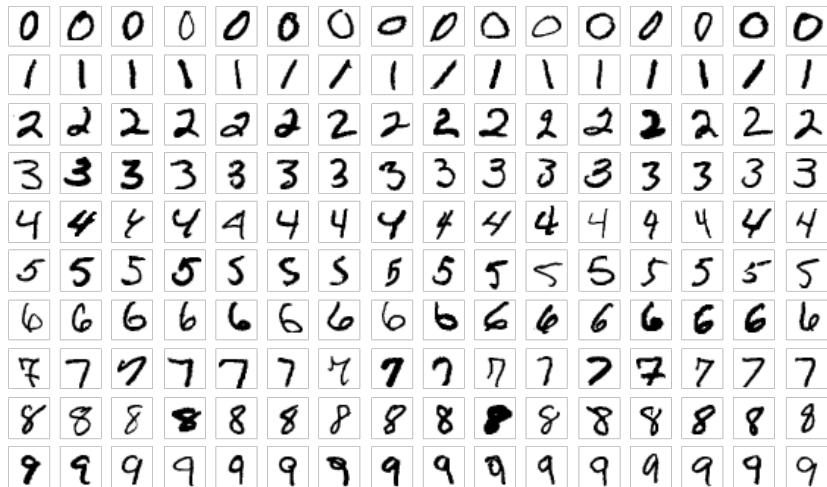


means, sigmas and scale factors of each gaussian are free parameters

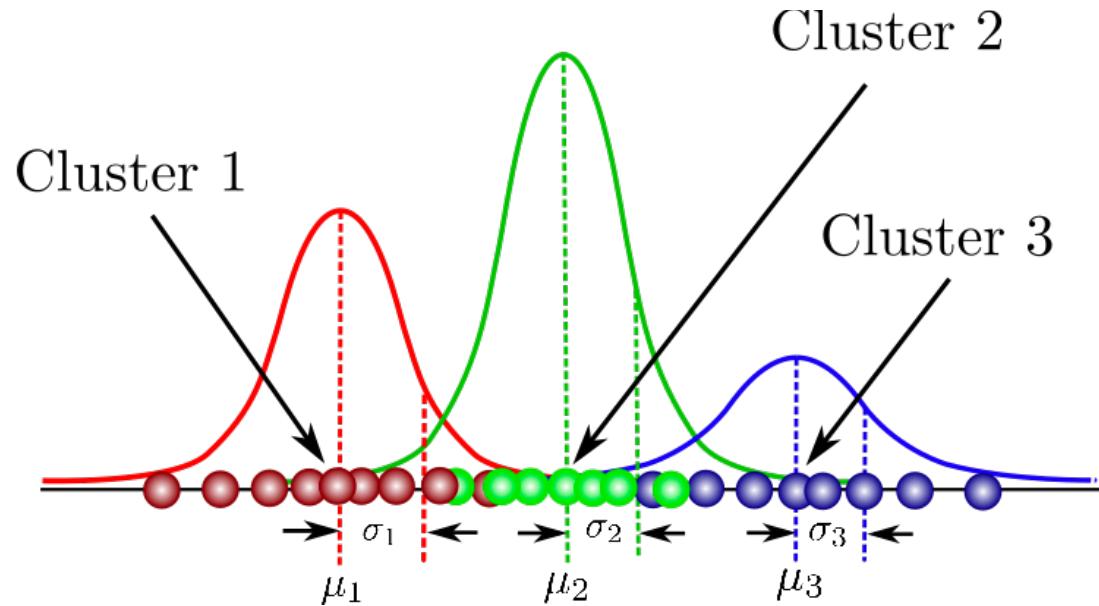
GAUSSIAN MIXTURE MODELS (GMMs) ARE DENSITY ESTIMATOR METHODS THAT FIT MULTIPLE GAUSSIANS TO THE REPRESENTATION



DATA



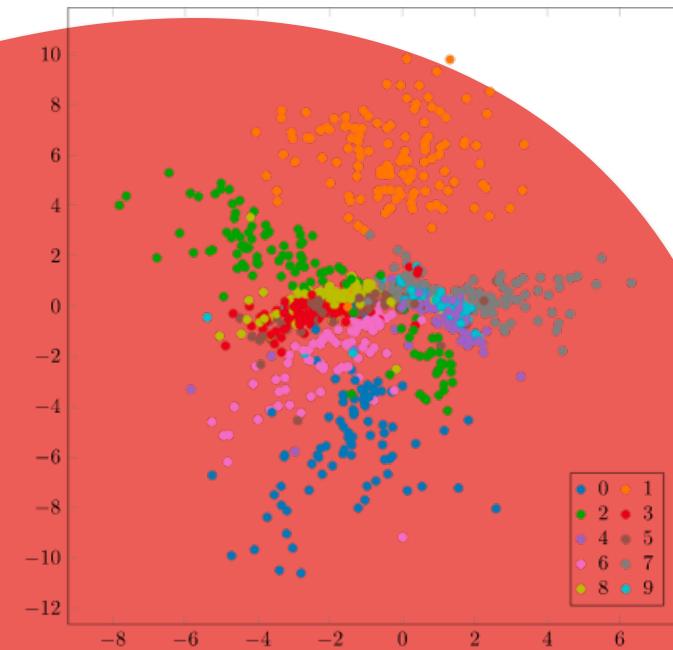
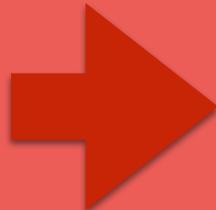
REPRESENTATION
(AUTOENCODER)



MODELING OF $P(X)$ WITH GMMs

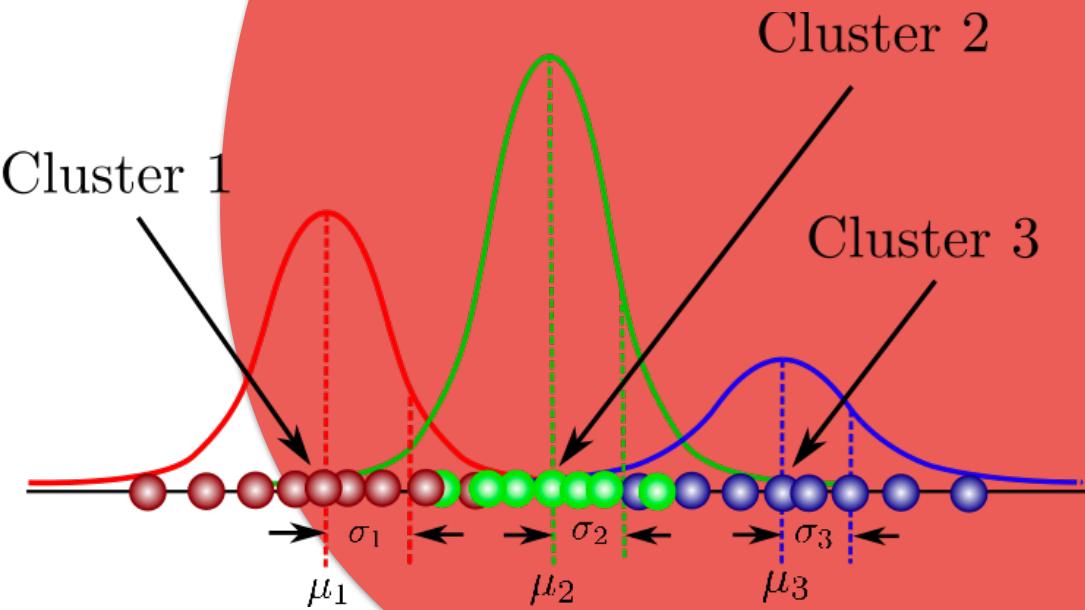
DATA

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9



REPRESENTATION (AUTOENCODER)

CAN WE COMBINE THESE 2 STEPS?



MODELING OF $P(X)$ WITH GMMs

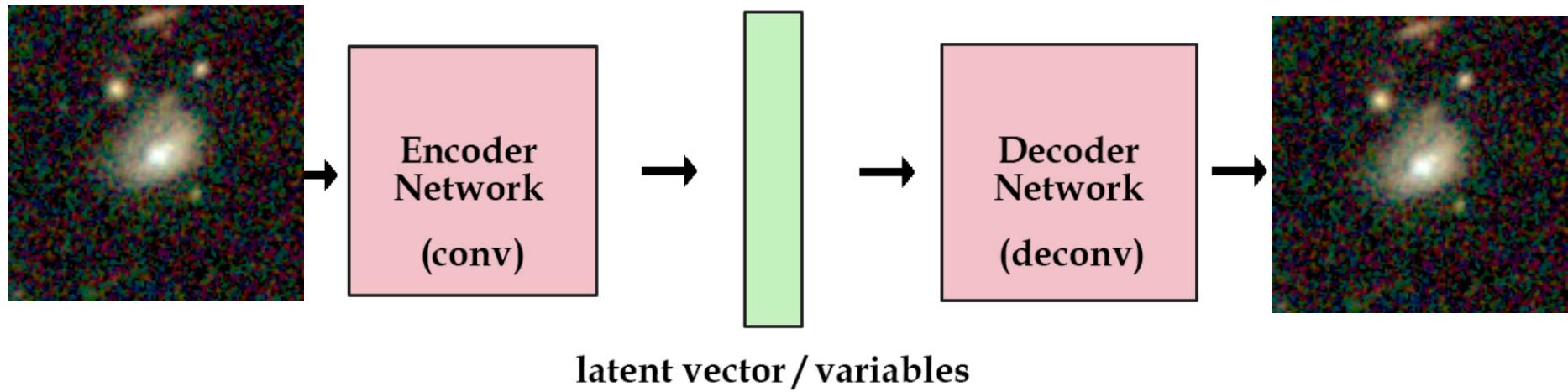
Much of the recent progress in unsupervised deep learning has been to invent network architectures that are capable of solving either or both of these related problems directly, without resorting to any auxiliary methods

VAE
(VARIATIONAL AUTOENCODER)

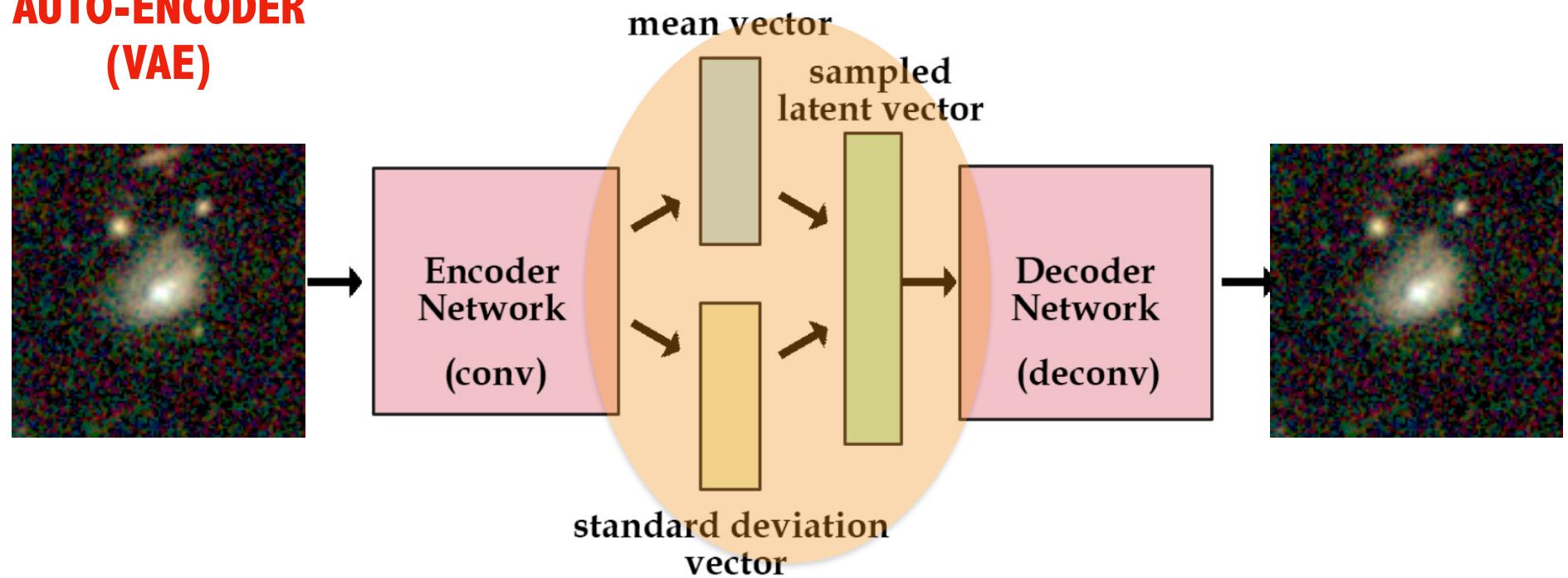
GAN
(GENERATIVE ADVERSARIAL NETWRK)

ARF
(AUTOREGRESSIVE FLOWS)

AUTO-ENCODER

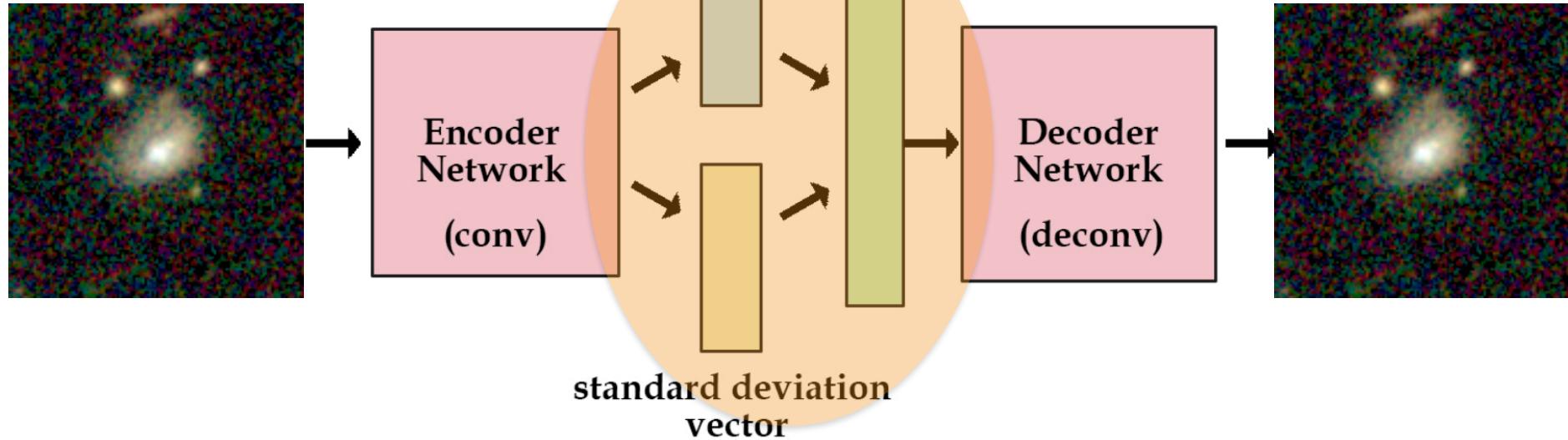


VARIATIONAL AUTO-ENCODER (VAE)



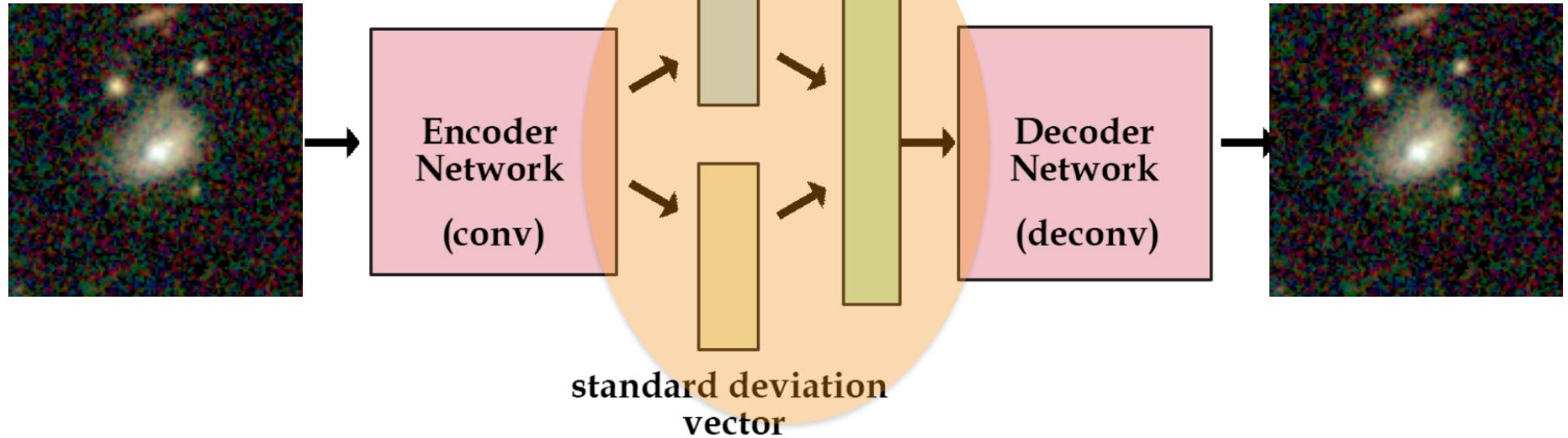
LET'S MODEL THE LATENT SPACE WITH A MIXTURE OF GAUSSIANS

VARIATIONAL AUTO-ENCODER (VAE)



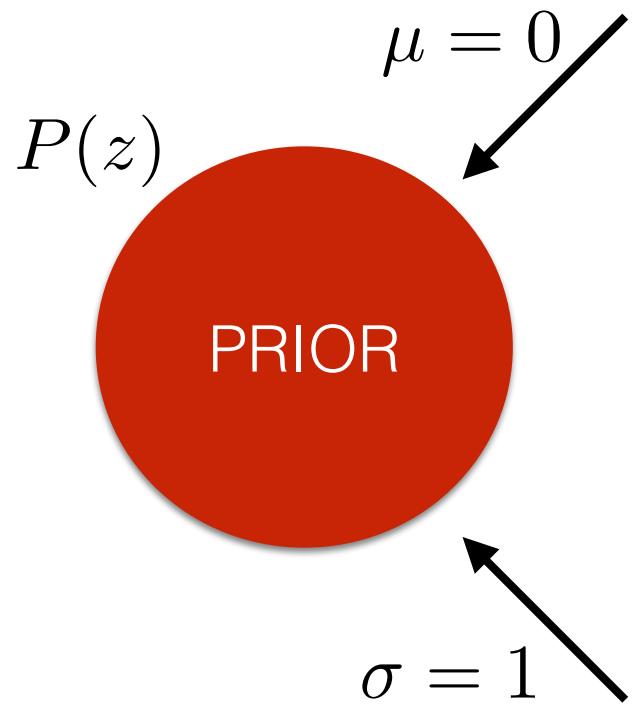
HOWEVER, NOTHING GUARANTEES US THAT THE LATENT SPACE CAN BE MODELLED BY A MIXTURE OF GAUSSIANS....

VARIATIONAL AUTO-ENCODER (VAE)

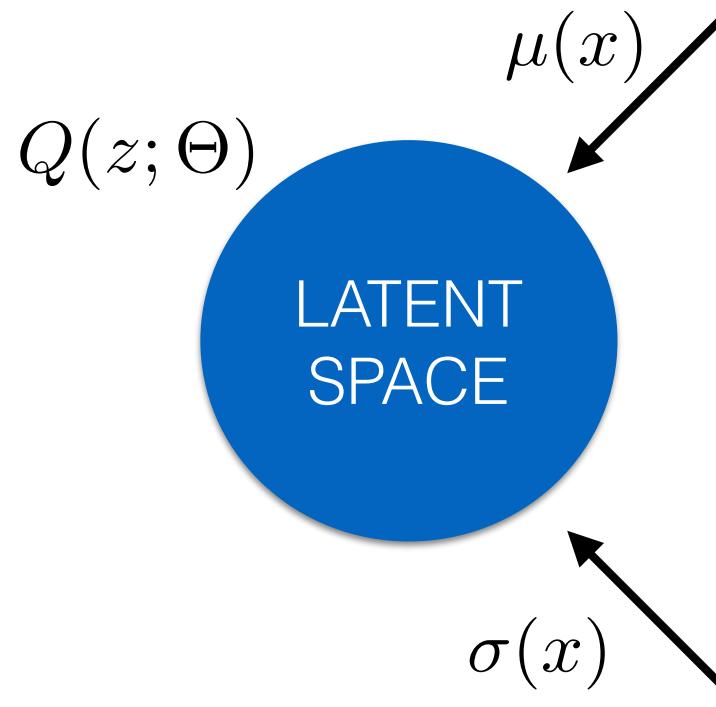
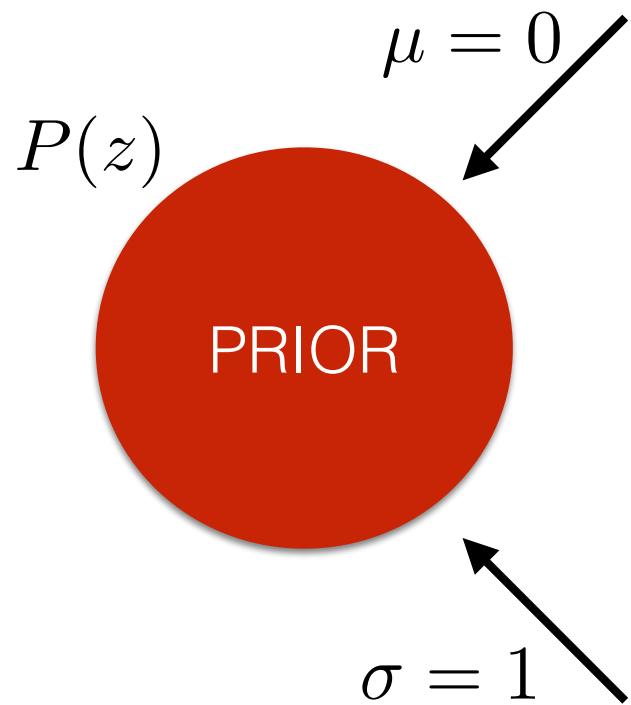


**HOWEVER, NOTHING GUARANTEES US THAT THE LATENT SPACE
CAN BE MODELLED BY A MIXTURE OF GAUSSIANS....**

... LET'S FORCE IT TO BE GAUSSIAN LIKE!

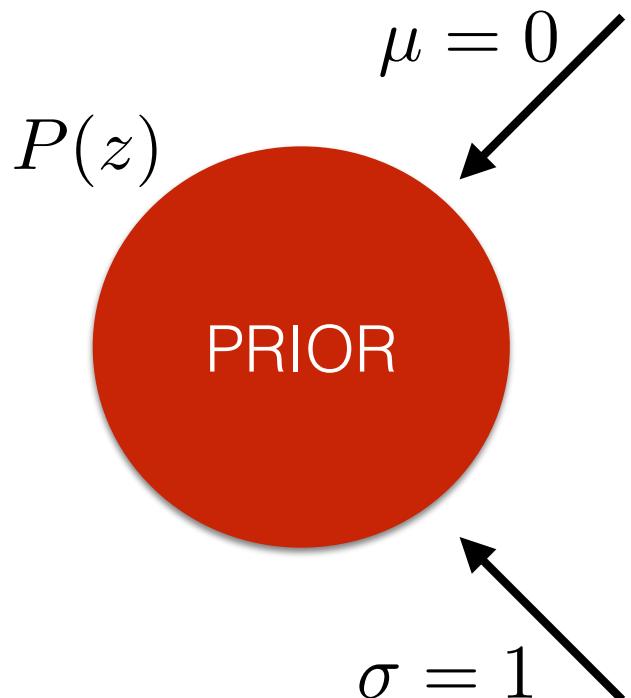


WE ASSUME A SIMPLE PRIOR

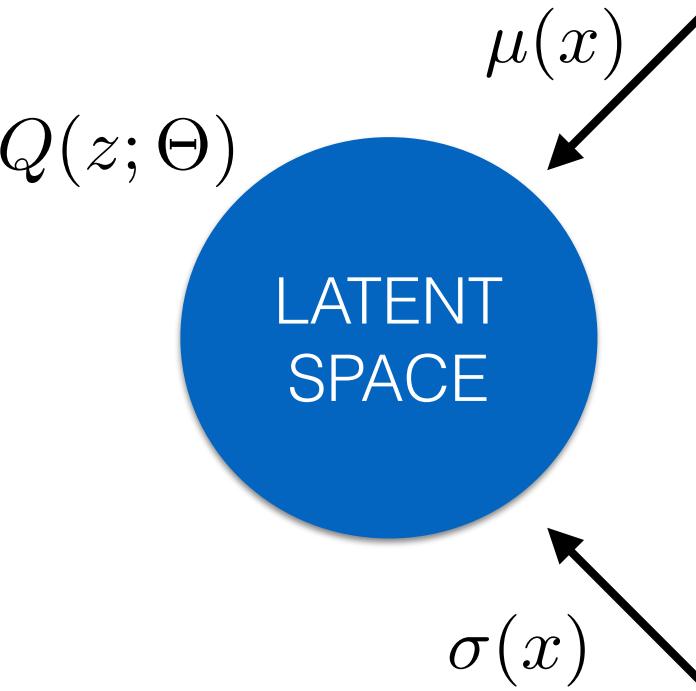


WE ASSUME A SIMPLE PRIOR

LATENT SPACE MODELING



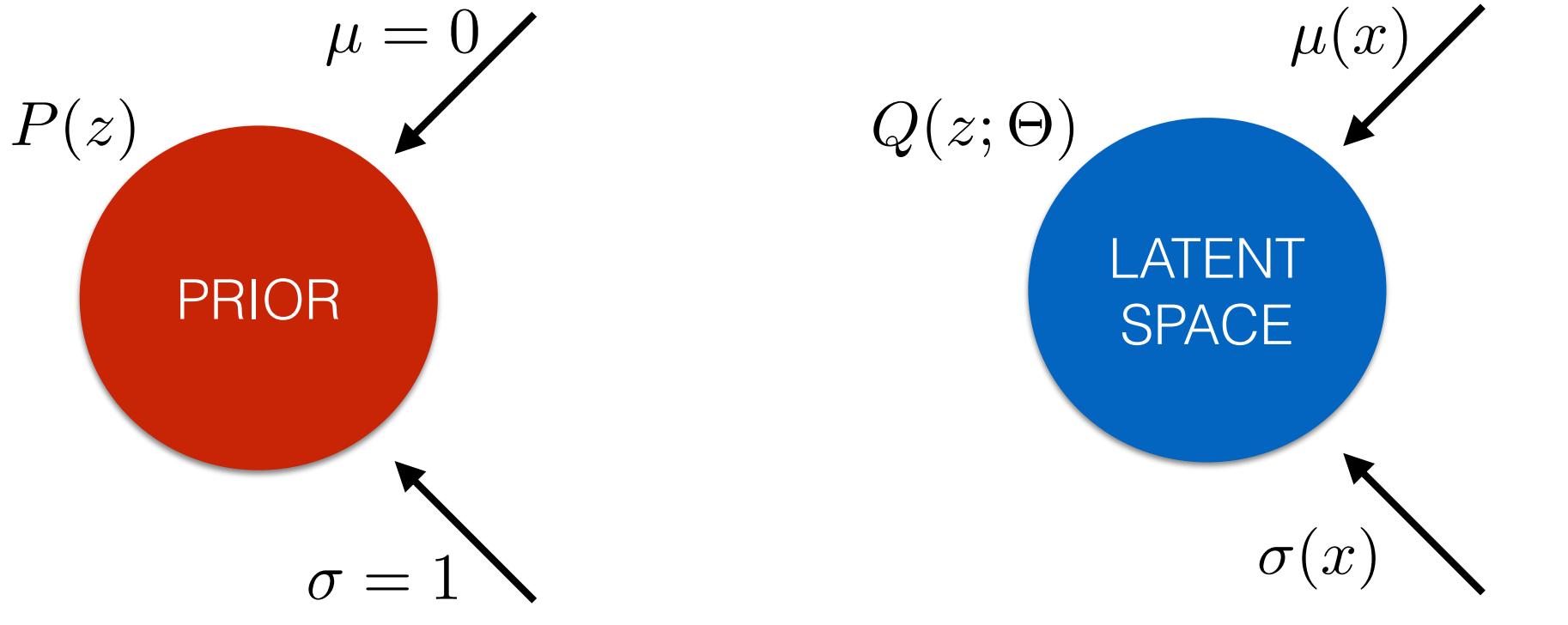
WE ASSUME A SIMPLE PRIOR



LATENT SPACE MODELING



WE WANT Q TO BE CLOSE TO THE PRIOR...



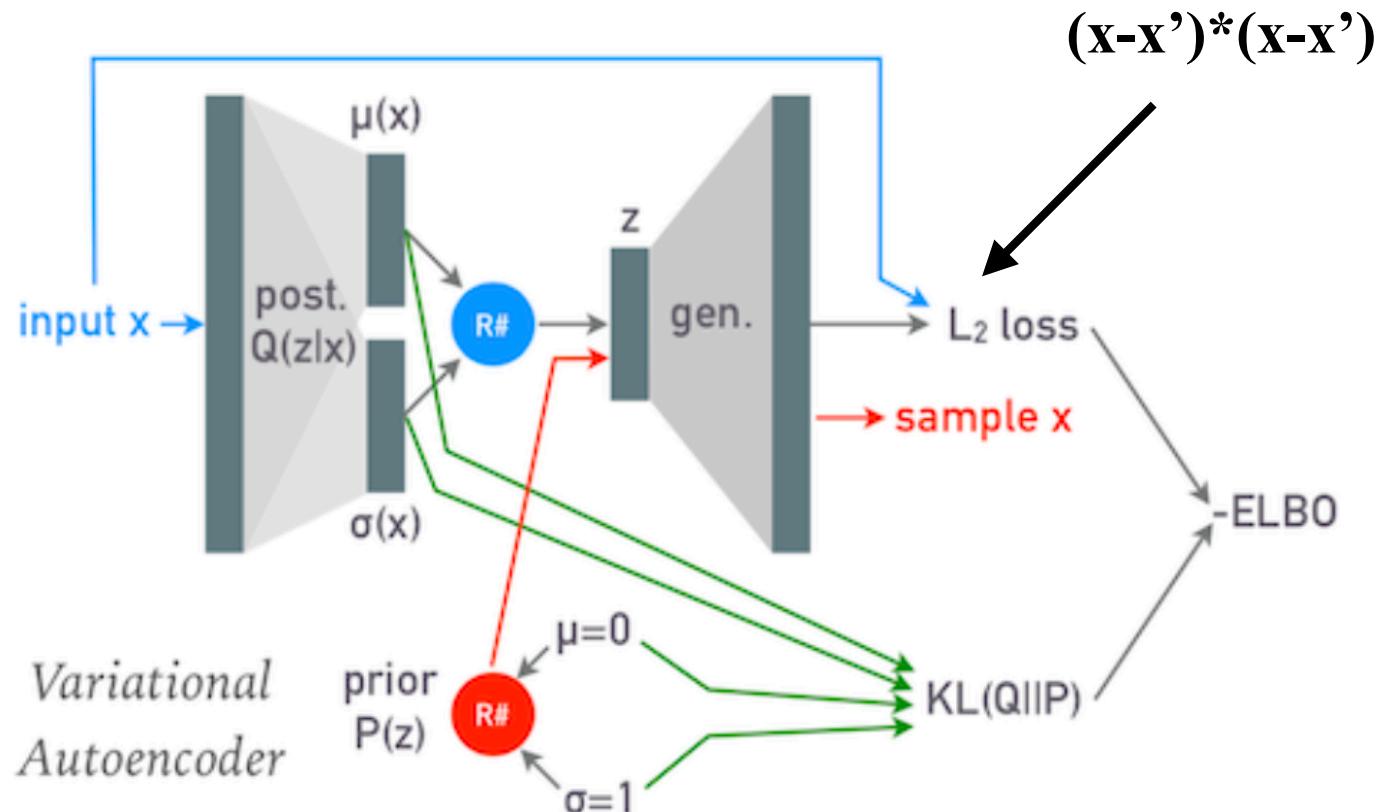
WE WANT Q TO BE CLOSE TO THE PRIOR...

$$D_{\text{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right).$$

$$D_{\text{KL}}(P \parallel Q) = \int_{\mathcal{X}} \log \left(\frac{dP}{dQ} \right) \frac{dP}{dQ} dQ,$$

WE MINIMIZE THE K-L DIVERGENCE BETWEEN P AND Q

WHAT WOULD BE THEN THE LOSS FUNCTION OF A VAE?



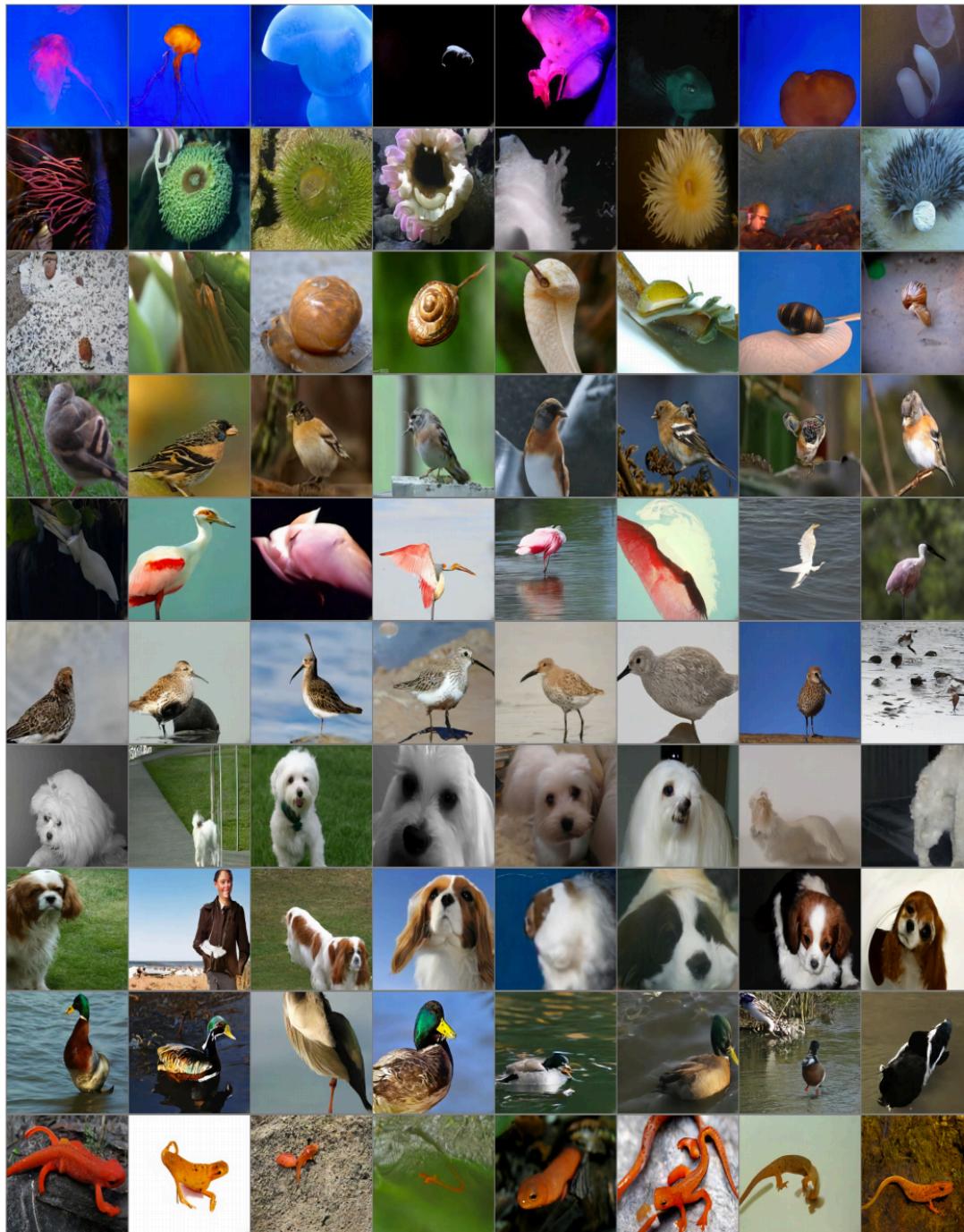
The key insight of VAE is that we are actually performing variational inference here, which then tells us what the loss function should be...

$$-\text{ELBO} = \langle \log P(\mathbf{x} | \mathbf{z}) \rangle_{\mathbf{z} \sim Q} + \text{KL}(Q(\mathbf{z}; \Theta) \parallel P(\mathbf{z})) ,$$

L2 LOSS

REGULARIZATION TERM

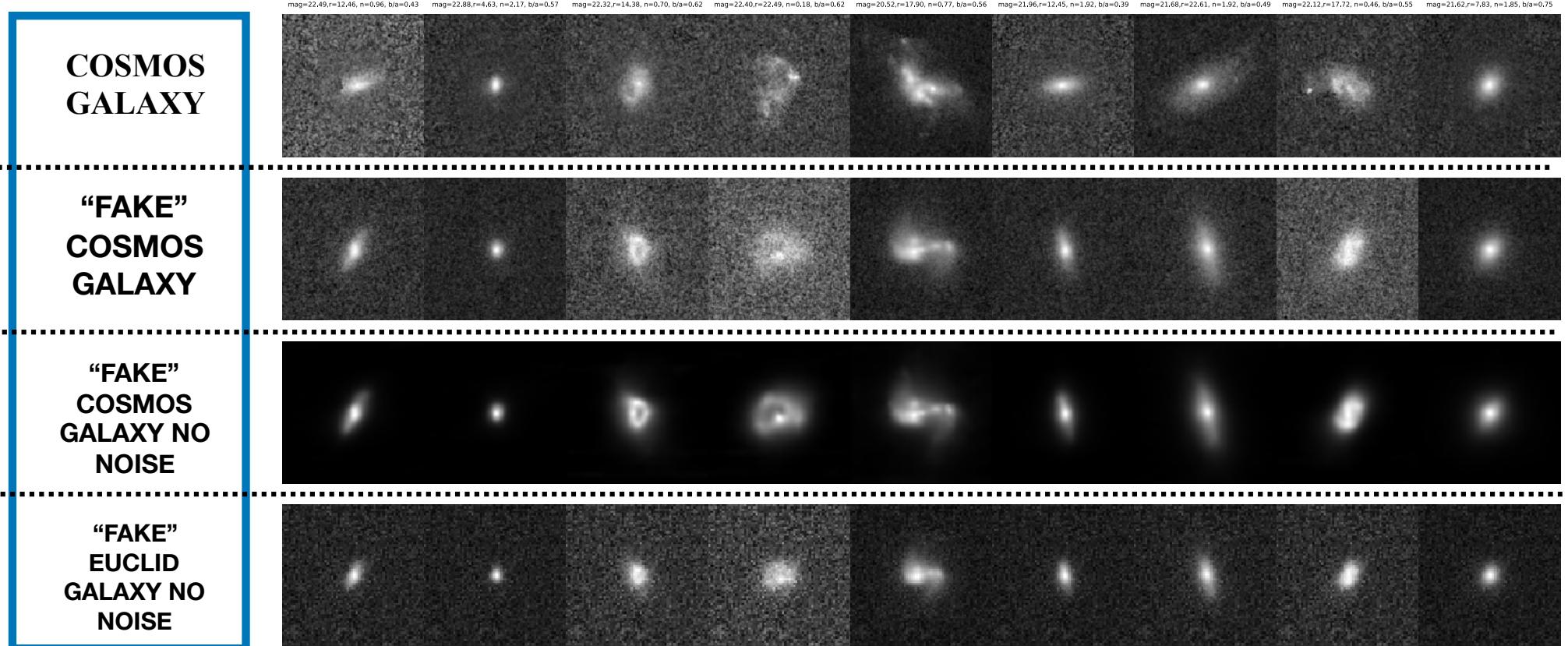
(VQ-VAE)



Razavi+19
(deepmind)



VAE GENERATED EUCLID REALISTIC GALAXIES



GO HERE:

[https://github.com/mhuertascompany/Saas-Fee/blob/main/
hands-on/session3/VAE_SDSS_spectra.ipynb](https://github.com/mhuertascompany/Saas-Fee/blob/main/hands-on/session3/VAE_SDSS_spectra.ipynb)