# OVA 2.0: Embedding Methods II

**CS3102 EDA**

UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA

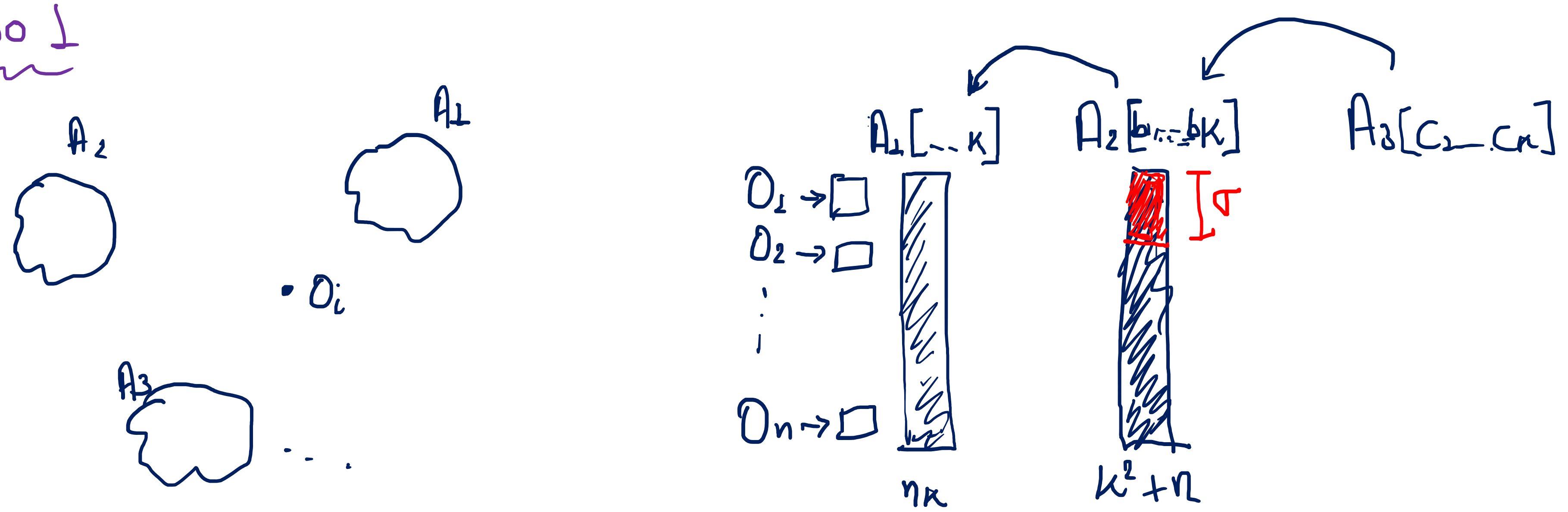
# Índice

1. SparseMap
2. t-SNE
3. UMAP
4. Autoencoder

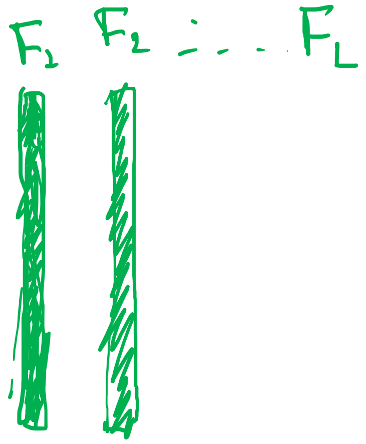
# 1. SparseMap

# **Sparse**_Map_ $d(F'(o_i), F'(b_i))$

Paso 1




Paso 2



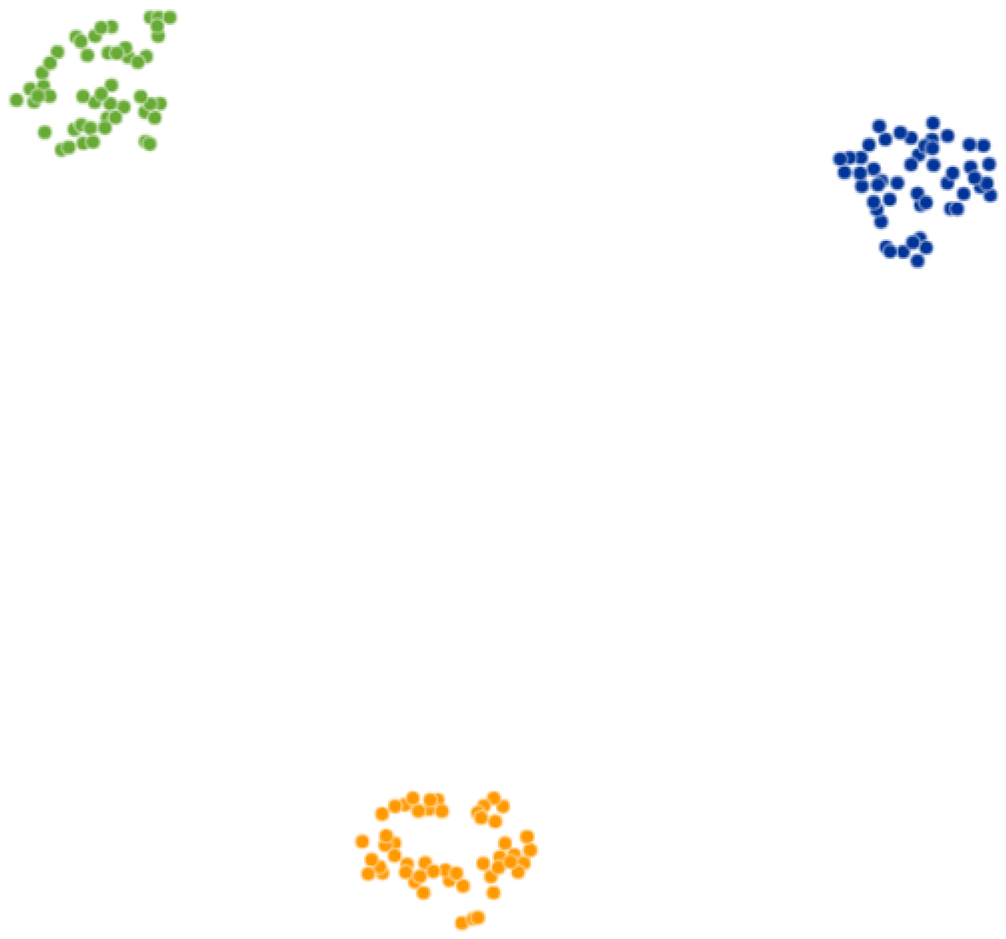

Seleccionamos los features con menor stress



Gabriela Hristescu and Martin Farach-Colton (1999) “Cluster-preserving embedding of proteins”.
Technical Report 99-50, Computer Science Department, Rutgers University

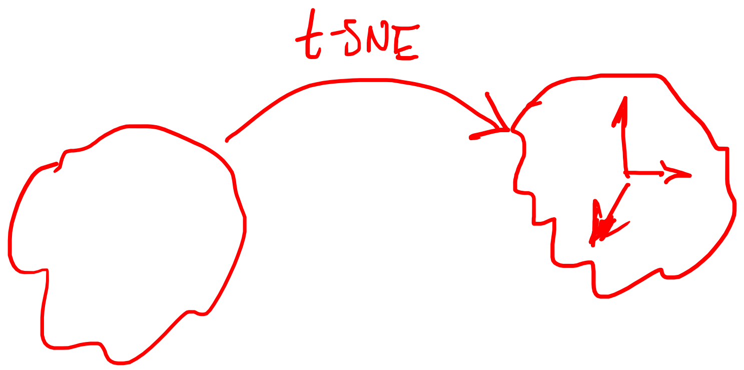
# 2. t-SNE

# t-SNE

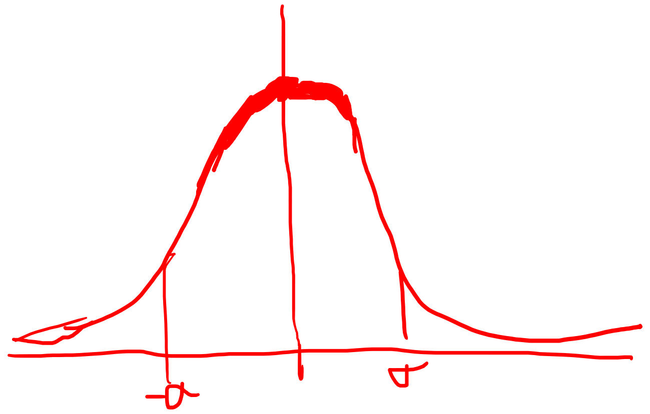




Laurens van der Maaten and Geoffrey Hinton (2008) "Visualizing Data using t-SNE".
Journal of machine learning research, vol. 9, no 11.




# t-SNE



Dado un conjunto de $N$ datos $x_1, \dots, x_N$, se calcula las probabilidades $p_{ij}$ que son proporcionales a la similaridad entre los datos

distribu datos reales

$$p(j|i) = \frac{e^{-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}}}{\boxed{\sum_{k\neq i} e^{-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}}}}$$

Definimos:

$$p_{ij} = \frac{p(j|i) + p(i|j)}{2N}$$



Laurens van der Maaten and Geoffrey Hinton (2008) "Visualizing Data using t-SNE".
Journal of machine learning research, vol. 9, no 11.

# t-SNE

Dado un conjunto de $N$ datos $x_1, \dots, x_N$, se calcula las probabilidades $p_{ij}$ que son proporcionales a la similaridad entre los datos

datos original

salida

$$p(j|i) = \frac{e^{-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}}} \qquad q(j|i) = \frac{\boxed{e^{-\frac{\|y_i - y_j\|^2}{2\sigma_i^2}}}}{\sum_{k \neq i} e^{-\frac{\|y_i - y_k\|^2}{2\sigma_i^2}}}$$

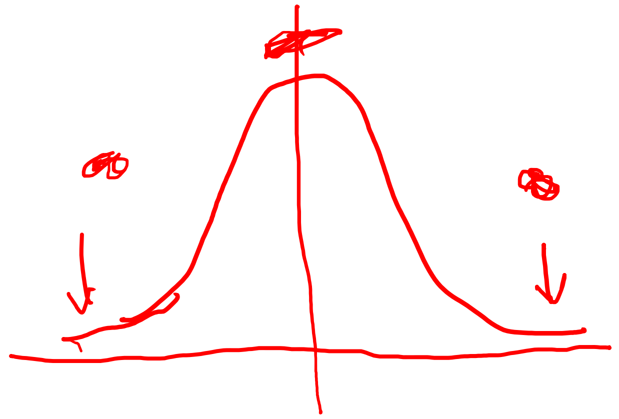
Definimos:

$$p_{ij} = \frac{p(j|i) + p(i|j)}{2N}$$



Laurens van der Maaten and Geoffrey Hinton (2008) "Visualizing Data using t-SNE".
Journal of machine learning research, vol. 9, no 11.

# t-SNE



Dado un conjunto de $N$ datos $x_1, \dots, x_N$, se calcula las probabilidades $p_{ij}$ que son proporcionales a la similaridad entre los datos

$$p(j|i) = \frac{e^{-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}}} \qquad q(j|i) = \frac{e^{-\frac{\|y_i - y_j\|^2}{2\sigma_i^2}}}{\sum_{k \neq i} e^{-\frac{\|y_i - y_k\|^2}{2\sigma_i^2}}}$$

Definimos:

$$p_{ij} = \frac{p(j|i) + p(i|j)}{2N} \qquad q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_k \sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$



Laurens van der Maaten and Geoffrey Hinton (2008) "Visualizing Data using t-SNE".
Journal of machine learning research, vol. 9, no 11.

# t-SNE

**Distribución de los datos originales**

$$p_{ij} = \frac{p(j|i) + p(i|j)}{2N}$$

**Distribución final**

$$q_{ij} = \frac{\left(1 + \left\|y_i - y_j\right\|^2\right)^{-1}}{\sum_k \sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

$$\mathrm{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

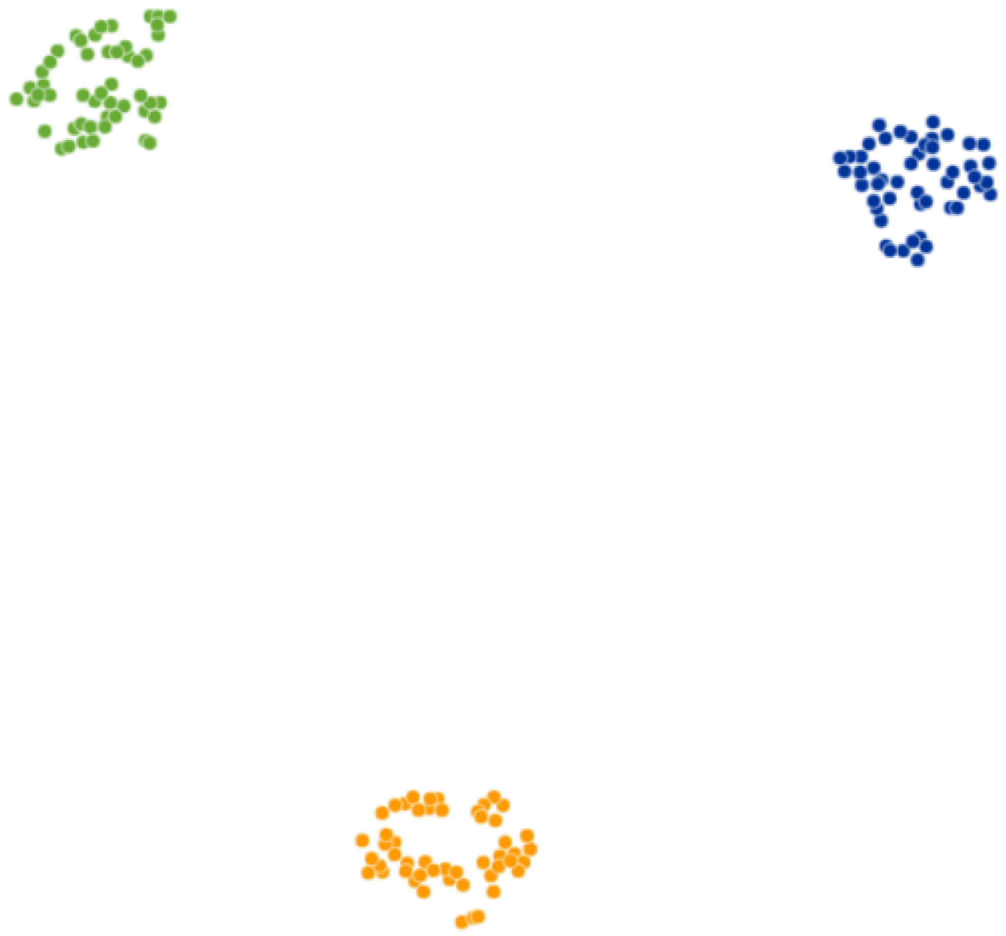


Laurens van der Maaten and Geoffrey Hinton (2008) “Visualizing Data using t-SNE”.
Journal of machine learning research, vol. 9, no 11.

# *t-SNE*

**Algorithm 1**: Simple version of t-Distributed Stochastic Neighbor Embedding.

**Data**: data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations $T$, learning rate $\eta$, momentum $\alpha(t)$.
**Result**: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
**begin**
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  **for** $t$=*1* **to** $T$ **do**
    compute low-dimensional affinities $q_{ij}$ (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
  **end**
**end**





Laurens van der Maaten and Geoffrey Hinton (2008) "Visualizing Data using t-SNE".
Journal of machine learning research, vol. 9, no 11.

# *t-SNE*

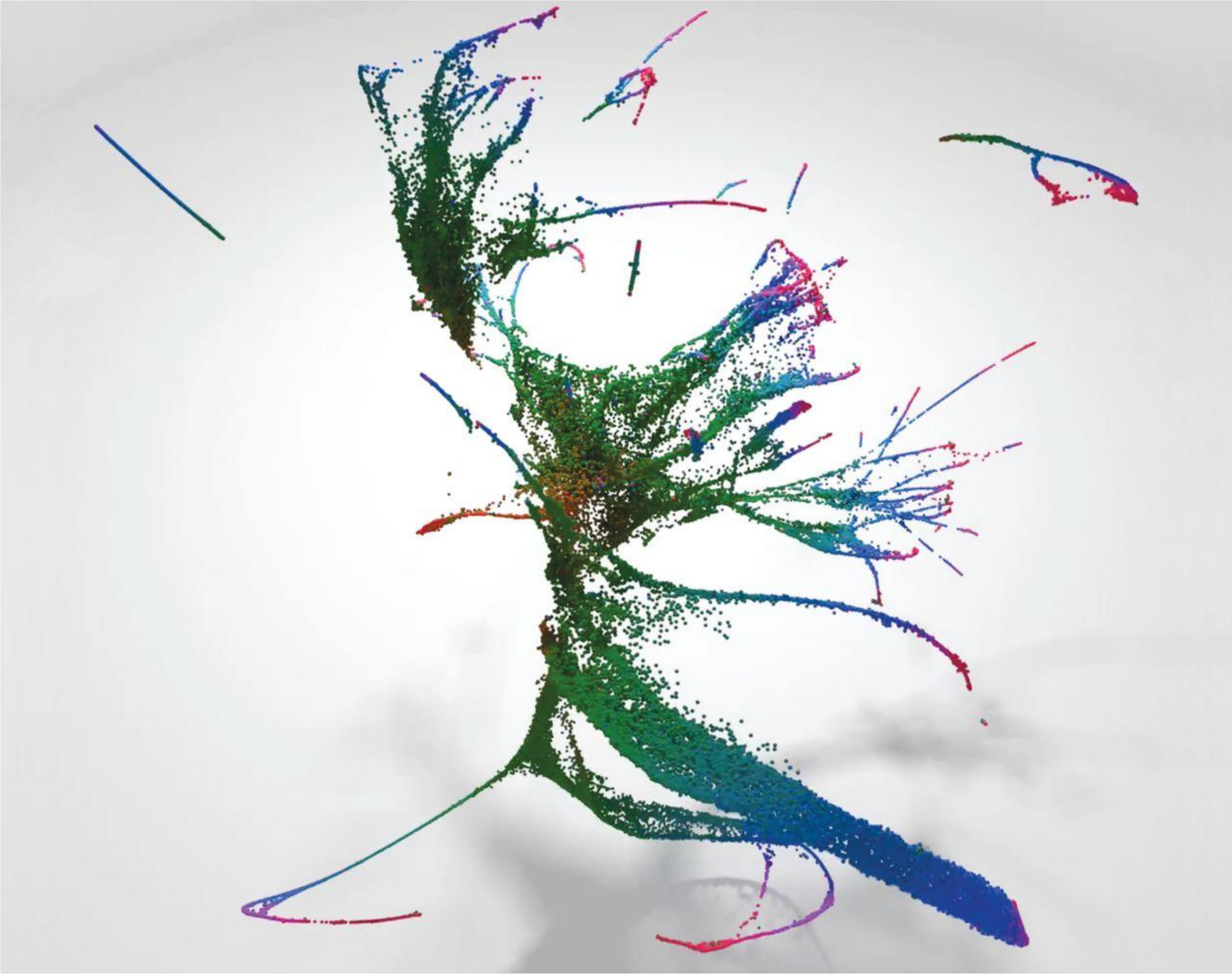
https://distill.pub/2016/misread-tsne/

# 3. UMAP

2 4

# UMAP

(Uniform Manifold Approximation and Projection for Dimension Reduction)

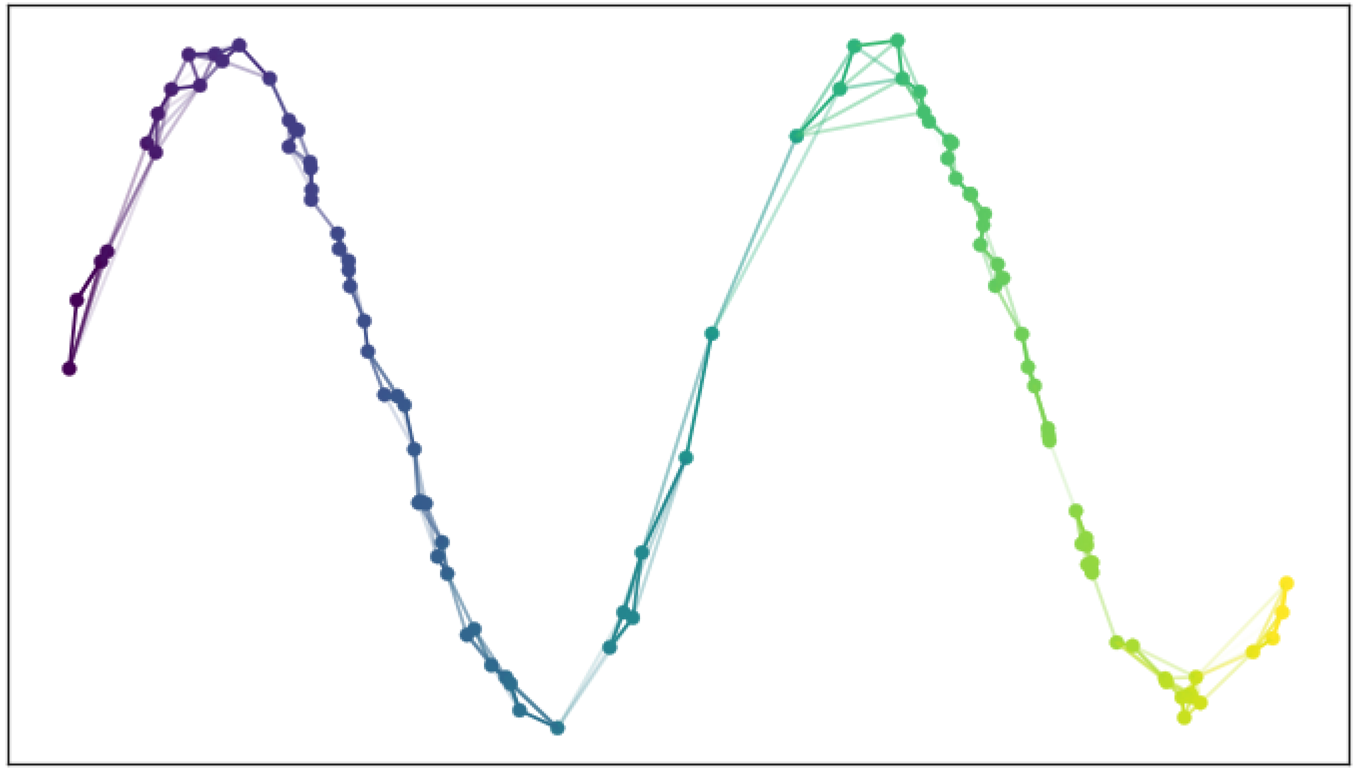




Leland McInnes, John Healy, James Melville (2018) “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. Journal of Open-Source Software, 3(29), 861.

# UMAP

(Uniform Manifold Approximation and Projection for Dimension Reduction)



**Binary cross-entropy**

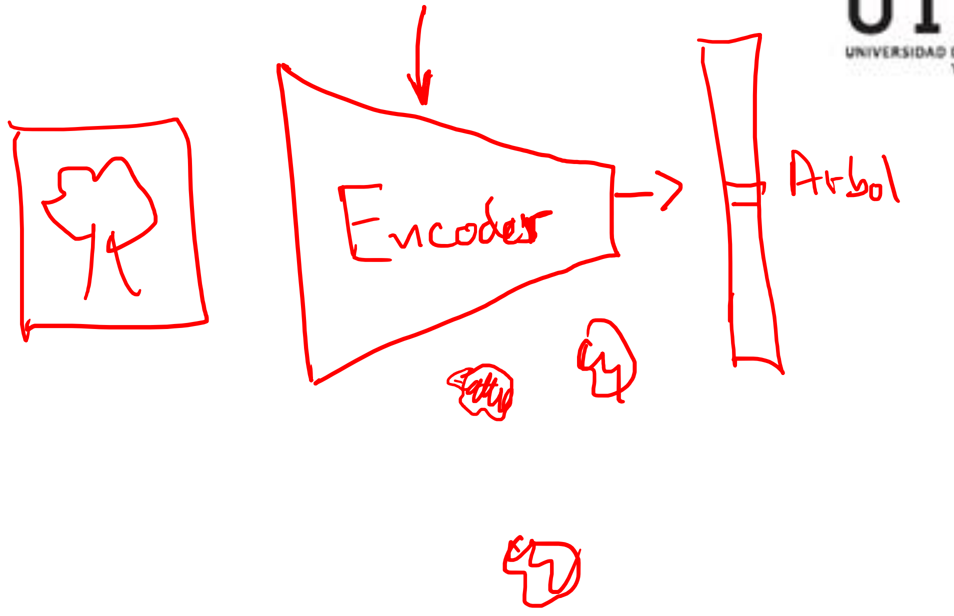
$$CE(X,Y) = \sum_i \sum_j \left[ p_{ij}(X) \log\left(\frac{p_{ij}(X)}{q_{ij}(Y)}\right) + (1 - p_{ij}(X)) \log\left(\frac{1 - p_{ij}(X)}{1 - q_{ij}(Y)}\right) \right]$$

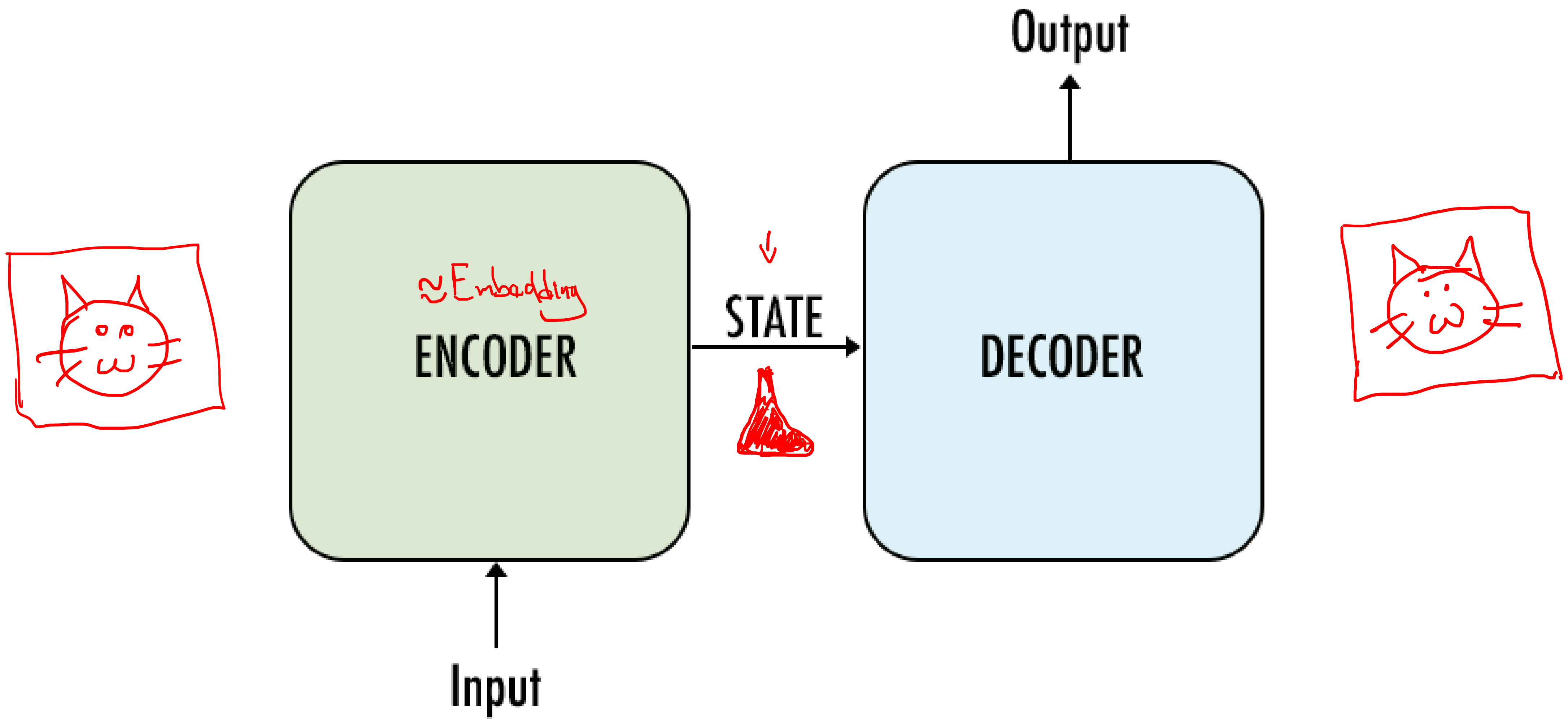


Leland McInnes, John Healy, James Melville (2018) "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". Journal of Open-Source Software, 3(29), 861.

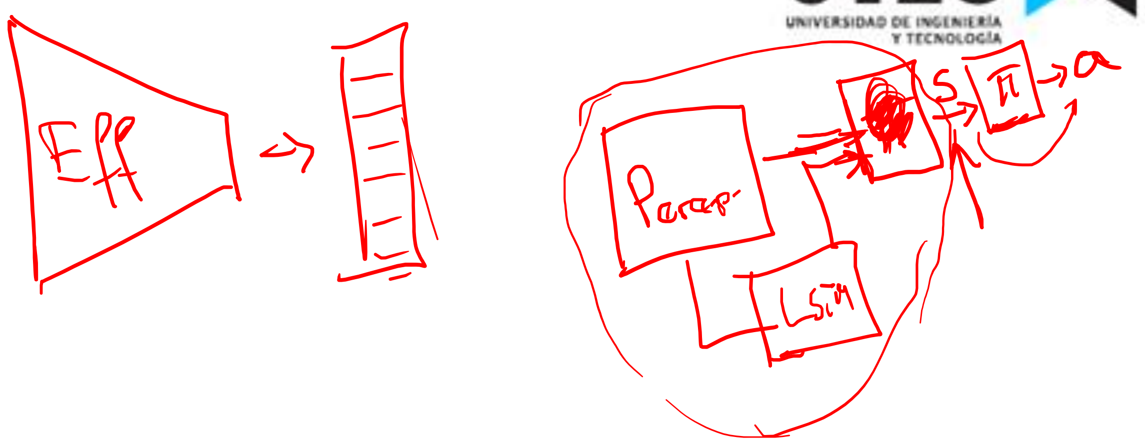
# 4. AutoEncoder

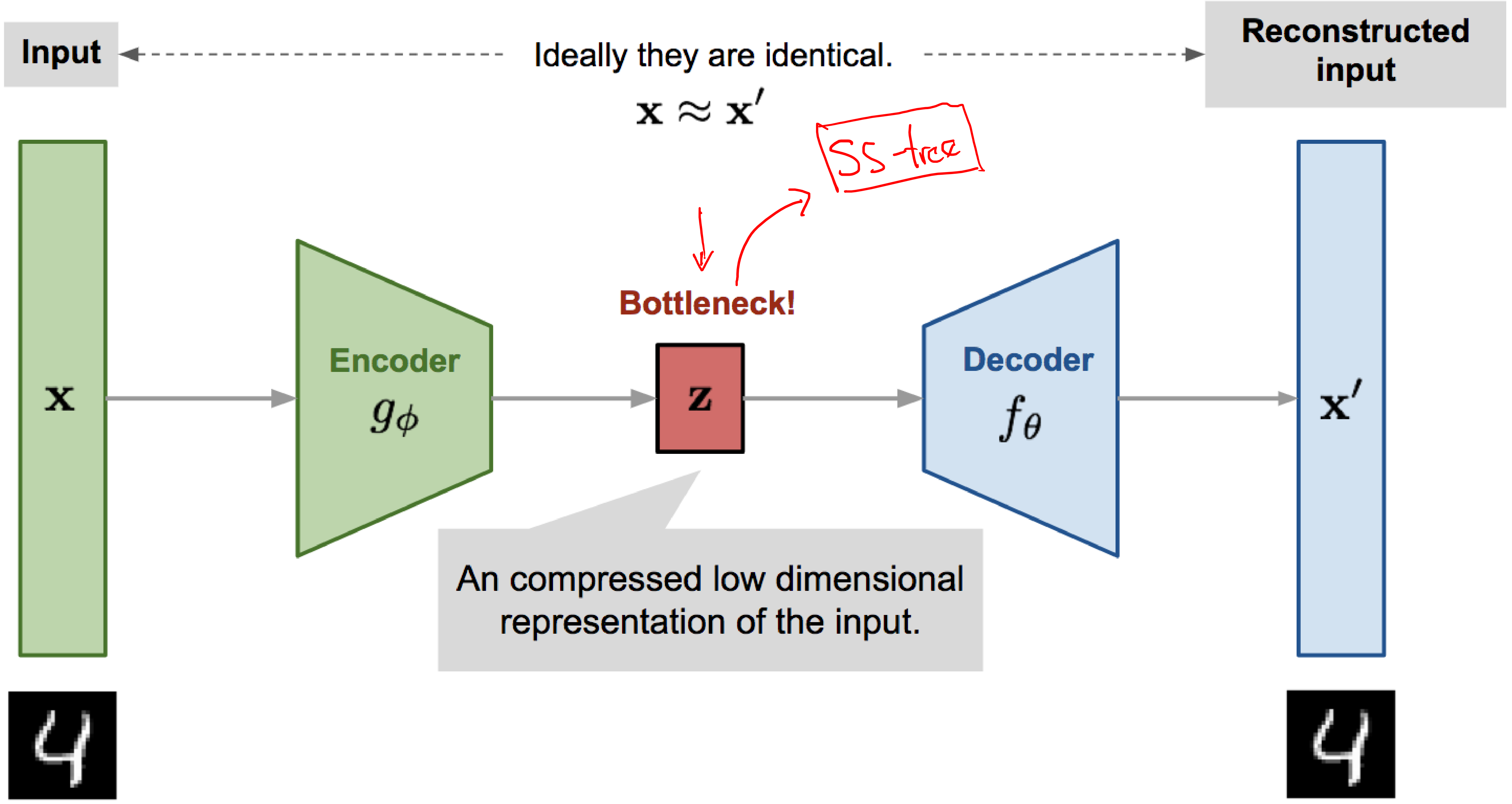
# **Auto***Encoder*



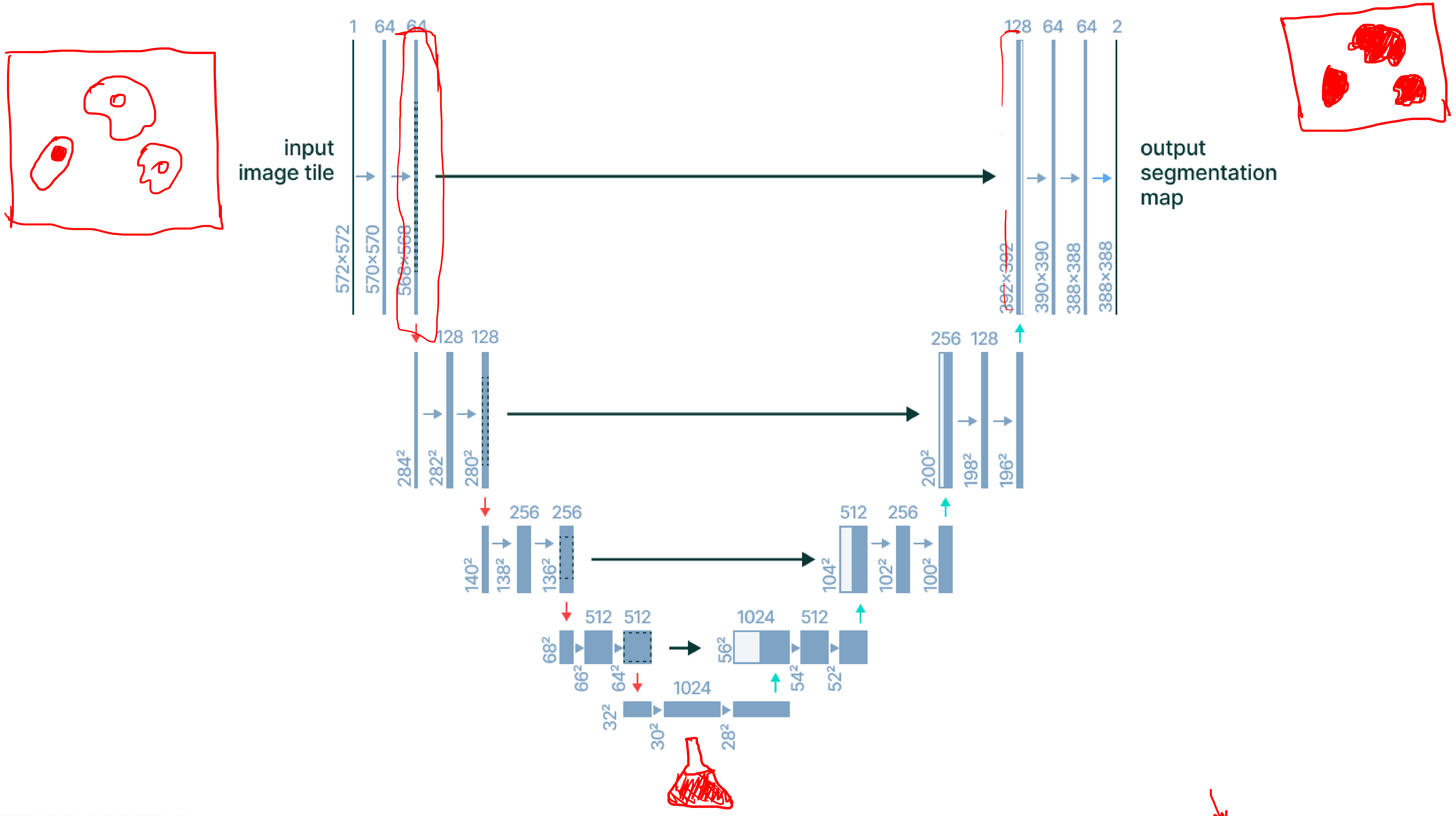



# AutoEncoder



# **Auto***Encoder*



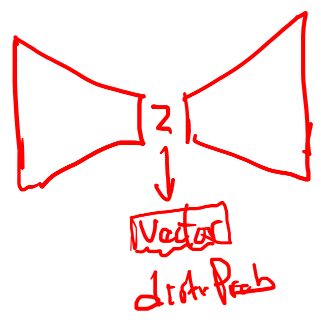


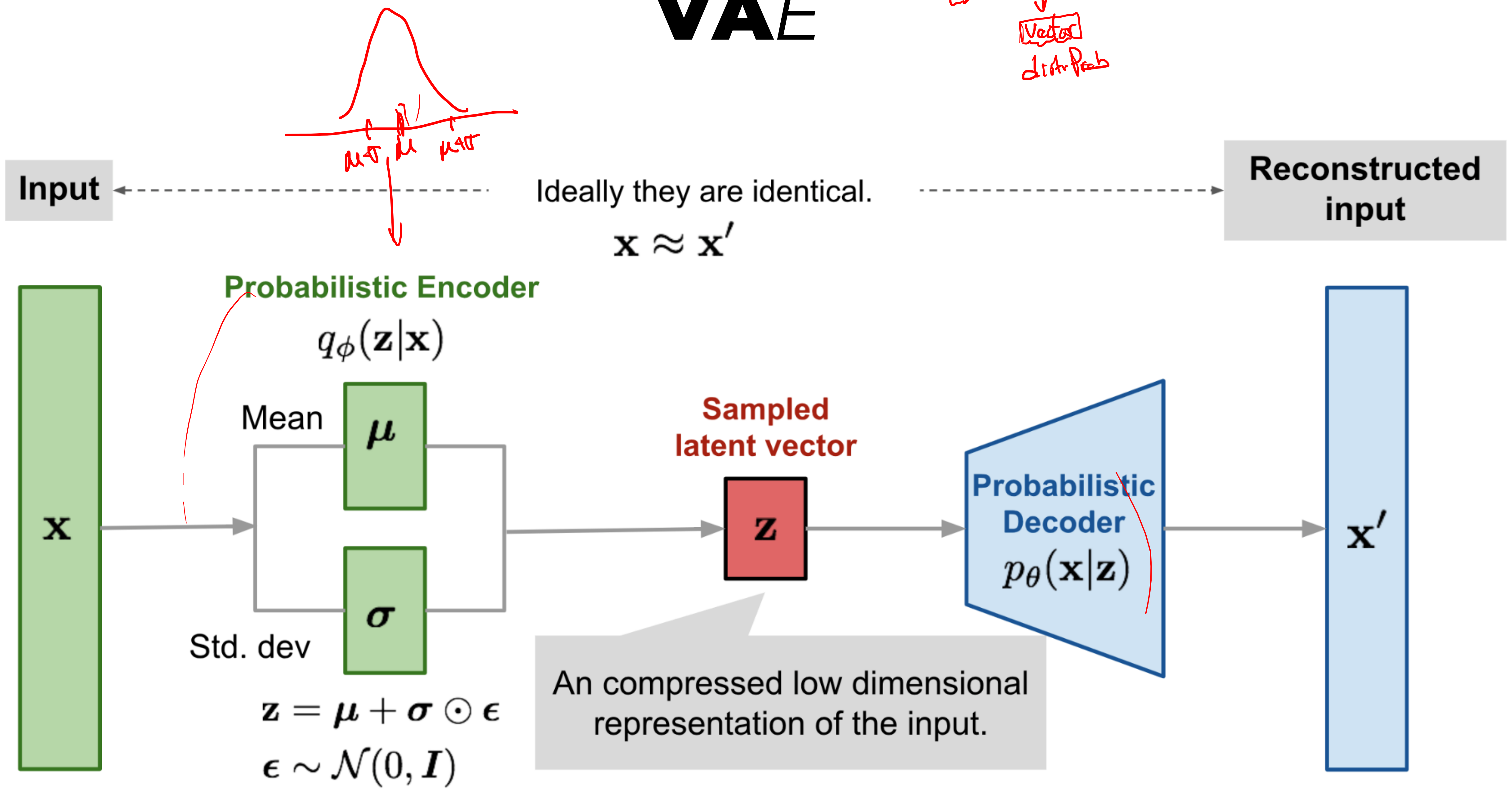

Olaf Ronneberger et al. (2015) "U-Net: Convolutional Networks for Biomedical Image Segmentation".
International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

# **Auto***Encoder*

https://projector.tensorflow.org/

# VAE



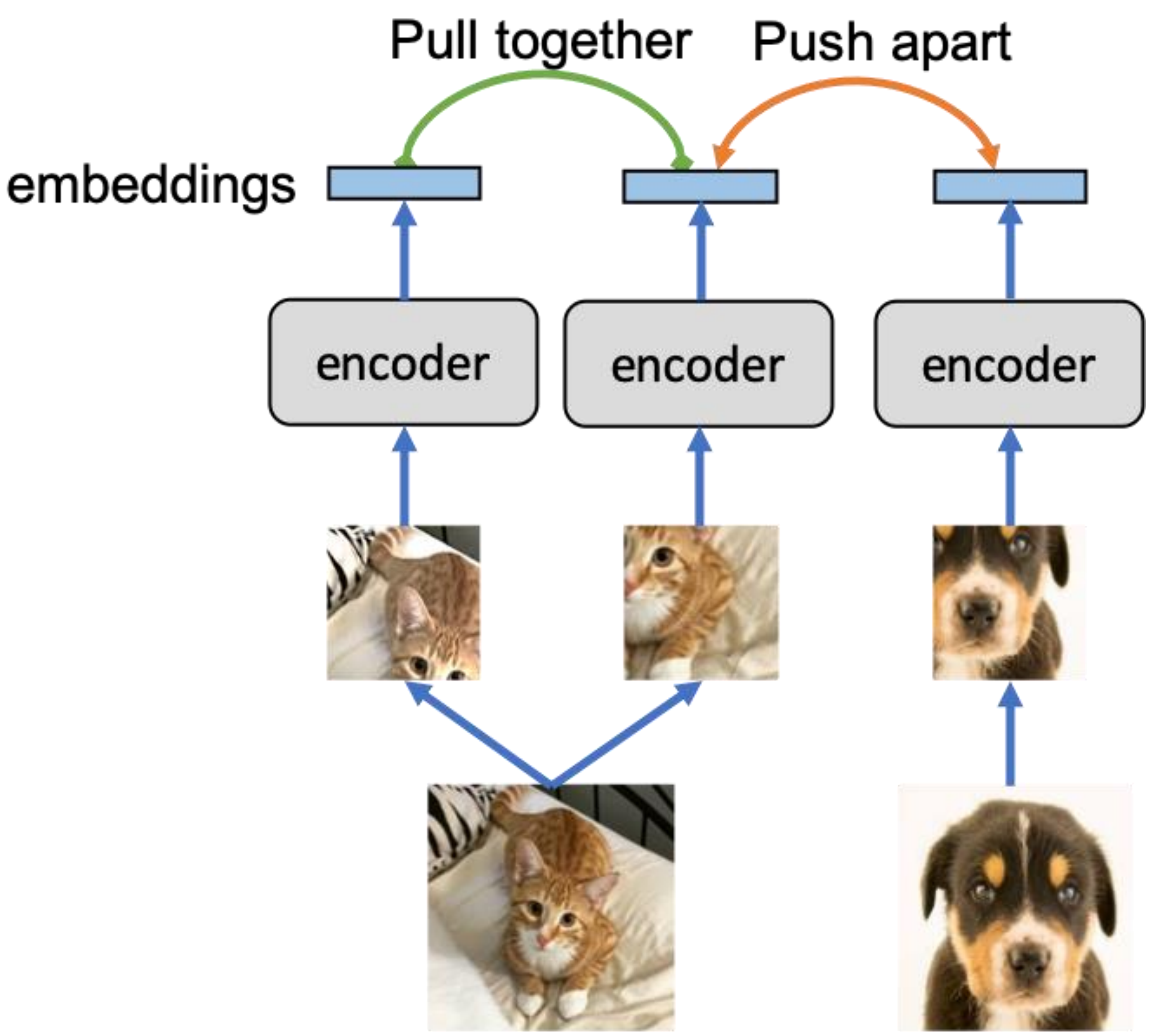



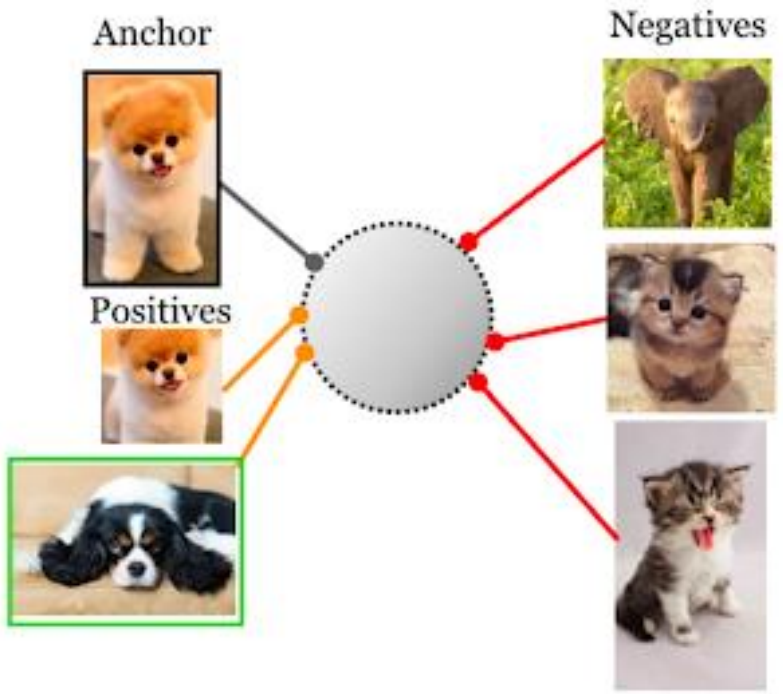

$$\mathcal{L}(\phi, \beta) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$$



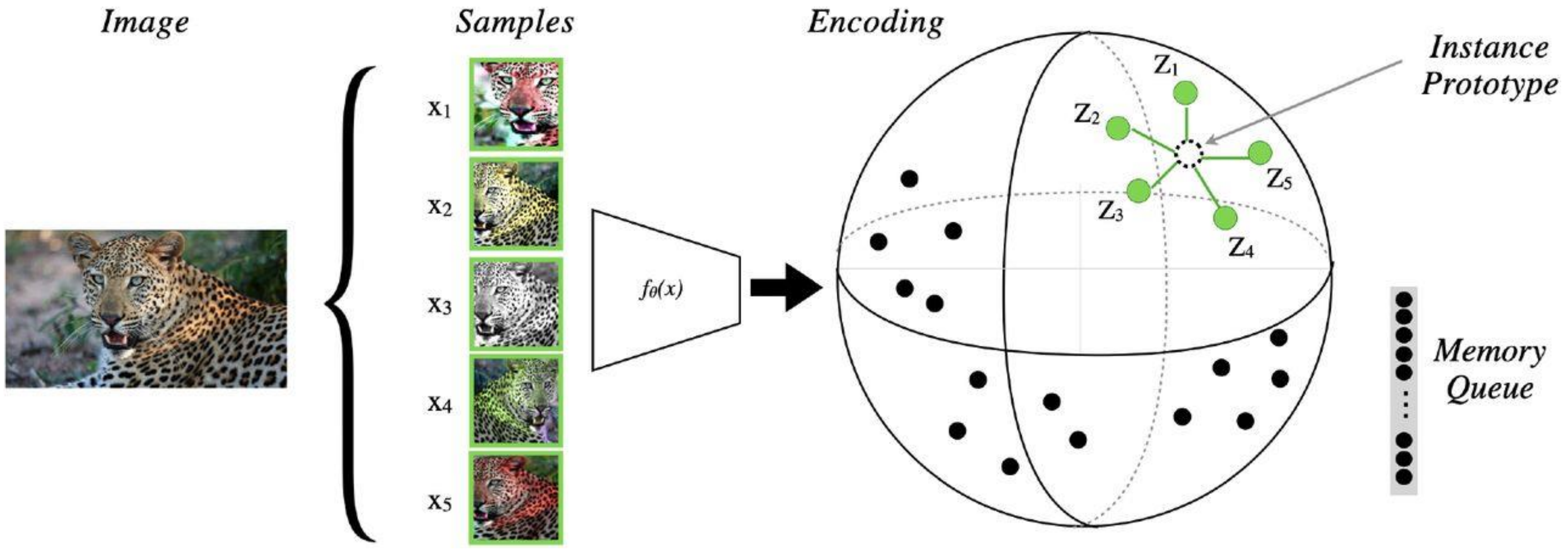
Diederik P. Kingma and Max Welling (2013) "Auto-Encoding Variational Bayes". arXiv preprint arXiv:1312.6114

# **Contrastive** *Learning*



# **Contrastive** *Learning*




UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA
INGENIERÍA
MECATRÓNICA
BIOINGENIERÍA
CIENCIA DE
LA COMPUTACIÓN