

# A Diffusion Approach to Image Generation

Nicolas Dunou  
Hippolyte Gisserot  
Lucas Marandat

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

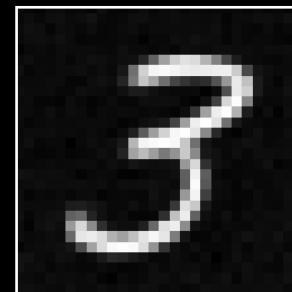
# Mathematical intuition

**We want to sample handwritten digit images, but...**

$$x \sim P(X)$$



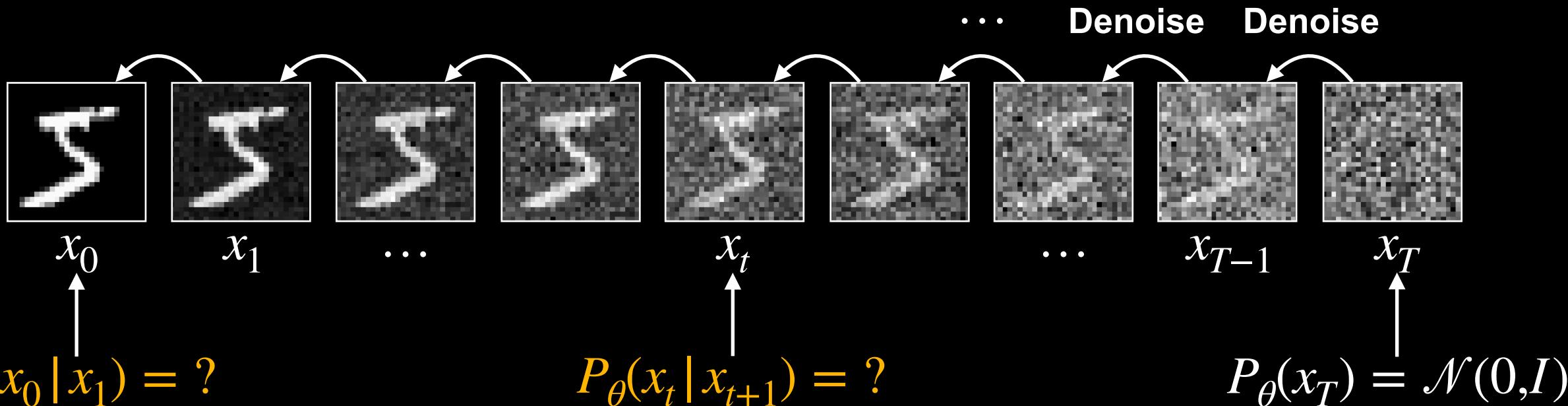
...



# Mathematical intuition

**What if we try to learn a stepwise denoiser from a known probability?**

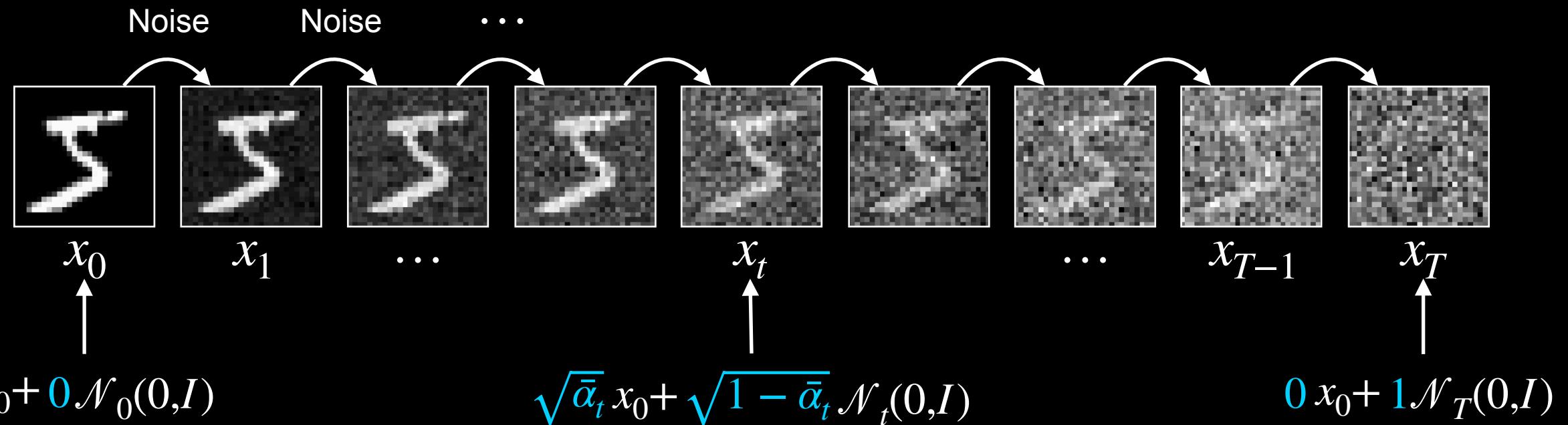
$$x_0 \sim P_\theta(x_0 | x_1) P_\theta(x_1 | x_2) \cdots P_\theta(x_T)$$



# Mathematical intuition

**Noising an image is quite easy!**

$$P(x_t | x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)I)$$

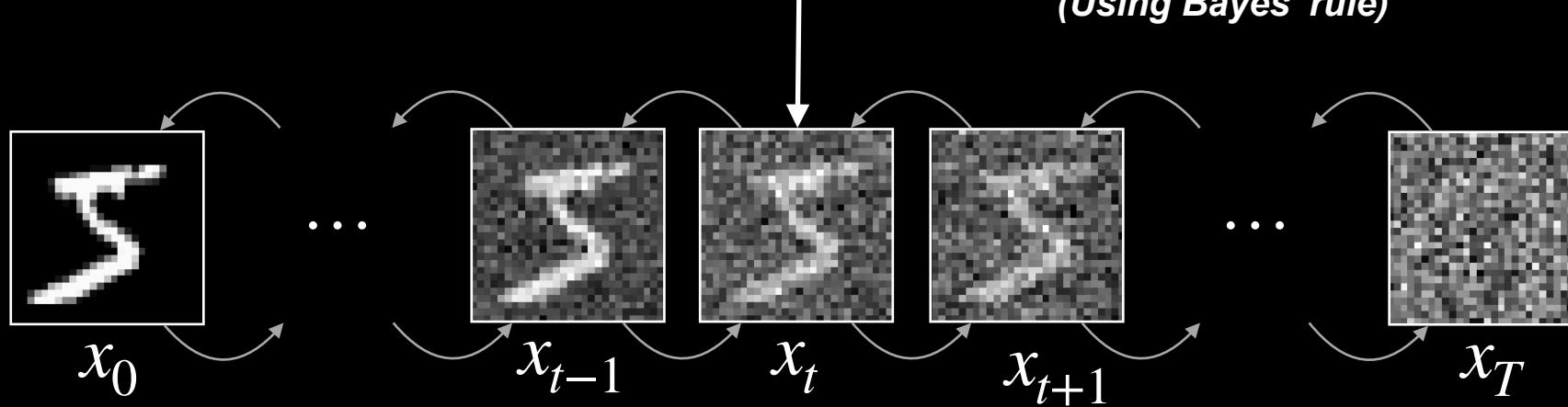


# Mathematical intuition

**Back to our problem**

$$P(x_t | x_{t+1}, x_0) = \mathcal{N}(\tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t)$$

*(Using Bayes' rule)*

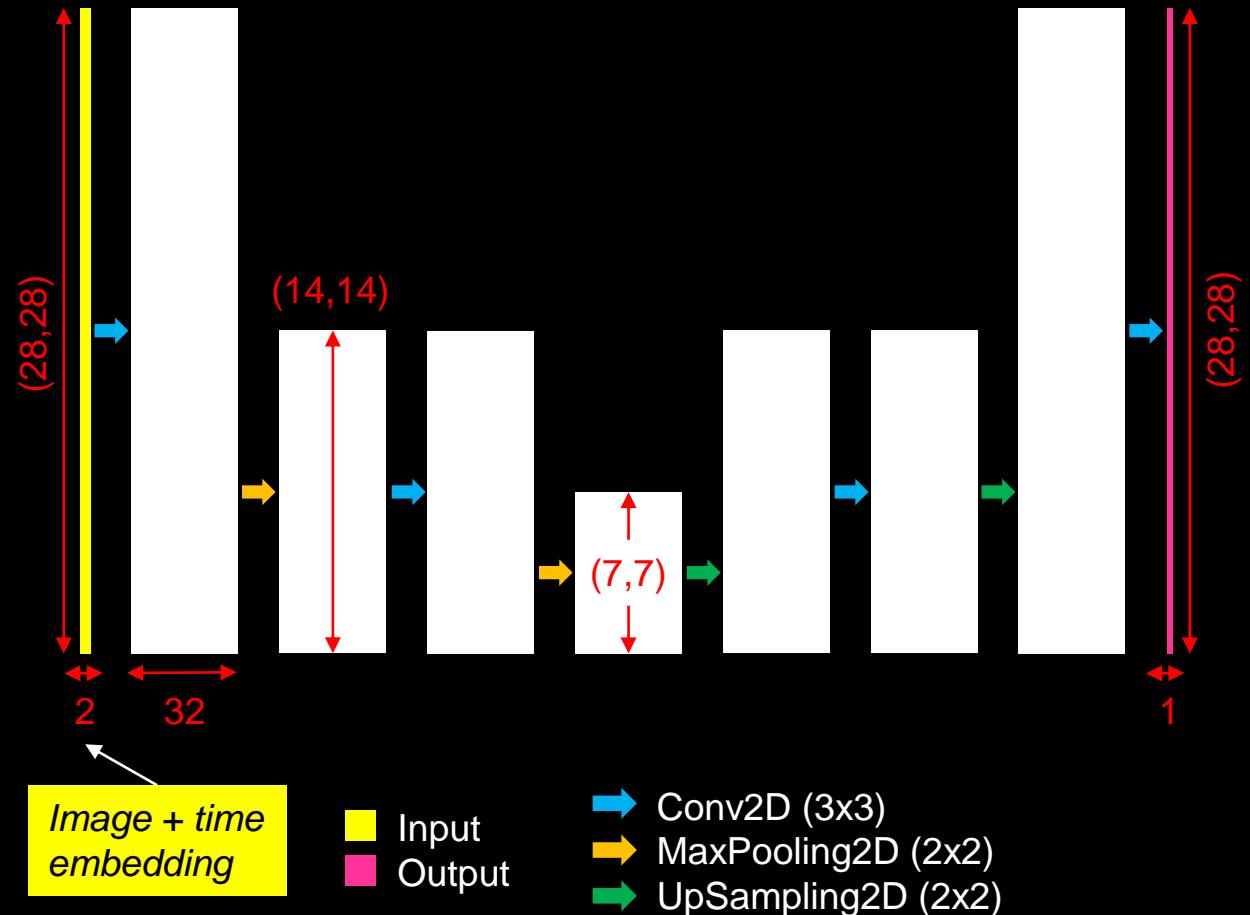


$$P_\theta(x_t | x_{t+1}) = \mathcal{N}(\mu_\theta(x_t, t), \beta_t) \implies \boxed{\mu_\theta(x_t, t) \approx \tilde{\mu}_t(x_t, x_0)}$$

*(Loss using Mean Squared Error)*

# Model 1

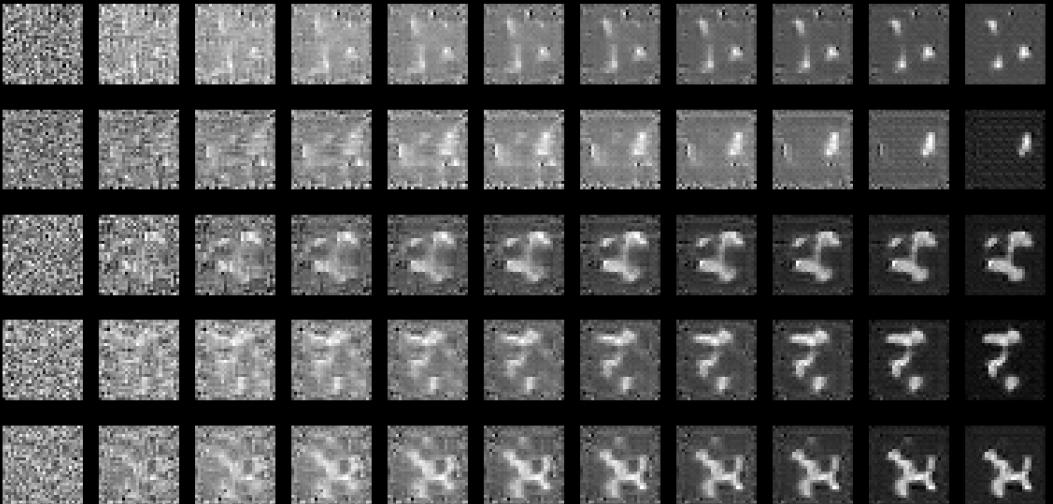
## Network architecture:



**Generation of 50 random images:**

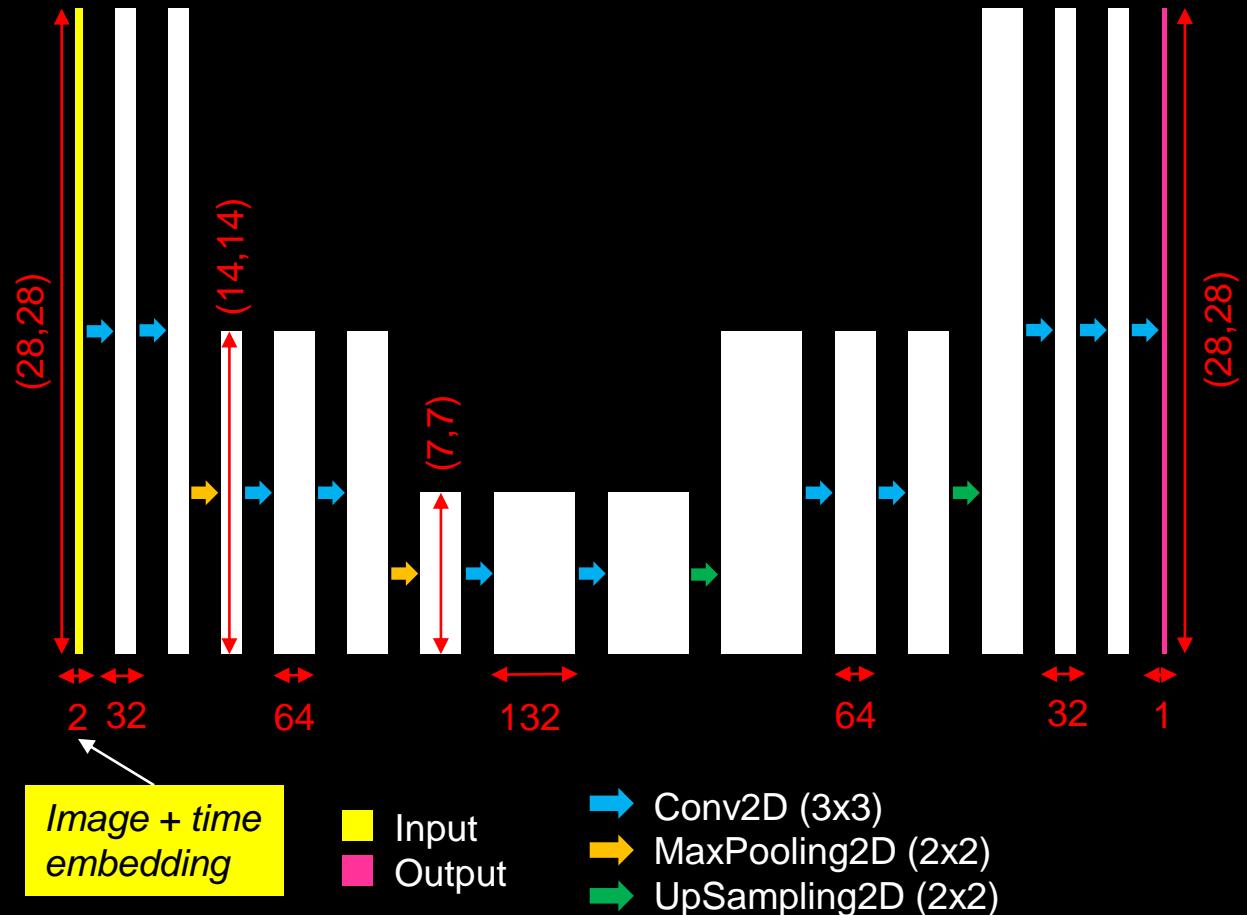


**Generation of 5 random denoising paths:**



# Model 2

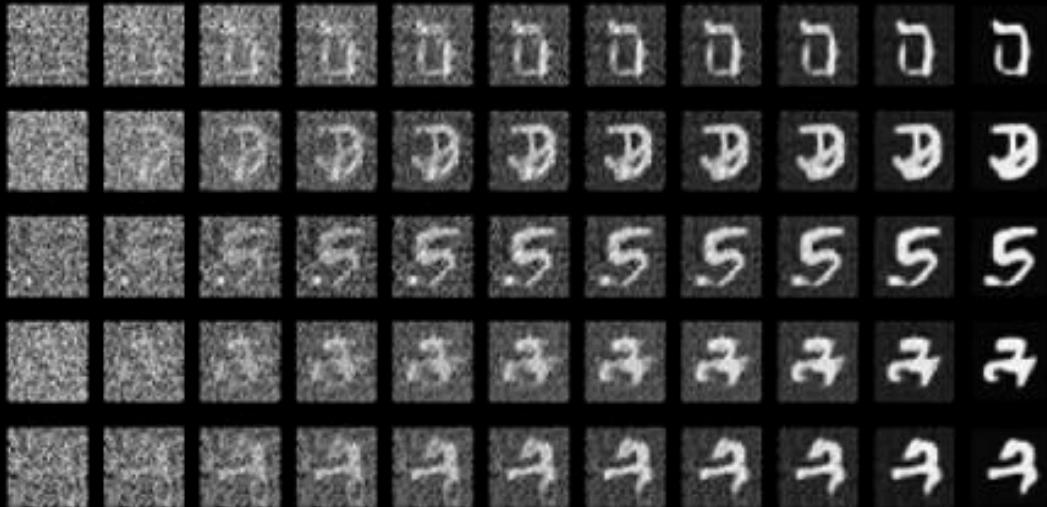
## Network architecture:



## Generation of 50 random images:

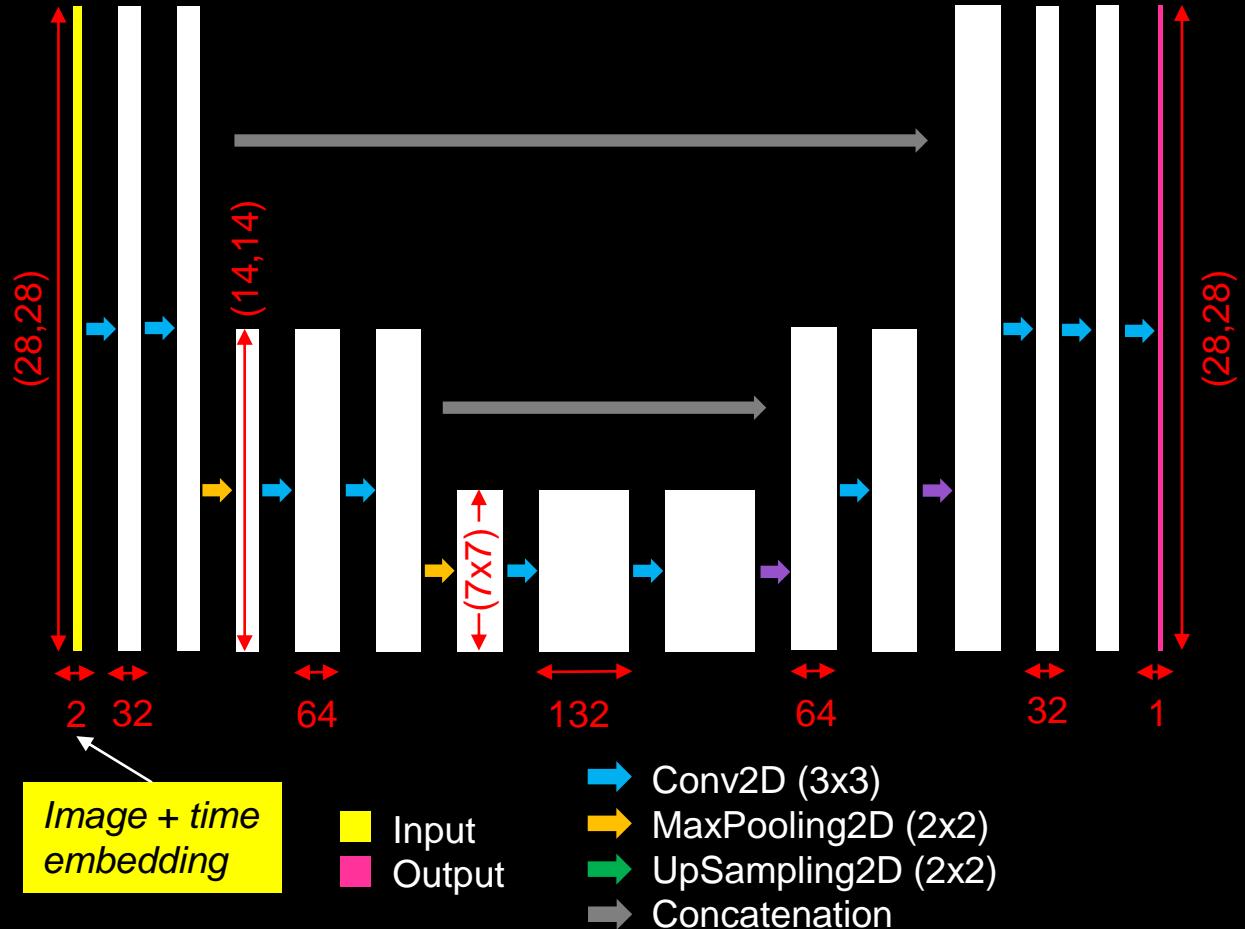


## Generation of 5 random denoising paths:



# Model 3

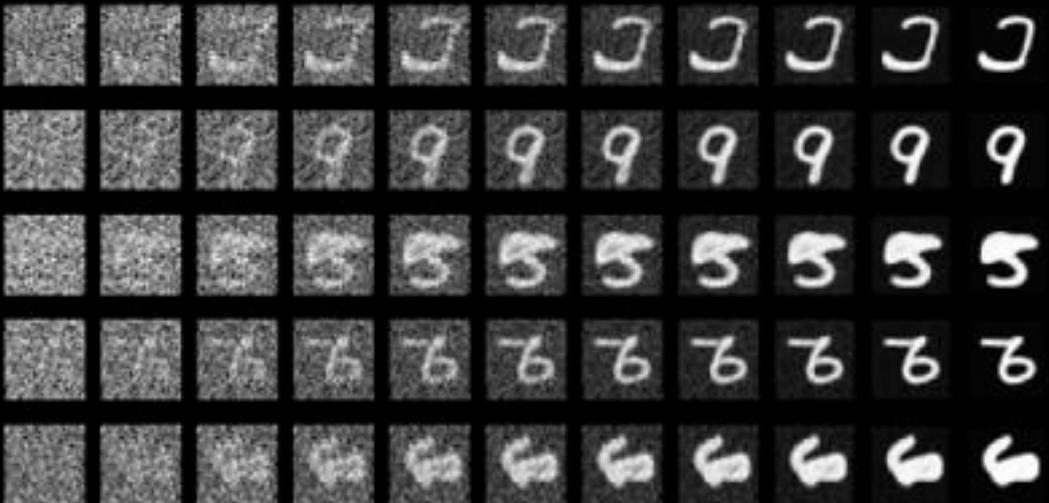
## Network architecture:



## Generation of 50 random images:

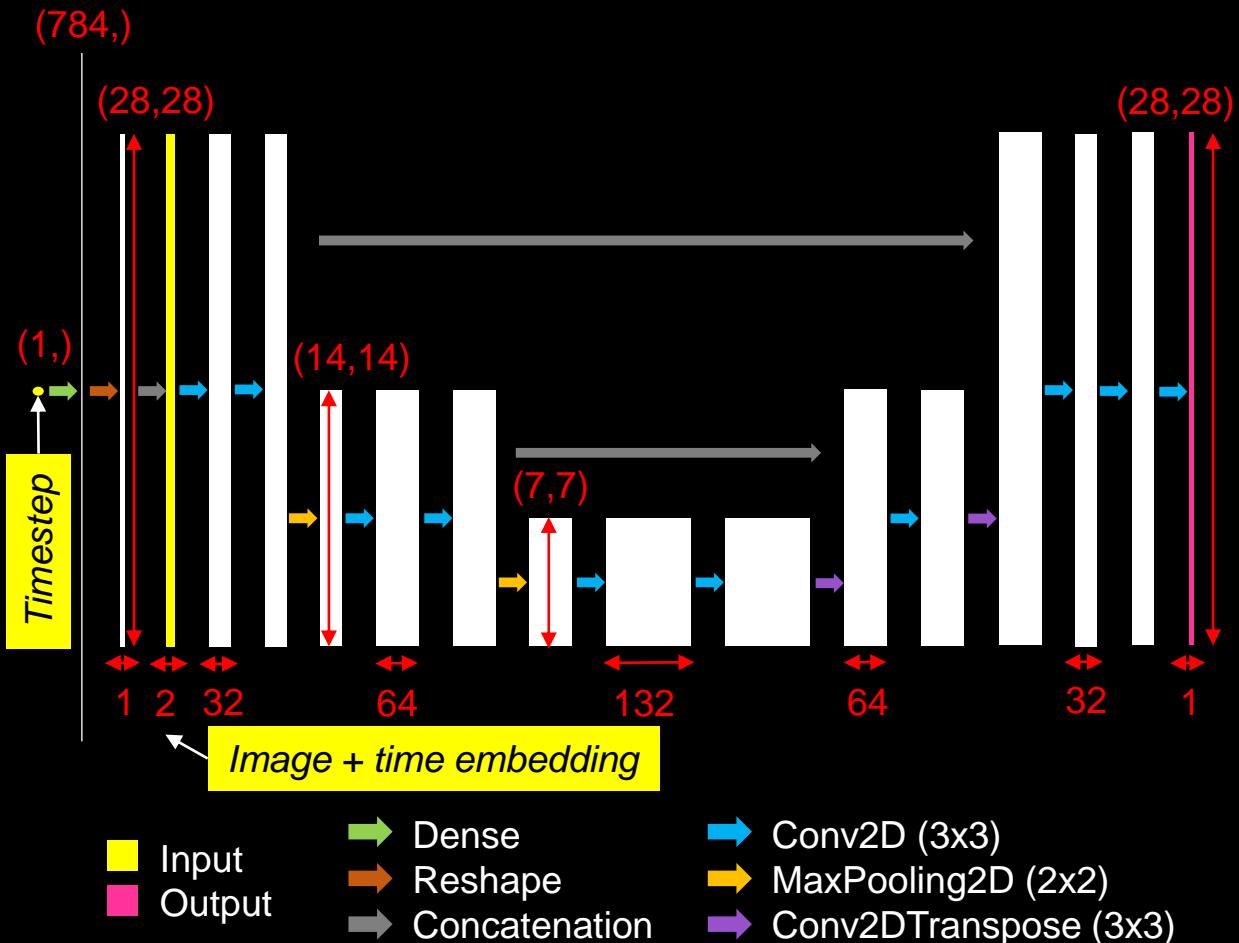


## Generation of 5 random denoising paths:



# Model 4

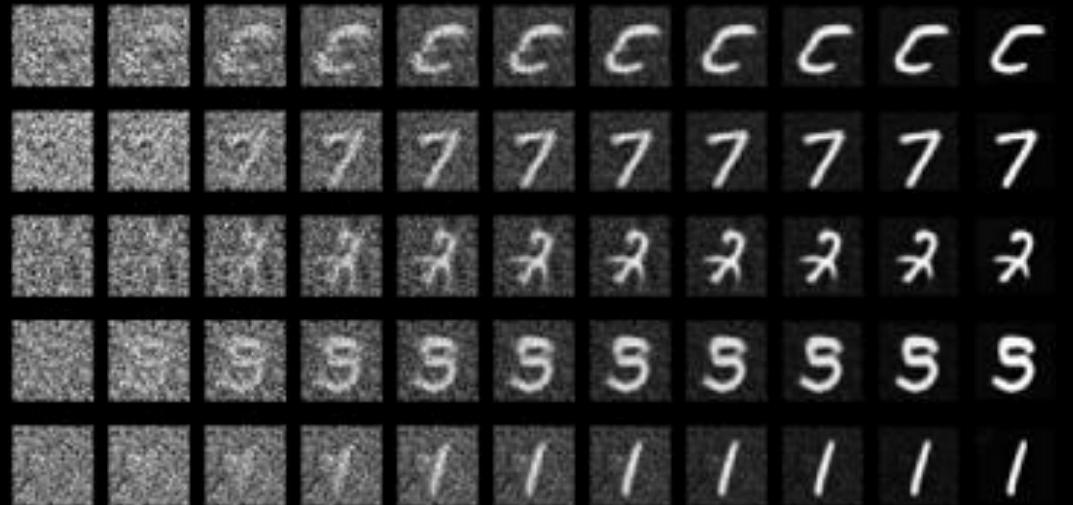
## Network architecture:



## Generation of 50 random images:

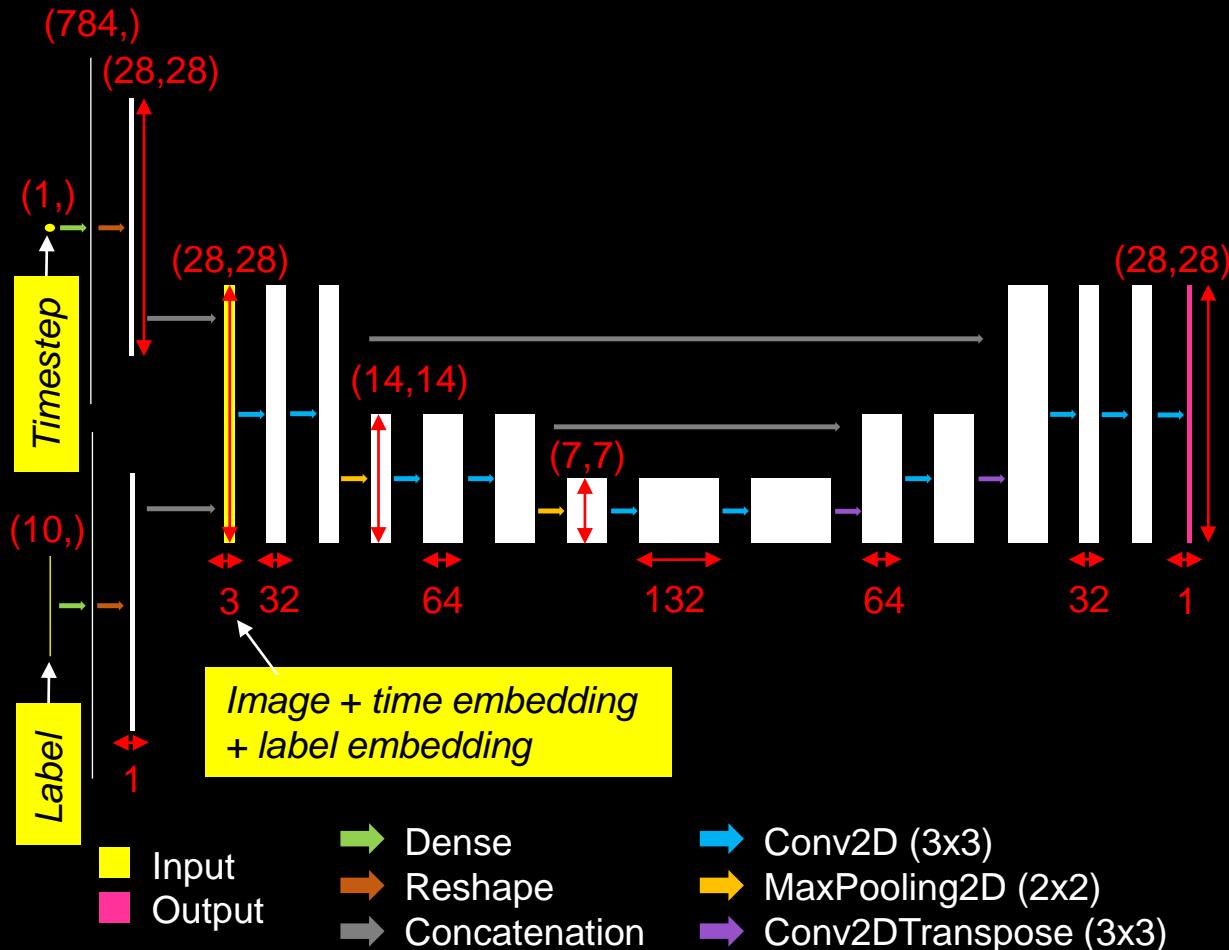


## Generation of 5 random denoising paths:



# Model 5

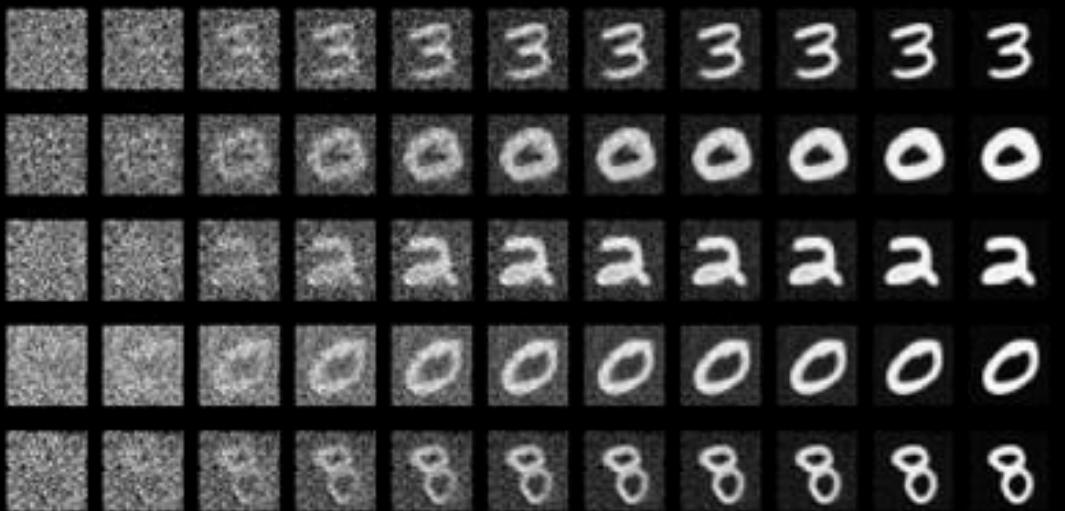
## Network architecture:



## Generation of 50 random images:



## Generation of 5 random denoising paths:



# Evaluation metric

We used the **Fréchet Distance** to evaluate our models.

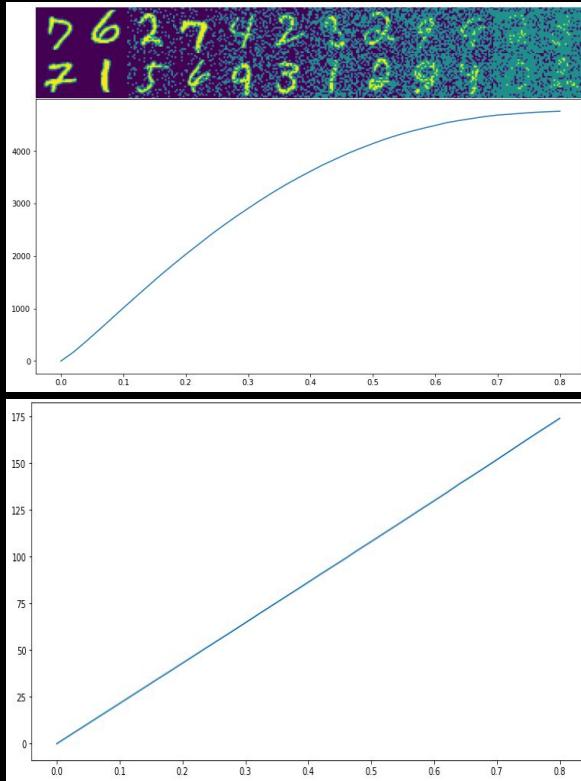
A distance between probability distribution that can be **estimated by sampling**.

$$FD(X, Y)^2 = \|\mu_X - \mu_Y\|^2 + \text{Tr}(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y})$$

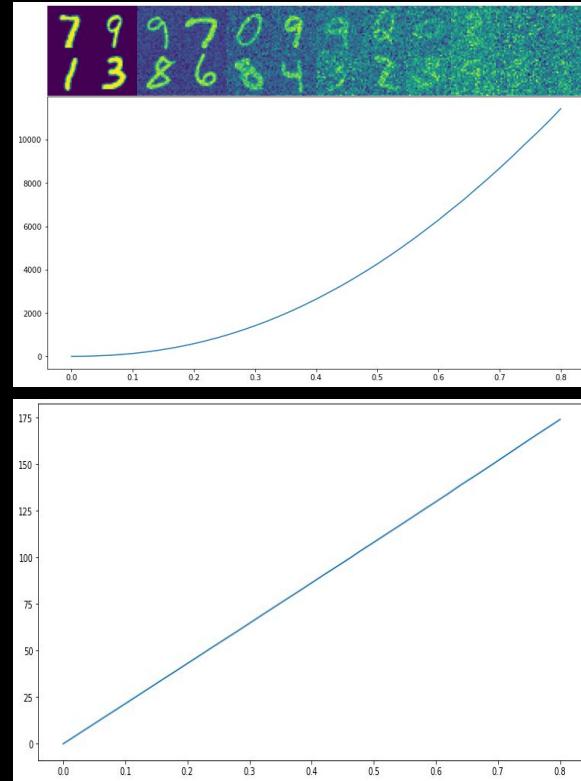
To obtain better results, we choose to apply the Fréchet Distance to an **embedding** obtained with a **Variational Auto Encoder**

# Metric comparison

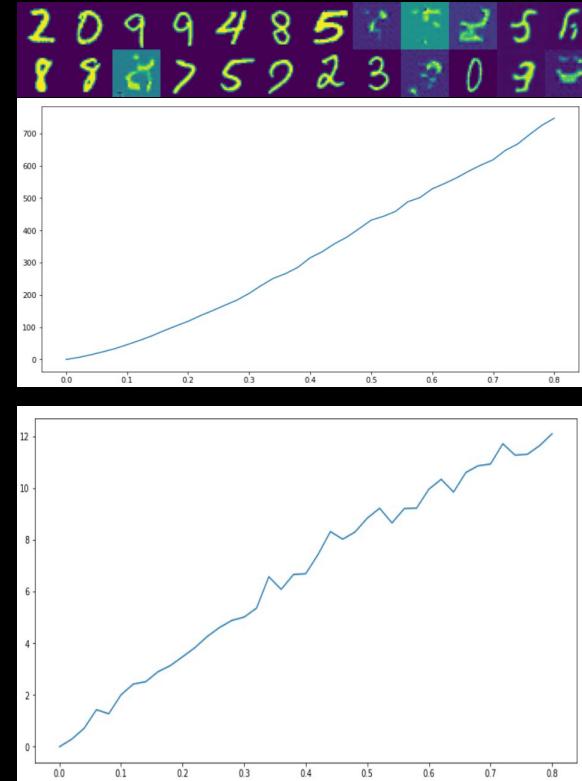
Dead pixels



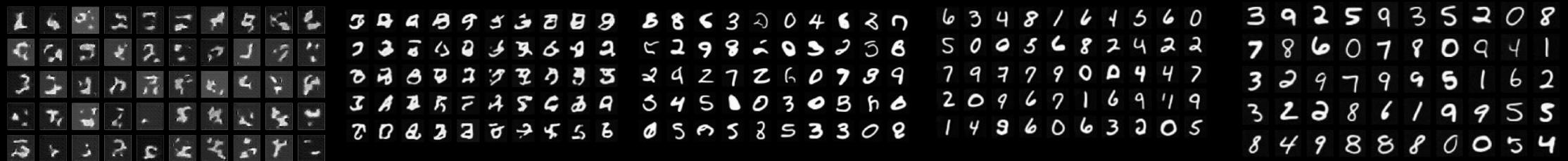
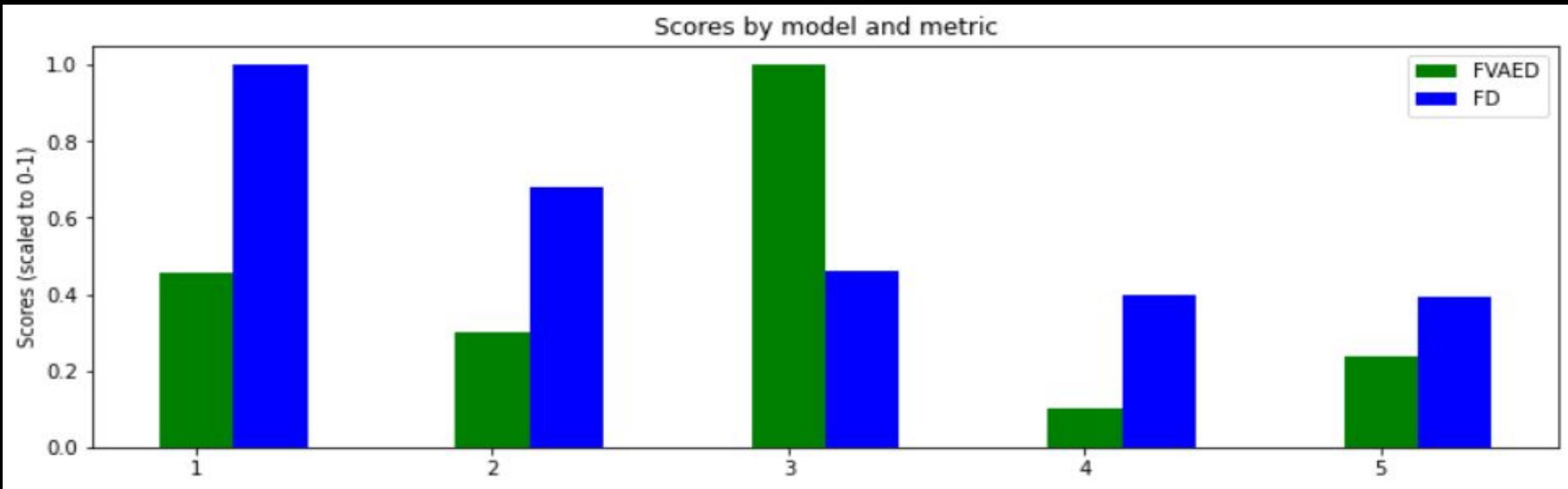
Gaussian noise



Invasion by another set



# Results evaluation



Thank you for  
your attention

