A theory of Creativity in Convolutional Diffusion Models

Luca Maria Del Bono

Based on the paper: "An analytic theory of creativity in convolutional diffusion models" by Kamb and Ganguli

The Age of Diffusion Models

Diffusion models are currently the primary way to generate images: e.g. ChatGPT uses **DALL·E**

The Age of Diffusion Models

Diffusion models are currently the primary way to generate images: e.g. ChatGPT uses **DALL·E**

“A realistic image of a dog”

The Age of Diffusion Models

Diffusion models are currently the primary way to generate images: e.g. ChatGPT uses **DALL·E**

"A realistic image of a dog"

"An impressionist style painting of a theoretical physicist at work"

The Age of Diffusion Models

Diffusion models are currently the primary way to generate images: e.g. ChatGPT uses **DALL·E**

“A realistic image of a dog”

“An impressionist style painting of a theoretical physicist at work”

"A crow-like version of Cthulhu, in comic book style"

The Age of Diffusion Models

Diffusion models are currently the primary way to generate images: e.g. ChatGPT uses **DALL·E**

“A realistic image of a dog”

“An impressionist style painting of a theoretical physicist at work”

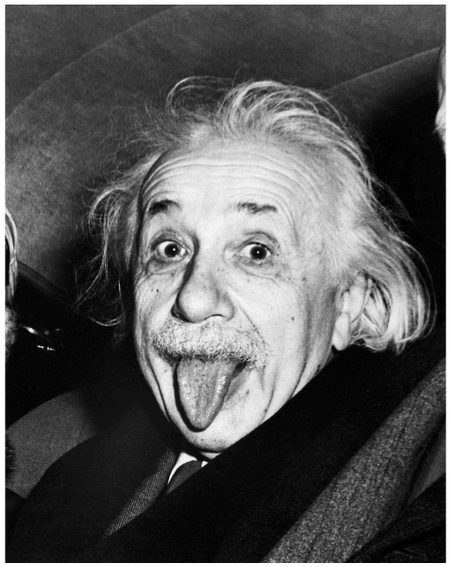
"A crow-like version of Cthulhu, in comic book style"

Diffusion models have a deep connection to physics!

A step back: diffusion in physics

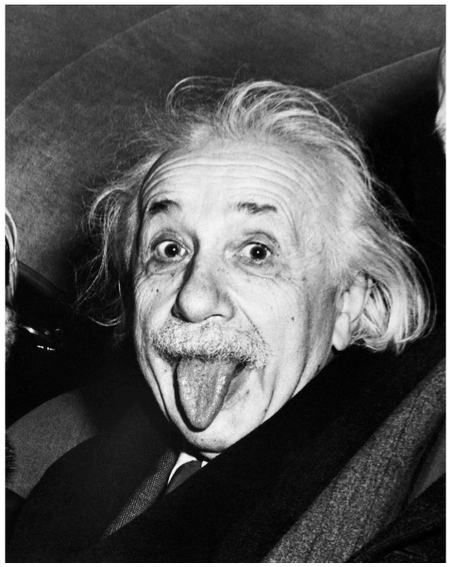
A step back: diffusion in physics

1905

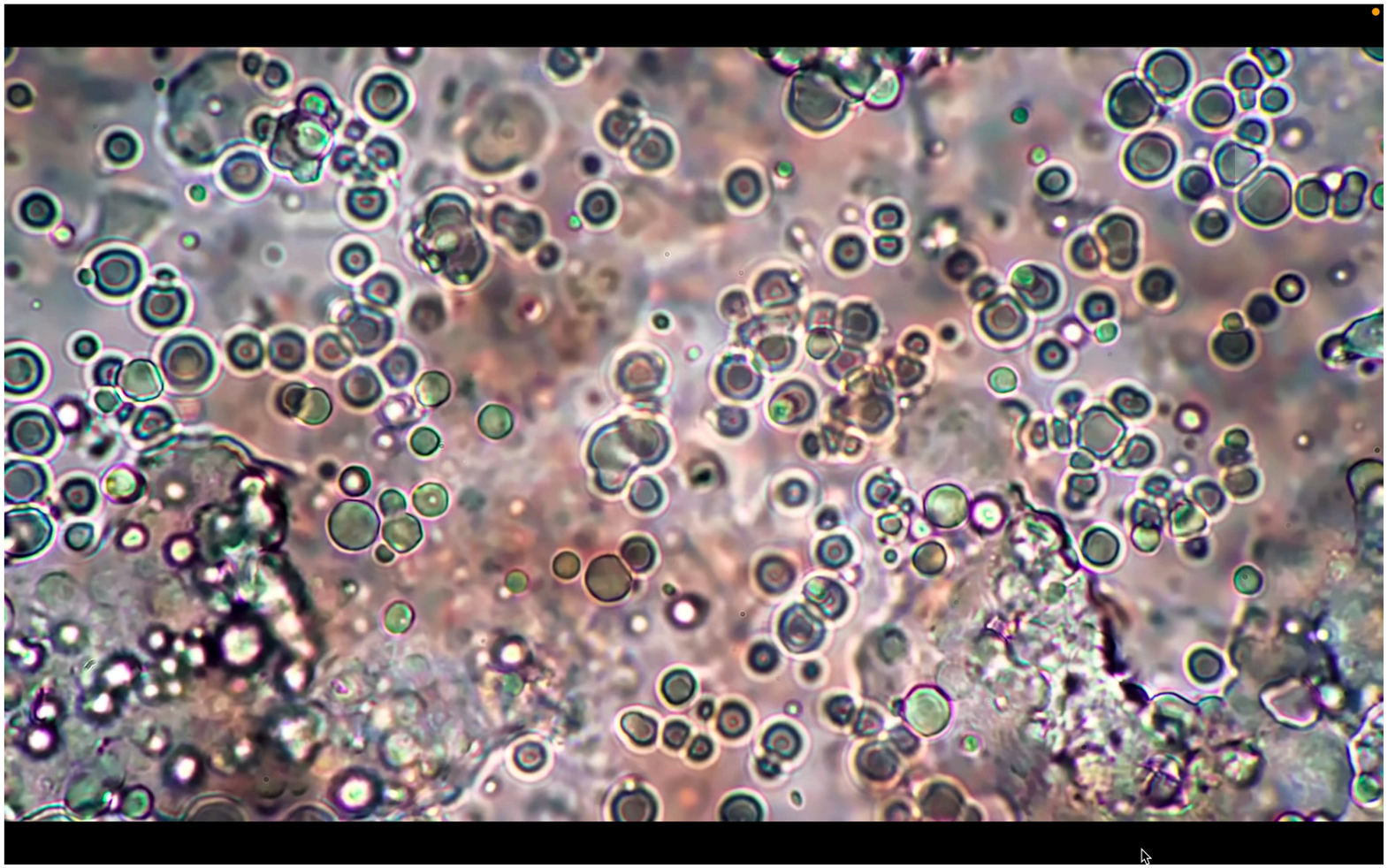


A step back: diffusion in physics

1905



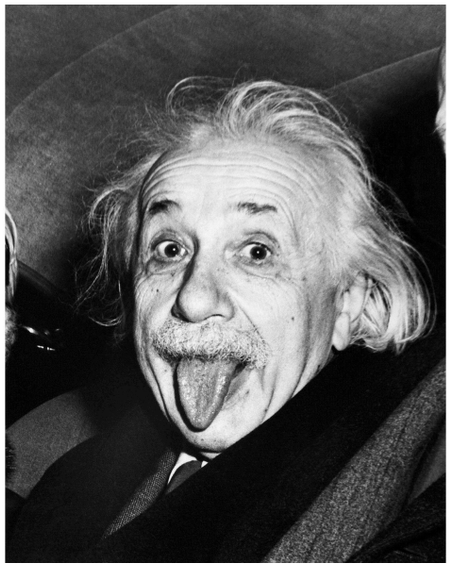
Brownian motion



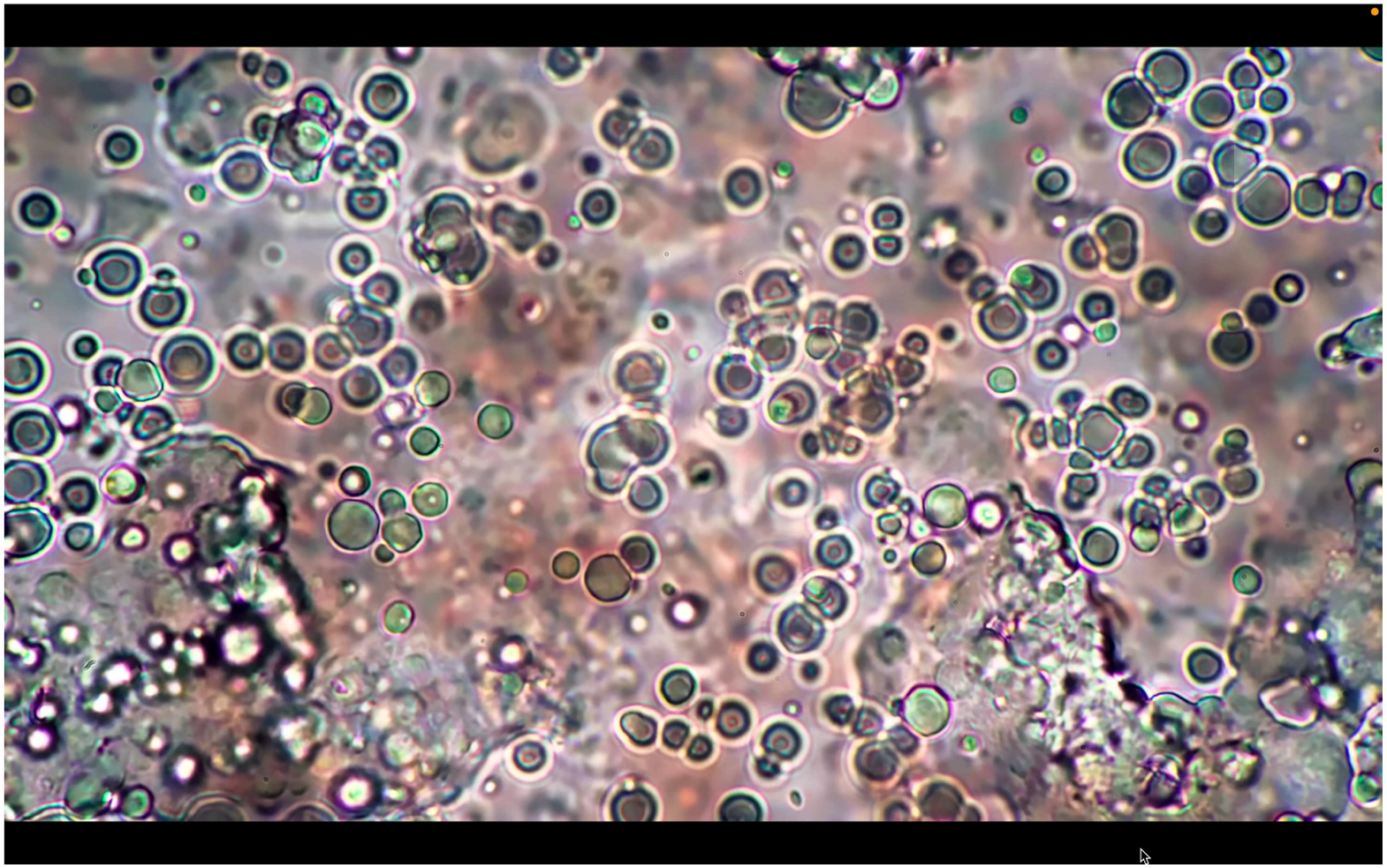
“The experiment that revealed the atomic world: Brownian Motion”

A step back: diffusion in physics

1905



Brownian motion



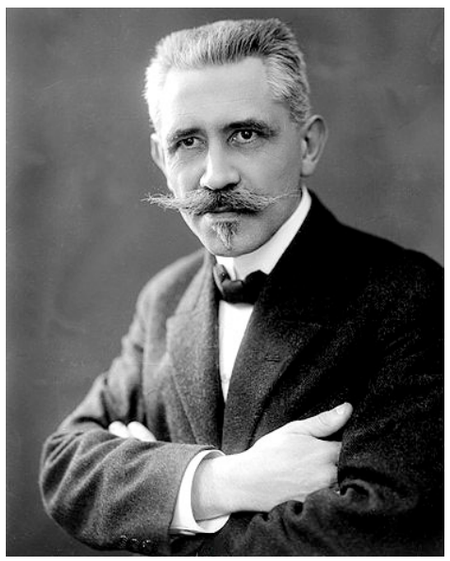
"The experiment that revealed the atomic world: Brownian Motion"

$P(x,t)$ is the probability of finding the grain at **position** x at **time** t and D is the **diffusion constant**

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial^2 P(x,t)}{\partial x^2} \Rightarrow P(x,t) = \frac{1}{\sqrt{4\pi D t}}\exp\left(-\frac{x^2}{4Dt}\right)$$

A step back: diffusion in physics

1908



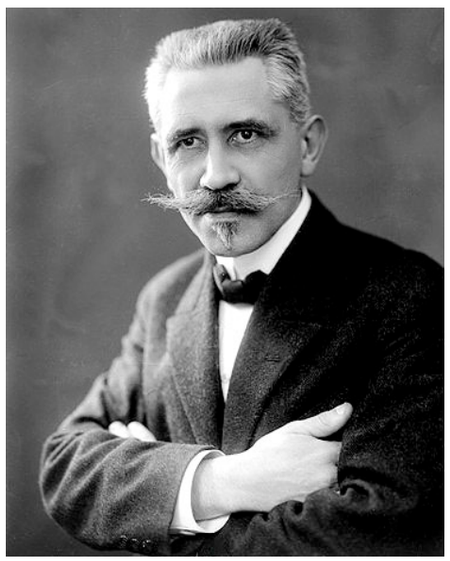
$$dx = vdt$$

$$dv = -\gamma v\, dt + \sigma dW_t$$

dx is the **displacement** in **time** dt. v is the velocity. γ and σ are **constants** ad dW_t is an **infinitesimal Wiener process**

A step back: diffusion in physics

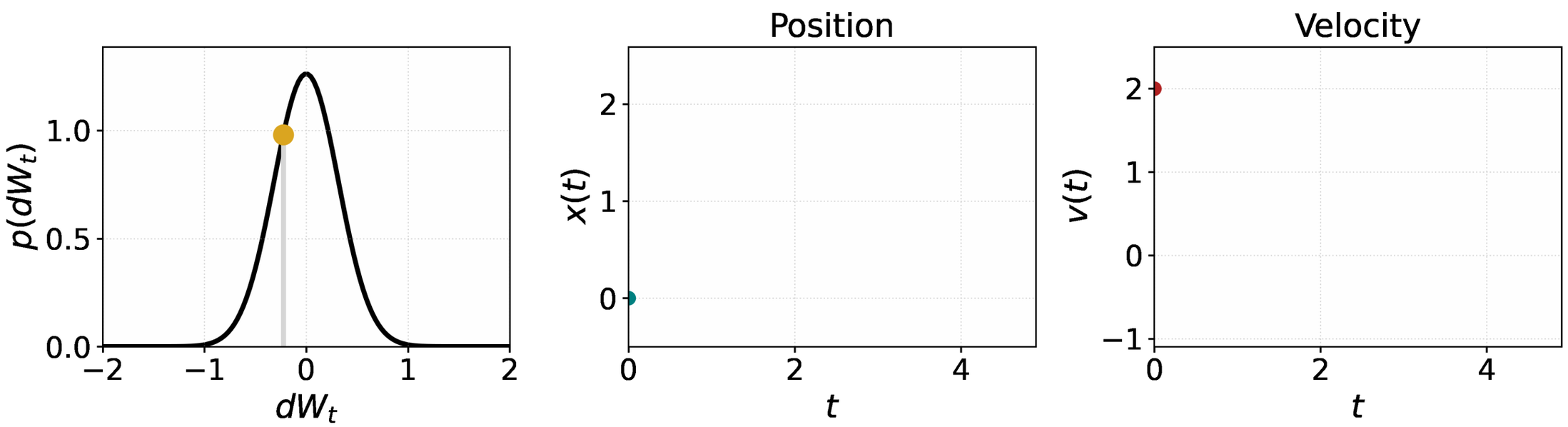
1908



$$dx = v\,dt$$

$$dv = -\gamma v\,dt + \sigma dW_t$$

dx is the **displacement** in **time** dt. v is the velocity. γ and σ are **constants** ad dW_t is an **infinitesimal Wiener process**



Look at the velocity: we started with something that had information and ended up losing it

Inverting the dynamics

It turns out that performing a **backwards process** according to

$$\frac{dv}{dt} = -\gamma\, v - \frac{1}{2}\sigma^2 s_t(v)$$

Score

$$s_t(v) = \partial_v \log \pi_t(v)$$

reverses the dynamics

Inverting the dynamics

It turns out that performing a **backwards process** according to

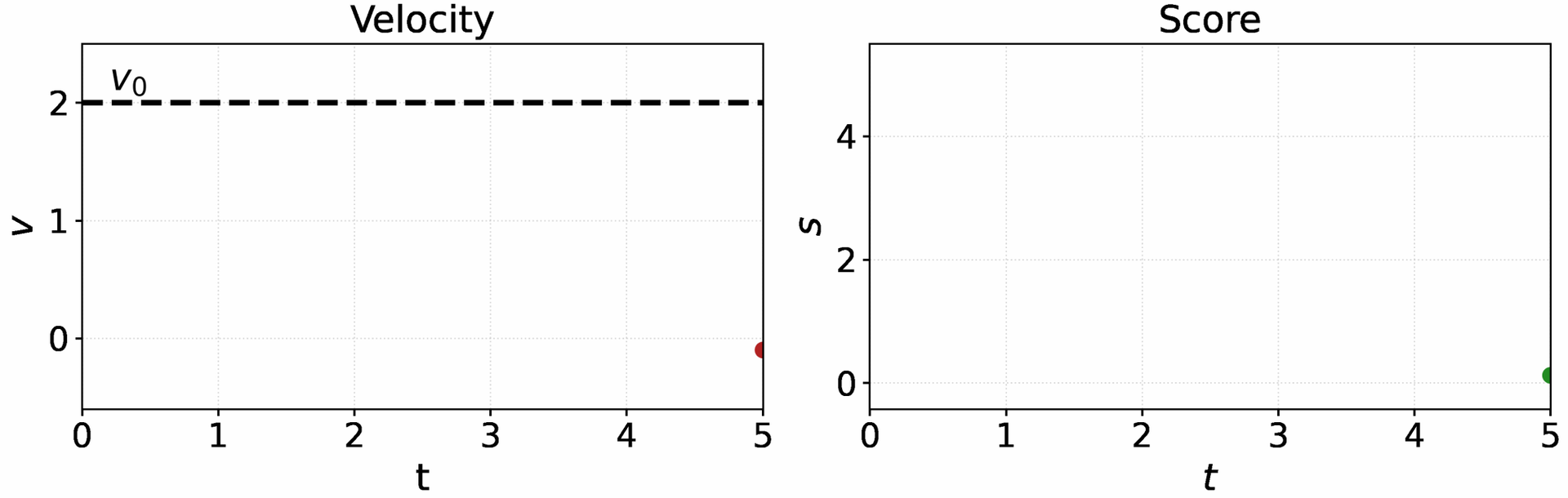
$$\frac{dv}{dt} = -\gamma\, v - \frac{1}{2}\sigma^2 s_t(v)$$

Score

$$s_t(v) = \partial_v \log \pi_t(v)$$

reverses the dynamics

$$s_t(v) = -\frac{2\gamma\left(v - v_0 e^{-\gamma t}\right)}{1 - e^{-2\gamma t}}$$



We started with something that did not have information and ended up with something that does!

Inverting the dynamics

It turns out that performing a **backwards process** according to

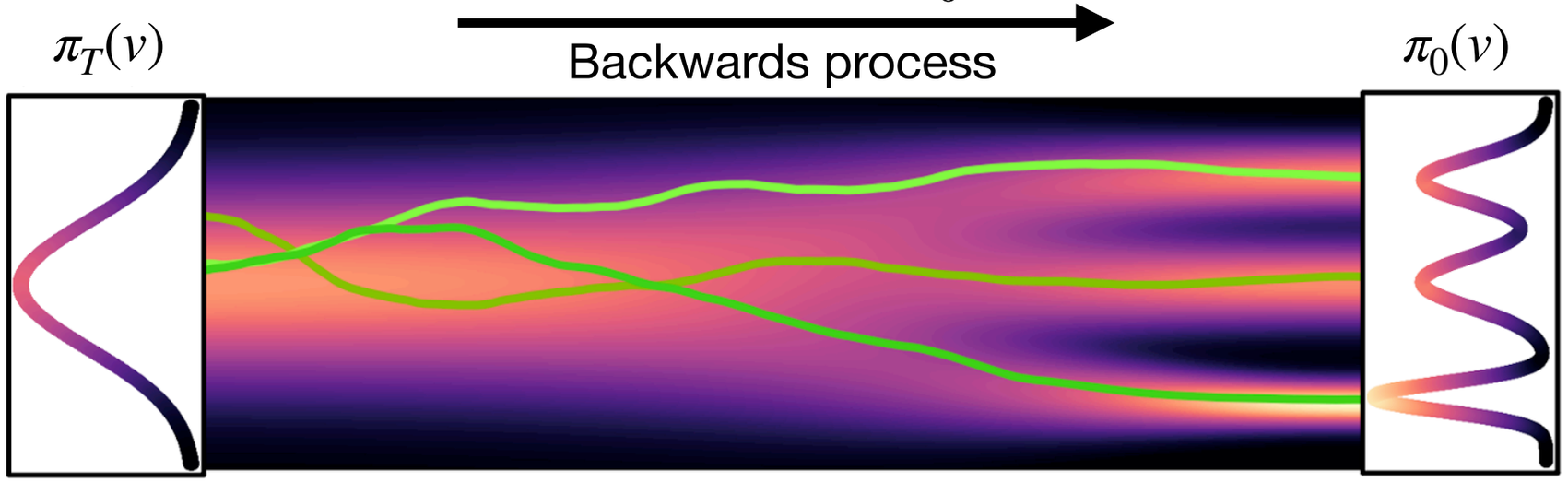
$$\frac{dv}{dt} = -\gamma v - \frac{1}{2}\sigma^2 s_t(v)$$

Score

$$s_t(v) = \partial_v \log \pi_t(v)$$

reverses the dynamics

In our example, we only had one starting velocity ($v_0 = 2$). In general, if we have a distribution over possible initial velocities $\pi_0(v)$, $s_t(v)$ will be more complicated, but reversing in time will generate data according to π_0



"https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-2/"

From diffusing particles to diffusing pixels

$$dv = -\gamma v\, dt + \sigma dW_t$$

$$d\phi_t = -\ \phi_t$$

The velocities are now vectors of pixels $\phi_t \in \mathbb{R}^N$

From diffusing particles to diffusing pixels

$$dv = -\gamma v\, dt + \sigma dW_t$$

$$d\phi_t = -\gamma_t \phi_t\, dt + \sqrt{2\gamma_t}\, dW_t$$

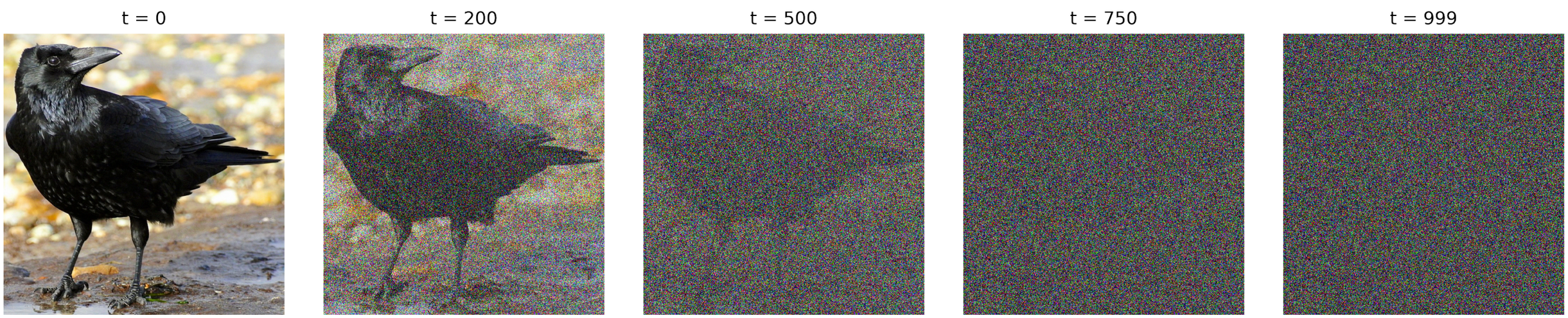
The coefficients are no longer homogenous in time, but are chosen according to a schedule γ_t

From diffusing particles to diffusing pixels

$$dv = -\gamma v\, dt + \sigma dW_t$$

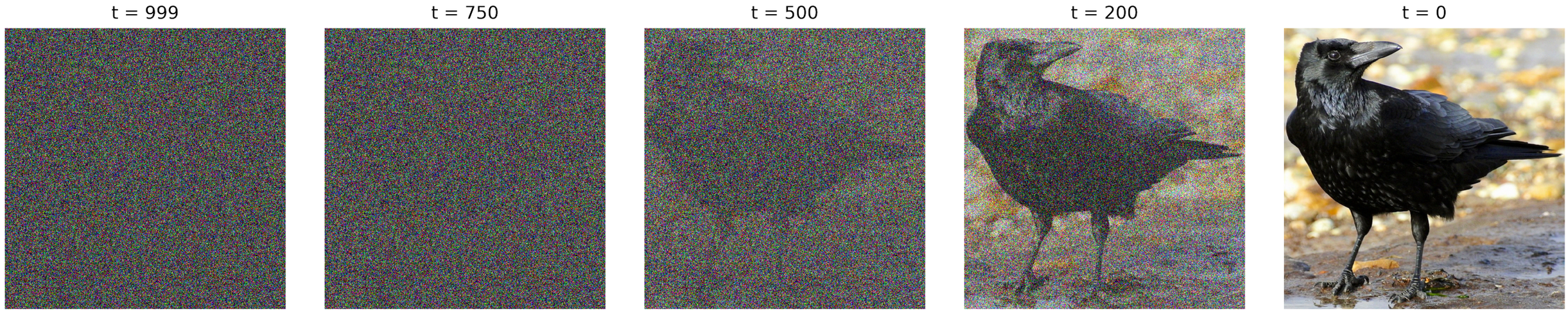
$$d\phi_t = -\gamma_t \phi_t\, dt + \sqrt{2\gamma_t}\, dW_t$$

The coefficients are no longer homogenous in time, but are chosen according to a schedule γ_t



From diffusing particles to diffusing pixels

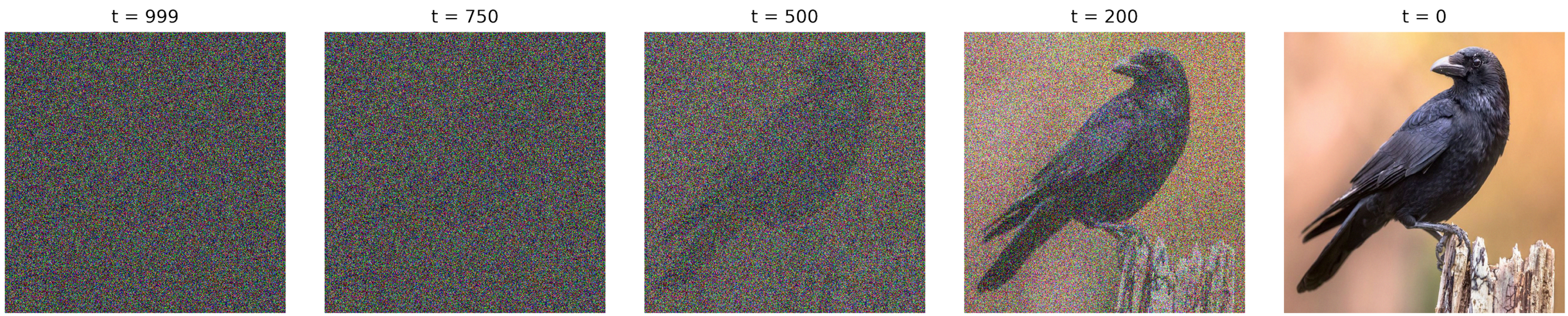
If we now reverse the process, we get the **same image**



Backwards process

From diffusing particles to diffusing pixels

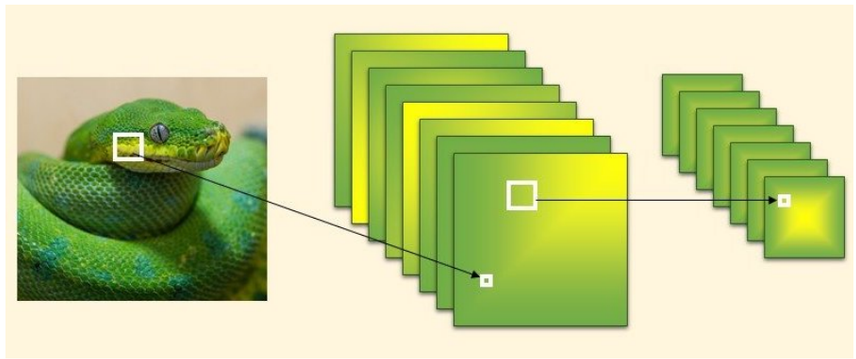
But the idea is that now, starting from **another realization** of the noise and using the score, we can get **new data**!



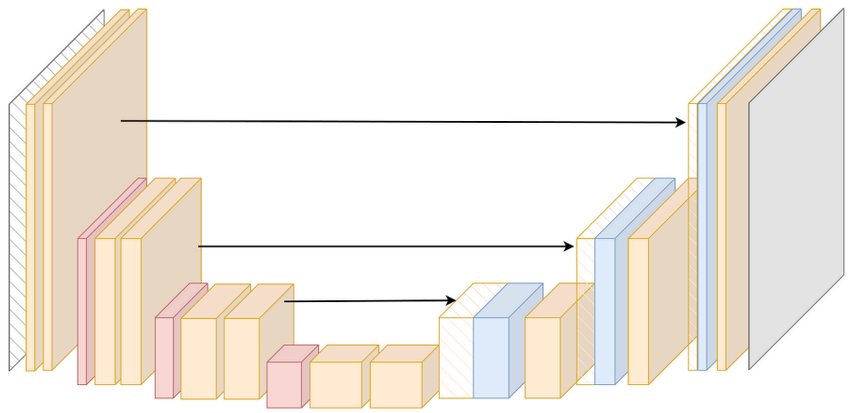
Backwards process

How does it work in practice?

The score is not know exactly, so one uses an approximation $\hat{s}_{\theta(t)}$ it using **a neural network**, e.g. a **Convolutional Neural Network (CNN)**



CNN convolution



UNet architecture

One trains the NN by minimizing the **loss**:

$$\mathscr{L}_t(\theta) = \int d\phi\, \pi_t(\phi) \left\| \hat{s}_{\theta(t)}(\phi) - s_t(\phi) \right\|^2$$

That is, you **simulate many trajectories and minimize**:

$$\mathscr{L}(\theta) = \mathbb{E}_{\phi,\phi_0} \left\| \hat{s}_{\theta(t)}(\phi) - s_t(\phi) \right\|^2$$

Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \,|\, \phi)$$

Posterior belief distribution

$$W_t(\varphi \,|\, \phi) = \frac{N(\phi \,|\, \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \,|\, \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix

Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

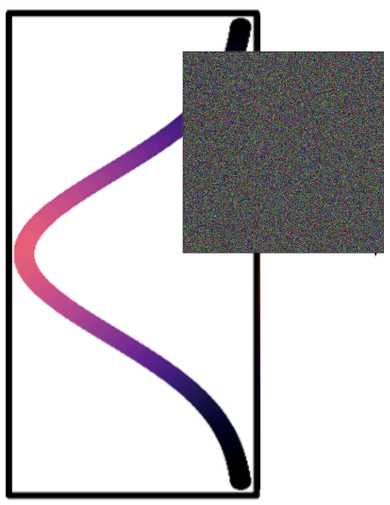
$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

Posterior belief distribution

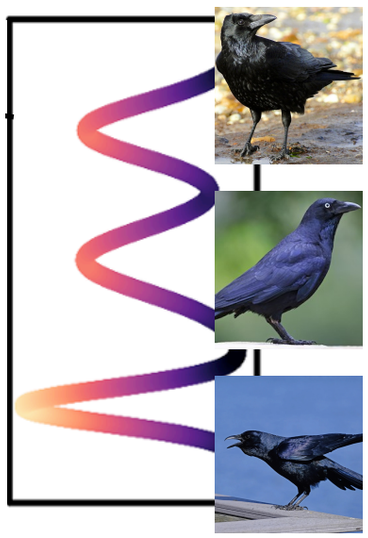
$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix



Training set



Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

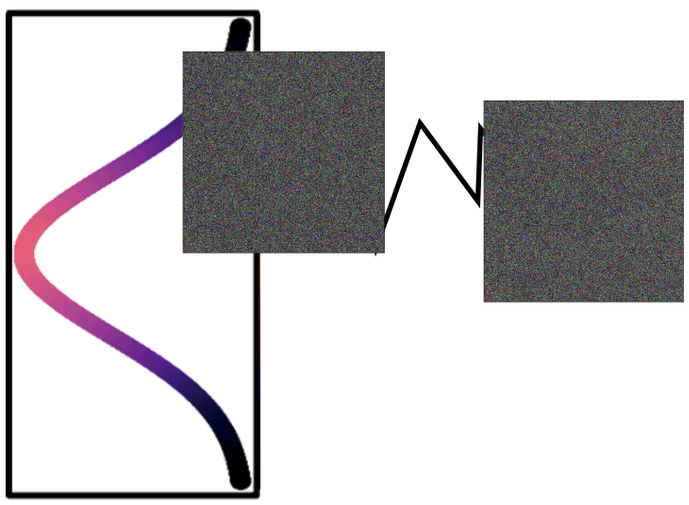
$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

Posterior belief distribution

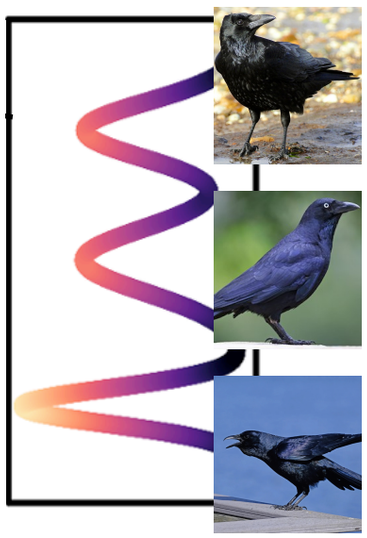
$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix



Training set



Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

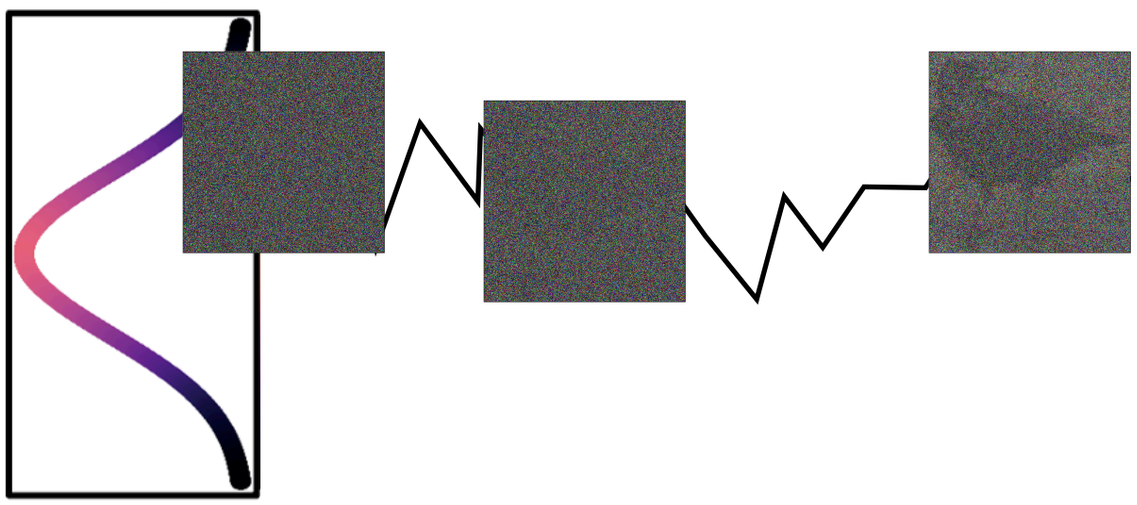
$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

Posterior belief distribution

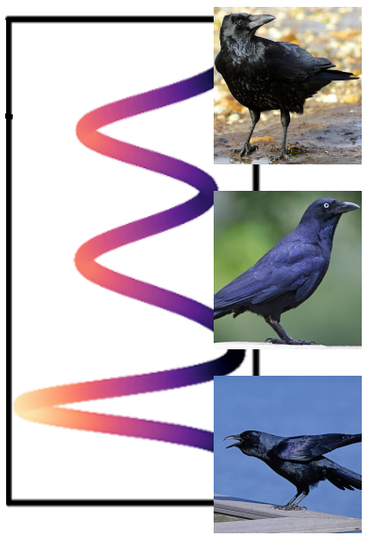
$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix



Training set



Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

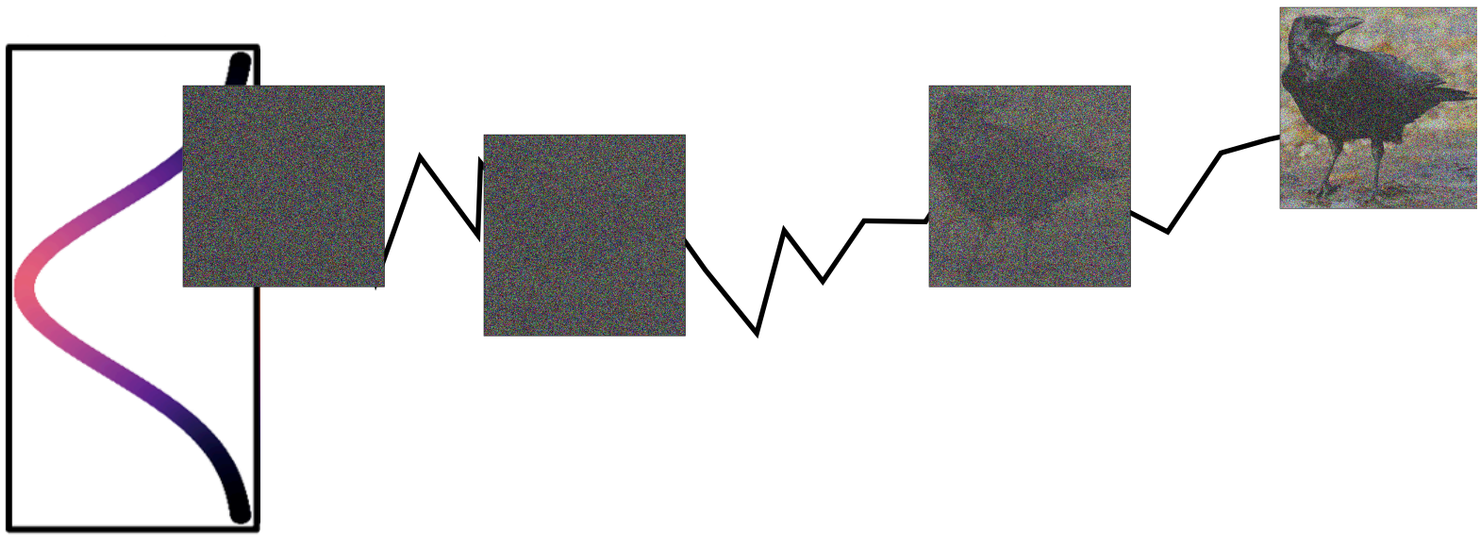
$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

Posterior belief distribution

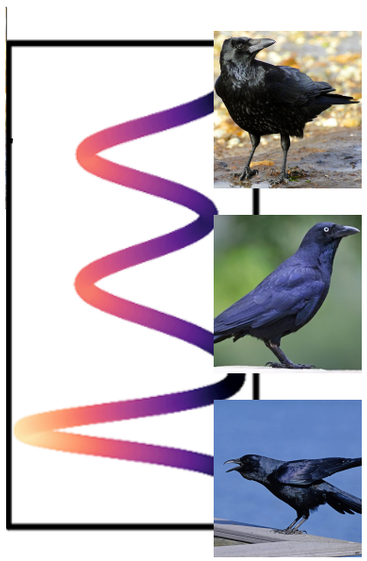
$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix



Training set



Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

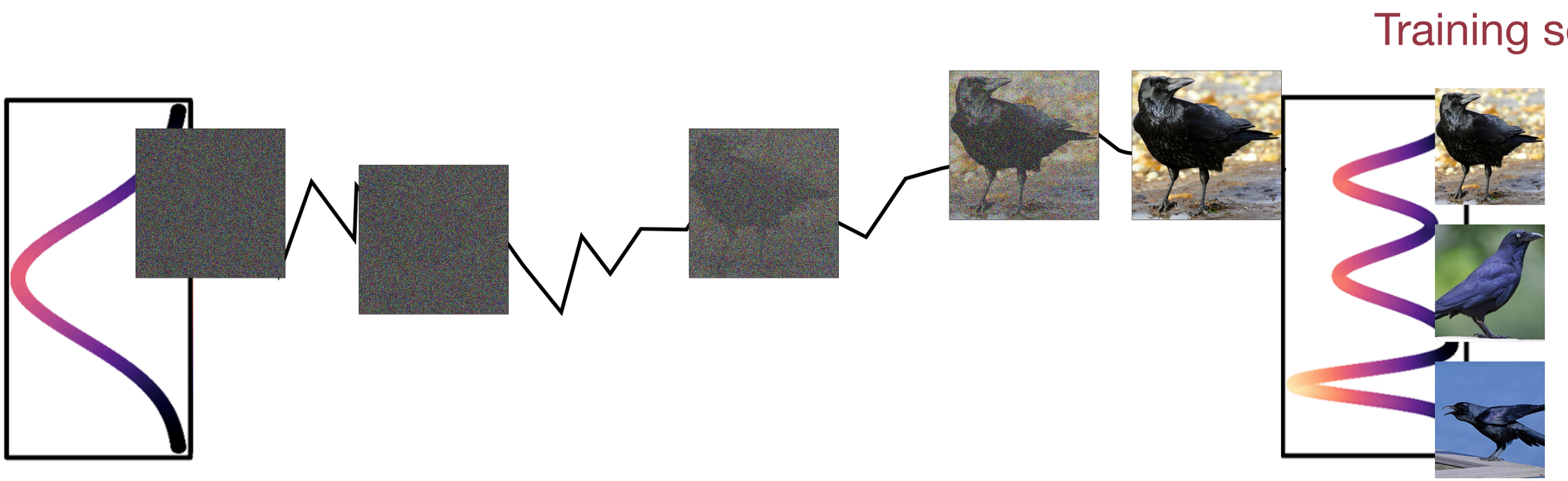
$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

Posterior belief distribution

$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix



Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

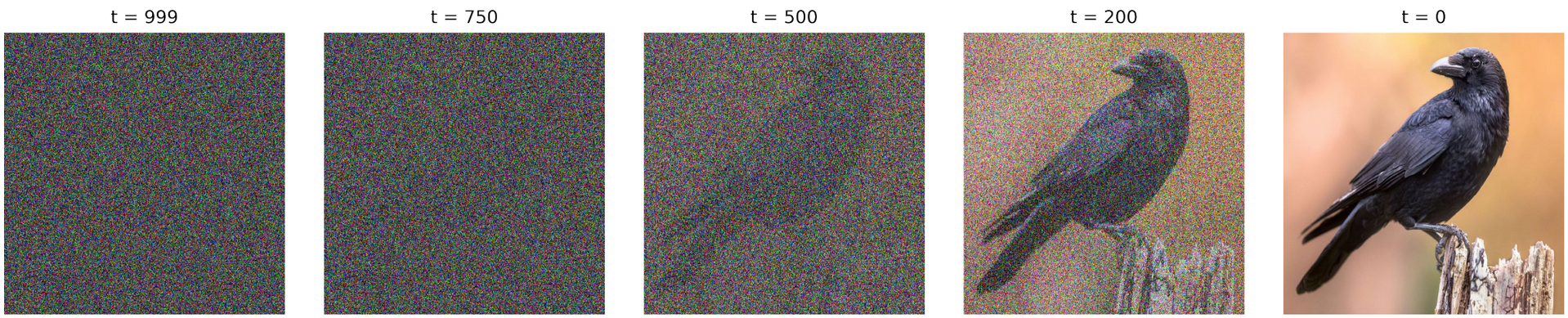
Posterior belief distribution

$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix

So, when performing backwards diffusion, we are only able to **generate elements from the dataset** (i.e. we only have **memory**, not "**creativity**")



Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

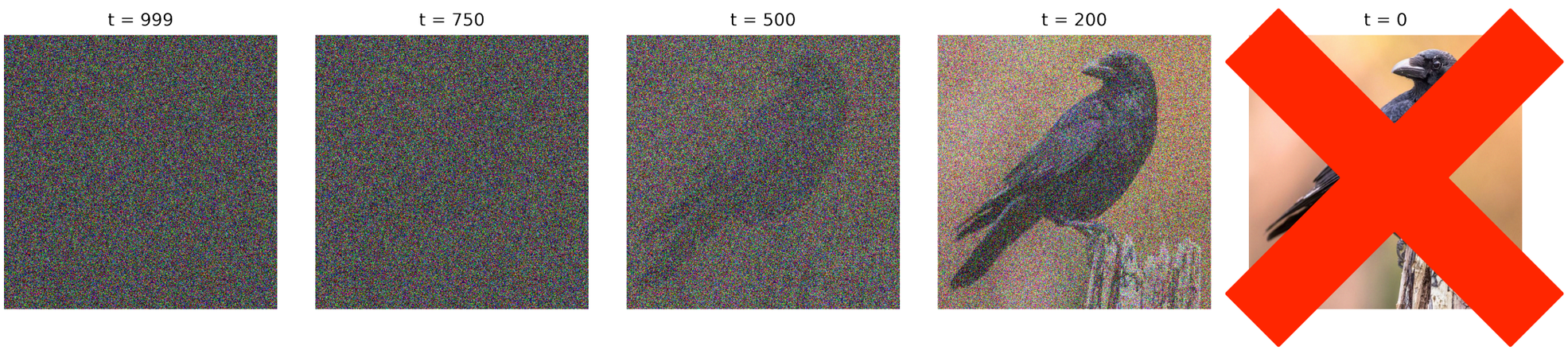
Posterior belief distribution

$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix

So, when performing backwards diffusion, we are only able to **generate elements from the dataset** (i.e. we only have **memory**, not "**creativity**")



Problem: memorization!

But there is a problem! For a given training dataset D the score can be written as

$$s_t(\phi) = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in D} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

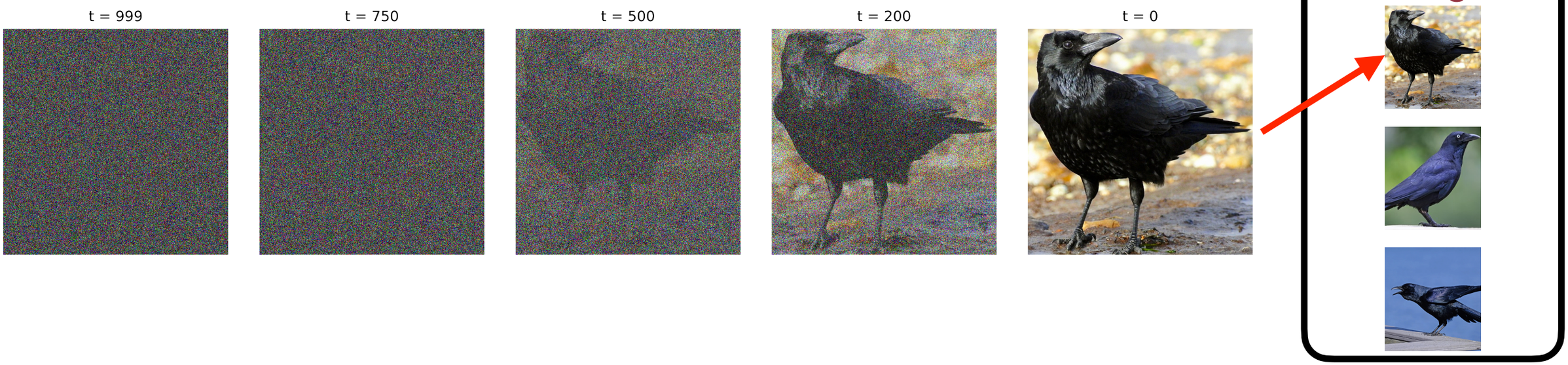
Posterior belief distribution

$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

$$\bar{\alpha}_t = \exp\left(-2 \int_0^t \gamma_t \, dt \right)$$

N : Gaussian pdf
I : Identity matrix

So, when performing backwards diffusion, we are only able to **generate elements from the dataset** (i.e. we only have **memory**, not "**creativity**")



Can inductive biases help?

Translational invariance

CNNs leverage the property of images of being **translationally invariant**

We can include this in the score by considering the **group of translations** G, obtaining:

$$s_t[\phi] = \frac{1}{1-\bar{\alpha}_t} \sum_{\varphi \in G(D)} \left(\sqrt{\bar{\alpha}_t}\varphi - \phi \right) W_t(\varphi \mid \phi)$$

$$W_t(\varphi \mid \phi) = \frac{N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in G(D)} N(\phi \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

So now, instead of flowing to images in the dataset, we flow to any possible translation of images in the dataset

Can inductive biases help?

Locality

In CNNs a pixel is only influenced by a **subset of nearby pixels**

We can introducing this locality by considering the dependence on a **pixel** x and on its **neighborhood** Ω_x

$$\hat{s}_t[\phi](x) = \sum_{\varphi \in D} \frac{\left(\sqrt{\bar{\alpha}_t}\varphi(x) - \phi(x)\right)}{1-\bar{\alpha}_t} W_t(\varphi_{\Omega_x} | \phi_{\Omega_x})$$

$$W_t(\varphi_{\Omega_x} | \phi_{\Omega_x}) = \frac{N(\phi_{\Omega_x} | \sqrt{\bar{\alpha}_t}\varphi_{\Omega_x}, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi_{\Omega_x} | \sqrt{\bar{\alpha}_t}\varphi'_{\Omega_x}, (1-\bar{\alpha}_t)I)}$$

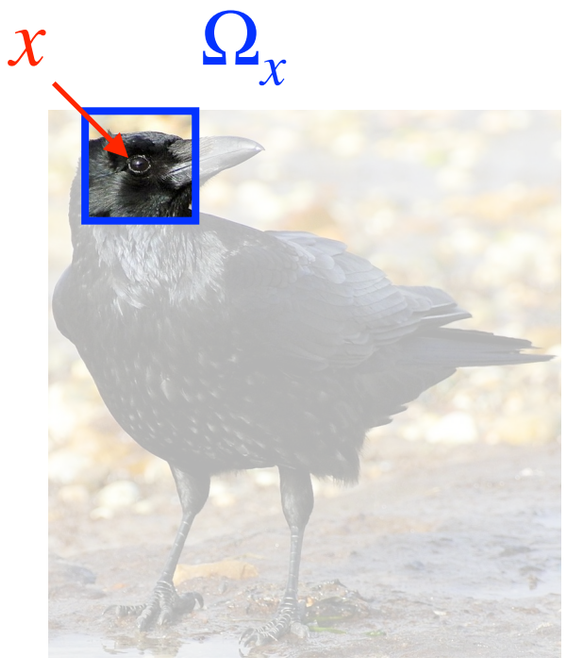
Far away pixels can flow independently to different images in the dataset

Can inductive biases help?

Locality

In CNNs a pixel is only influenced by a **subset of nearby pixels**

We can introducing this locality by considering the dependence on a **pixel** x and on its **neighborhood** Ω_x



$$\hat{s}_t[\phi](x) = \sum_{\varphi \in D} \frac{\left(\sqrt{\bar{\alpha}_t}\varphi(x) - \phi(x)\right)}{1 - \bar{\alpha}_t} W_t(\varphi_{\Omega_x} | \phi_{\Omega_x})$$

$$W_t(\varphi_{\Omega_x} | \phi_{\Omega_x}) = \frac{N(\phi_{\Omega_x} | \sqrt{\bar{\alpha}_t}\varphi_{\Omega_x}, (1 - \bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi_{\Omega_x} | \sqrt{\bar{\alpha}_t}\varphi'_{\Omega_x}, (1 - \bar{\alpha}_t)I)}$$

Far away pixels can flow independently to different images in the dataset

Can inductive biases help?

Locality

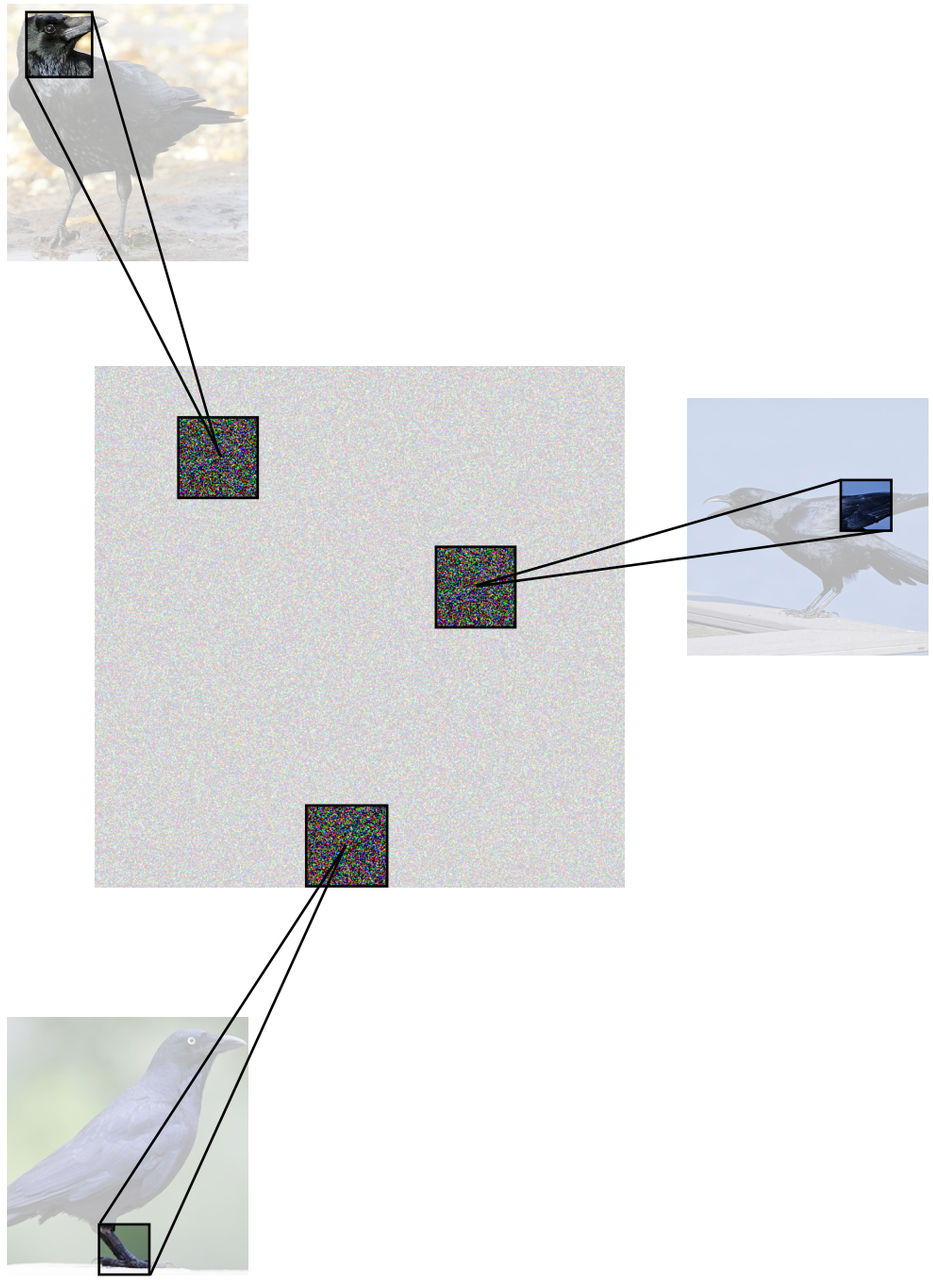
In CNNs a pixel is only influenced by a **subset of nearby pixels**

We can introducing this locality by considering the dependence on a **pixel** x and on its **neighborhood** Ω_x

$$\hat{s}_t[\phi](x) = \sum_{\varphi \in D} \frac{\left(\sqrt{\bar{\alpha}_t}\varphi(x) - \phi(x)\right)}{1-\bar{\alpha}_t} W_t(\varphi_{\Omega_x} | \phi_{\Omega_x})$$

$$W_t(\varphi_{\Omega_x} | \phi_{\Omega_x}) = \frac{N(\phi_{\Omega_x} | \sqrt{\bar{\alpha}_t}\varphi_{\Omega_x}, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in D} N(\phi_{\Omega_x} | \sqrt{\bar{\alpha}_t}\varphi'_{\Omega_x}, (1-\bar{\alpha}_t)I)}$$

Far away pixels can flow independently to different images in the dataset



Putting all together

Putting together the biases, we find the **Equivariant Local Score (ELS)** machine

$$s_t[\phi](x) = \sum_{\varphi \in P_\Omega(D)} \frac{\left(\sqrt{\bar{\alpha}_t}\varphi(0) - \phi(x)\right)}{1-\bar{\alpha}_t} W_t(\varphi \,|\, \phi, x)$$

$$W_t(\varphi \,|\, \phi, x) = \frac{N(\phi_{\Omega_x} \,|\, \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in P_\Omega(D)} N(\phi_{\Omega_x} \,|\, \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

The ELS mixes pixels coming from **different patches** of **different images.**

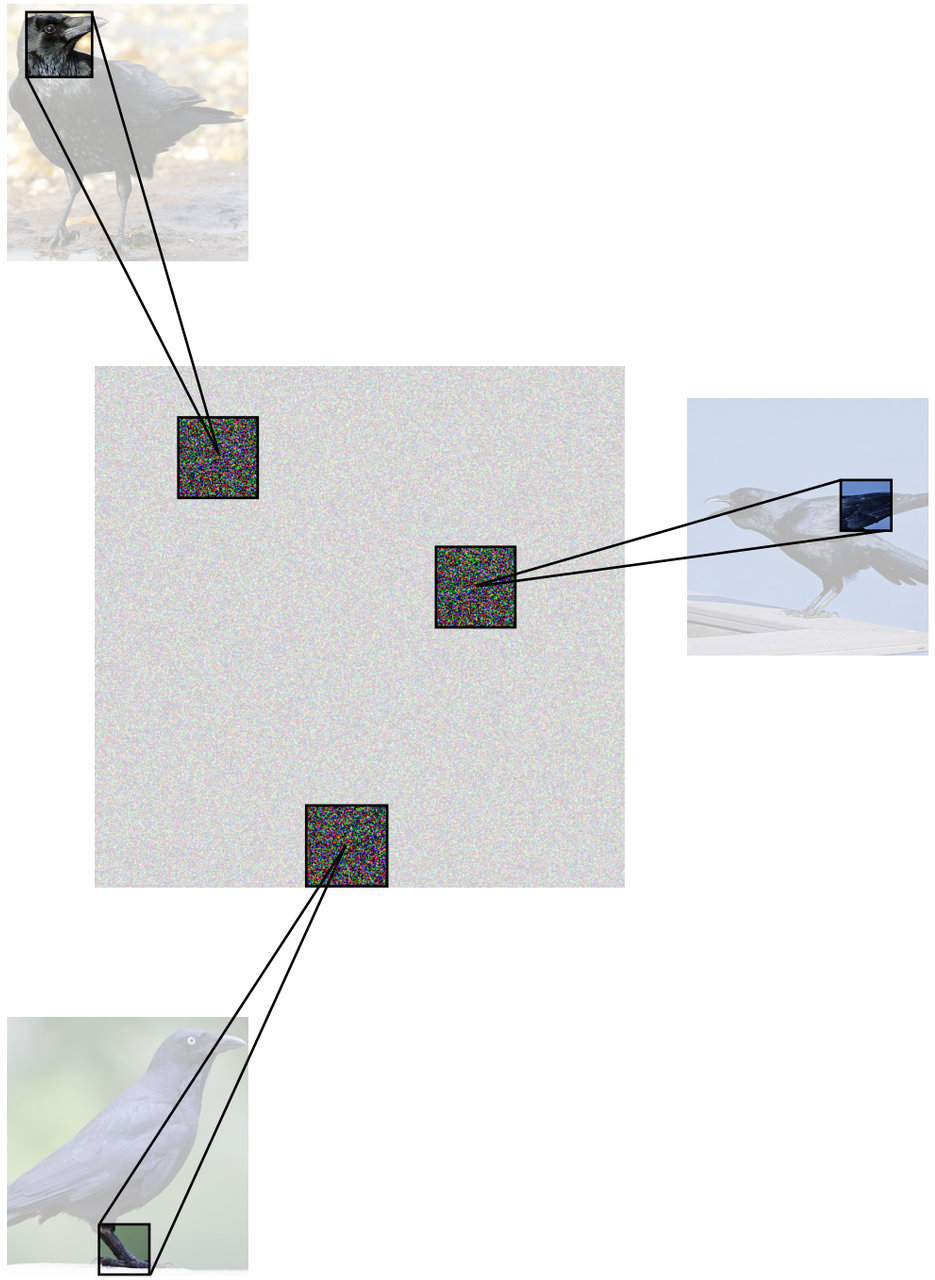
Putting all together

Putting together the biases, we find the **Equivariant Local Score (ELS)** machine

$$s_t[\phi](x) = \sum_{\varphi \in P_\Omega(D)} \frac{\left(\sqrt{\bar{\alpha}_t}\varphi(0) - \phi(x)\right)}{1-\bar{\alpha}_t} W_t(\varphi \,|\, \phi, x)$$

$$W_t(\varphi \,|\, \phi, x) = \frac{N(\phi_{\Omega_x} \,|\, \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in P_\Omega(D)} N(\phi_{\Omega_x} \,|\, \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

The ELS mixes pixels coming from **different patches** of **different images.**



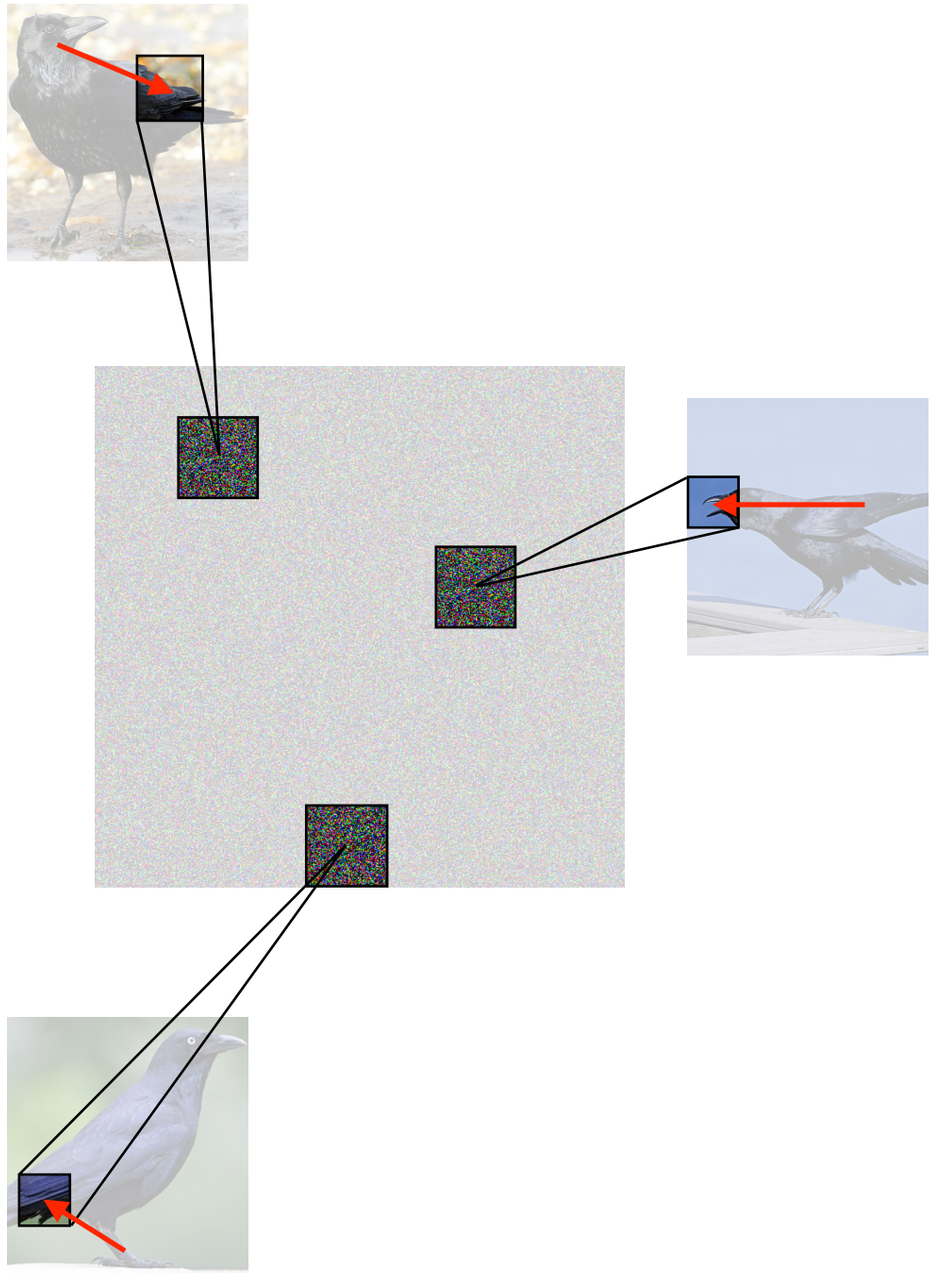
Putting all together

Putting together the biases, we find the **Equivariant Local Score (ELS)** machine

$$s_t[\phi](x) = \sum_{\varphi \in P_\Omega(D)} \frac{\left(\sqrt{\bar{\alpha}_t}\varphi(0) - \phi(x)\right)}{1-\bar{\alpha}_t} W_t(\varphi \mid \phi, x)$$

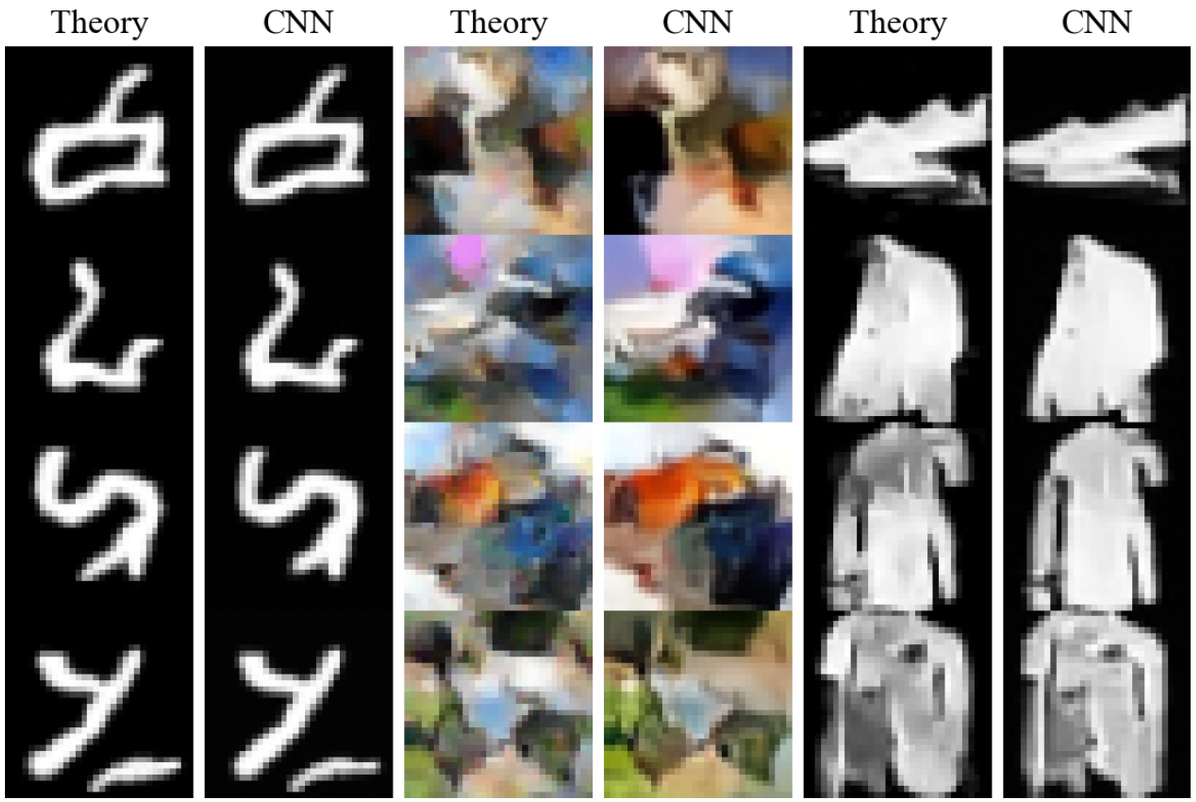
$$W_t(\varphi \mid \phi, x) = \frac{N(\phi_{\Omega_x} \mid \sqrt{\bar{\alpha}_t}\varphi, (1-\bar{\alpha}_t)I)}{\sum_{\varphi' \in P_\Omega(D)} N(\phi_{\Omega_x} \mid \sqrt{\bar{\alpha}_t}\varphi', (1-\bar{\alpha}_t)I)}$$

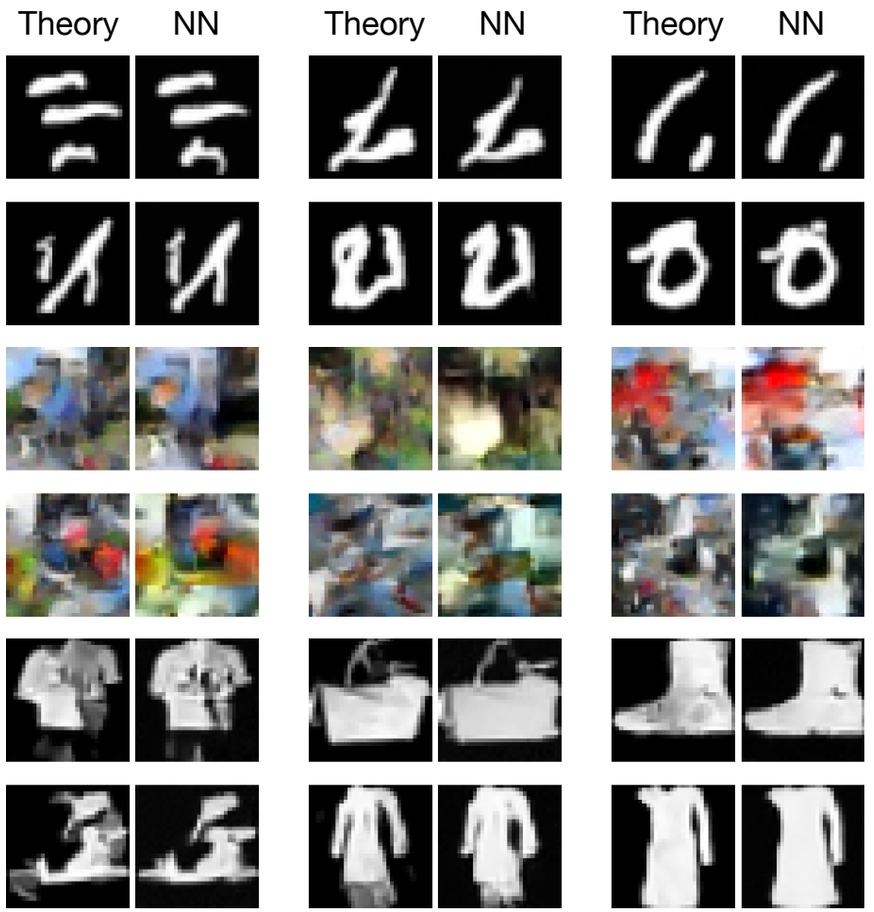
The ELS mixes pixels coming from **different patches** of **different images.**



Numerical experiments

The outputs of ELS can be compared with the outputs of different convolutional diffusion models, with striking similarity



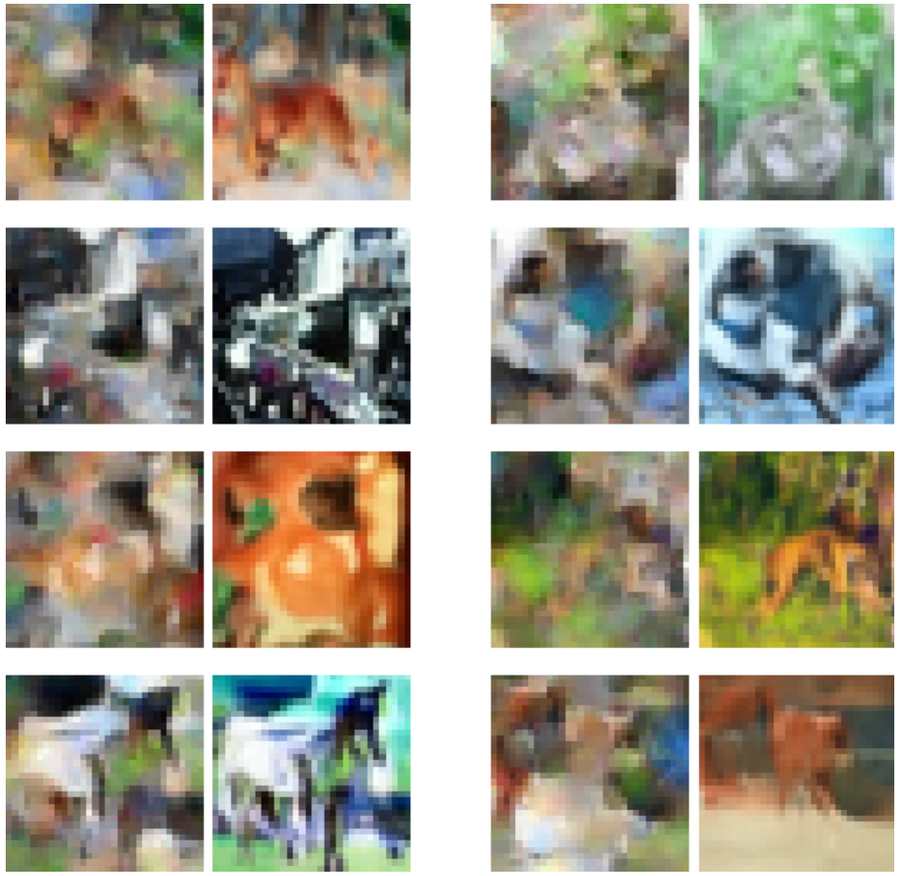


Comparison on **MNIST**, **CIFAR10** and **FashionMNIST** for **ResNet** and **UNet**

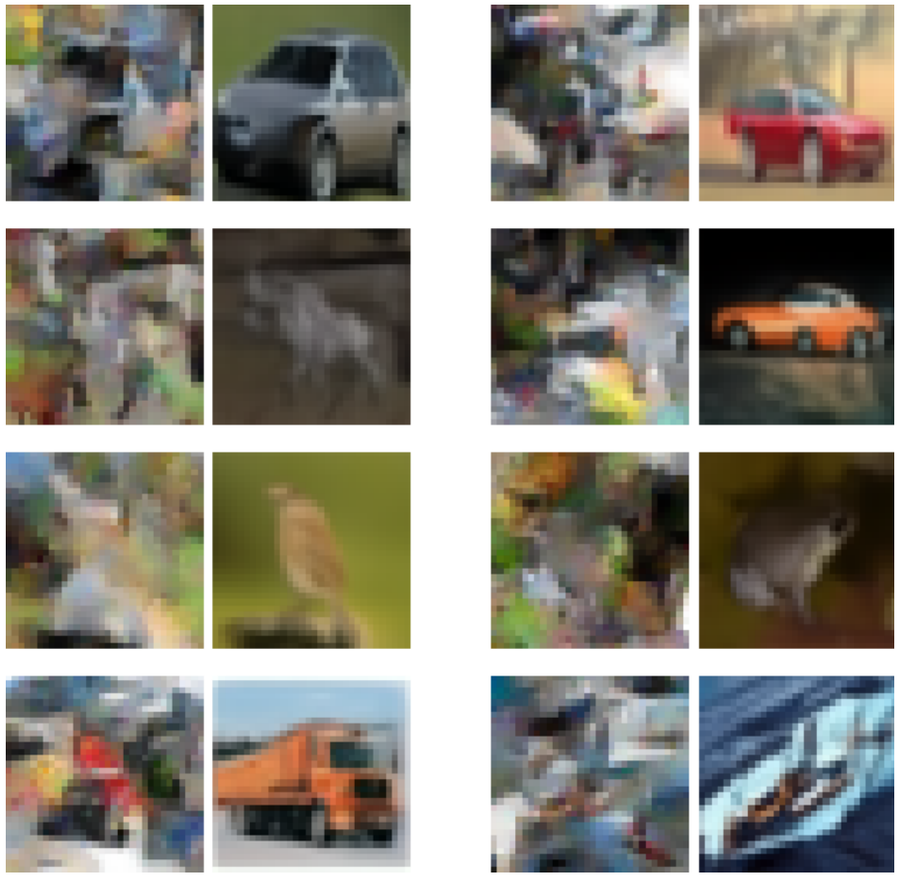
Numerical experiments

A comparison can also be made with architectures that use the **attention mechanism** (which is not taken into account in the theory). Interestingly, incoherent outputs show similarities with the model

Incoherent outputs



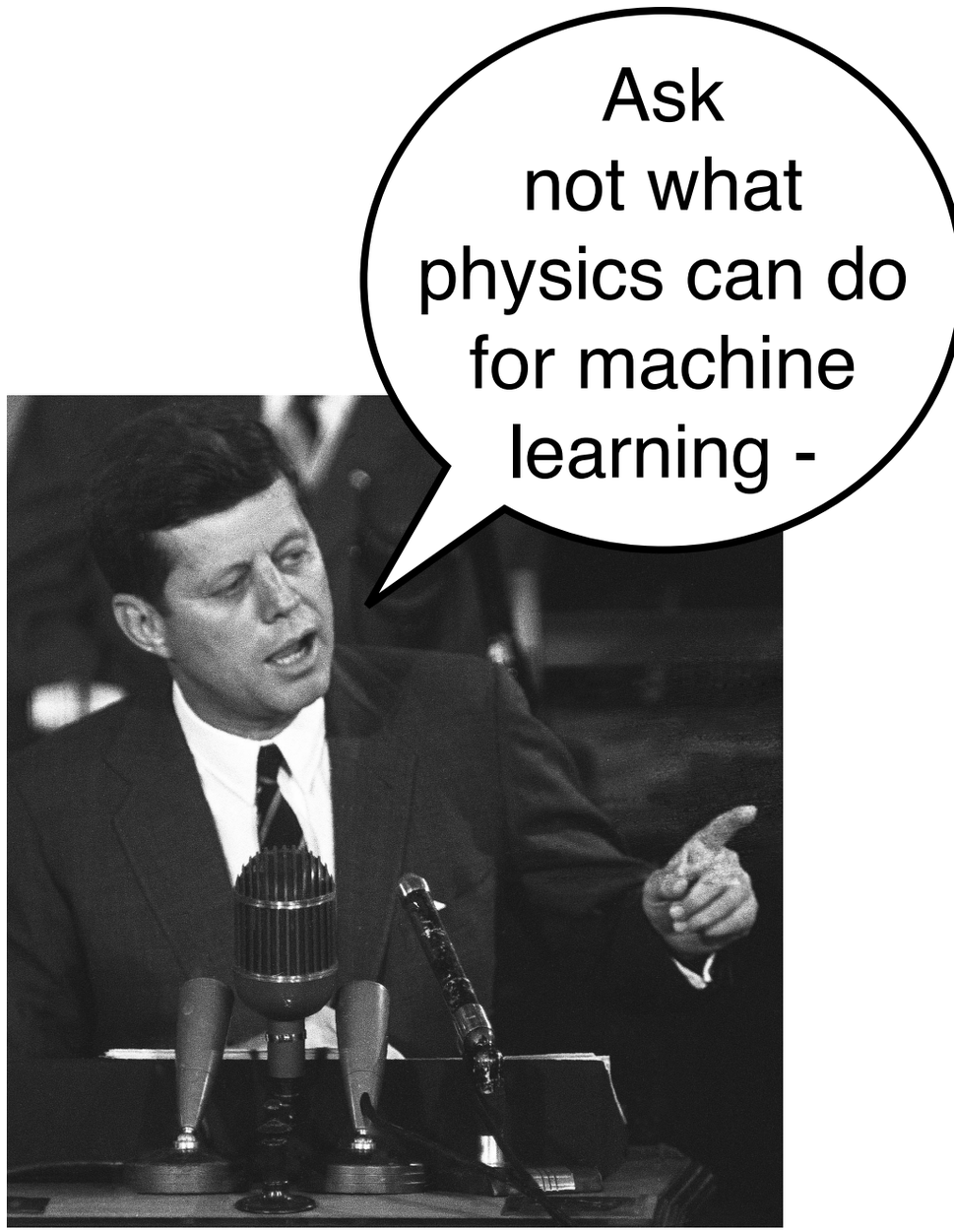
Coherent outputs



Comparison on **CIFAR10** between the theoretical model and a UNet with the attention mechanism

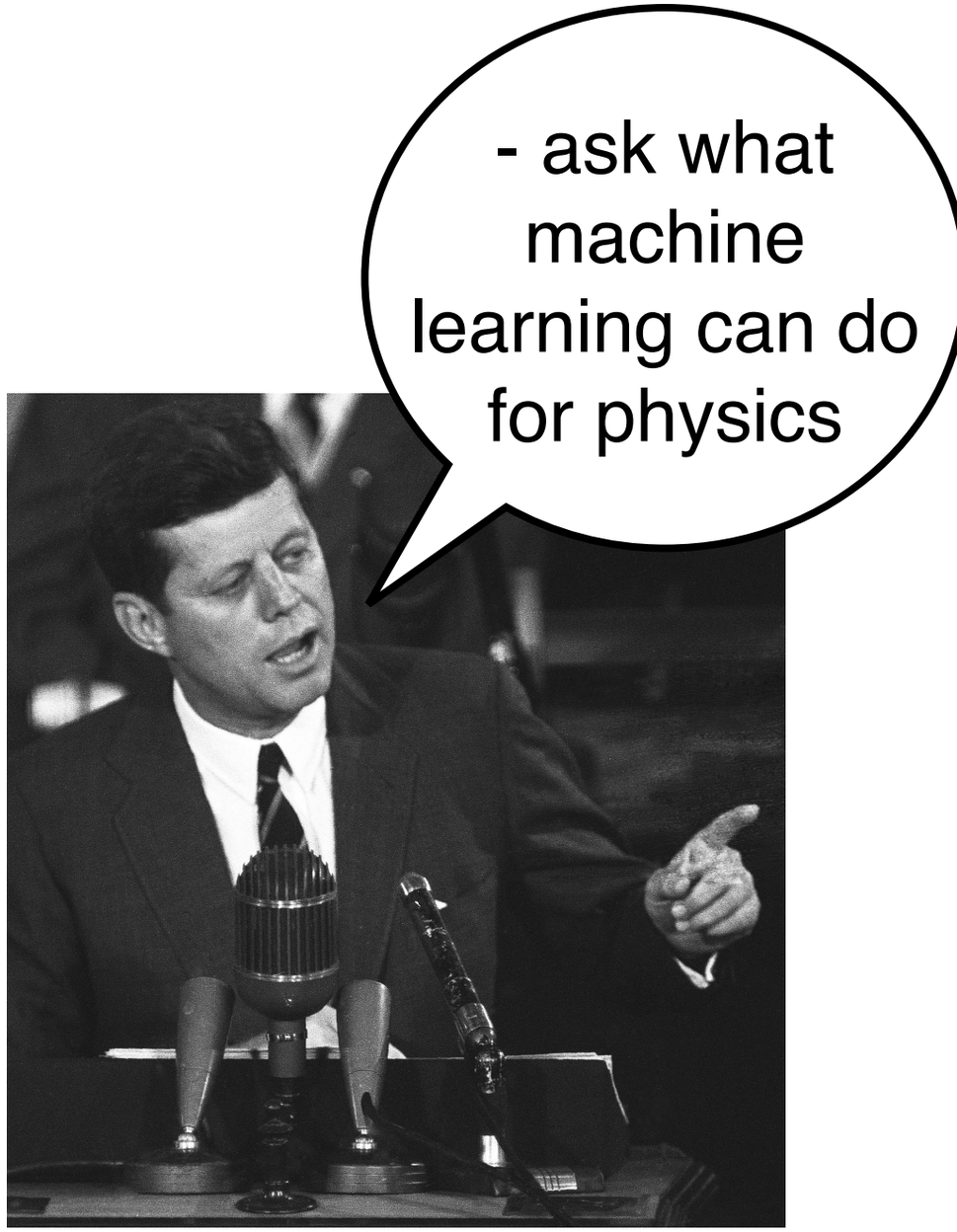
Where to go from here?

- Diffusion models are deeply rooted in **physics**, and a **physical approach** can help better understand **how they work**



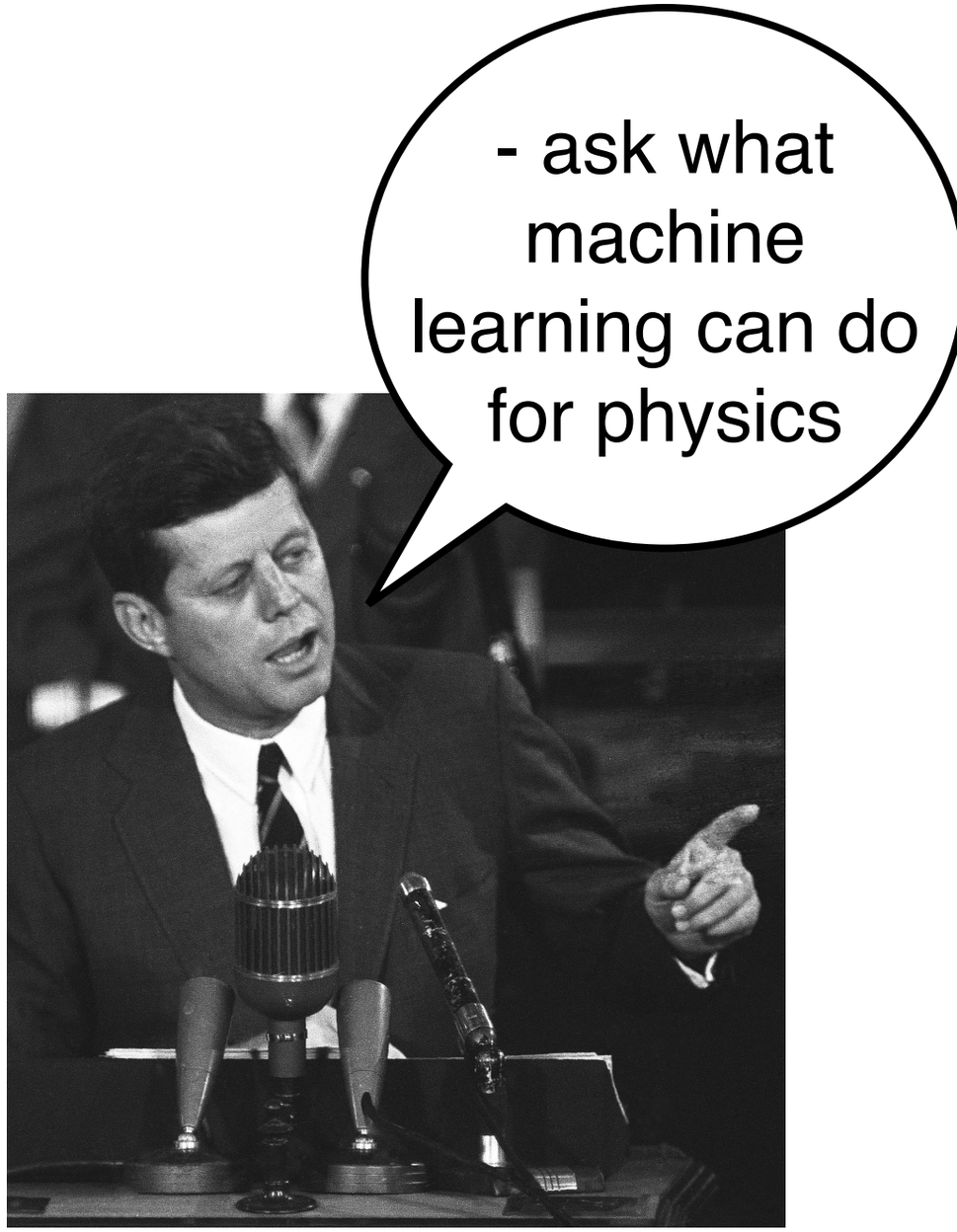
Where to go from here?

- Diffusion models are deeply rooted in **physics**, and a **physical approach** can help better understand **how they work**
- But one can also use Machine Learning to **aid research in physics**. For example, generative models can be used to study disordered systems ad setup **powered-up Monte Carlo methods** (my PhD!)



Where to go from here?

- Diffusion models are deeply rooted in **physics**, and a **physical approach** can help better understand **how they work**
- But one can also use Machine Learning to **aid research in physics**. For example, generative models can be used to study disordered systems ad setup **powered-up Monte Carlo methods** (my PhD!)
- There will be a strong **cross-fertilization** between **statistical mechanics** (a physics in general) and **machine learning**



Thank you for your attention!

“Ma se in vece fossimo riusciti ad annoiarvi, credete che non s’è fatto apposta”

“But if instead we have succeeded in boring you, believe that it was not done on purpose”

- Alessandro Manzoni