

1. Introduction to Generative Modelling

M. Ravasi

AI Summer School @ KAUST

A short intro about myself

Assistant Professor of Applied Geophysics, ErSE, KAUST

Head of **Deep Imaging Group**

Co-PI of **DeepWave** consortium

Member of **Extreme Computing Research Center**



Research Geophysicist – Statoil/Equinor
Intern – Schlumberger, WesternGeco



Phd, Geophysics – University of Edinburgh



Inverse Problems, Seismic Imaging,
Machine Learning, HPC, Open Software

A short intro about my group



Deep imaging group.

→ <https://dig.kaust.edu.sa>

→ <http://github.com/dig-kaust>

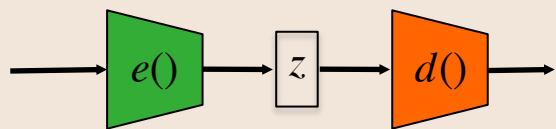
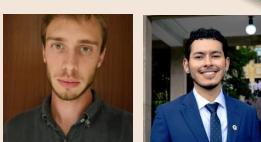
*Deepen our understanding of the subsurface
(possibly with the help of Deep Learning)*

Applications in oil & gas, CCUS, geothermal, near surface

$$-g^-(x_R, x_{VS}, \omega) = f^-(x_R, x_{VS}, \omega) + \int_S R(x_R, x_s, \omega) f^+(x_s, x_{VS}, \omega) dx_s$$

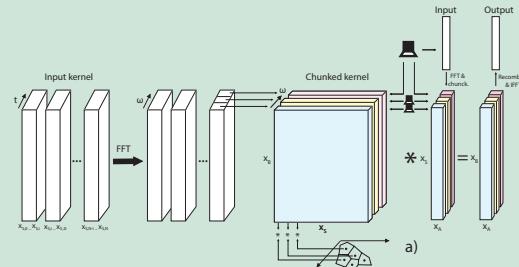
Wave theory and inverse problems

- Marchenko-based processing and imaging
- Target-oriented full-wavefield inversion
- Proximal solvers for seismic inversion problems



ML-aided inverse problems

- Physics-Informed NN for geophysical applications
- Self-supervised learning
- Deep-learning assisted UQ



HPC for geoscience

- Open-source software tools for large-scale inversion
- Seismic processing with SVD-based compression and mixed precision



PyL $\left[\begin{smallmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{smallmatrix}\right]$ ps

A Linear-Operator
Python Library



CuPy

CPU/GPU backends

Linear
Operators
(CPU/GPU)

Solvers

HTracker
PyMarchenko

PyProximal

SPGL1

PyL $\left[\begin{smallmatrix} \star \\ \vdots \end{smallmatrix}\right]$ ps

Distributed operators

PyL $\left[\begin{smallmatrix} GPU \\ \star \end{smallmatrix}\right]$ ps

Operators on GPUs +
AutoGrad of PyLops operators


DASK

Powering distributed
computing in Python



PyTorch

Tensors on GPUs +
AutoGrad (ML-oriented)

Generative Modelling: course material

All slides and codes available at:

<https://github.com/DIG-Kaust/GenModelling>

Generative Modelling: examples

- Can you imagine a **dog driving a car**?
- Can you imagine a **story of a young boy travelling to the moon**?
- Can you imagine a **soundtrack of a videogame**?
- Can you imagine a **panda surfing in the sea**?

*As humans, we have **great imagination** -> and sometimes can put it into practice...*

Generative Modelling: examples

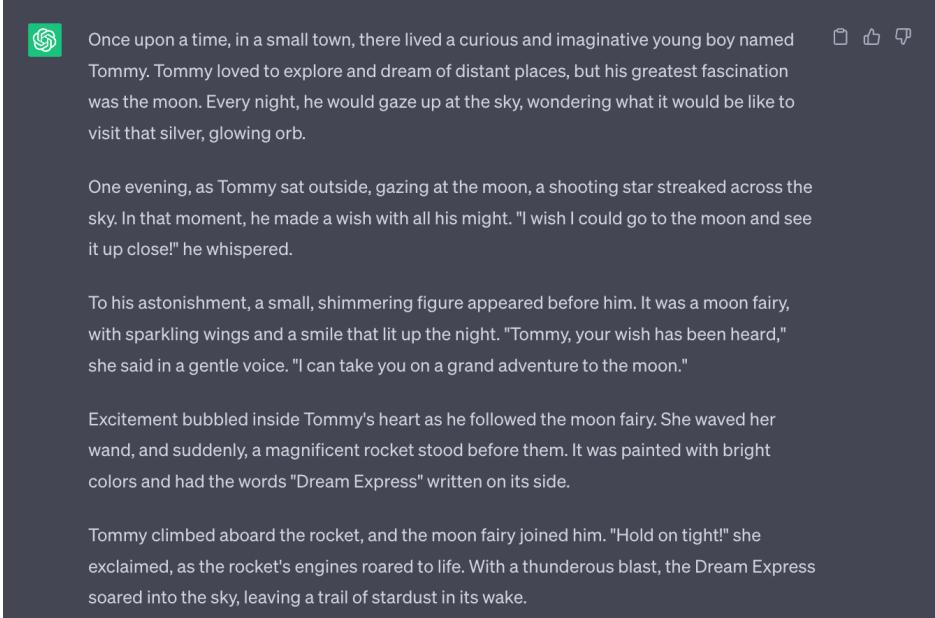
- Can you imagine a **dog driving a car?**



Adobe Firefly (Beta)

Generative Modelling: examples

- Can you imagine a **story of a young boy travelling to the moon?**



Once upon a time, in a small town, there lived a curious and imaginative young boy named Tommy. Tommy loved to explore and dream of distant places, but his greatest fascination was the moon. Every night, he would gaze up at the sky, wondering what it would be like to visit that silver, glowing orb.

One evening, as Tommy sat outside, gazing at the moon, a shooting star streaked across the sky. In that moment, he made a wish with all his might. "I wish I could go to the moon and see it up close!" he whispered.

To his astonishment, a small, shimmering figure appeared before him. It was a moon fairy, with sparkling wings and a smile that lit up the night. "Tommy, your wish has been heard," she said in a gentle voice. "I can take you on a grand adventure to the moon."

Excitement bubbled inside Tommy's heart as he followed the moon fairy. She waved her wand, and suddenly, a magnificent rocket stood before them. It was painted with bright colors and had the words "Dream Express" written on its side.

Tommy climbed aboard the rocket, and the moon fairy joined him. "Hold on tight!" she exclaimed, as the rocket's engines roared to life. With a thunderous blast, the Dream Express soared into the sky, leaving a trail of stardust in its wake.



Generative Modelling: examples

- Can you imagine a **soundtrack of a videogame?**

The main soundtrack of an arcade game. It is fast-paced and upbeat, with a catchy electric guitar riff. The music is repetitive and easy to remember, but with unexpected sounds, like cymbal crashes or drum rolls.



Generative Modelling: examples

- Can you imagine a **panda surfing in the sea?**



Generative Modelling: what and how?

*Generative modelling is a subset of artificial intelligence that aims to **create new and original content** (images, text, music, videos, etc.) **using machine learning tools**.*

Generative Modelling: what and how?

*Generative modelling is a subset of artificial intelligence that aims to **create new and original content** (images, text, music, videos, etc.) **using machine learning tools**.*

- Some training data: $\mathbb{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N_s)}\}$
- A model with learnable parameters (which learns the data): f_θ

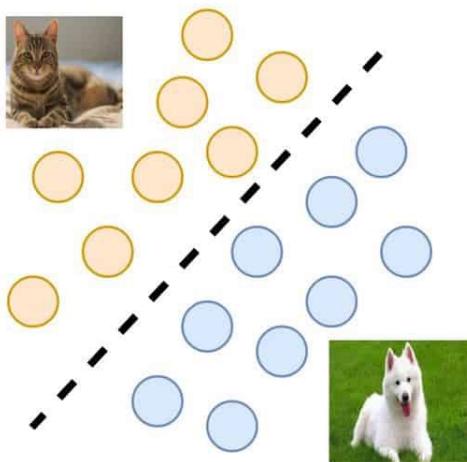
Generative Modelling: what and how?

*Generative modelling is a subset of artificial intelligence that aims to **create new and original content** (images, text, music, videos, etc.) **using machine learning tools**.*

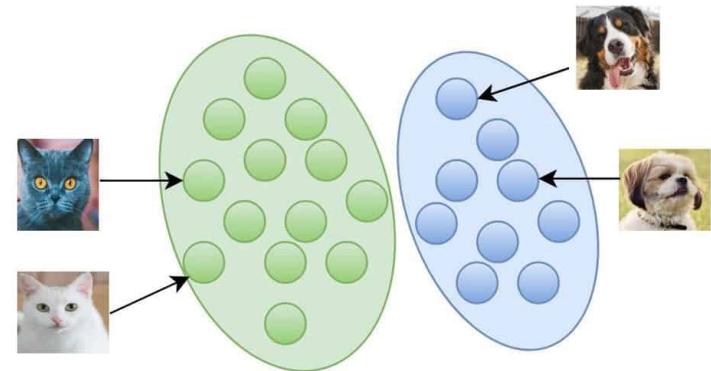
- Some training data: $\mathbb{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N_s)}\}$
- A model with learnable parameters (which learns the data): f_θ
- A procedure for the model to learn: $\mathbf{x}^{(i)} \rightarrow f_\theta$
- A procedure to generate data (or evaluate prob): $f_\theta \rightarrow \mathbf{x}^{(new)}/p(\mathbf{x}^{(new)})$

Generative Modelling: what's new?

Two types of Machine Learning models:



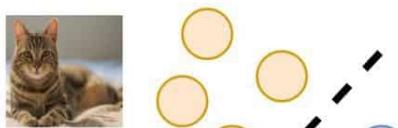
Discriminative



Generative

Generative Modelling: what's new?

Two types of Machine Learning models:



$p(y|\mathbf{x})$ from $(\mathbf{x}^{(i)}, y^{(i)})$



Discriminative

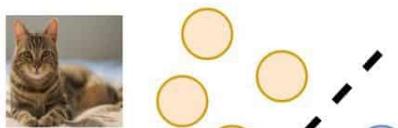
$p(\mathbf{x})$ from $\mathbf{x}^{(i)}$



Generative

Generative Modelling: what's new?

Two types of Machine Learning models:



$p(y|\mathbf{x})$ from $(\mathbf{x}^{(i)}, y^{(i)})$



Discriminative



$p(\mathbf{x})$ from $\mathbf{x}^{(i)}$
or $p(\mathbf{x}|y)$ from $(\mathbf{x}^{(i)}, y^{(i)})$

Generative

Generative Modelling: what's new?

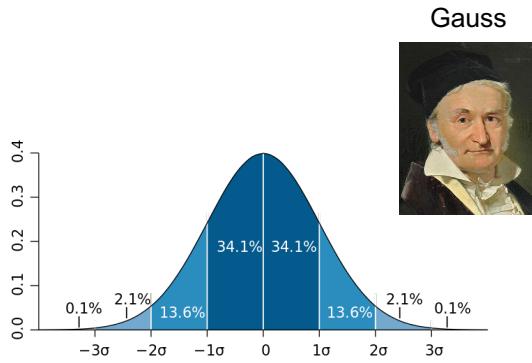
Ideally a generative modelling is a model solving the following problem

$$\operatorname{argmax}_{\theta} p(\mathbf{x}) \text{ from } \mathbf{x}^{(i)}$$

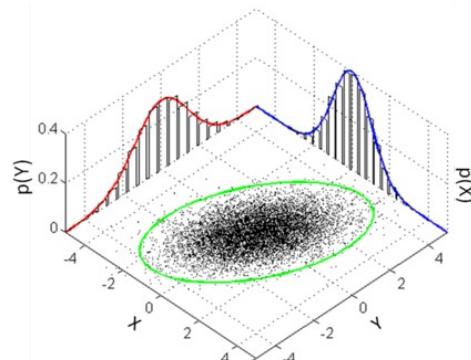
Maximum-likelihood estimator

We will see that this is not easy to do, usually approximations are required.

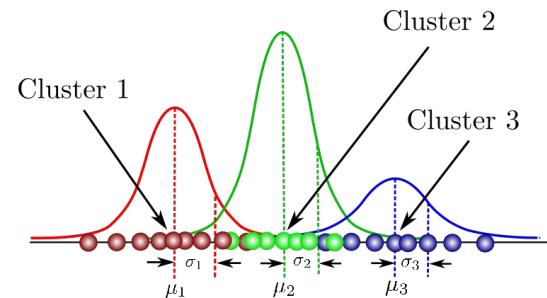
History of Generative Modelling



1809: Gaussian distribution

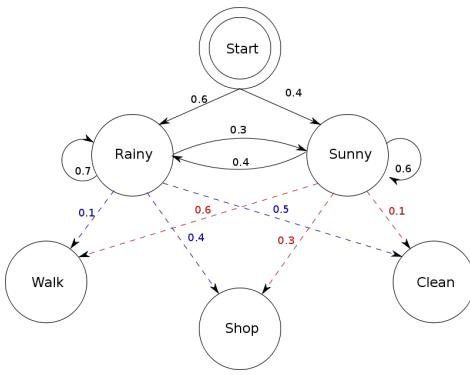


Multi-variate Gaussian distribution

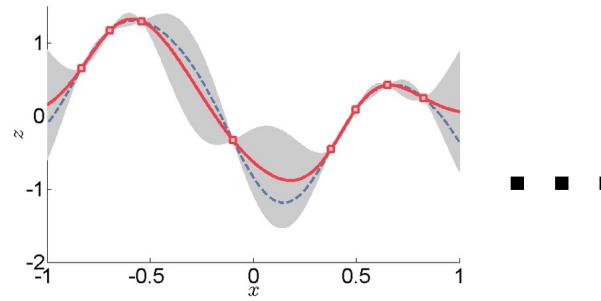


50'/60': Gaussian Mixture Models (GGMs)

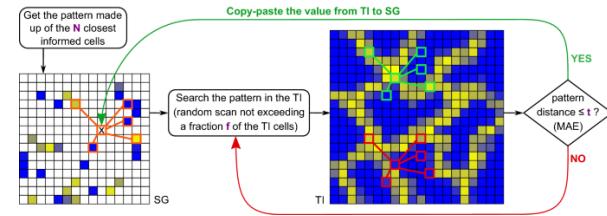
History of Generative Modelling



60': Hidden Markov Models (HMMs)

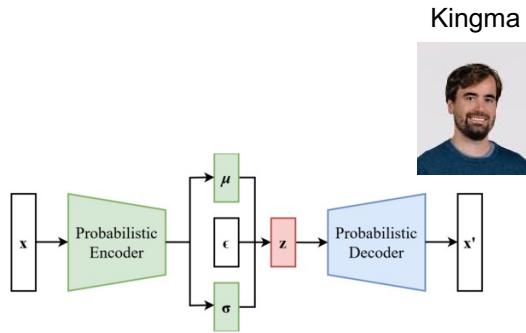


70': Kriging – later popularized as Gaussian Process regression (GP)

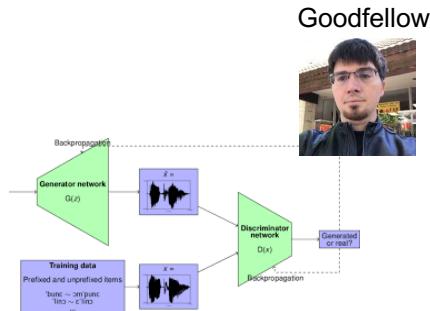


90': Geostatistics – two-point and multiple-point statistics

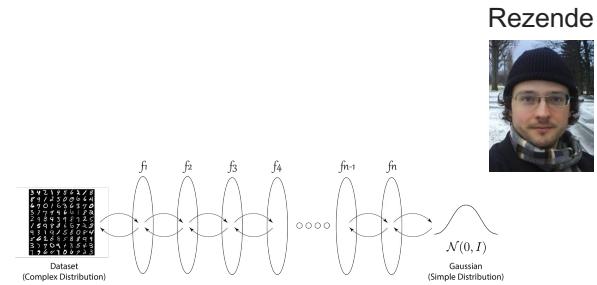
History of Generative Modelling



2014: Variational
AutoEncoders



2014: Generative
Adversarial Networks
(GANs)

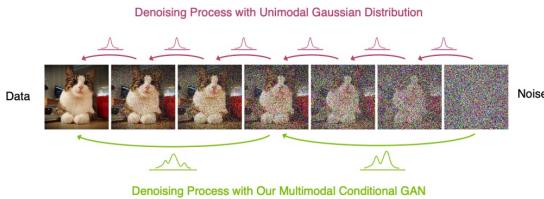


2015: Normalizing flows

....

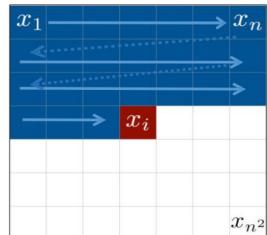
History of Generative Modelling

Sohl-Dickstein



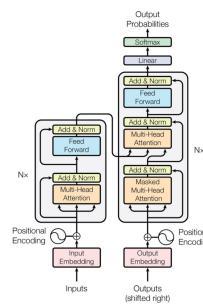
2015: Diffusion models

Aaron van den Oord



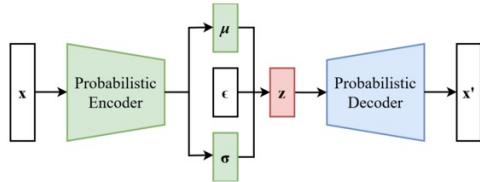
2015: Autoregressive (image) models

Bahdanau

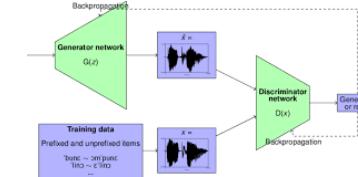


2015/7: Attention mechanism and Transformers

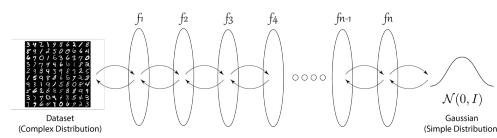
Generative Modelling in this week



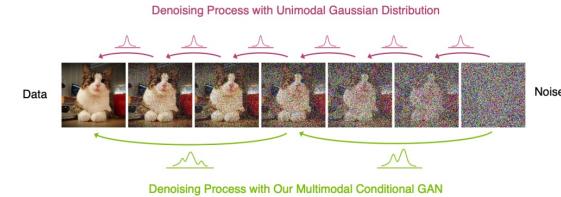
2014: Variational
AutoEncoders



2014: Generative Adversarial
Networks (GANs)

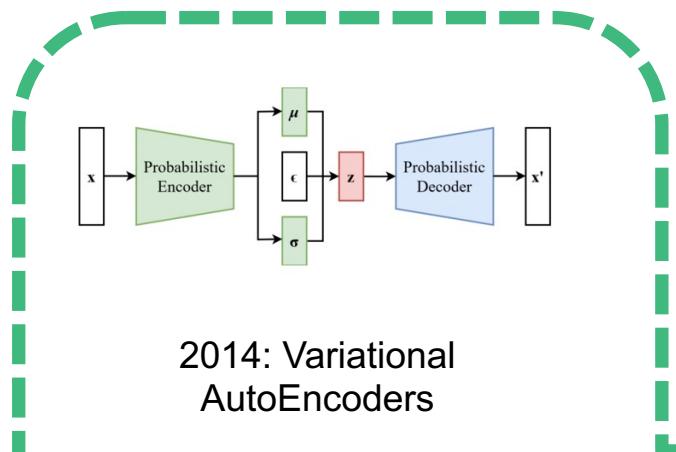


2015: Normalizing flows

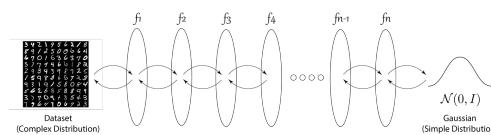


2015: Diffusion models

Generative Modelling in this week

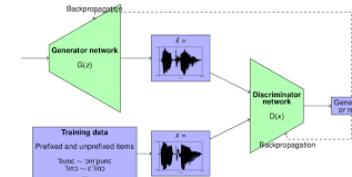


2014: Variational
AutoEncoders

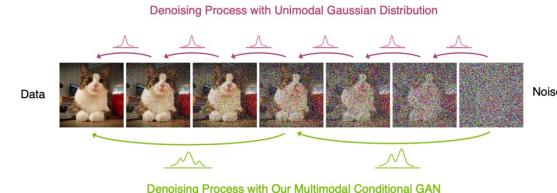


2015: Normalizing flows

Implicit models: distribution is implicitly represented by a model of its sampling process



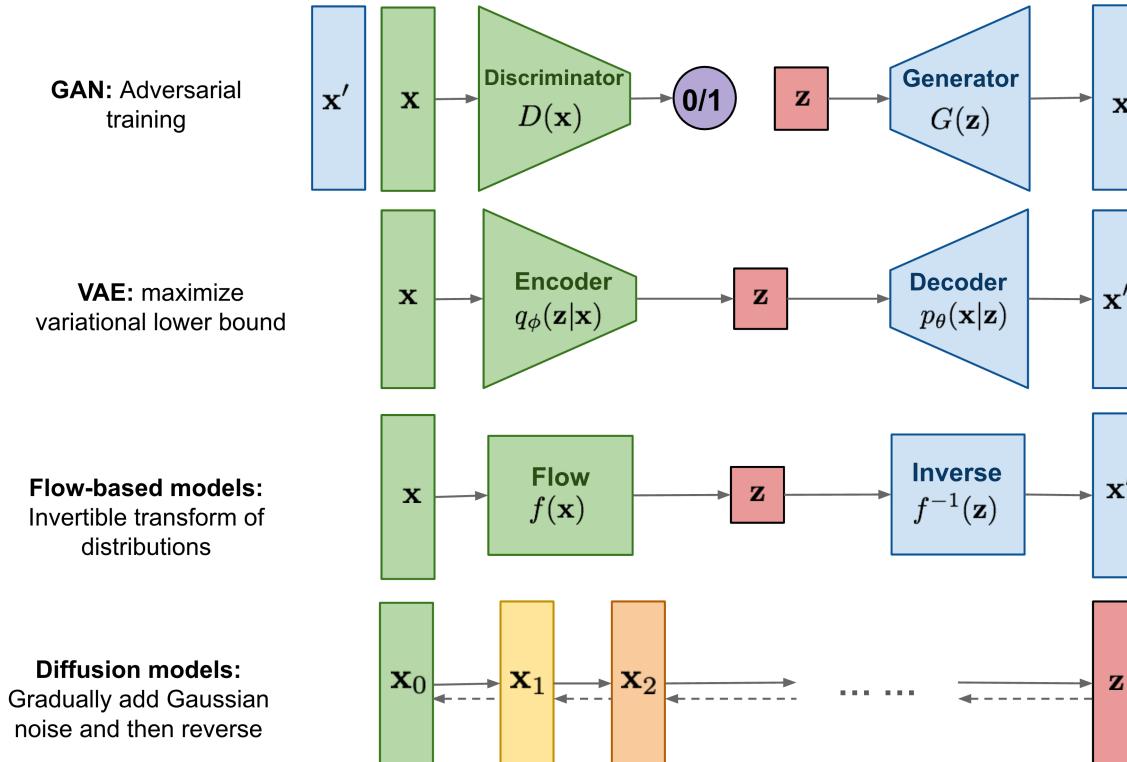
2014: Generative Adversarial
Networks (GANs)



2015: Diffusion models

Likelihood-based models: learn $p(x)$ via (approx.) Max-Likelihood

Generative Modelling in this week



From <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>

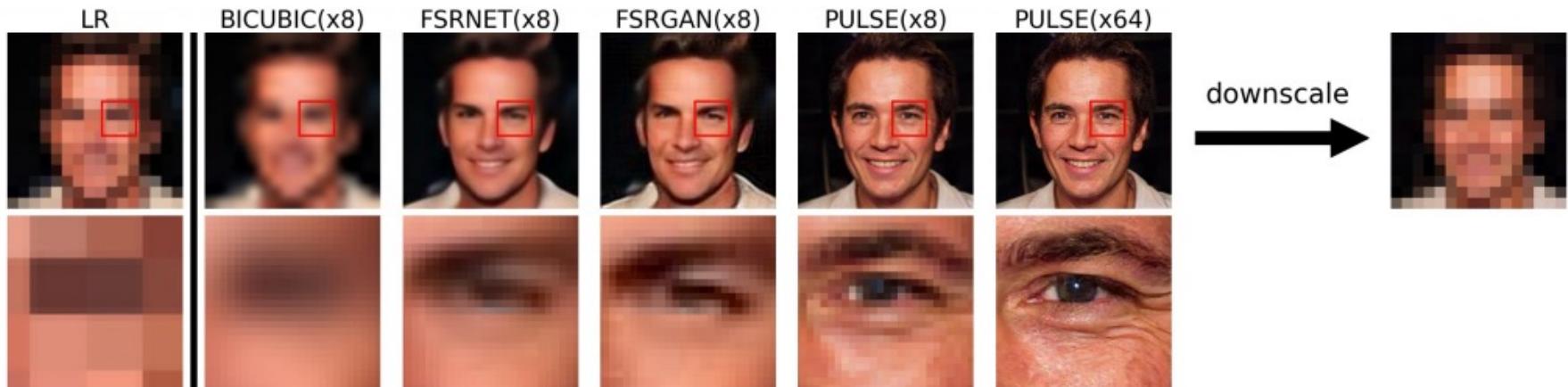
Success stories of GM: Face generation



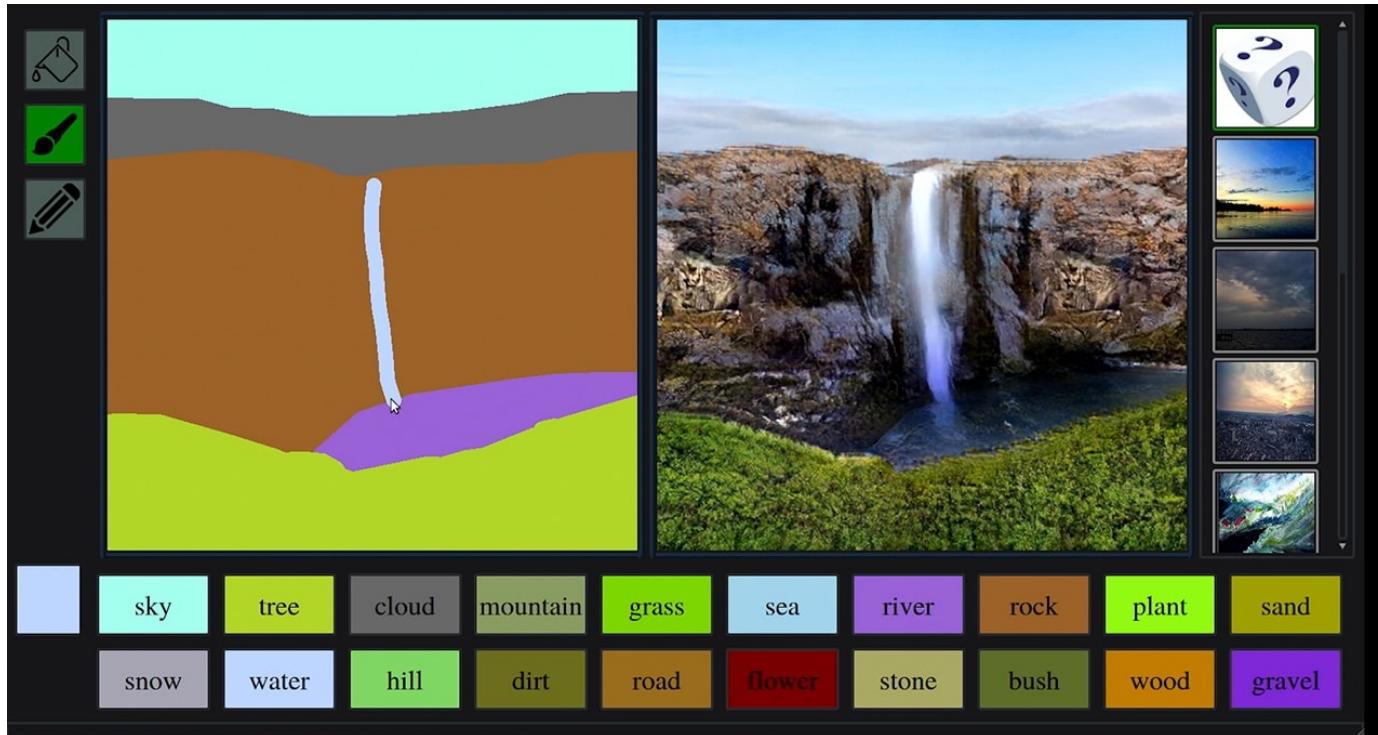
Success stories of GM: Impainting



Success stories of GM: Super-resolution



Success stories of GM: Sketch-to-image



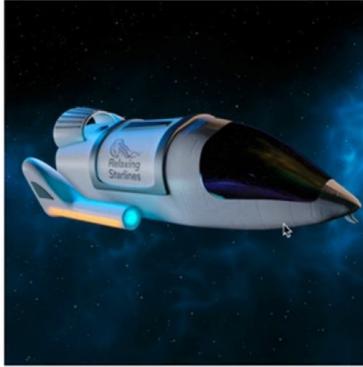
Success stories of GM: Text generation

MA Write a short description of the AI Summer School to be held in KAUST in June 

 Join us at the AI Summer School, hosted by KAUST (King Abdullah University of Science and Technology), a renowned research university located on the shores of the Red Sea in Saudi Arabia. Taking place in June, this immersive educational event offers a unique opportunity to dive deep into the world of Artificial Intelligence.

Designed for students, researchers, and professionals interested in AI, the AI Summer School at KAUST brings together experts, industry leaders, and passionate individuals to explore the latest advancements, applications, and challenges in the field of AI. Whether you're a beginner or an experienced practitioner, this program caters to a wide range of backgrounds and skill levels.

Success stories of GM: Code generation



```
/* Add this image of a rocketship:  
https://i1.sndcdn.com/artworks-j8xjG7zc1wmTe07b-06183w-t500x500.jpg */  
var rocketship =  
document.createElement('img');  
rocketship.src =  
'https://i1.sndcdn.com/artworks-j8xjG7zc1wmTe07b-06183w-t500x500.jpg';  
document.body.appendChild(rock  
etship);
```

Add this image of a rocketship:

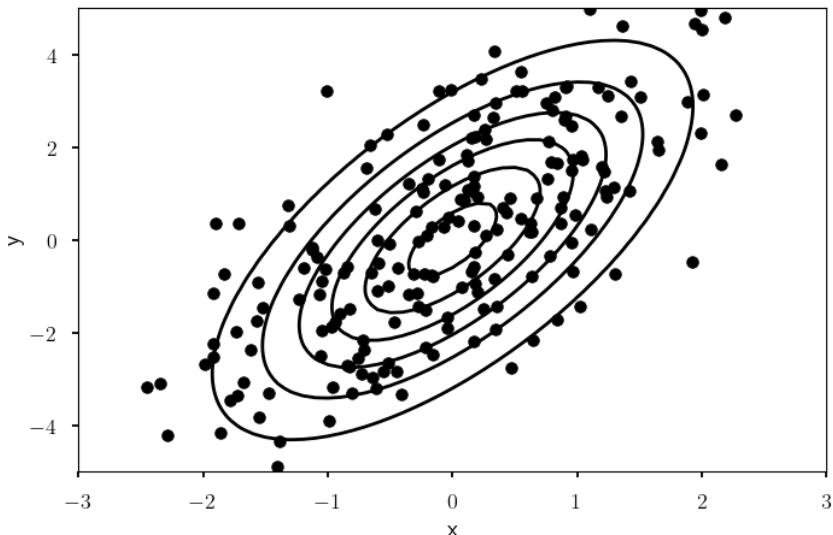
<https://i1.sndcdn.com/artworks-j8xjG7zc1wmTe07b-06183w-t500x500.jpg>

→

🔗

Generative Modelling: Gauss's legacy

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{k/2} \det(\Sigma)^{1/2}} e^{-\frac{1}{2} (\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

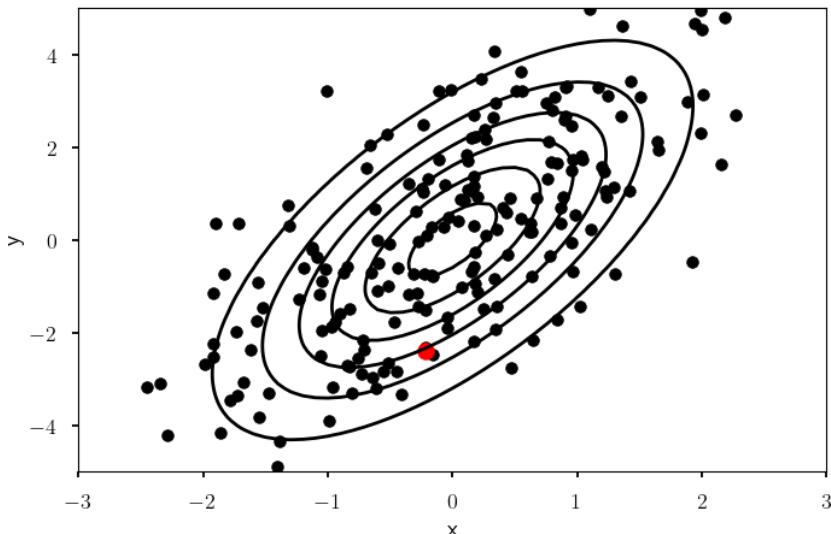


$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 5 & 1.5 \\ 1.5 & 1 \end{bmatrix}$$

$$\theta = \{\boldsymbol{\mu}, \Sigma\}$$

Generative Modelling: Gauss's legacy

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{k/2} \det(\Sigma)^{1/2}} e^{-\frac{1}{2} (\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$



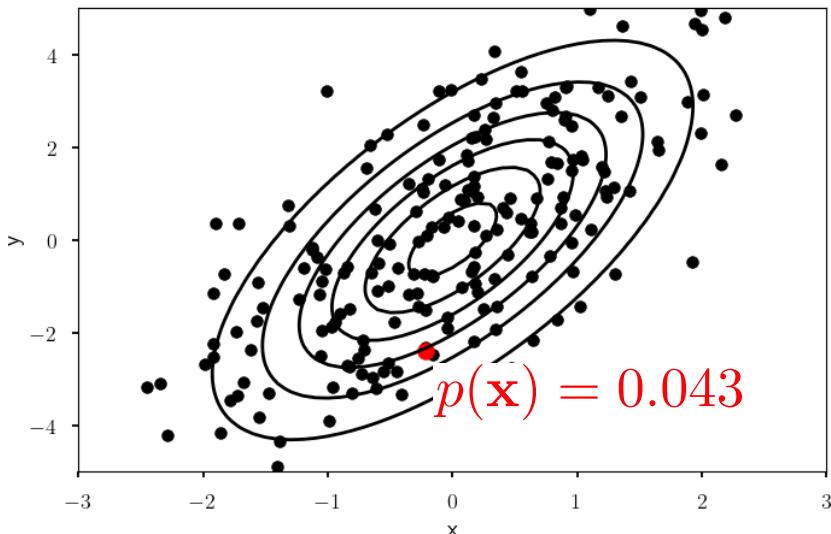
Sample distribution:

Being able to draw a sample from the distribution (red dot)

If we sample many points (e.g., 1000), we can then estimate sample statistics - mean, cov

Generative Modelling: Gauss's legacy

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{k/2} \det(\boldsymbol{\Sigma})^{1/2}} e^{-\frac{1}{2} (\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$



Evaluate distribution:

Being able to insert \mathbf{x} in the probability density function and get the probability of occurrence of that sample

Generative Modelling: Gauss's legacy

Sample a Gaussian distribution:

Training

1. Compute the sample mean and covariance from the training samples: μ, Σ
2. Apply Cholesky decomposition of the covariance matrix: $\Sigma = \mathbf{L}^T \mathbf{L}$

Inference/generation

1. Sample a vector from normal distribution: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2. Create new sample from true distribution by scaling and shifting: $\mathbf{x}^{(new)} = \mathbf{L}\mathbf{z} + \mu$

Generative Modelling: Gauss's legacy

There is a bit of Gauss in every modern generative model:

- **VAEs**: latent code distribution
- **GANs**: noise input to generator
- **Normalizing flows**: basic distribution (to be warped into $p(x)$)
- **Diffusion models**: noise added at every step