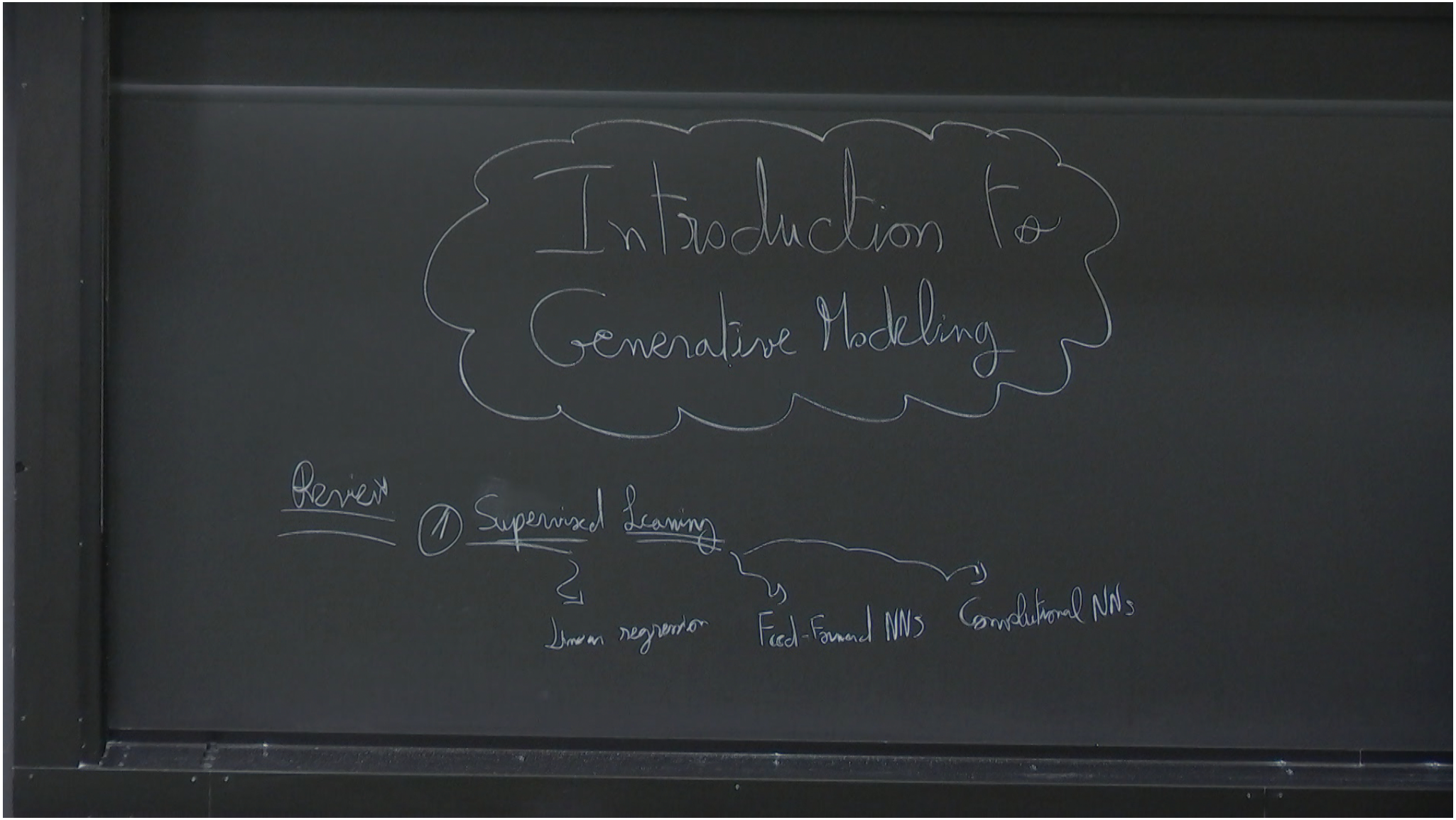
Title: Machine Learning Lecture - 230328

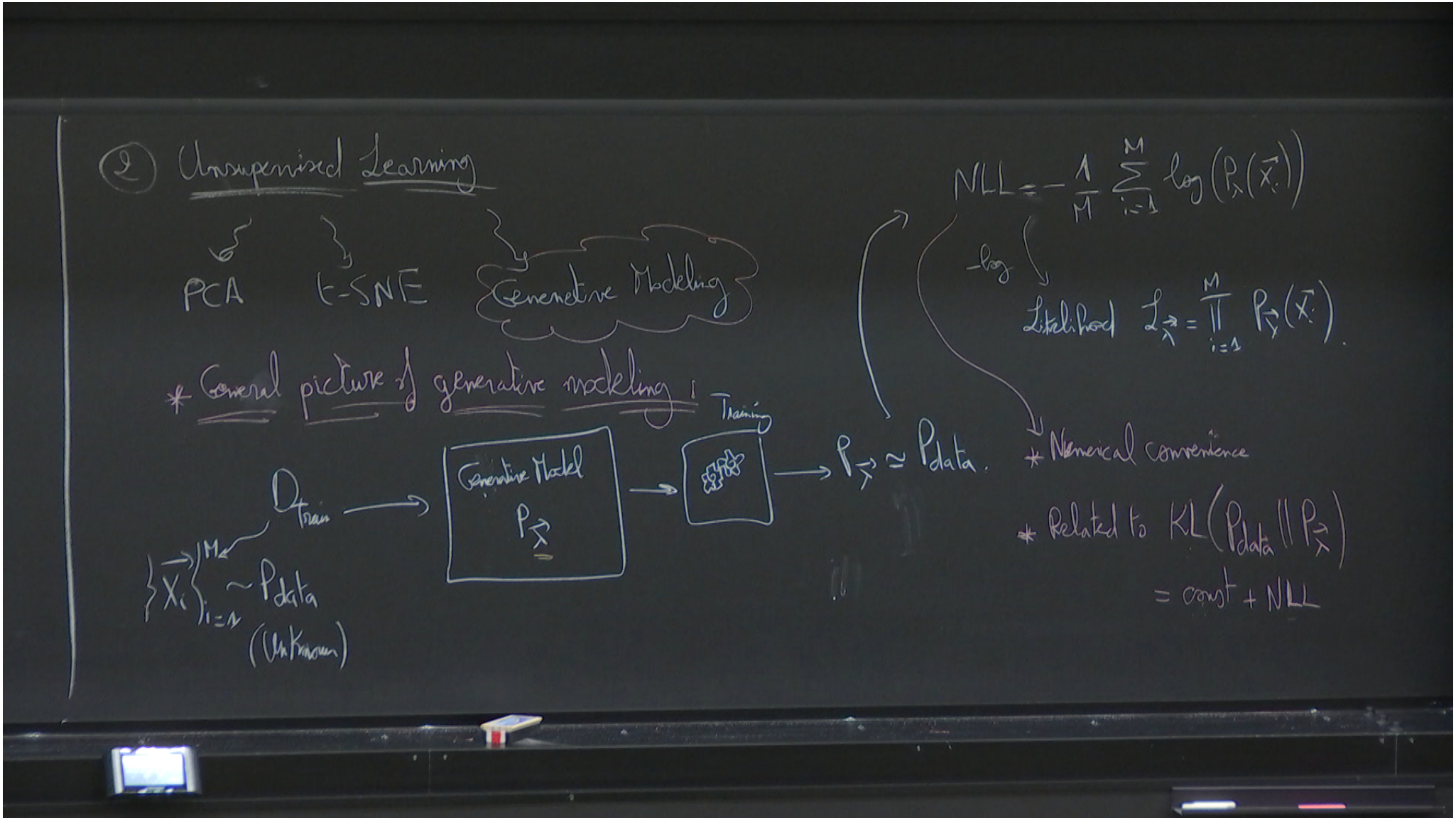
Speakers: Lauren Hayward

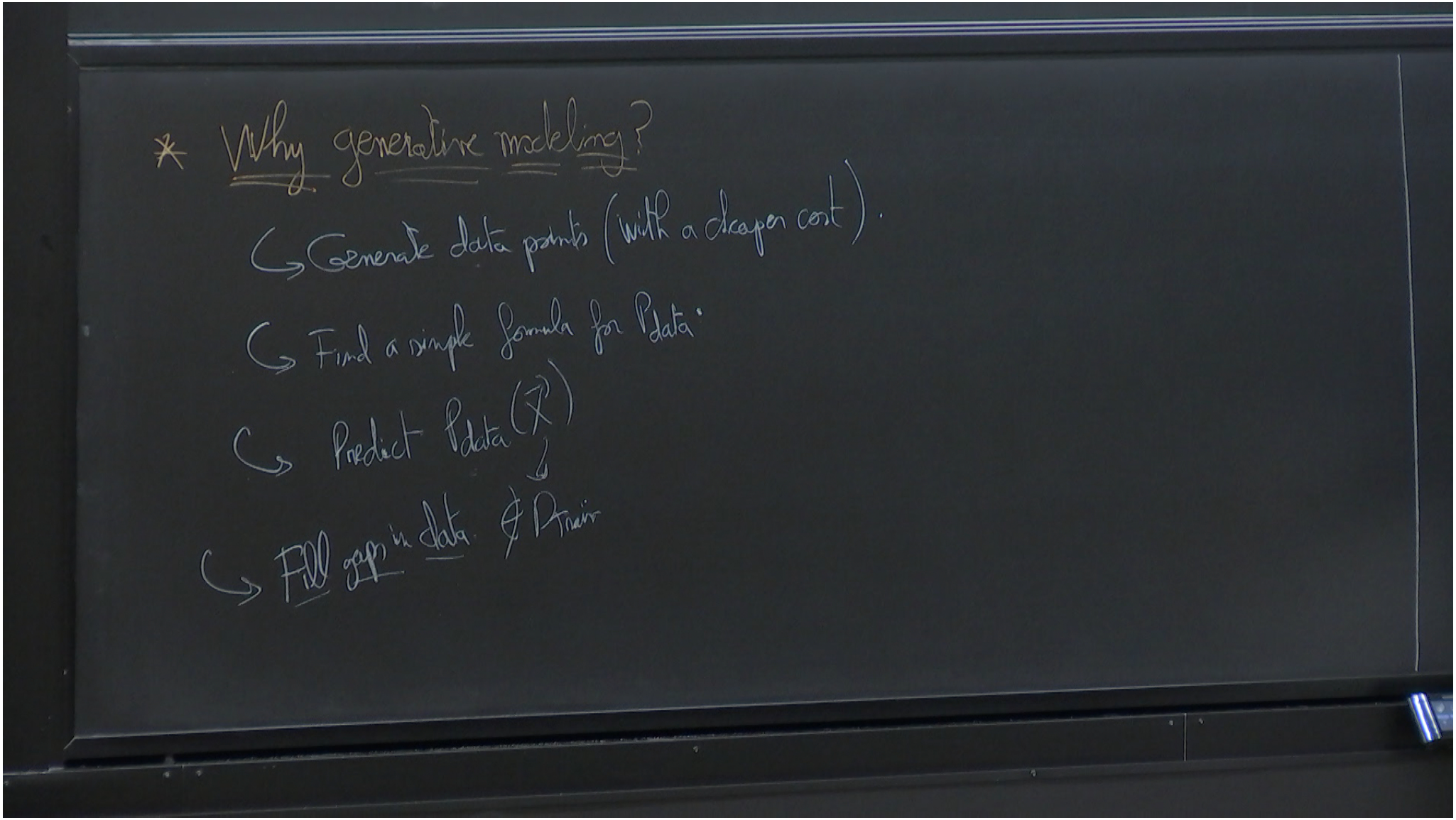
Collection: Machine Learning for Many-Body Physics (2022/2023)

Date: March 28, 2023 - 11:30 AM

URL: https://pirsa.org/23030036


Introduction to
Generative Modeling
Review
1 Supervised Learning
Linear regression
Feed-Forward NNs
Convolutional NNs

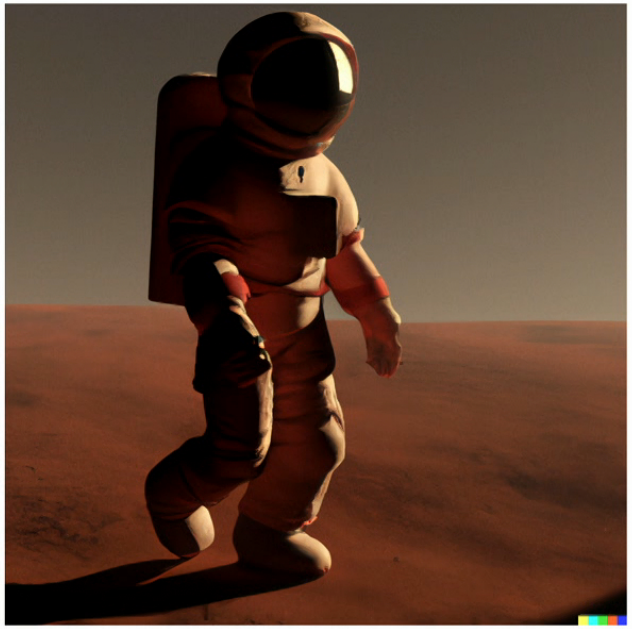

(2) Unsupervised Learning
PCA
t-SNE
Generative Modeling
* General picture of generative modeling :
Training
Generative Model
$P_{\vec{X}}$
D_{train}
$\{\vec{X}_i\}_{i=1}^{M} \sim P_{data}$
(Unknown)
$P_{\vec{X}} \simeq P_{data}$.
$NLL = -\frac{1}{M}\sum_{i=1}^{M} \log\left(P_{\vec{X}}(\vec{x}_i)\right)$
-log
Likelihood $\mathcal{L}_{\vec{X}} = \prod_{i=1}^{M} P_{\vec{X}}(\vec{x}_i)$.
* Numerical convenience
* Related to $KL(P_{data} \| P_{\vec{X}})$
$= const + NLL$


* Why generative modeling?
↳ Generate data points (with a cheaper cost).
↳ Find a simple formula for Pdata.
↳ Predict Pdata(x)
↳ Fill gaps in data ∉ Ptrain

Generating faces

Credit: this-person-does-not-exist.com

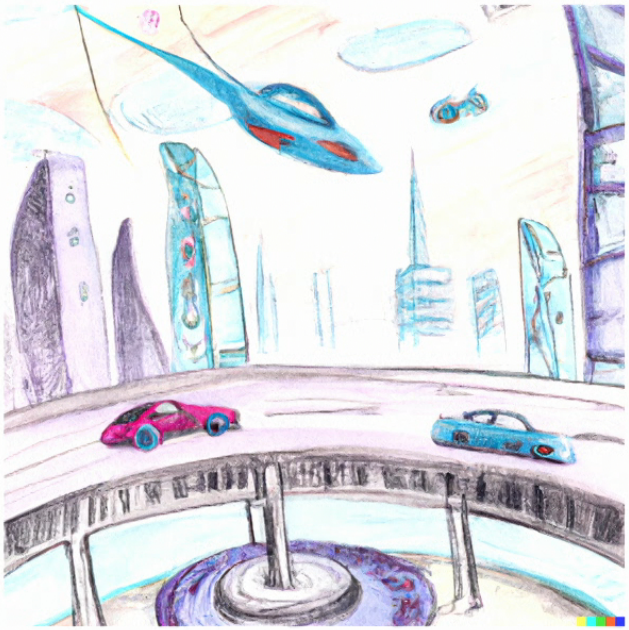
Dall-E



A 3d realistic render of an astronaut walking on Mars

https://openai.com/product/dall-e-2

Dall-E



A pencil and watercolor drawing of a bright city in the future with flying cars

Artificial Intelligence Virtual Artist

Chat-GPT

M

Explain generative modeling to five years-old kid

Generative modeling is like playing pretend with your friends. Imagine you have a box of toys, and you want to make a new toy that looks like the ones in the box. You can use your imagination to think of what the new toy might look like, and then try to make it with the materials you have.

Generative modeling is kind of like that, but with a computer instead of toys. The computer tries to imagine new things based on what it has seen before, and then creates them. It's like the computer is using its imagination to make something new!

Cosmology

CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks

Mustafa Mustafa[1*], Deborah Bard[1], Wahid Bhimji[1], Zarija Lukić[1], Rami Al-Rfou[2] and Jan M. Kratochvil[3]

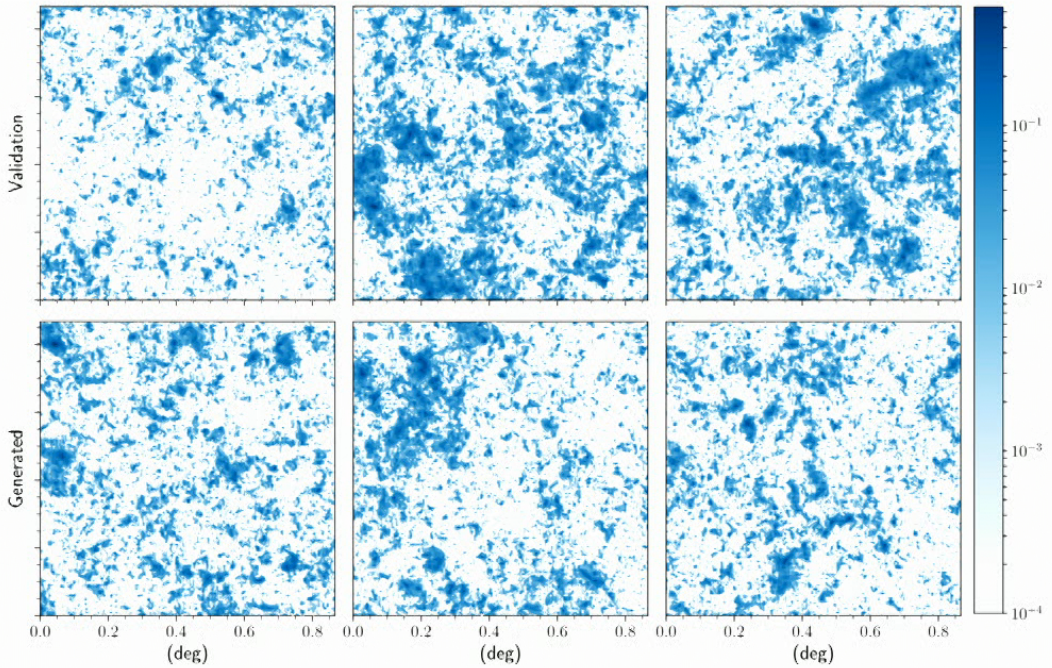


Figure 1 Weak lensing convergence maps for our ΛCDM cosmological model. Randomly selected maps from validation dataset (top) and GAN generated examples (bottom)

Quantum state tomography

Attention-based quantum tomography

Peter Cha[6,1], Paul Ginsparg[2], Felix Wu[2], Juan Carrasquilla[3,4], Peter L McMahon[5] and Eun-Ah Kim[1]

Published 23 November 2021 • © 2021 The Author(s). Published by IOP Publishing Ltd

Machine Learning: Science and Technology, Volume 3, Number 1

Citation Peter Cha *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 01LT01

DOI 10.1088/2632-2153/ac362b

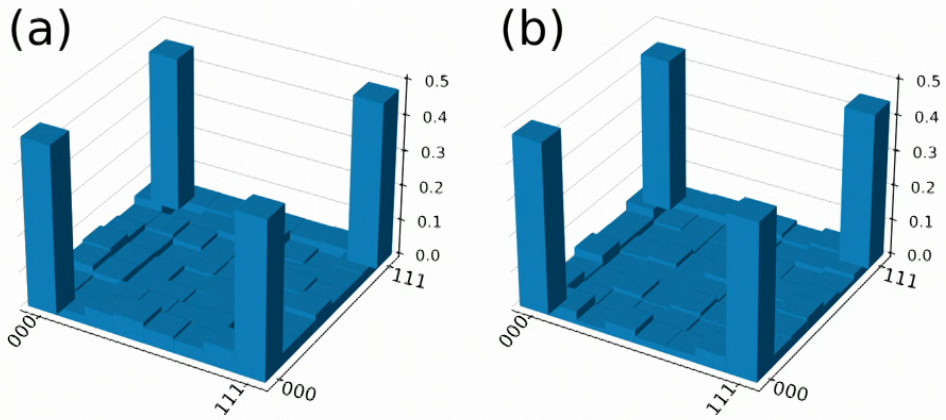
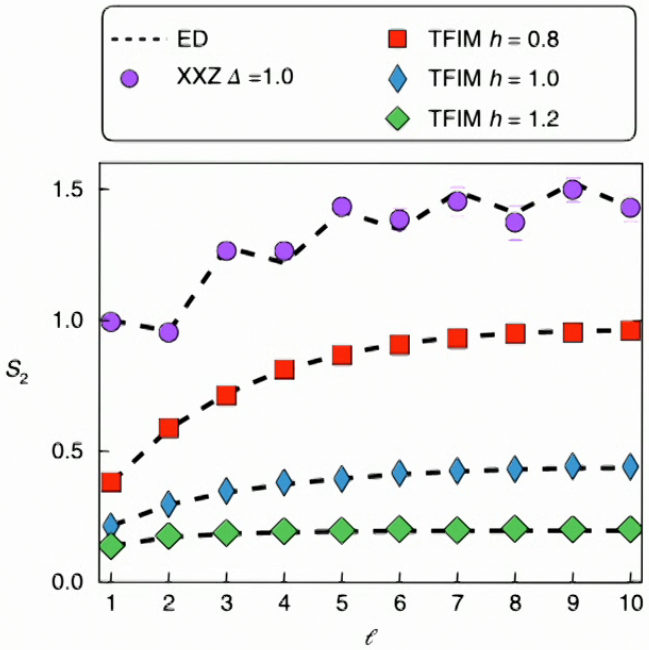


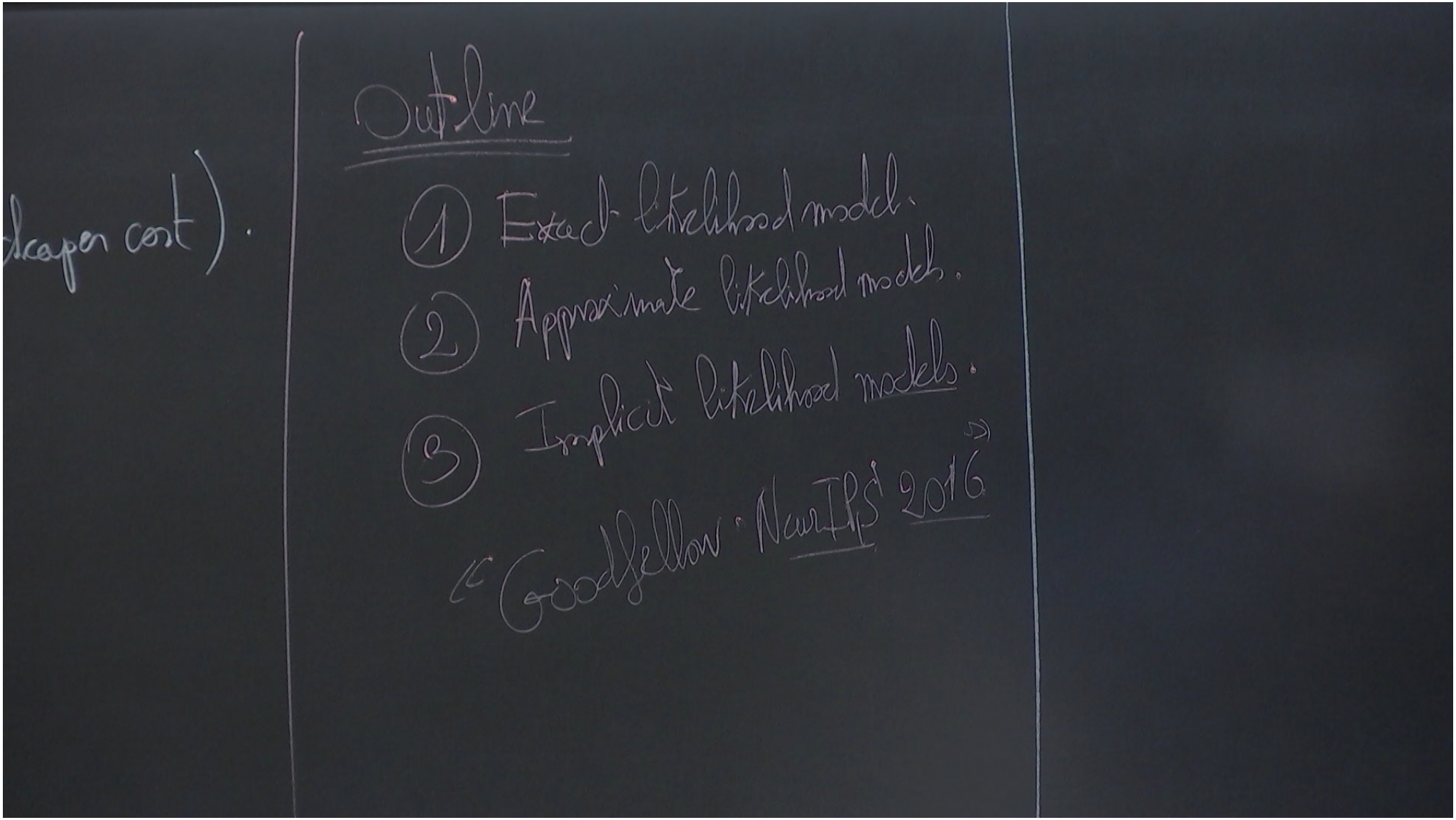
Figure 3. Benchmarking AQT (a) to MLE tomography offered by IBM's Qiskit library (b) for a noisy 3-qubit GHZ state data generated on the IBMQ_OURENSE quantum computer. Each bar represents the absolute value of a density matrix (DM) element.

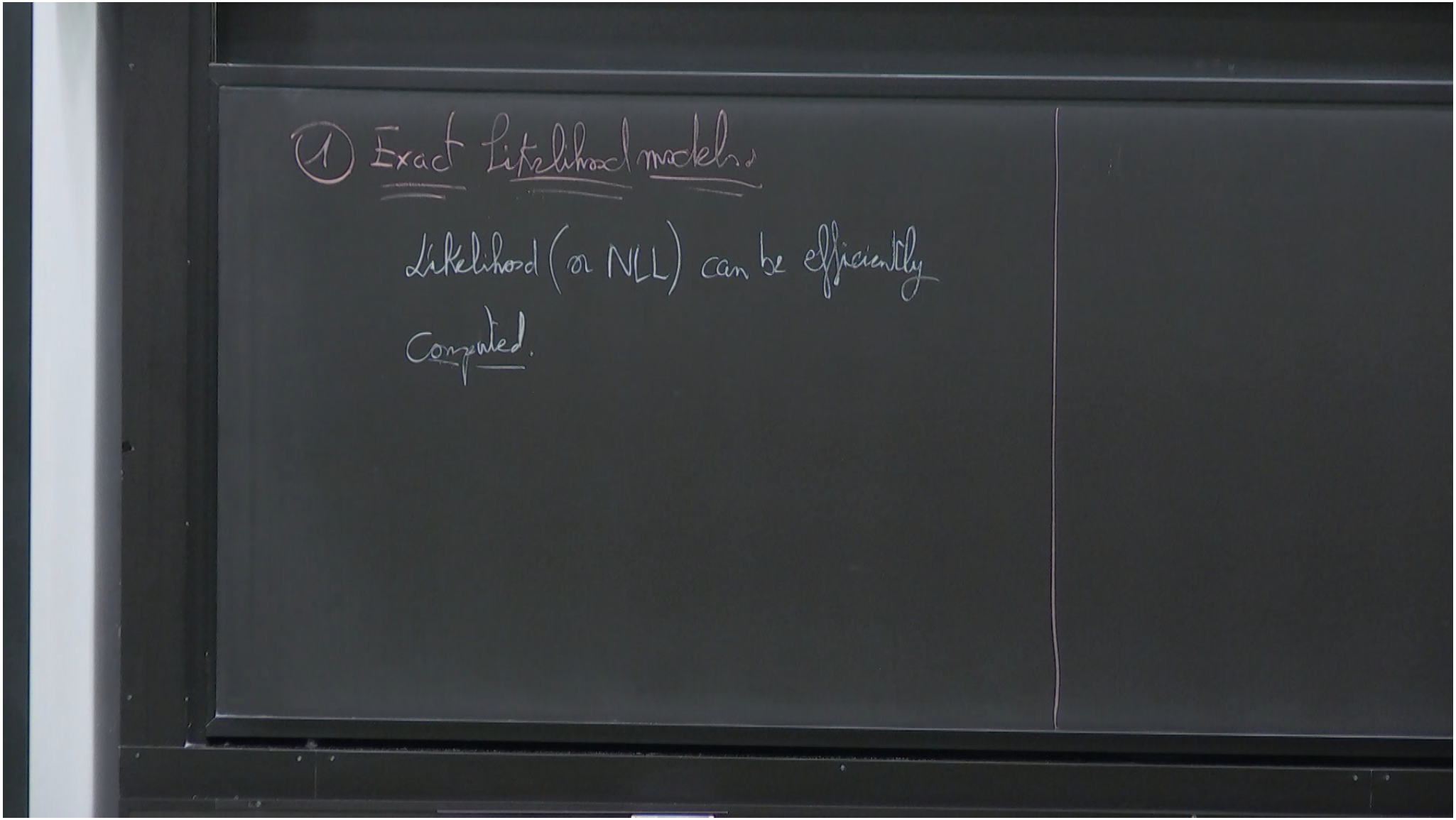
Quantum state tomography

Neural-network quantum state tomography

Giacomo Torlai[1,2], Guglielmo Mazzola[3], Juan Carrasquilla[4,5], Matthias Troyer[3,6], Roger Melko[1,2] and Giuseppe Carleo[3,7*]




Outline
1 Exact likelihood models.
2 Approximate likelihood models.
3 Implicit likelihood models.
"Goodfellow NeurIPS 2016"

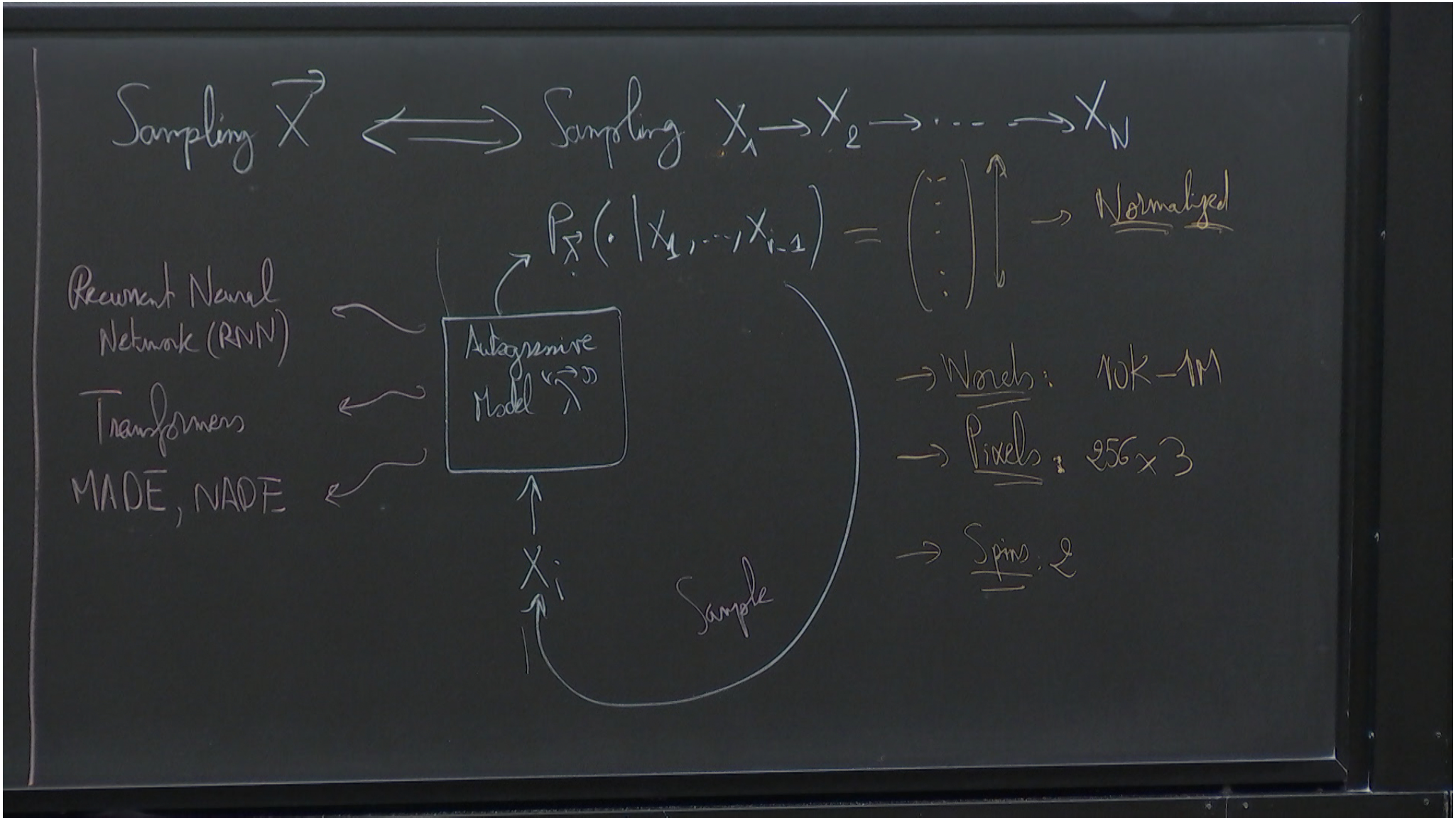

① Exact Likelihood models
Likelihood (or NLL) can be efficiently
computed.

1_1_Autoregressive models:

↳ A prediction of a new variable is based on previous variable.

$$P_{\vec{X}}(\vec{X}) = P_X(X_1) \cdot P(X_2|X_1) \cdots P(X_N|X_{N-1}, \cdots, X_1)$$

$$(X_1, X_2, \cdots, X_N)$$


Sampling $\vec{X}$ $\Longleftrightarrow$ Sampling $X_1 \to X_2 \to \cdots \to X_N$
$P_{\vec{X}}(\cdot \mid X_1, \cdots, X_{i-1})$ = (vector) → Normalized
Recurrent Neural Network (RNN)
Transformers
MADE, NADE
Autoregressive Model
X_i
Sample
→ Words: 10K–1M
→ Pixels: 256 × 3
→ Spins: 2

be efficiently

1-1- Autoregressive models:

↳ A prediction of a new variable is based on previous variable.

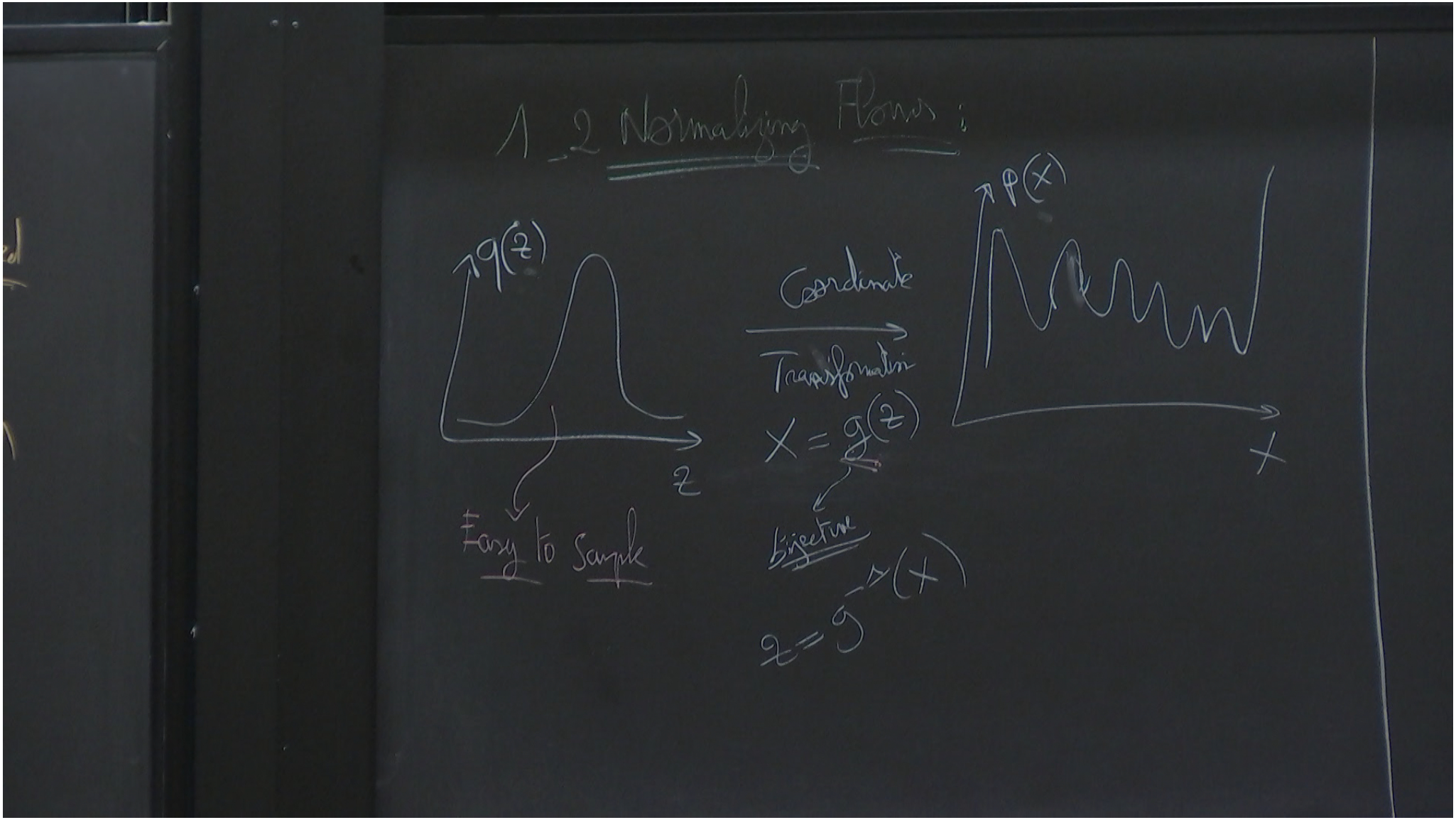
Normalized

$$P_{\vec{X}}(\vec{X}) = P_{X}(X_1) \cdot P(X_2 | X_1) \cdots P(X_N | X_{N-1}, \cdots, X_1)$$

$$(X_1, X_2, \cdots, X_N)$$

Sam

Recurre
Net


1_2 Normalizing Flows :
q(z)
z
Coordinate
Transformation
X = g(z)
Easy to Sample
bijective
z = g^{-1}(x)
p(x)
x

$$\int q(z)\,dz = 1 \iff \int \underbrace{q\left(g^{-1}(x)\right)\left|\frac{dz}{dx}\right|}_{p(x)}\,dx$$

* In general

$$p(\vec{x}) = q\left(g^{-1}(\vec{x})\right)\left|\det\left(\frac{\partial \vec{z}}{\partial \vec{x}}\right)\right|$$

$$\frac{\partial \vec{z}}{\partial \vec{x}} = \begin{pmatrix} * & & & \\ * & * & & \\ * & * & * & \\ * & * & * & * \end{pmatrix}$$

Example.

$$g\begin{cases} x_1 = z_1 \\ x_2 = z_2 \cdot \exp\big(s(z_1)\big) + t(z_1) \end{cases}$$

$$g^{-1}\begin{cases} z_1 = x_1 \\ z_2 = \big(x_2 - t(x_1)\big) \cdot \exp\big(-s(x_1)\big) \end{cases}$$

$$J = \frac{\partial \vec{x}}{\partial \vec{x}} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_2}{\partial x_1} \\ \frac{\partial z_1}{\partial x_2} & \frac{\partial z_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1 & * \\ 0 & \exp(-s(x_1)) \end{pmatrix}$$

$$g_1 \circ g_2 \cdots g_M$$

$$\det\left(J_1 \times J_2 \times \cdots \times J_M\right)$$

$$\begin{pmatrix} 1 & * \\ 0 & \exp(-S(x_1)) \end{pmatrix}$$

(2) Approximate Likelihood models:

2.1 Energy-based models

$$P_{\vec{X}}(\vec{x}) = \frac{\exp\left(-E_{\vec{X}}(\vec{x})\right)}{Z_{\vec{X}}} \qquad \sum_{ij} W_{ij} x_i x_j + \sum_i b_i x_i$$

$$Z_{\vec{X}} = \sum_{\vec{x}} \exp\left(-E_{\vec{X}}(\vec{x})\right)$$

"Intractable"

$$\frac{P(\vec{x}_2)}{P(\vec{x}_1)} = \exp\left(-\left(E_\lambda(\vec{x}_2) - E_\lambda(\vec{x}_1)\right)\right)$$

↳ Use Metropolis sampling

Intractable

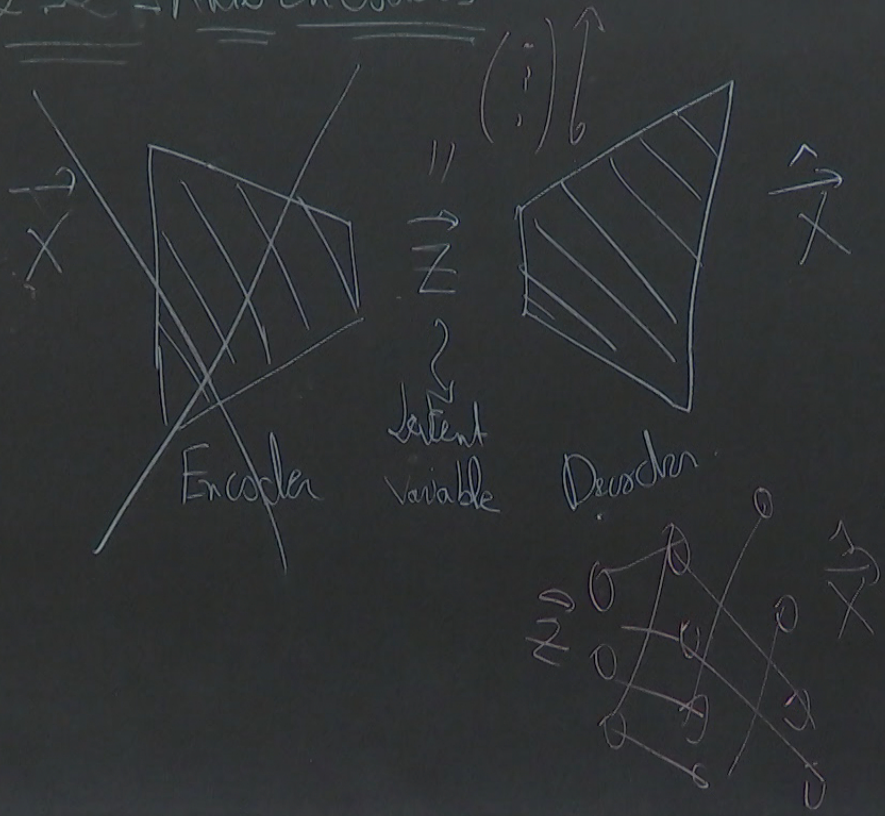
↳ Cost function

$$C = \|\vec{x} - \vec{\hat{x}}\|^2 \sim \text{NLL}$$

$$\mathcal{L} = \exp(-C) = \exp\left(-\|\vec{x} - \vec{\hat{x}}\|^2\right)$$

$\left(E_{\vec{x}}(\vec{x}_2) - E_{\vec{x}}(\vec{x}_1)\right)$

2.2 - Autoencoders



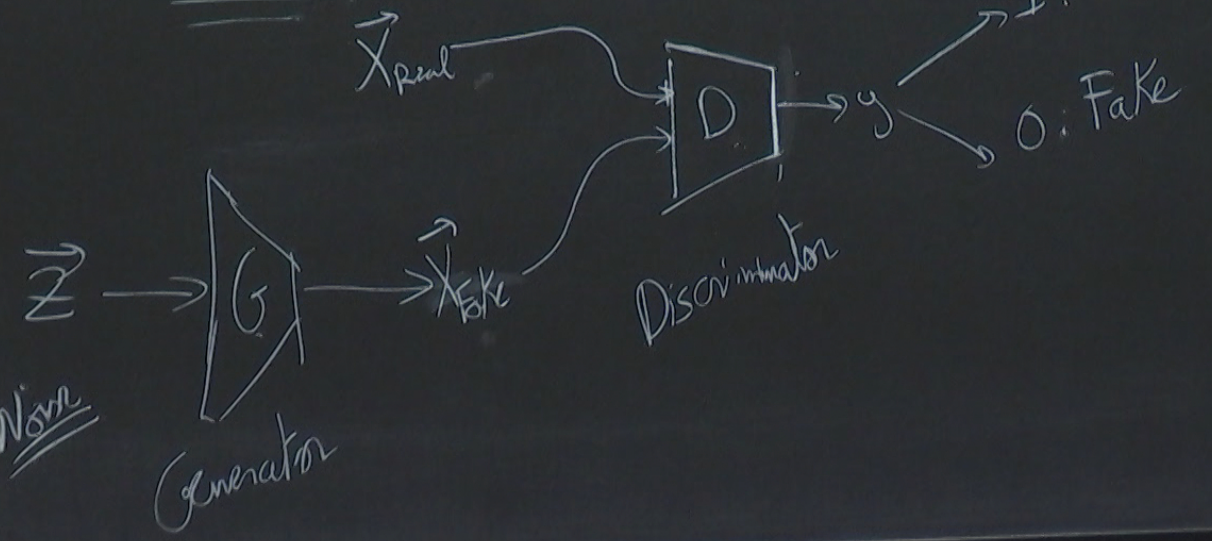
↳ Cost

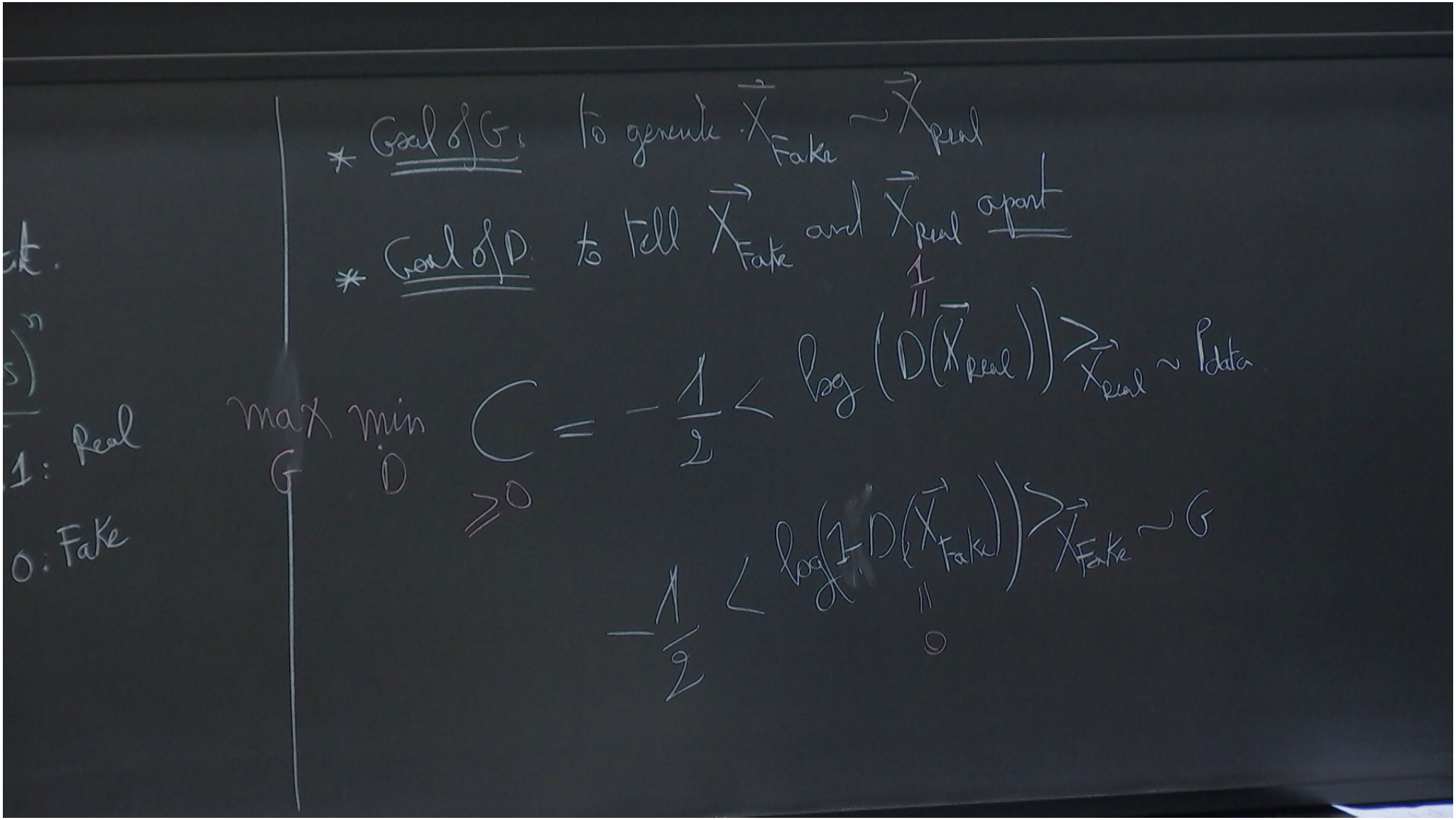
$\mathcal{L}$

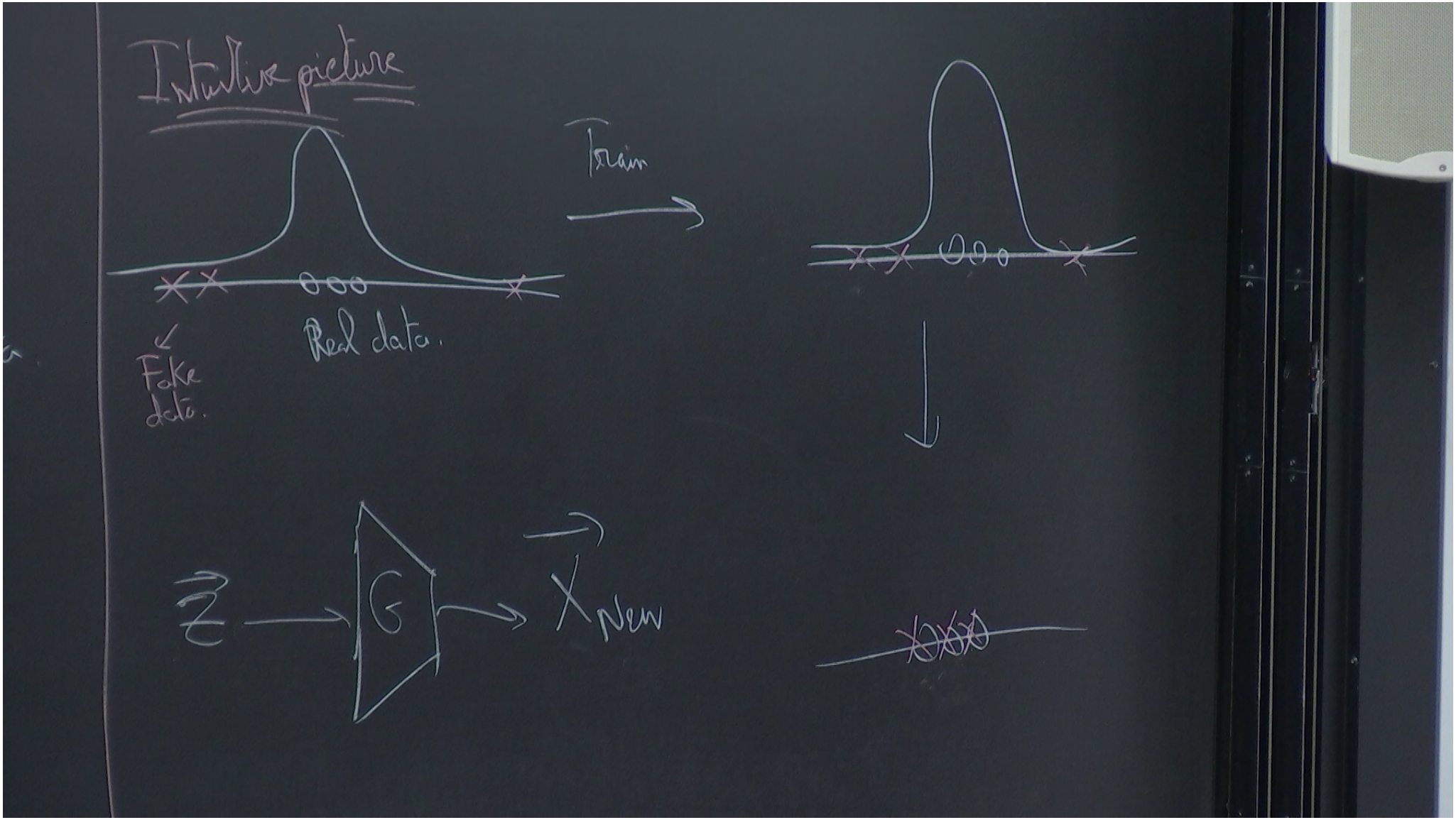
③ Implicit Likelihood models.

↳ Knowledge of likelihood is implicit.

« Generative Adversarial Networks (GANs) »




* Goal of G: to generate $\vec{X}_{Fake} \sim \vec{X}_{real}$
* Goal of D: to tell $\vec{X}_{Fake}$ and $\vec{X}_{real}$ apart
$\max_G \min_D C = -\frac{1}{2} < \log\left(D(\vec{X}_{real})\right) >_{\vec{X}_{real} \sim P_{data}}$
≥ 0
$= 1$
$-\frac{1}{2} < \log\left(1 - D(\vec{X}_{Fake})\right) >_{\vec{X}_{Fake} \sim G}$
$= 0$
1: Real
0: Fake


Intuitive picture
Train
Real data.
Fake data.
G
X_{New}