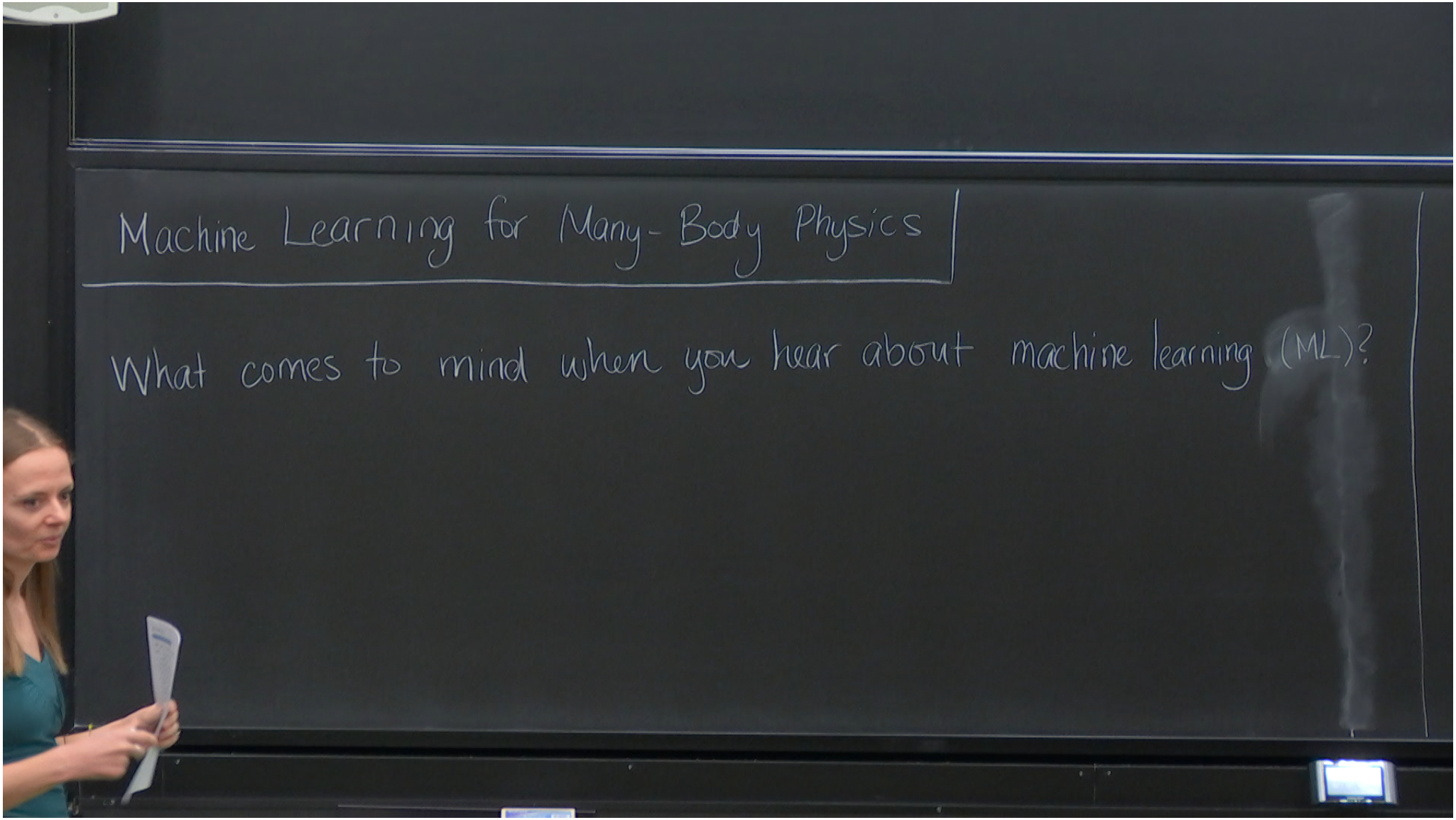
Title: Machine Learning Lecture - 230228

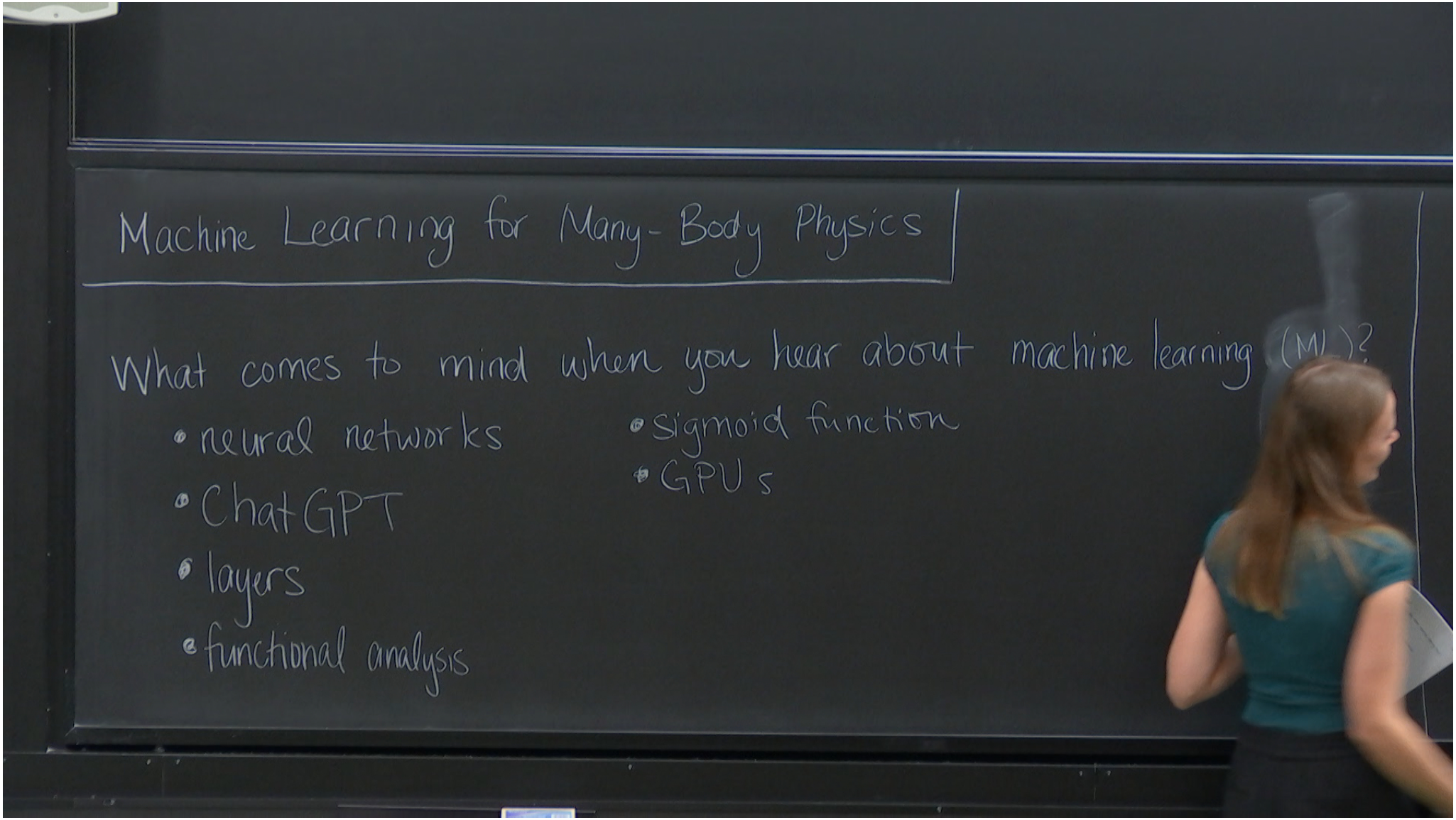
Speakers: Lauren Hayward

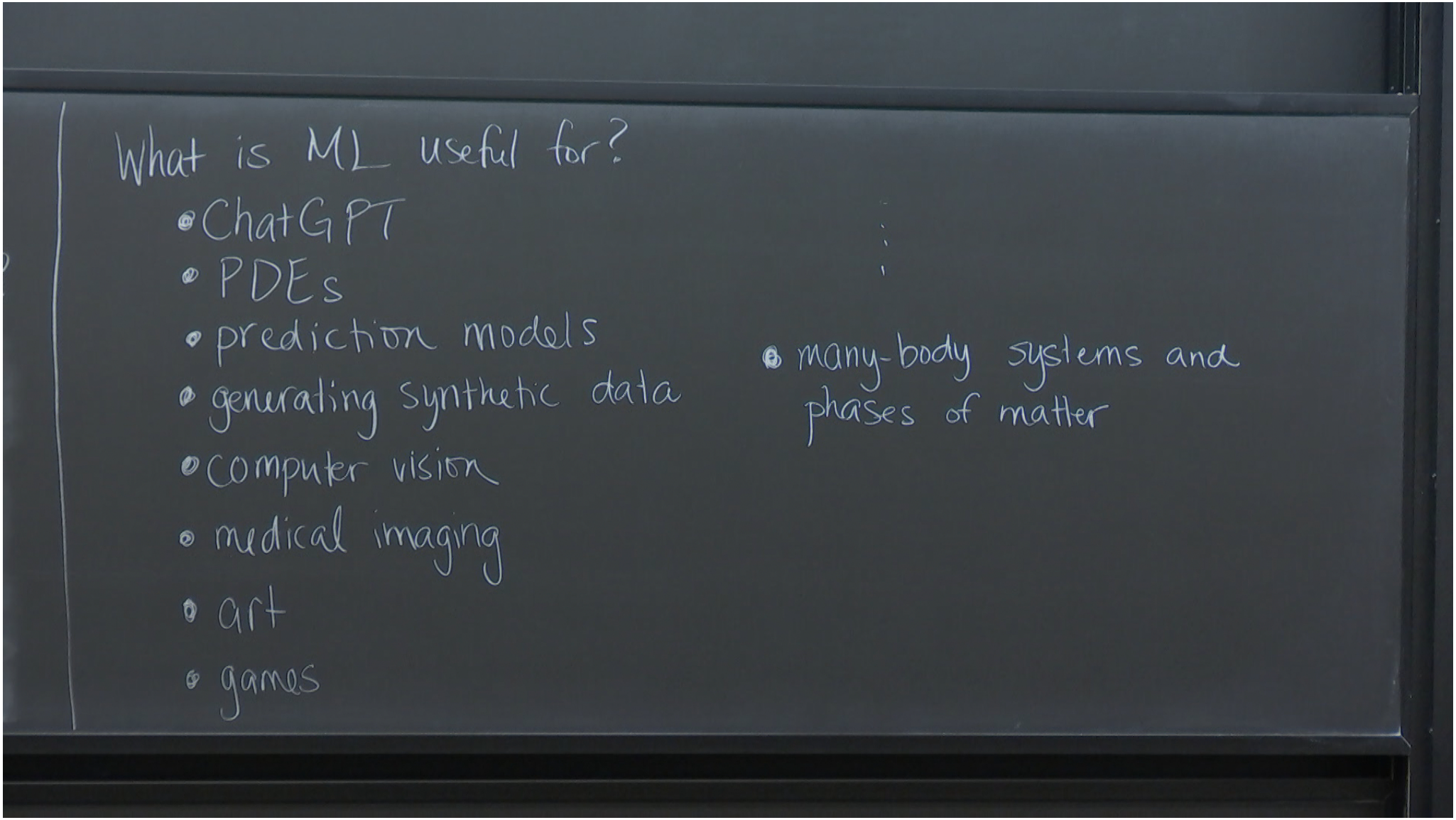
Collection: Machine Learning for Many-Body Physics (2022/2023)

Date: February 28, 2023 - 9:00 AM

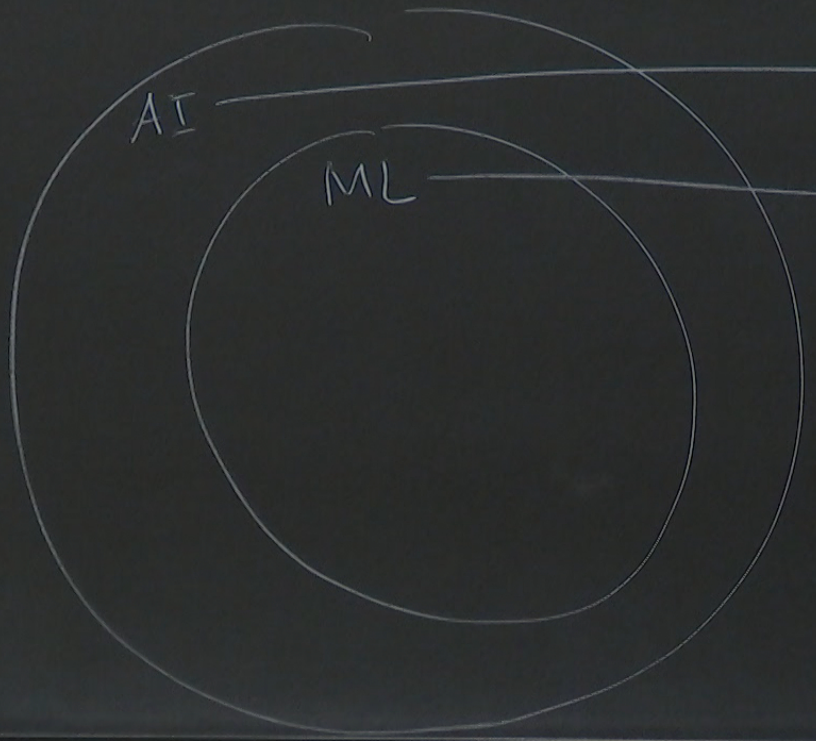
URL: https://pirsa.org/23020018


Machine Learning for Many-Body Physics
What comes to mind when you hear about machine learning (ML)?


Machine Learning for Many-Body Physics
What comes to mind when you hear about machine learning (ML)?
• neural networks
• ChatGPT
• layers
• functional analysis
• sigmoid function
• GPUs

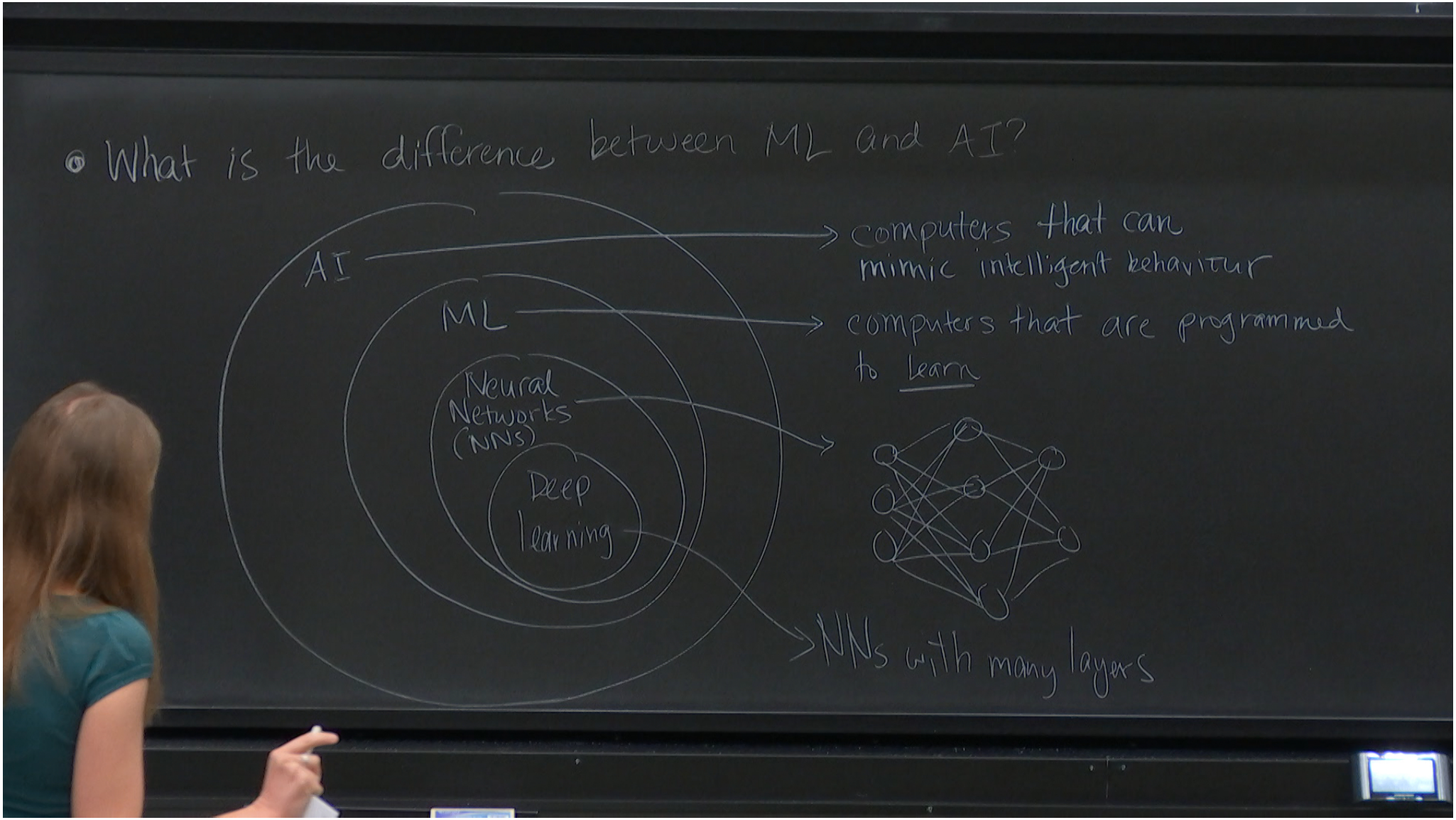

What is ML useful for?
ChatGPT
PDEs
prediction models
generating synthetic data
computer vision
medical imaging
art
games
many-body systems and phases of matter

- What is the difference between ML and AI?



AI → computers that can mimic intelligent behaviour

ML → computers that are programmed to learn


What is the difference between ML and AI?
AI
computers that can mimic intelligent behaviour
ML
computers that are programmed to learn
Neural Networks (NNs)
Deep learning
NNs with many layers

- games

ML: training computers to detect and characterize features from data

Goal of many-body physics: predicting/explaining macroscopic phenomena (features) from microscopic quantities (data)

learning

NNs with many layers

Many-body physics:

- fundamentally relies on many particles interacting
- needed to describe high-T superconductors, fractional quantum Hall effect, ...
- not needed to describe phenomena where single quasiparticle theories exist (eg. electron conduction in most metals and semiconductor)

Solving the



Solving the Schrödinger eq. becomes exponentially difficult

Other computational methods available:

- Monte Carlo
- tensor networks
- exact diagonalization

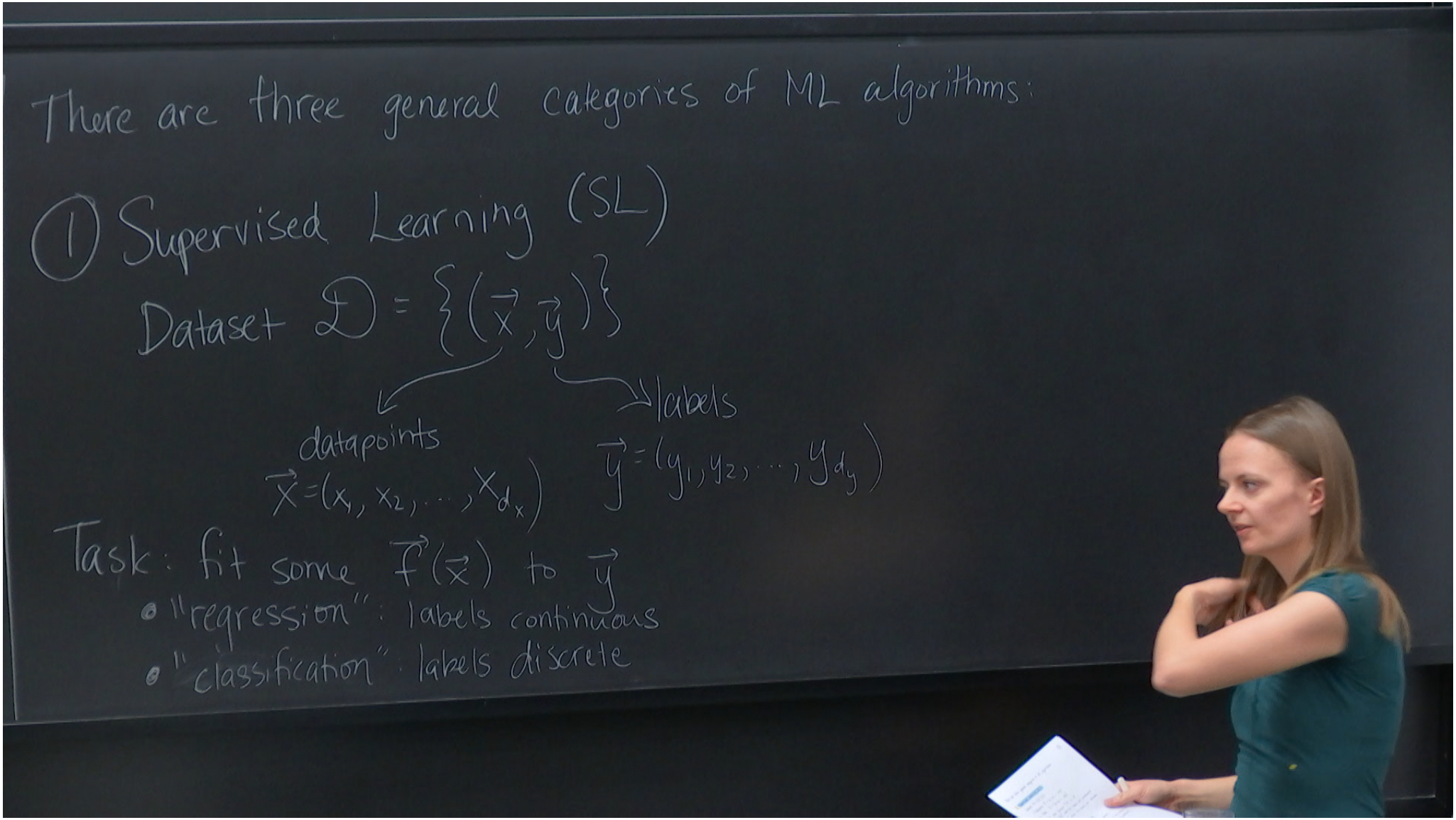
Solving the Schrödinger eq. becomes exponentially difficult

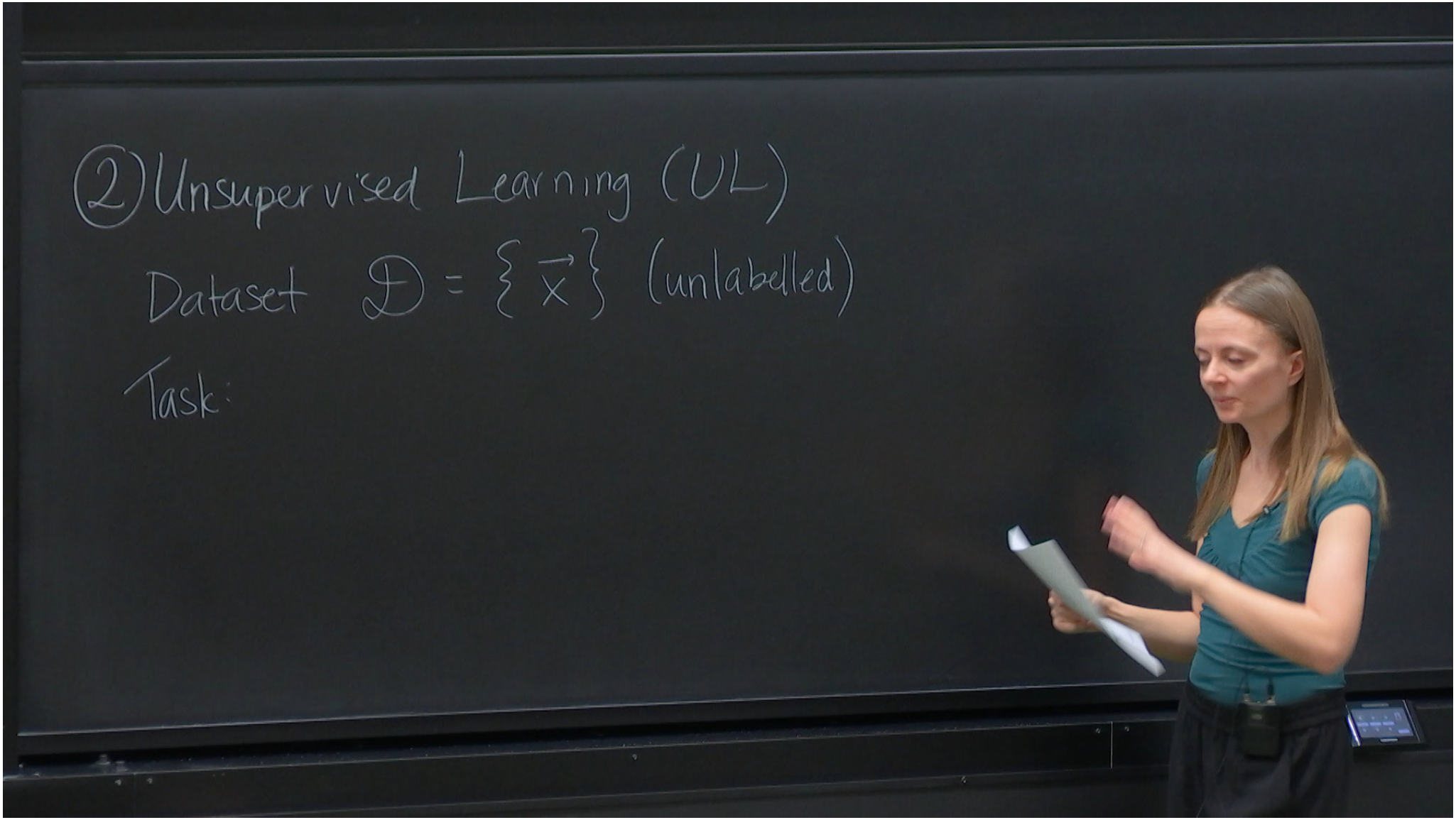
Other computational methods available:

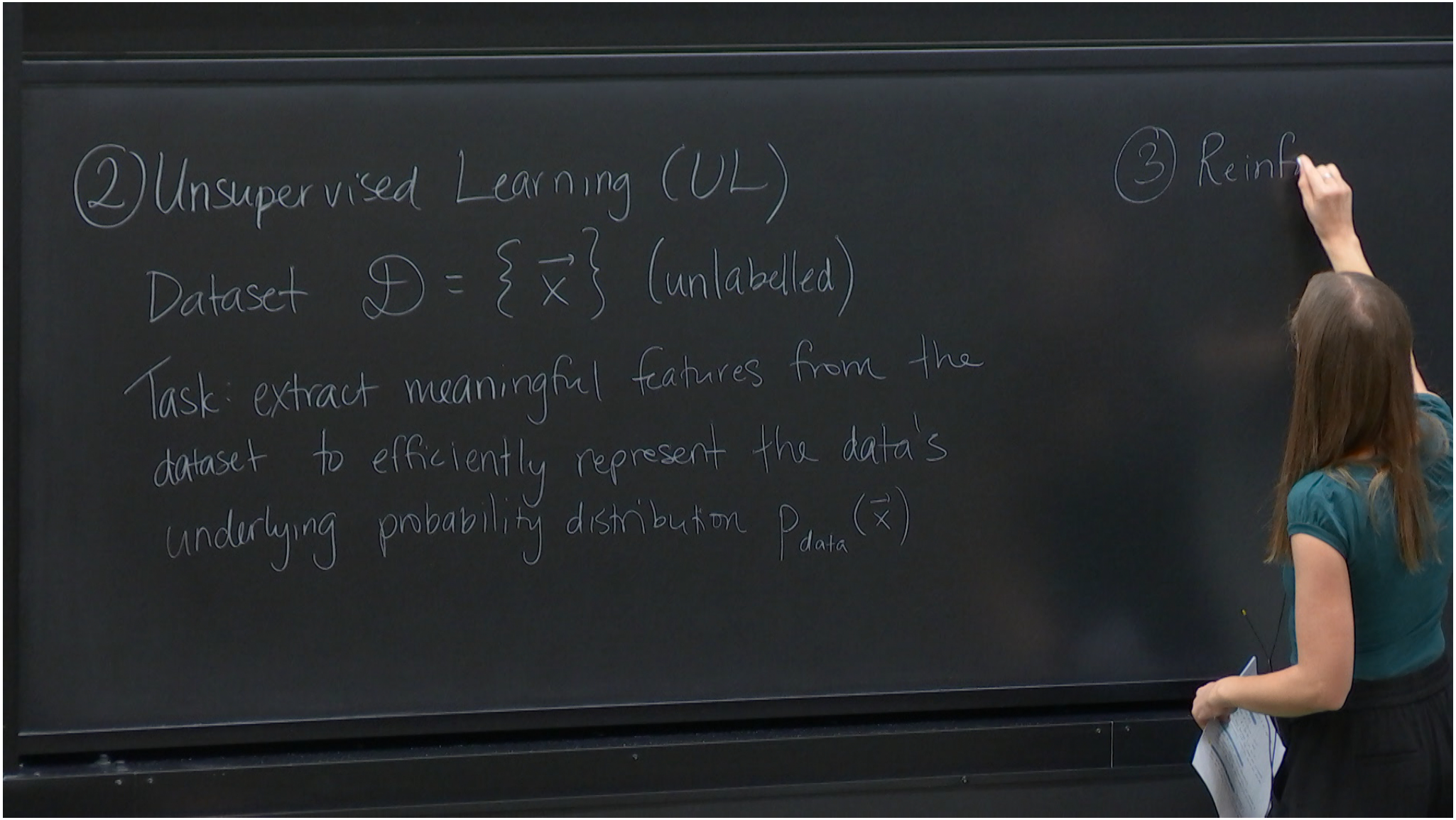
- Monte Carlo
- tensor networks
- exact diagonalization
- density functional theory
- ⋮

} Start from a physical theory, use it to generate data describing the system

ML methods tend to be more data driven: start from physical measurements (data) and then learn about macroscopic properties (features) such as phases.


There are three general categories of ML algorithms:
① Supervised Learning (SL)
Dataset $\mathcal{D} = \{(\vec{x}, \vec{y})\}$
datapoints
$\vec{x} = (x_1, x_2, \dots, x_{d_x})$
labels
$\vec{y} = (y_1, y_2, \dots, y_{d_y})$
Task: fit some $\vec{f}(\vec{x})$ to $\vec{y}$
• "regression": labels continuous
• "classification": labels discrete


② Unsupervised Learning (UL)
Dataset $\mathcal{D} = \{\vec{x}\}$ (unlabelled)
Task:


② Unsupervised Learning (UL)
Dataset $\mathcal{D} = \{\vec{x}\}$ (unlabelled)
Task: extract meaningful features from the dataset to efficiently represent the data's underlying probability distribution $p_{data}(\vec{x})$
③ Reinf

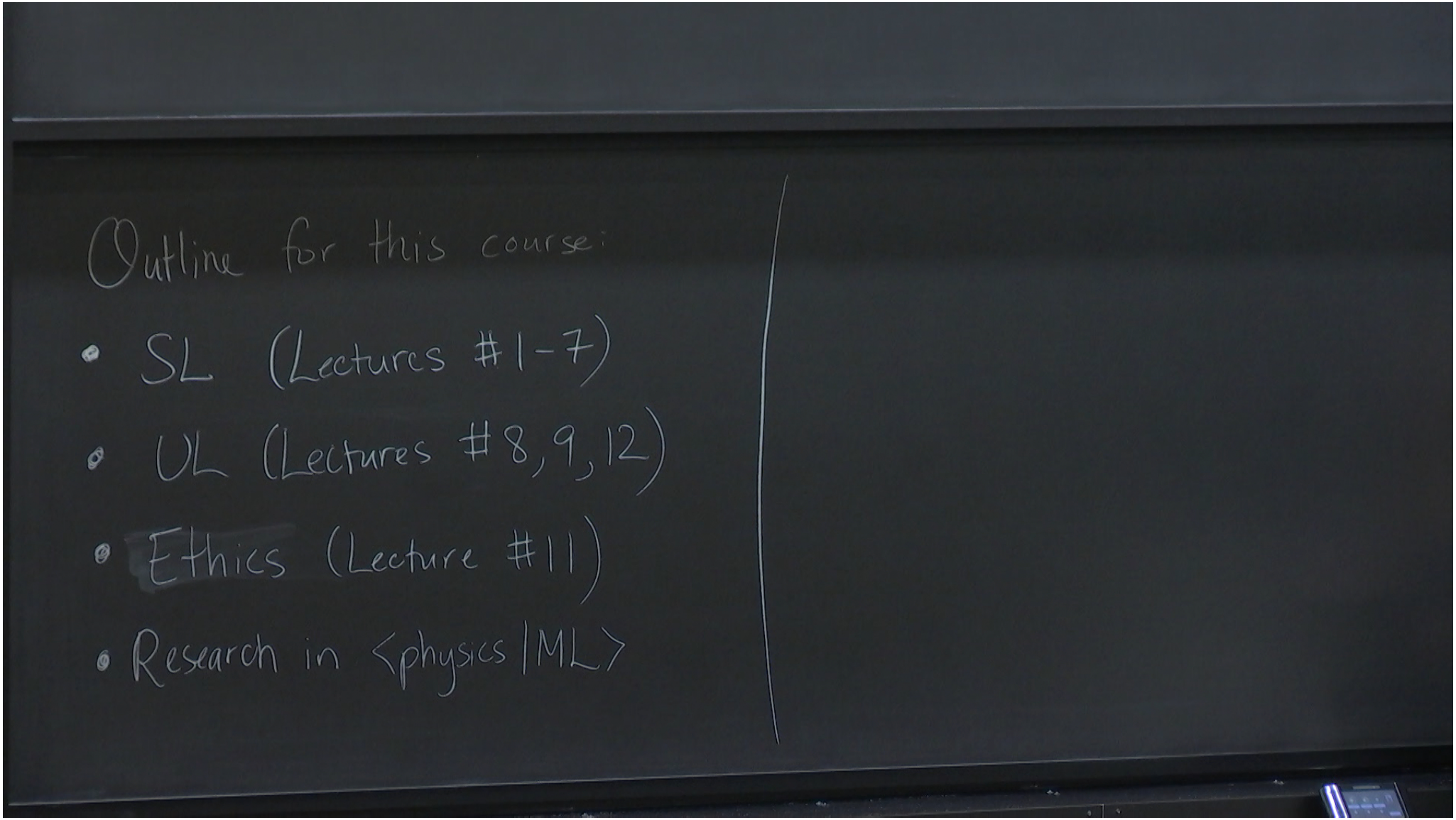
Task: fit some $\vec{f}(\vec{x})$ to $\vec{y}$

- "regression": labels continuous
- "classification": labels discrete

(3) Reinforcement Learning (RL)

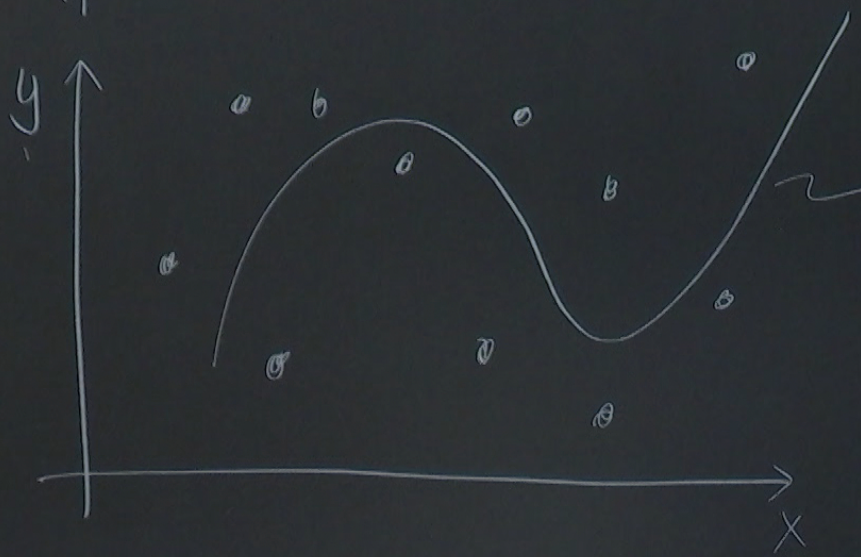
Not covered in this course

Task: given an "environment", take an action so the resulting "reward" is maximized

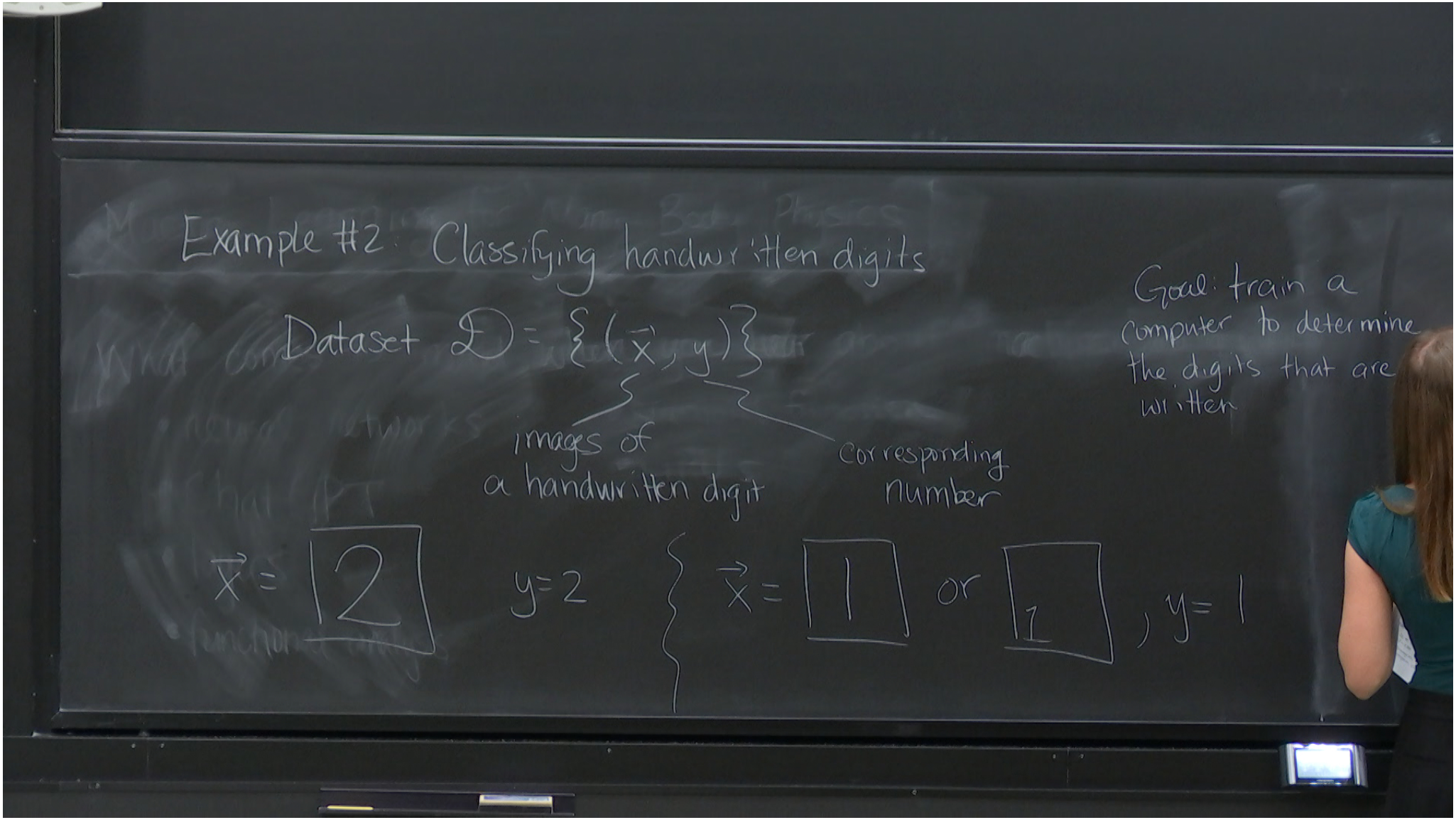

Outline for this course:
• SL (Lectures #1-7)
• UL (Lectures #8, 9, 12)
• Ethics (Lecture #11)
• Research in <physics/ML>

SL examples

Example #1: 1D regression

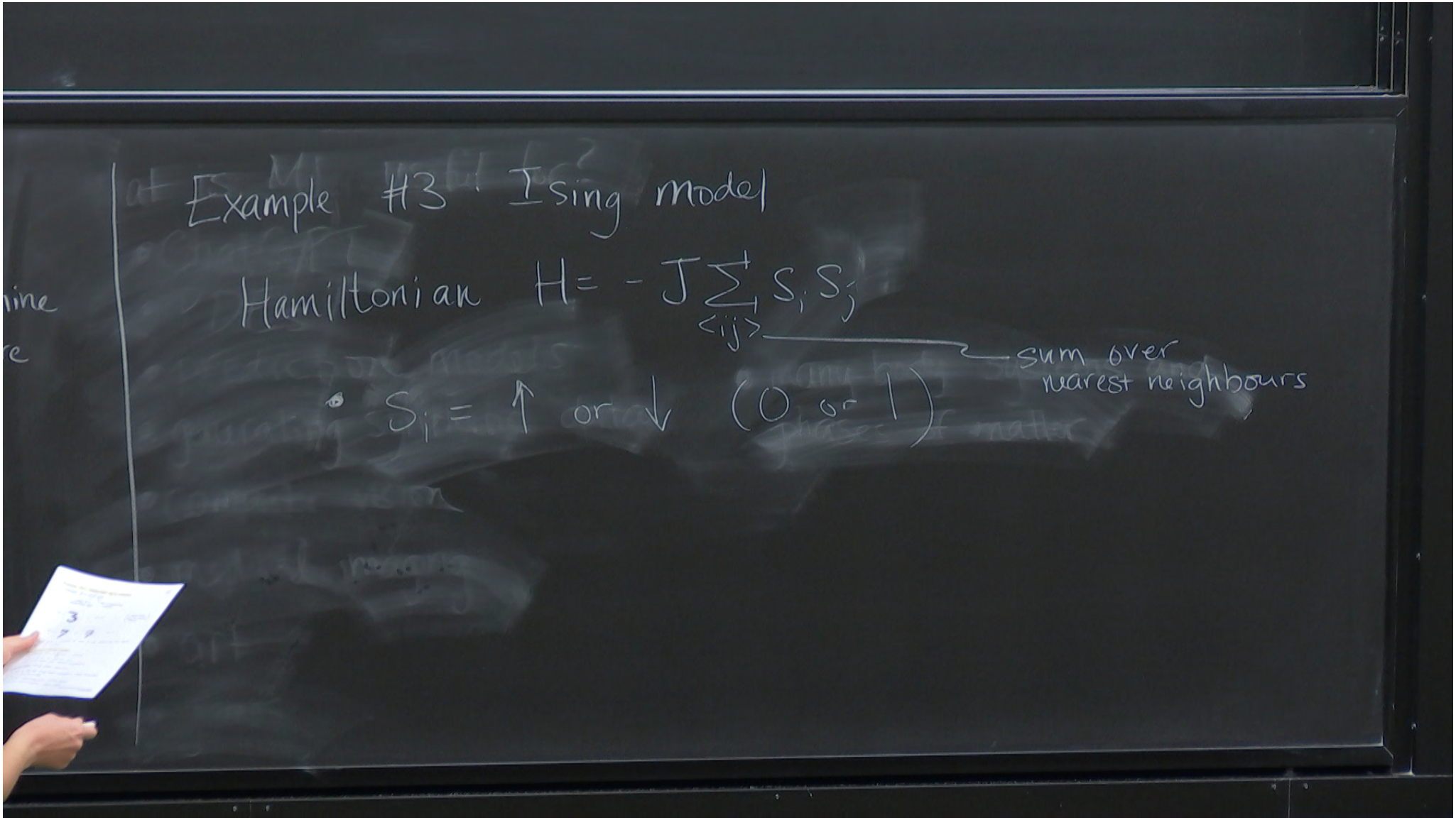


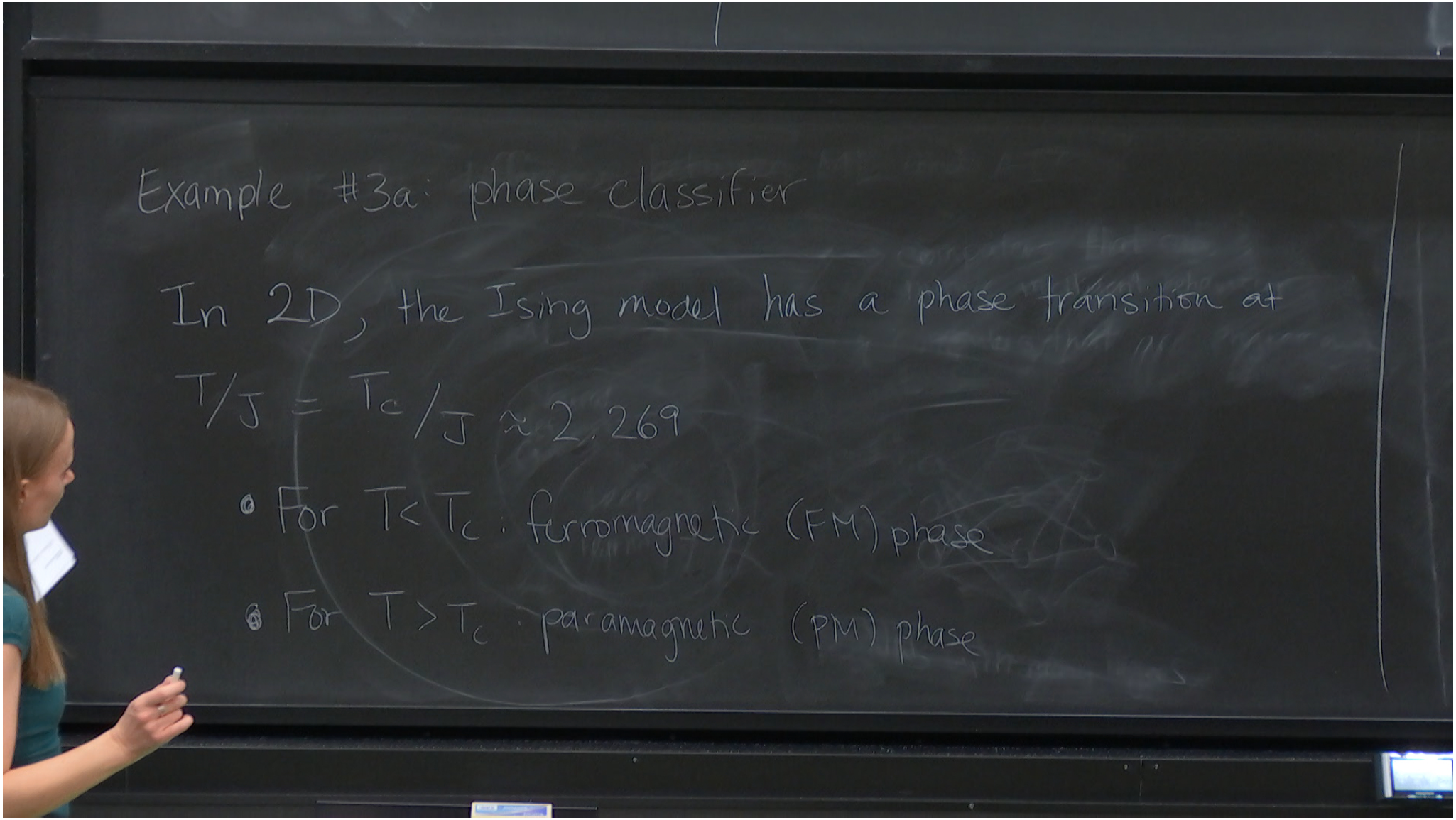
Goal: determine a curve to describe the data


Example #2: Classifying handwritten digits
Dataset $\mathcal{D} = \{(\vec{x}, y)\}$
images of a handwritten digit
corresponding number
Goal: train a computer to determine the digits that are written
$\vec{x} =$
$y = 2$
$\vec{x} =$ or , $y = 1$

Example #3: Ising model

Hamiltonian $H = -J \sum_{\langle ij \rangle} S_i S_j$


Example #3 : Ising model
Hamiltonian $H = -J \sum_{\langle ij \rangle} S_i S_j$
sum over nearest neighbours
$S_i = \uparrow$ or $\downarrow$ (0 or 1)


Example #3a: phase classifier
In 2D, the Ising model has a phase transition at
$T/J = T_c/J \approx 2.269$
For $T < T_c$: ferromagnetic (FM) phase
For $T > T_c$: paramagnetic (PM) phase

$\mathcal{D} = \{(\vec{x}, y)\}$

Spin config

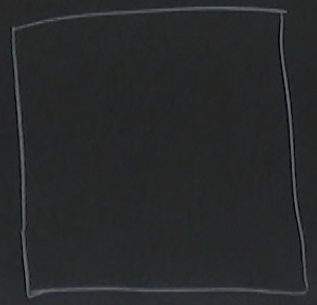
$(S_1, S_2, \ldots, S_N)$

N = # of spins

label is FM or PM

(0 or 1)

$\vec{x} =$



$\mathcal{D} = \{(\vec{x}, y)\}$

Spin config

$(S_1, S_2, \ldots, S_N)$

N = # of spins

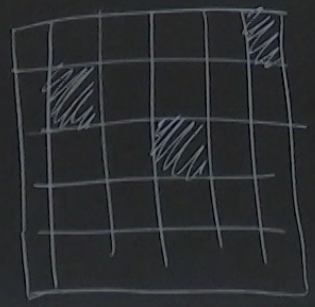
label is FM or PM

(0 or 1)

□ = ↑ = 1

▨ = ↓ = 0

$\vec{x} =$



$\mathcal{D} = \{(\vec{x}, y)\}$

Spin config

$(S_1, S_2, \ldots, S_N)$

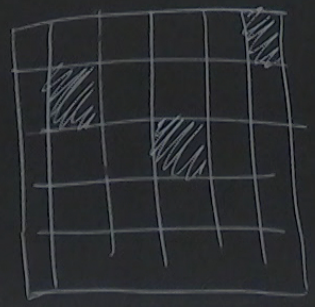
N = # of spins

label is FM or PM

(0 or 1)

□ = ↑ = 1

▨ = ↓ = 0

$\vec{x}$ = 

y = FM

(0)

$\mathcal{D} = \{(\vec{x}, y)\}$

Spin config

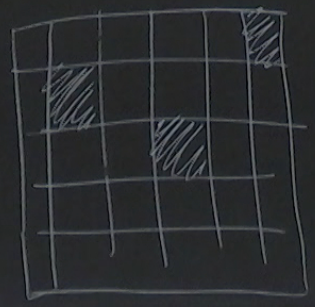
label is FM or PM (0 or 1)

$(S_1, S_2, \ldots, S_N)$

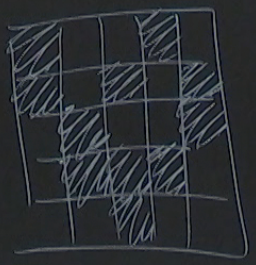
N = # of spins

□ = ↑ = 1

▨ = ↓ = 0

$\vec{x} =$ 

y = FM (0)

$\vec{x} =$ 

, y = PM (1)

See Homework #1

- For $T > T_c$: paramagnetic (PM) phase

$\vec{X} =$

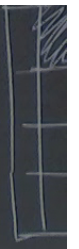
Example #3b: regression to learn the Ising Hamiltonian

$$\mathcal{D} = \{(\vec{x}, y)\}$$

$\vec{x}$: spin config

y: energy of the spin config

- For $T > T_c$: paramagnetic (PM) phase

$\vec{x} =$ 

Example #3b: regression to learn the Ising Hamiltonian

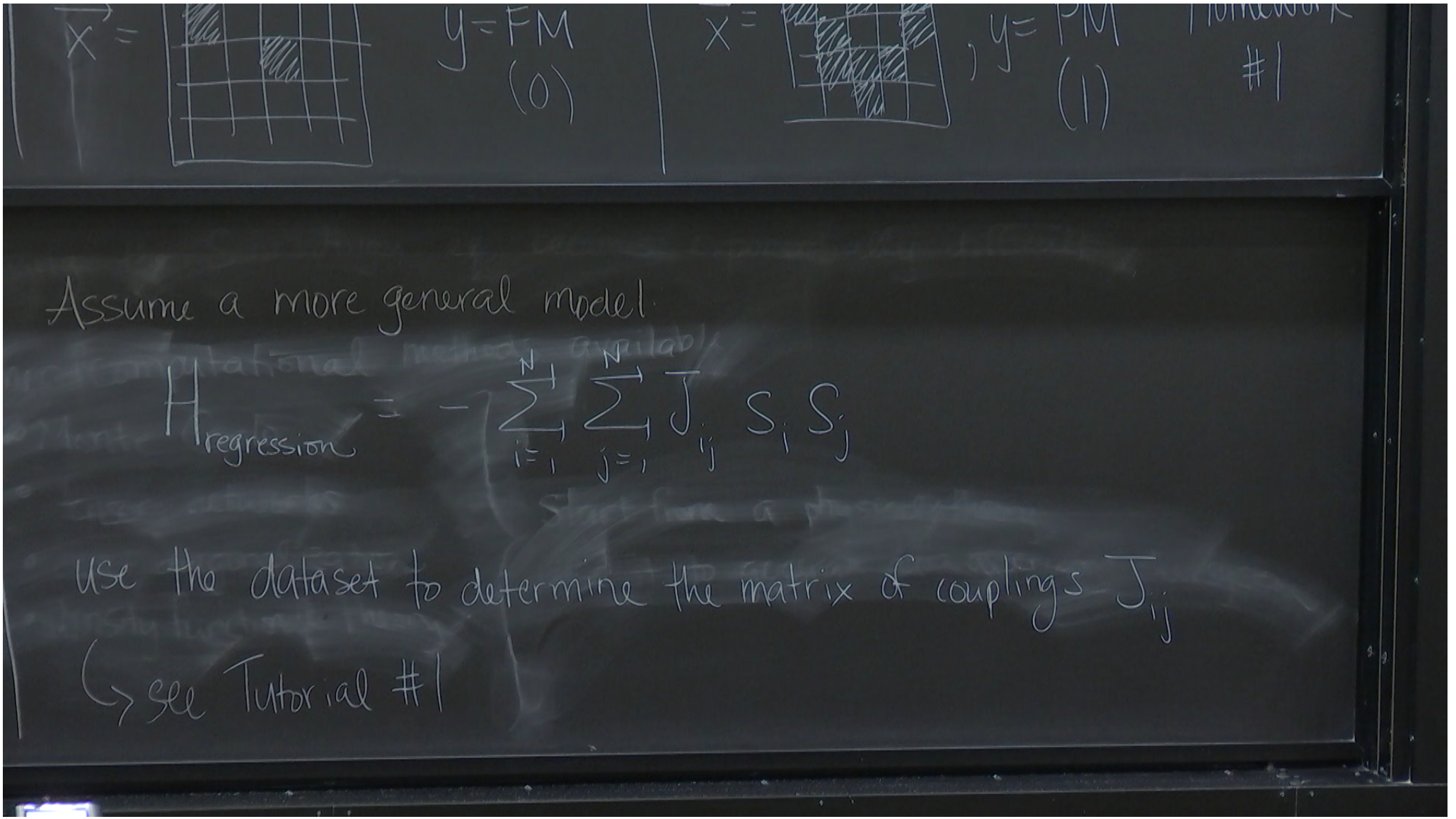
$$\mathcal{D} = \{(\vec{x}, y)\}$$

$\vec{x}$: spin config

y: energy of the spin config

Assume we are given $\mathcal{D}$ and we don't know where it came from

Assume


$\vec{x}$ =
y = FM
(0)
x =
, y = PM
(1)
#1
Assume a more general model
$H_{regression} = -\sum_{i=1}^{N}\sum_{j=1}^{N} J_{ij} S_i S_j$
use the dataset to determine the matrix of couplings J_{ij}
↳ see Tutorial #1