Title: Machine Learning (2021/2022)

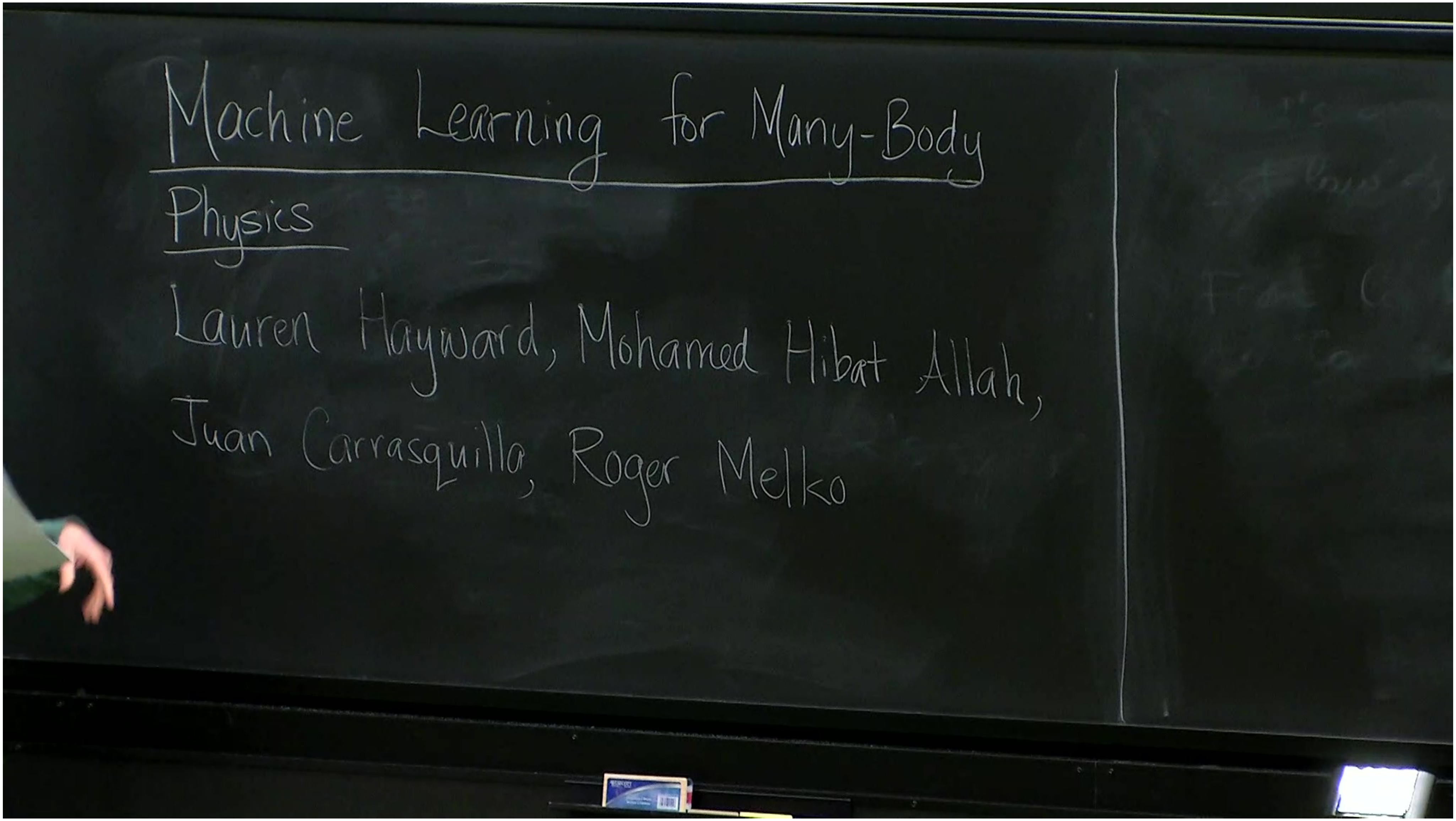
Speakers: Lauren Hayward

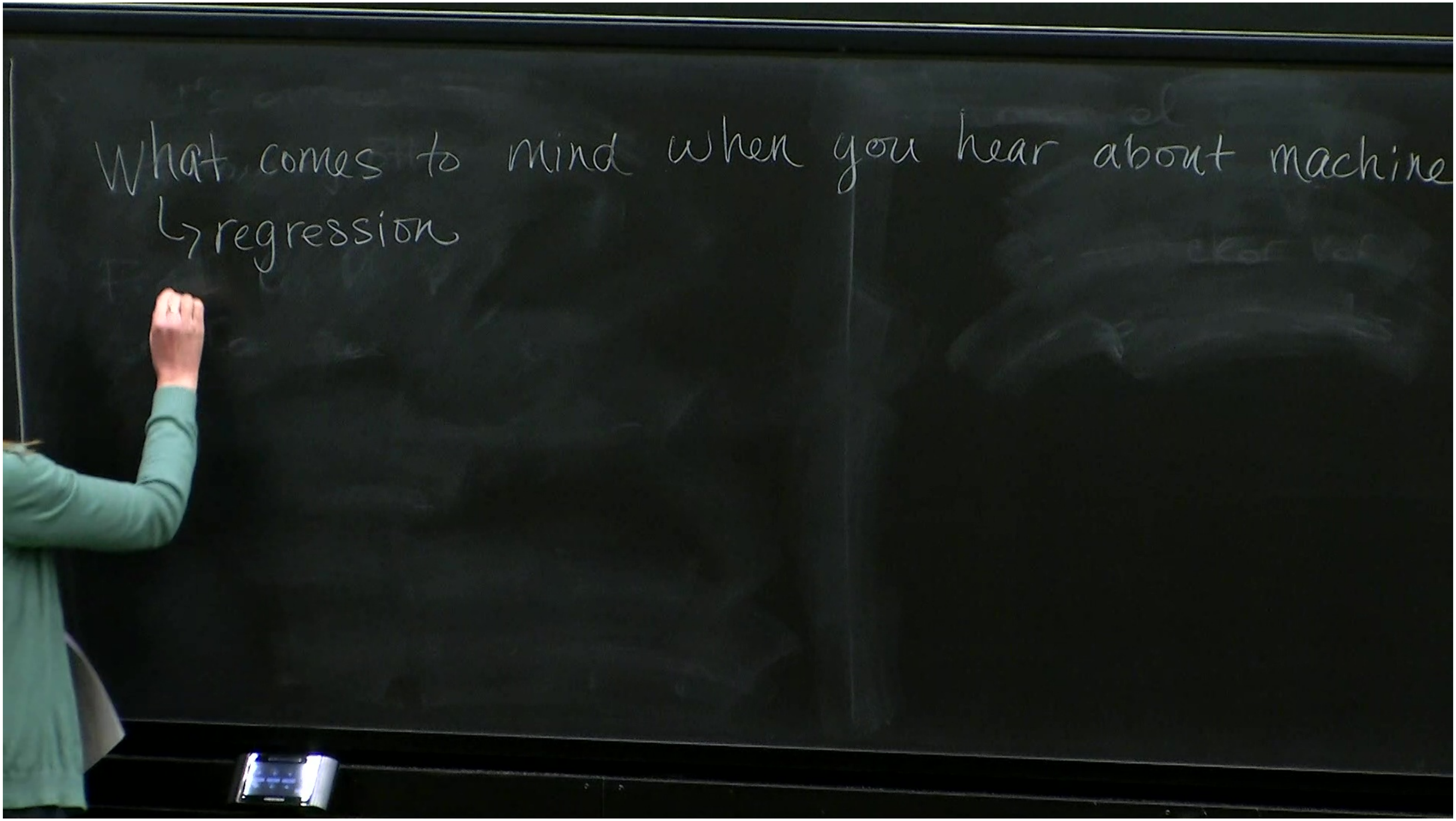
Collection: Machine Learning (2021/2022)

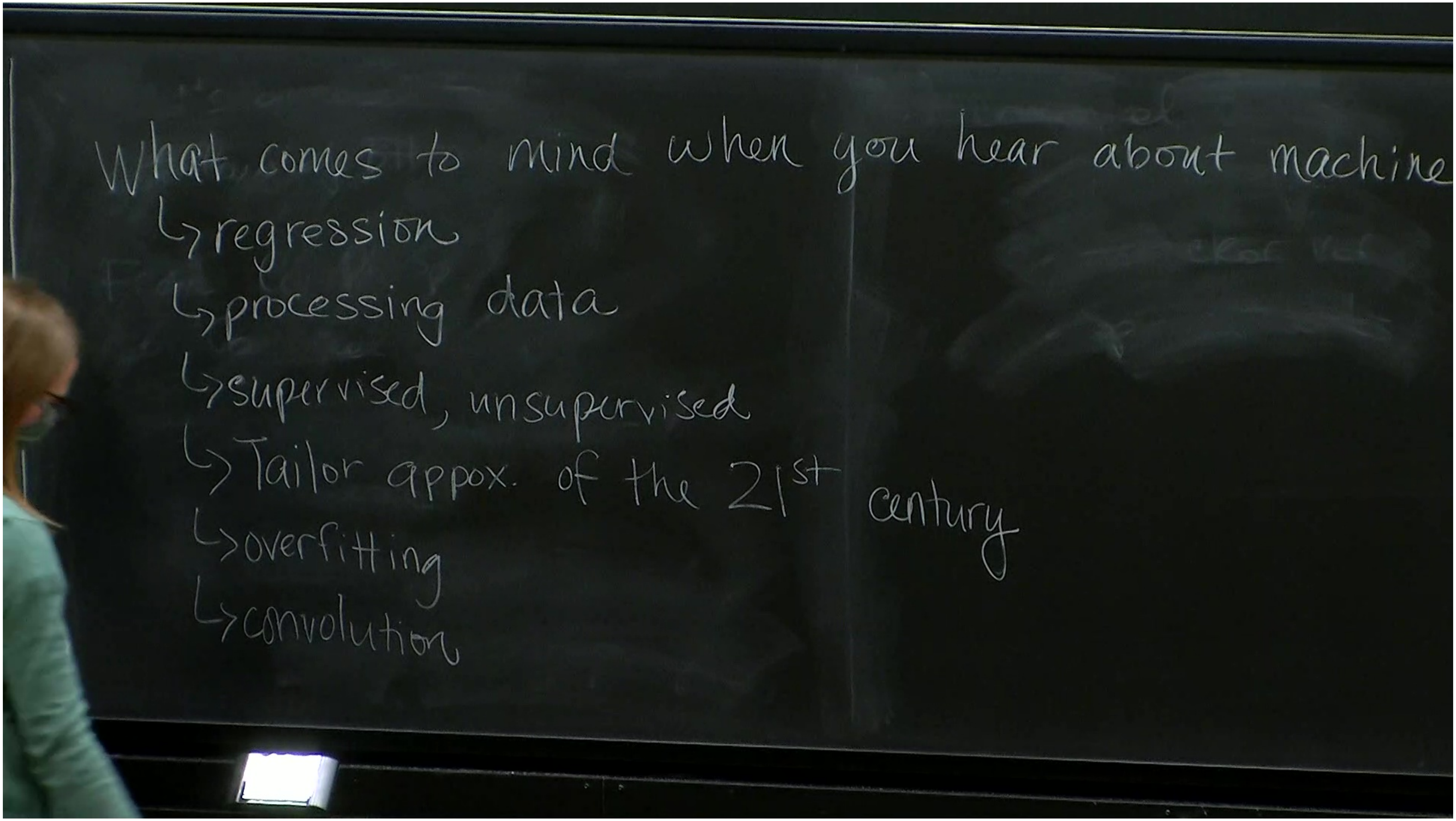
Date: April 04, 2022 - 11:30 AM

URL: https://pirsa.org/22040064

Abstract: This course is designed to introduce modern machine learning techniques for studying classical and quantum many-body problems encountered in condensed matter, quantum information, and related fields of physics. Lectures will focus on introducing machine learning algorithms and discussing how they can be applied to solve problem in statistical physics. Tutorials and homework assignments will concentrate on developing programming skills to study the problems presented in lecture.


Machine Learning for Many-Body Physics
Lauren Hayward, Mohamed Hibat Allah,
Juan Carrasquilla, Roger Melko


What comes to mind when you hear about machine
↳ regression


What comes to mind when you hear about machine
↳ regression
↳ processing data
↳ supervised, unsupervised
↳ Tailor appox. of the 21st century
↳ overfitting
↳ convolution

What is ML useful for?

- ↳ classification (images, etc.)
- ↳ pattern recognition
- ↳ function approx.
- ↳ learning prob distributions
- ↳ generating images, art, music, ...

↳ processing large amounts of data

↳ language translation

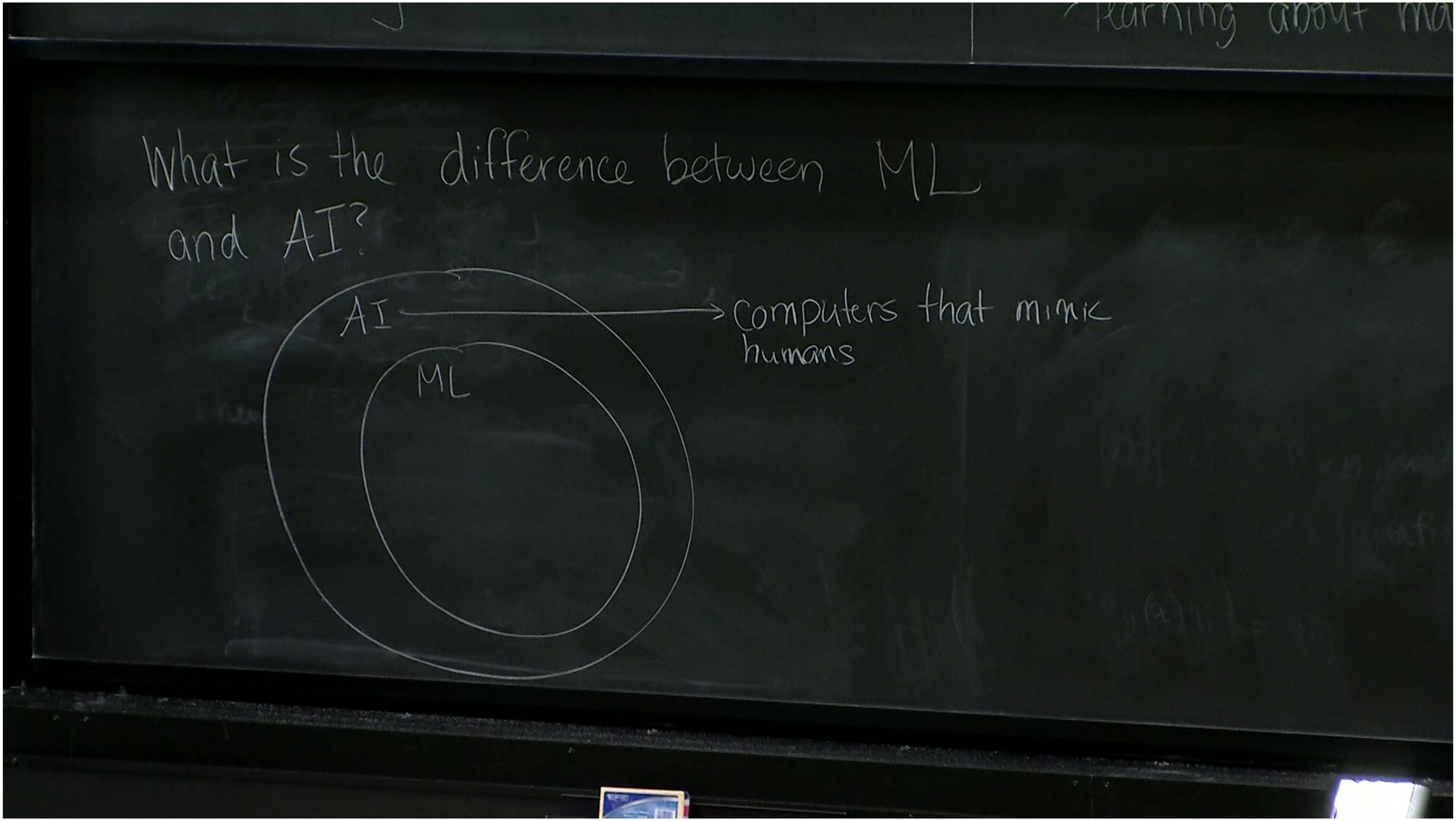
↳ recommender algorithms (Netflix, ...)

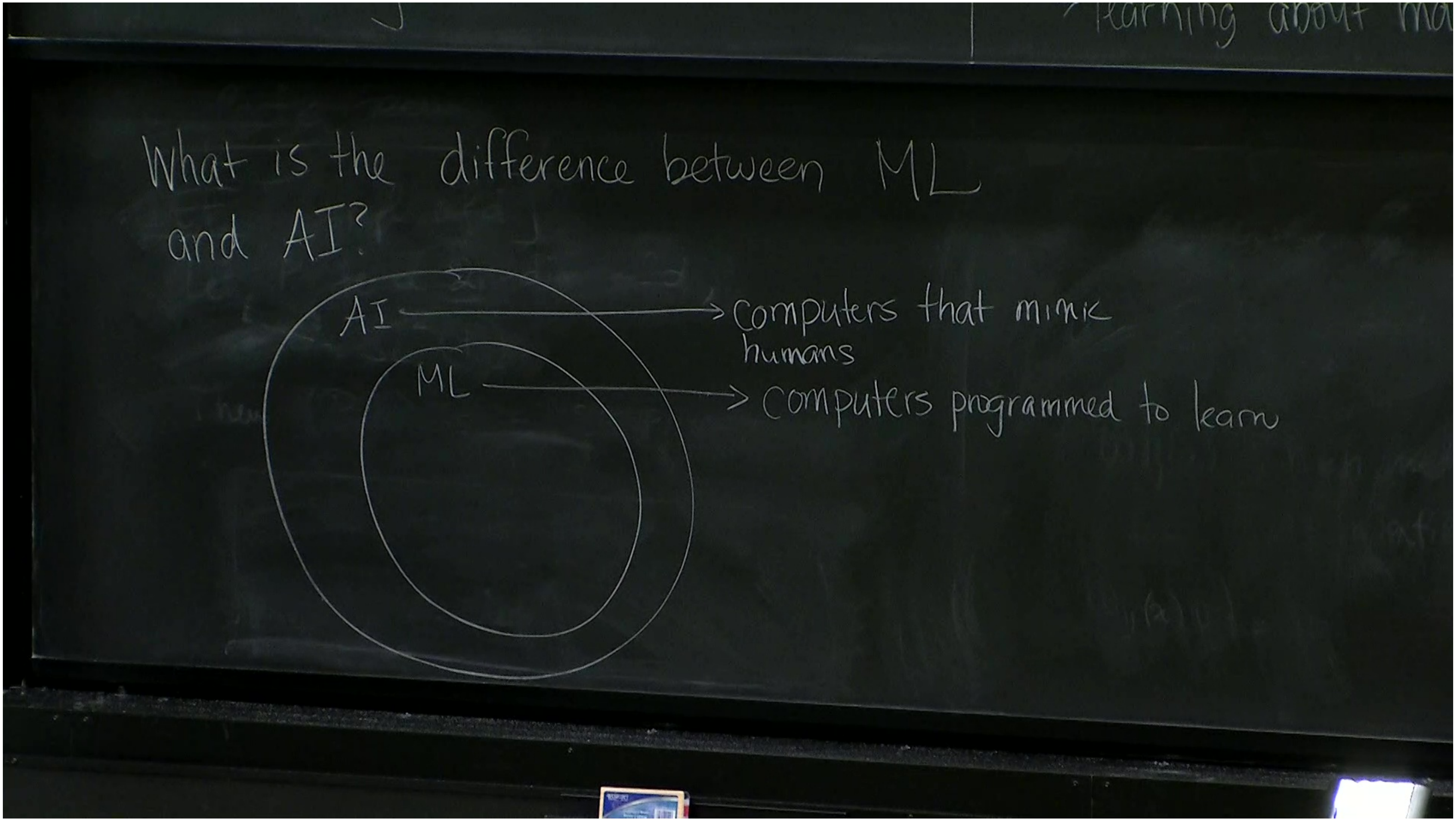
↳ playing games (eg: AlphaGo)

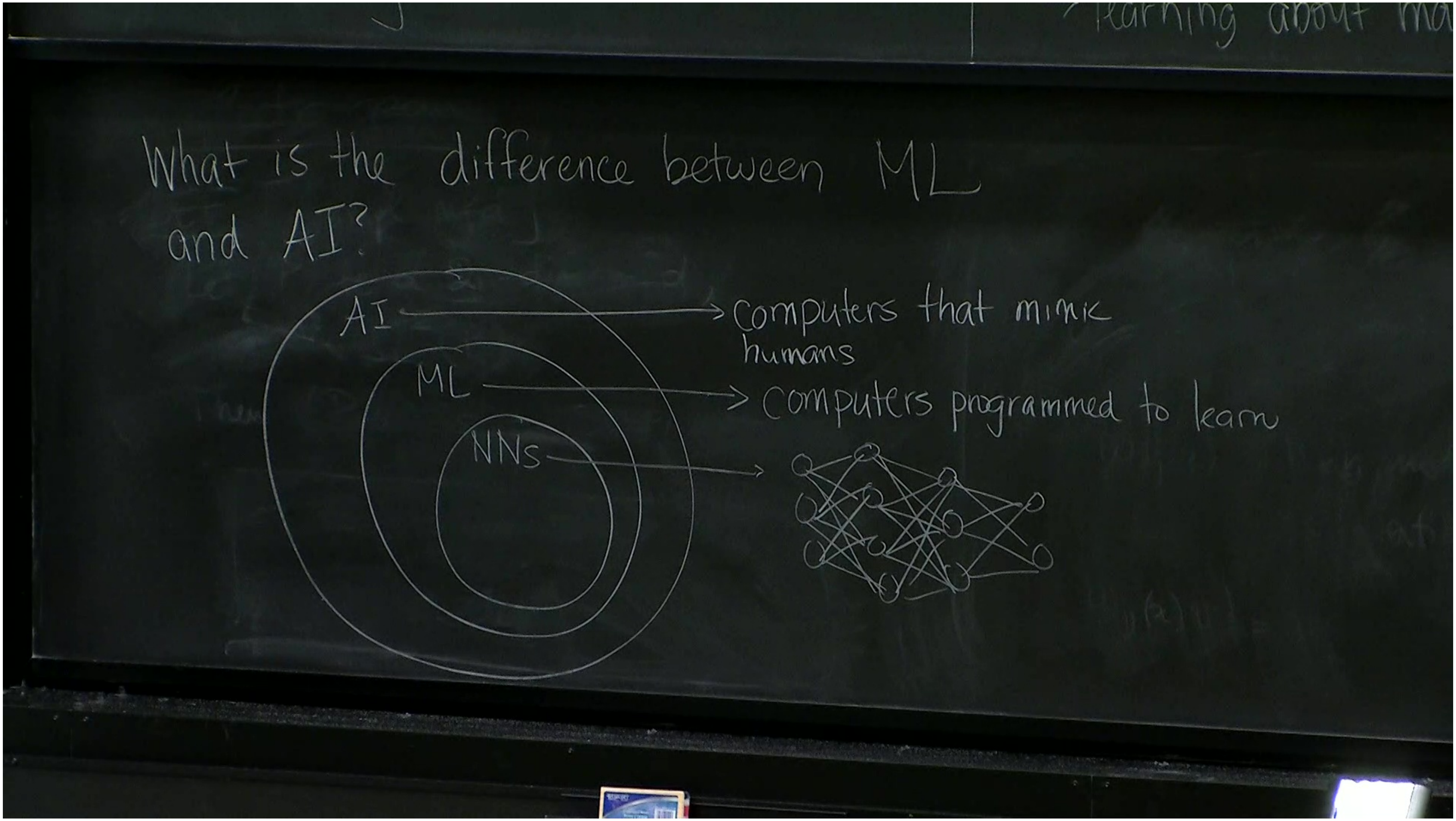
↳ self-driving cars

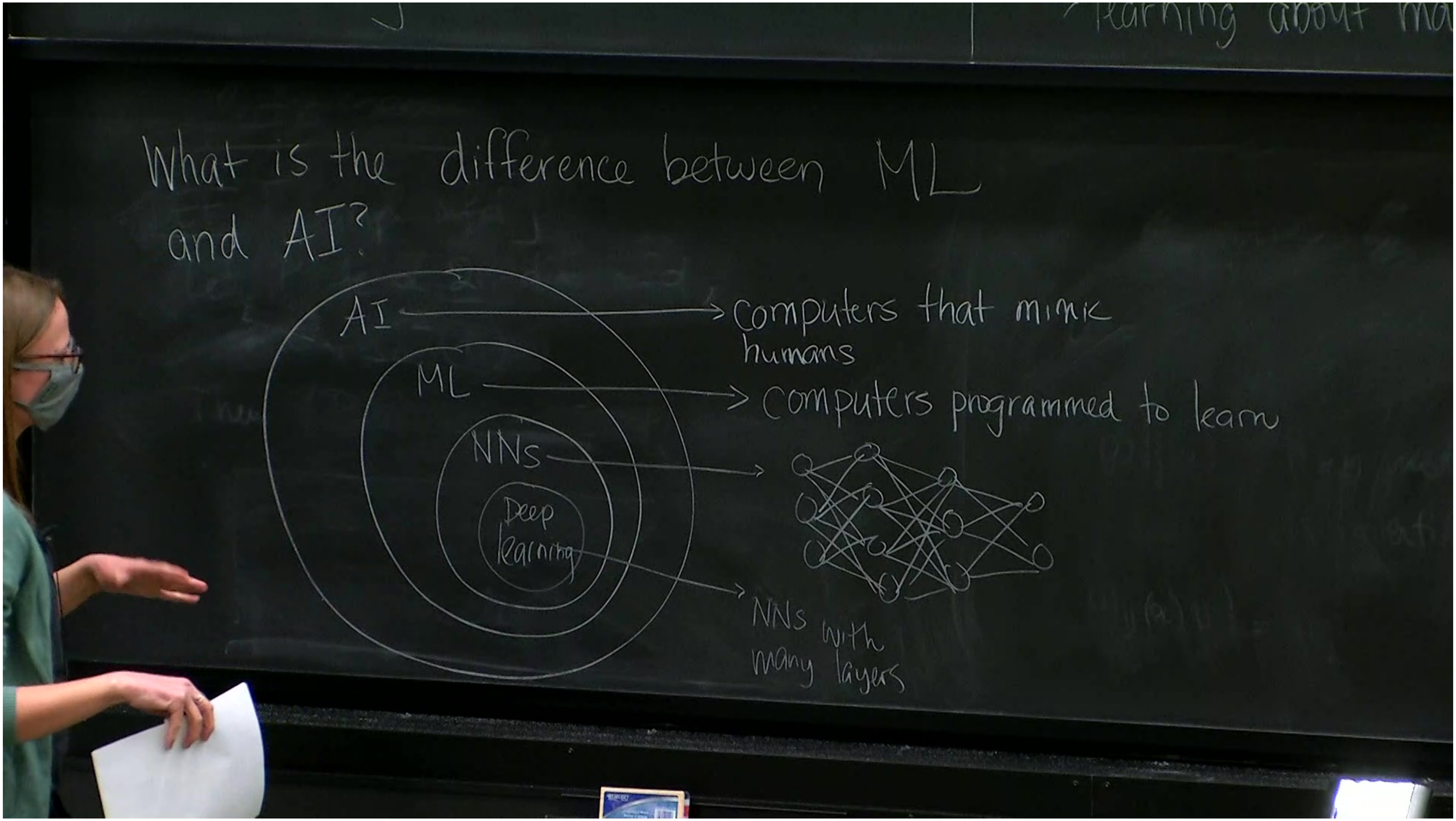
⋮

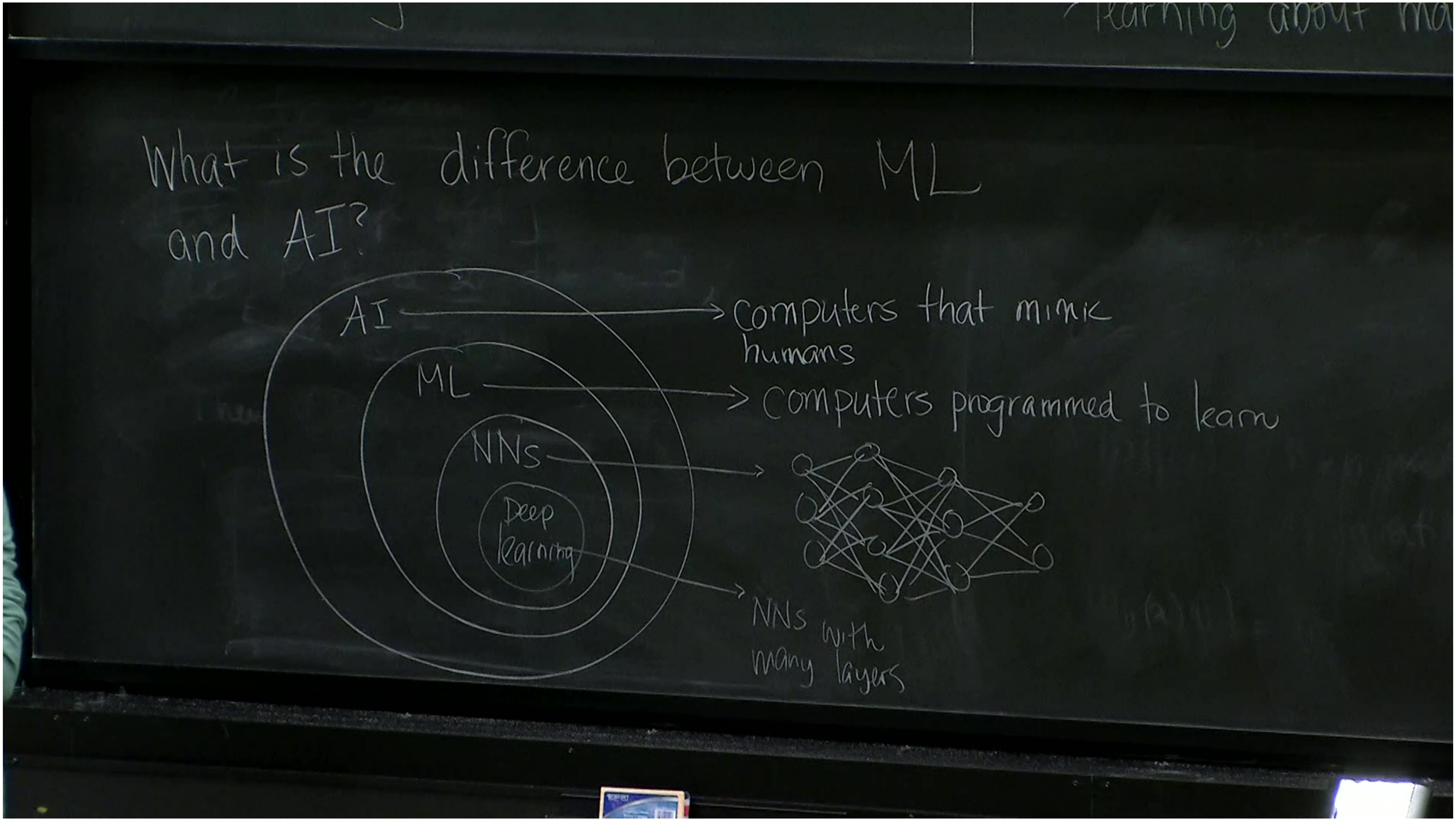
↳ learning about many-body systems and phases of matter

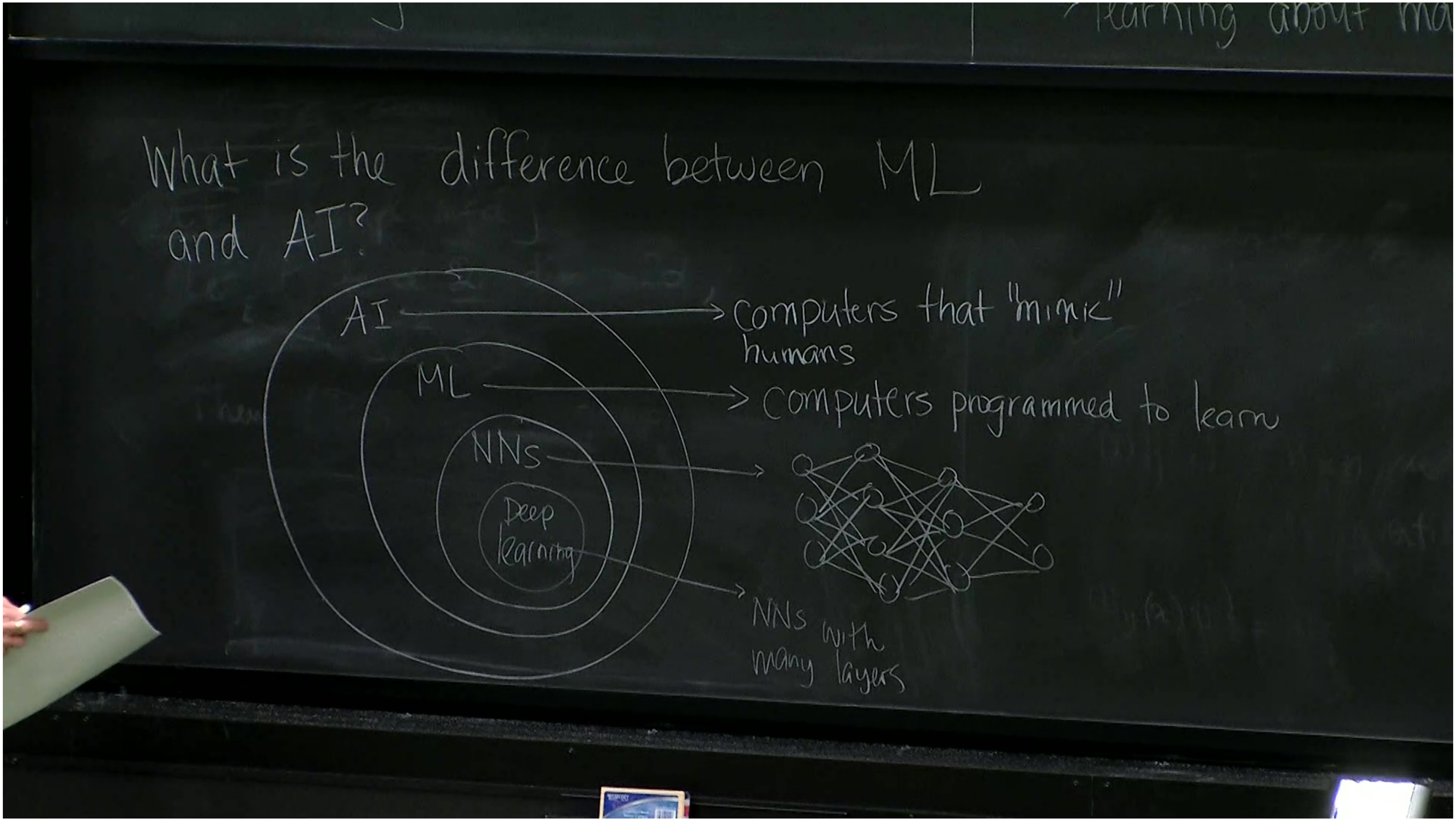

What is the difference between ML and AI?
AI
ML
computers that mimic humans


learning about
What is the difference between ML and AI?
AI
computers that mimic humans
ML
computers programmed to learn


What is the difference between ML and AI?
AI
computers that mimic humans
ML
computers programmed to learn
NNs


What is the difference between ML and AI?
AI
computers that mimic humans
ML
computers programmed to learn
NNs
Deep learning
NNs with many layers


learning about ma
What is the difference between ML and AI?
AI
computers that mimic humans
ML
computers programmed to learn
NNs
Deep learning
NNs with many layers


What is the difference between ML and AI?
AI
computers that "mimic" humans
ML
computers programmed to learn
NNs
Deep learning
NNs with many layers

<u>ML</u>: training computers to detect and characterize features from data

<u>Many-body physics</u>: predict and explain $\underbrace{\text{macroscopic phenomena}}_{\text{(features)}}$

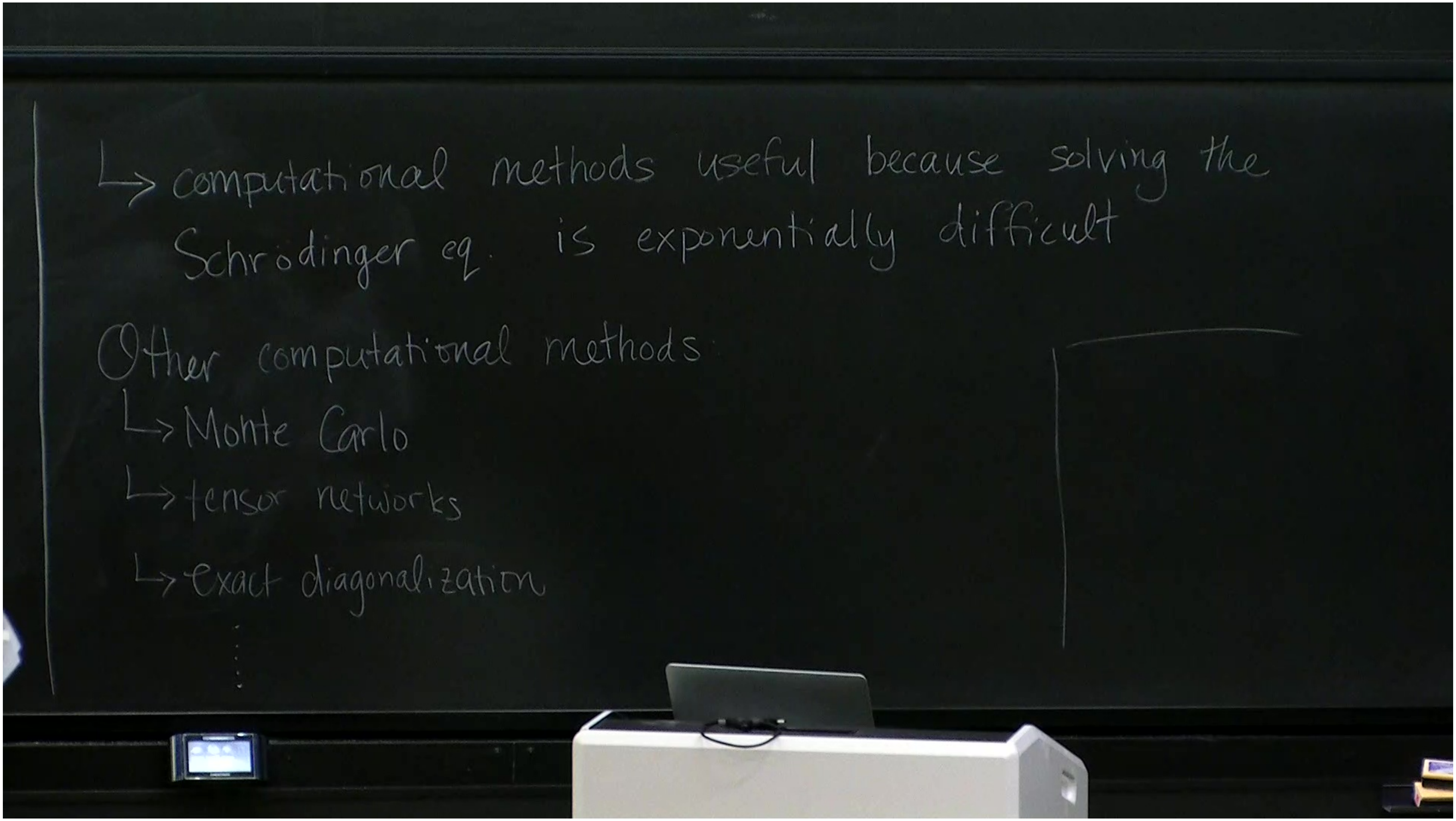
from $\underbrace{\text{microscopic quantities}}_{\text{(data)}}$

Many-body physics:

↳ fundamentally relies on many particles interacting

↳ needed to describe high-T_c superconductors, fractional quantum Hall effect, many qubits interacting

↳ not needed to describe electron conduction in most metals, semiconductors (single-quasiparticle theories exist)


↳ computational methods useful because solving the
Schrödinger eq. is exponentially difficult
Other computational methods:
↳ Monte Carlo
↳ tensor networks
↳ exact diagonalization
⋮

ML is typically more data driven.
Start from physical data, then
learn about properties of the system

Three general categories of ML:

1. Supervised Learning (SL): Dataset $\mathcal{D} = \{(\vec{x}, \vec{y})\}$

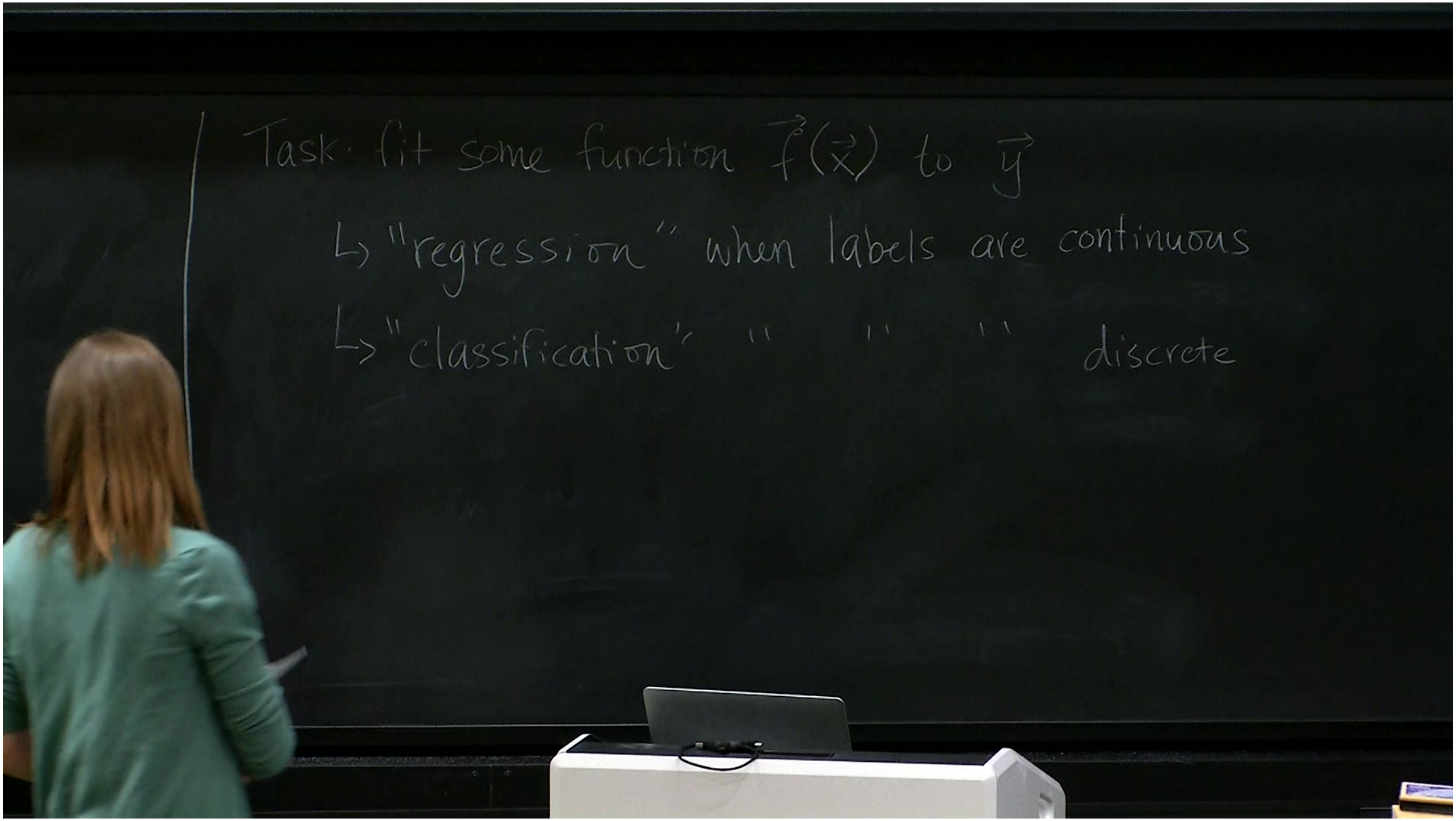
ML is typically more data driven.
Start from physical data, then learn about properties of the system

Three general categories of ML:

1. Supervised Learning (SL): Dataset $\mathcal{D} = \{(\vec{x}, \vec{y})\}$

datapoints $\vec{x} = (x_1, x_2, \ldots, x_{d_x})$

labels $\vec{y} = (y_1, y_2, \ldots, y_{d_y}$


Task: fit some function $\vec{f}(\vec{x})$ to $\vec{y}$
↳ "regression" when labels are continuous
↳ "classification" " " " discrete

datapoints

$\vec{x} = (x_1, x_2, \ldots, x_{d_x})$

$\vec{y} = (y_1, y_2, \ldots, y_{d_y})$

② Unsupervised Learning (UL):

Dataset $\mathcal{D} = \{\vec{x}\}$

Task: extract meaningful features from the dataset to efficiently represent the underlying prob. dist. $P_{data}(\vec{x})$

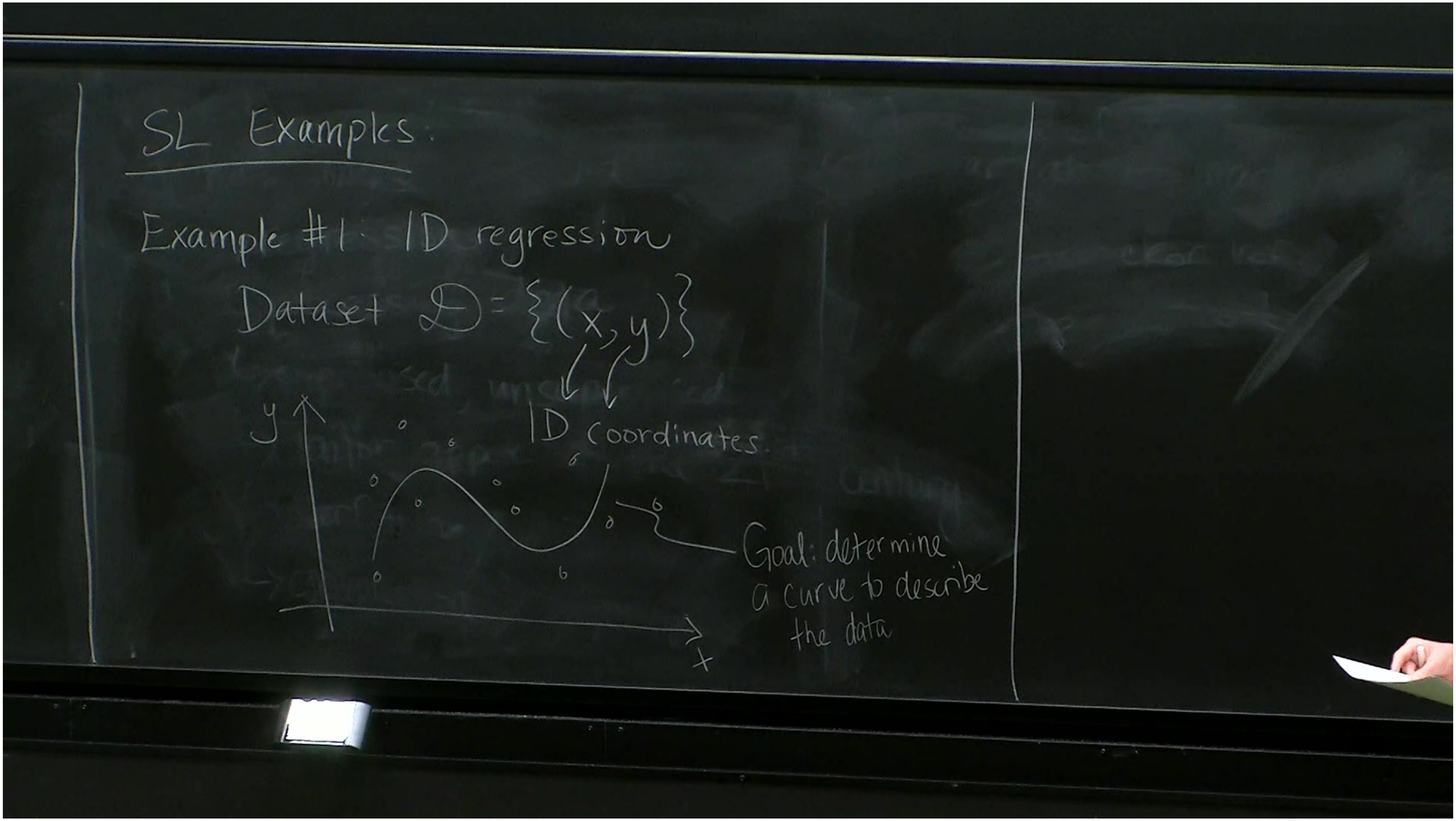
③ Reinforcement Learning (RL) (Not covered in this course)

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

Outline:

- Programming in Python (Lecture #2)
- SL (Lectures #1, #3-8)
- UL (Lectures #9-12)
- Quantum ML (Lecture #13)
- Research in <physics|ML>

classical ML methods


SL Examples:
Example #1: 1D regression
Dataset $\mathcal{D} = \{(x, y)\}$
1D coordinates.
y
x
Goal: determine a curve to describe the data

al: determine
urve to describe
the data

Example #2 Handwritten digits

Classifier

Dataset $\mathcal{D} = \{(\underbrace{\vec{x}}_{\text{image of a handwritten digit}}, \underbrace{y}_{\text{the corresponding number}})\}$

② Unsu

Datas

Task:

the d

underl

al: determine
curve to describe
the data

Example #2 Handwritten digits

Classifier

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

$\vec{x}$: image of a handwritten digit

y: the corresponding number

$\vec{x} = \boxed{9}$, $y = 9$

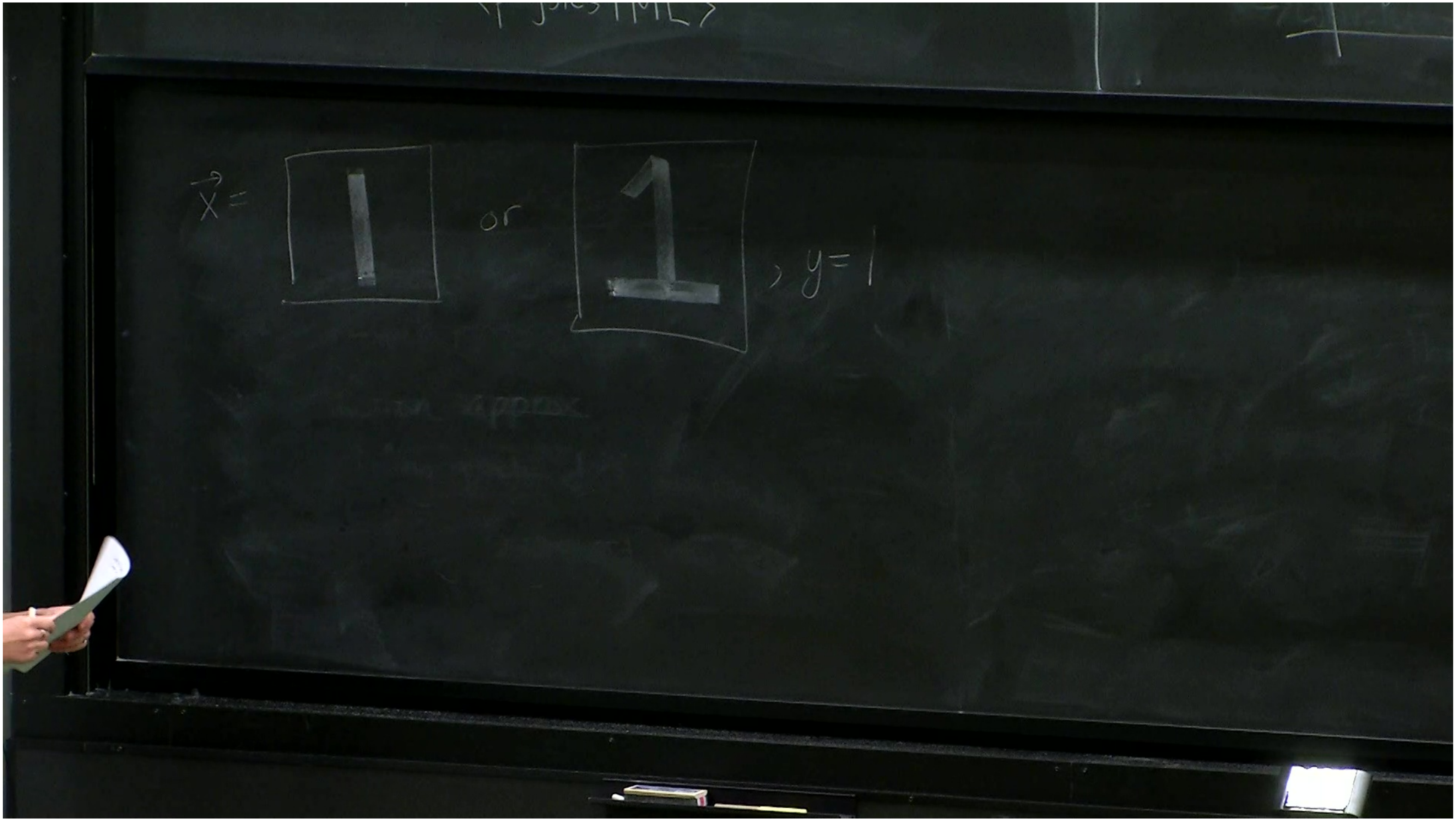
2 Unsu

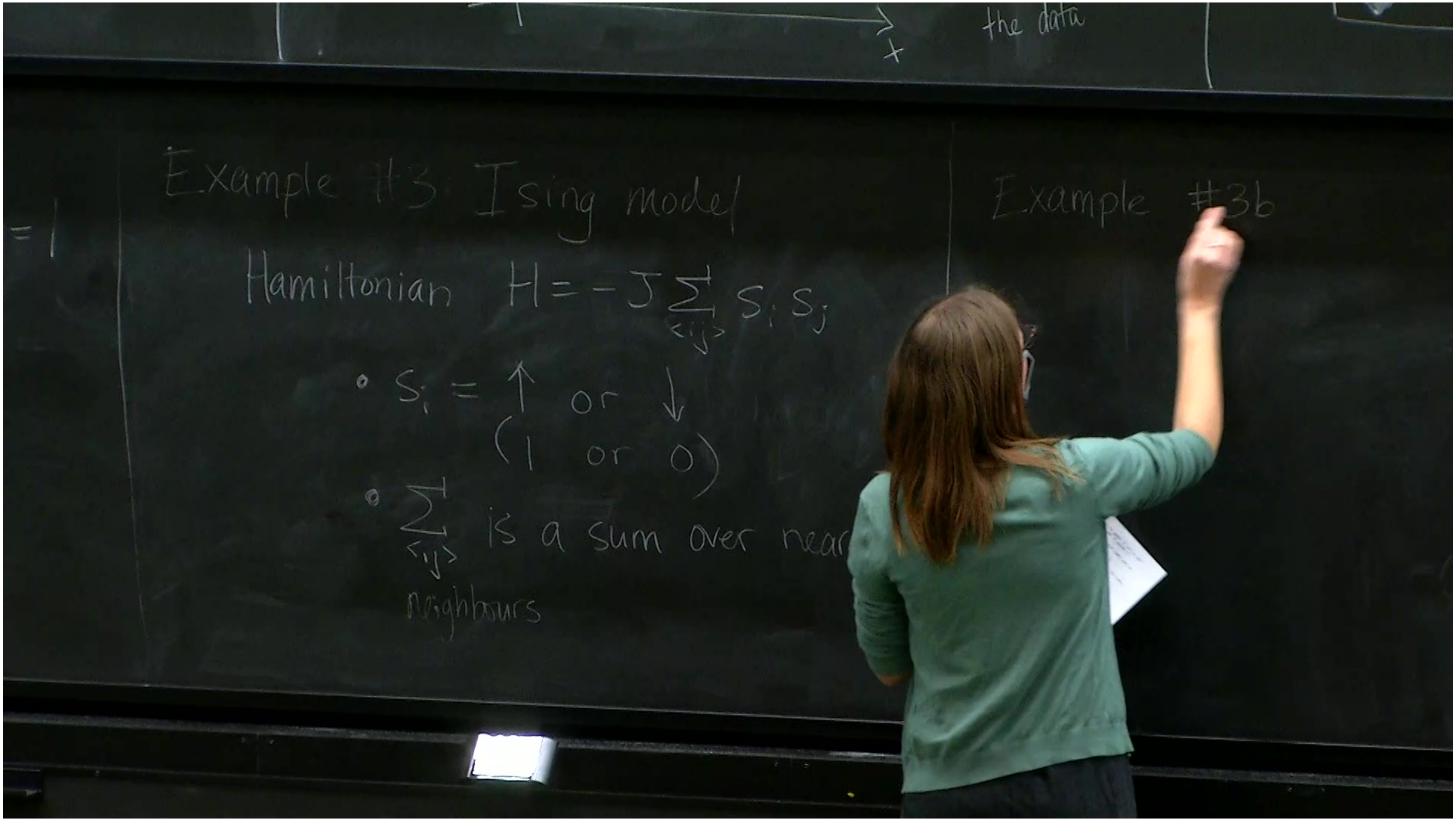
Datas

Task:

the d

underl


x =
1
or
1
, y = 1


the data
Example #3 Ising model
Hamiltonian $H = -J \sum_{\langle ij \rangle} s_i s_j$
• s_i = ↑ or ↓
(1 or 0)
• $\sum_{\langle ij \rangle}$ is a sum over near
neighbours
Example #3b

el

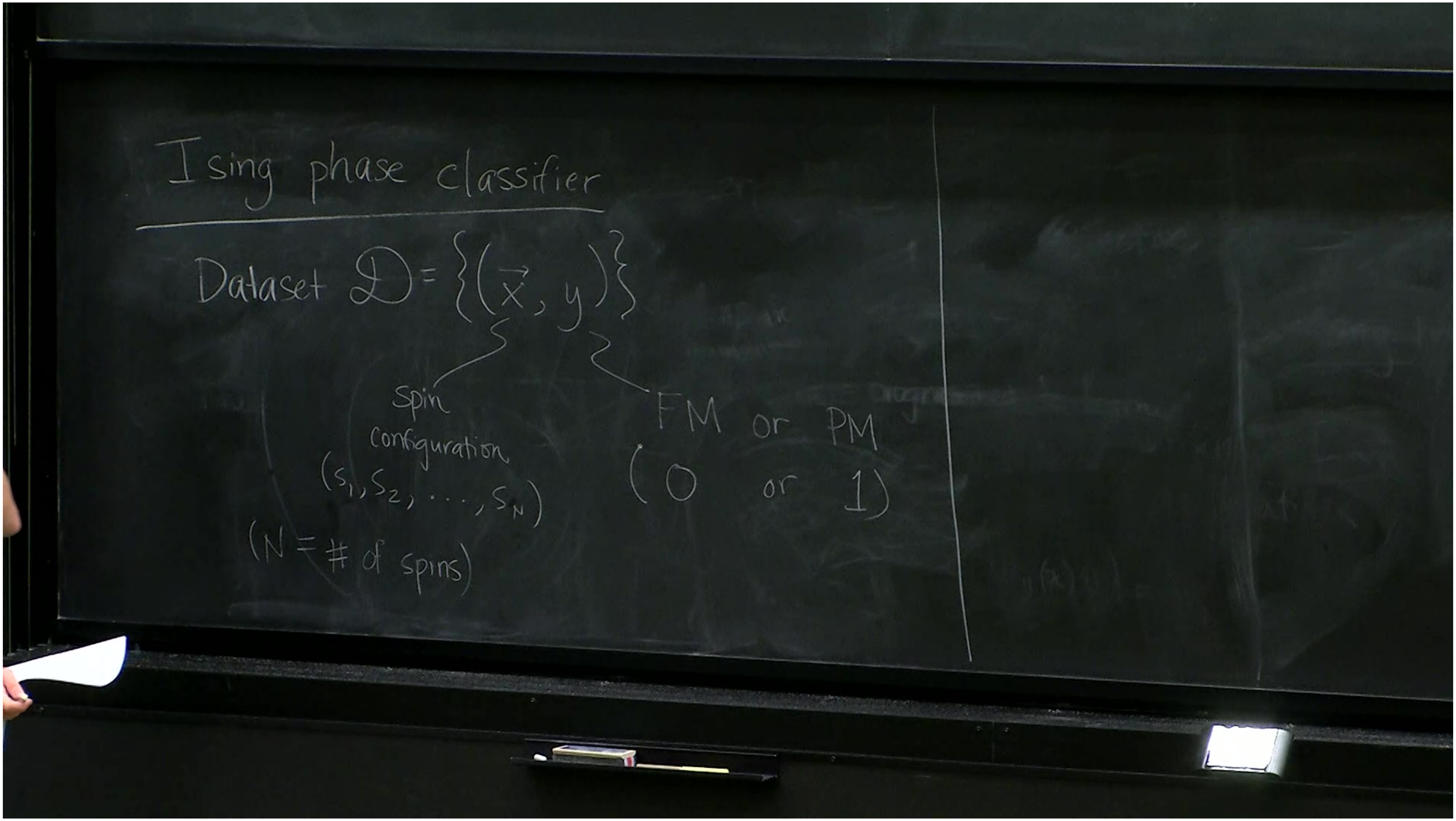
$\sum_{\langle ij \rangle} S_i S_j$

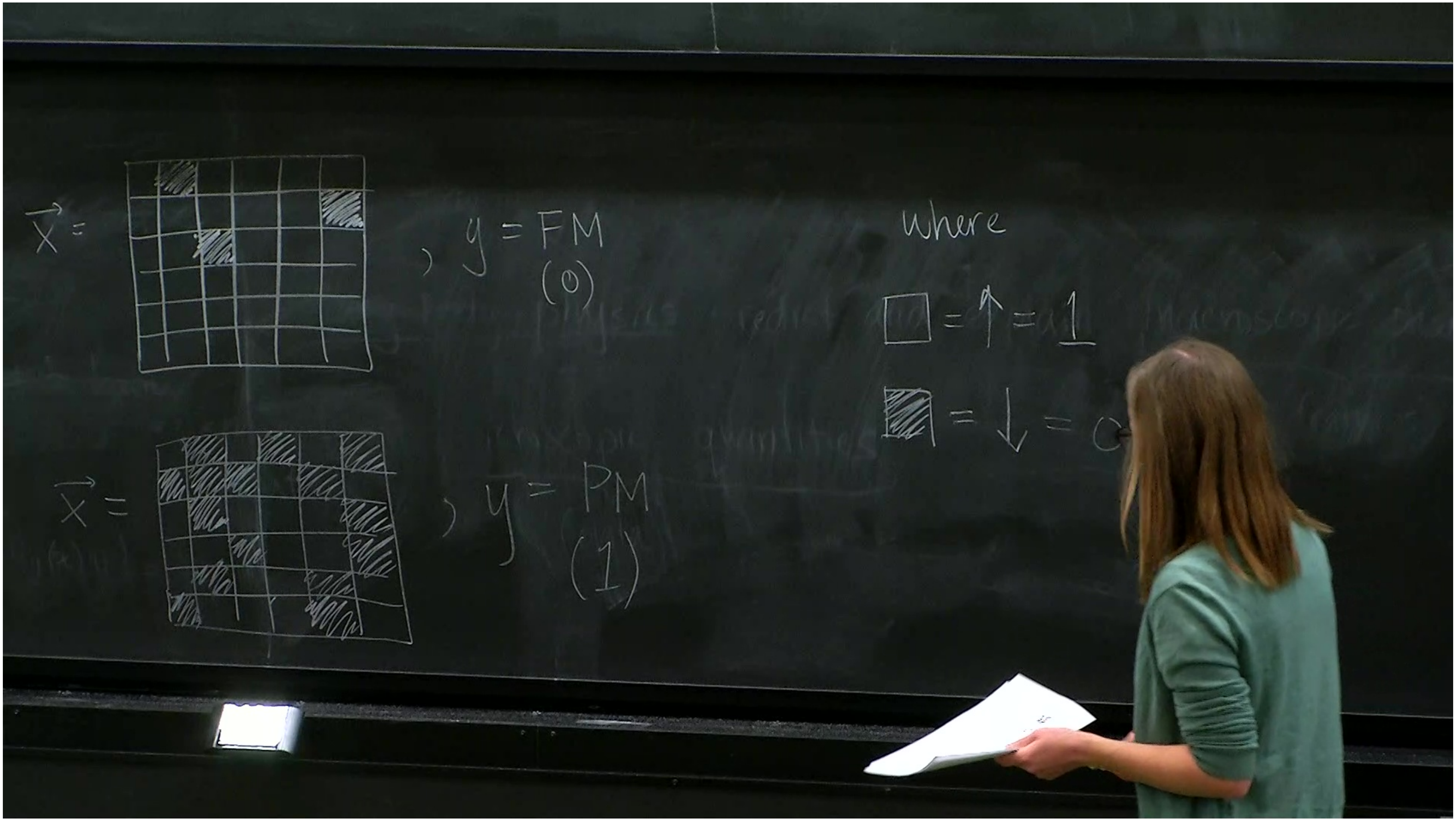
over nearest

Example #3b:

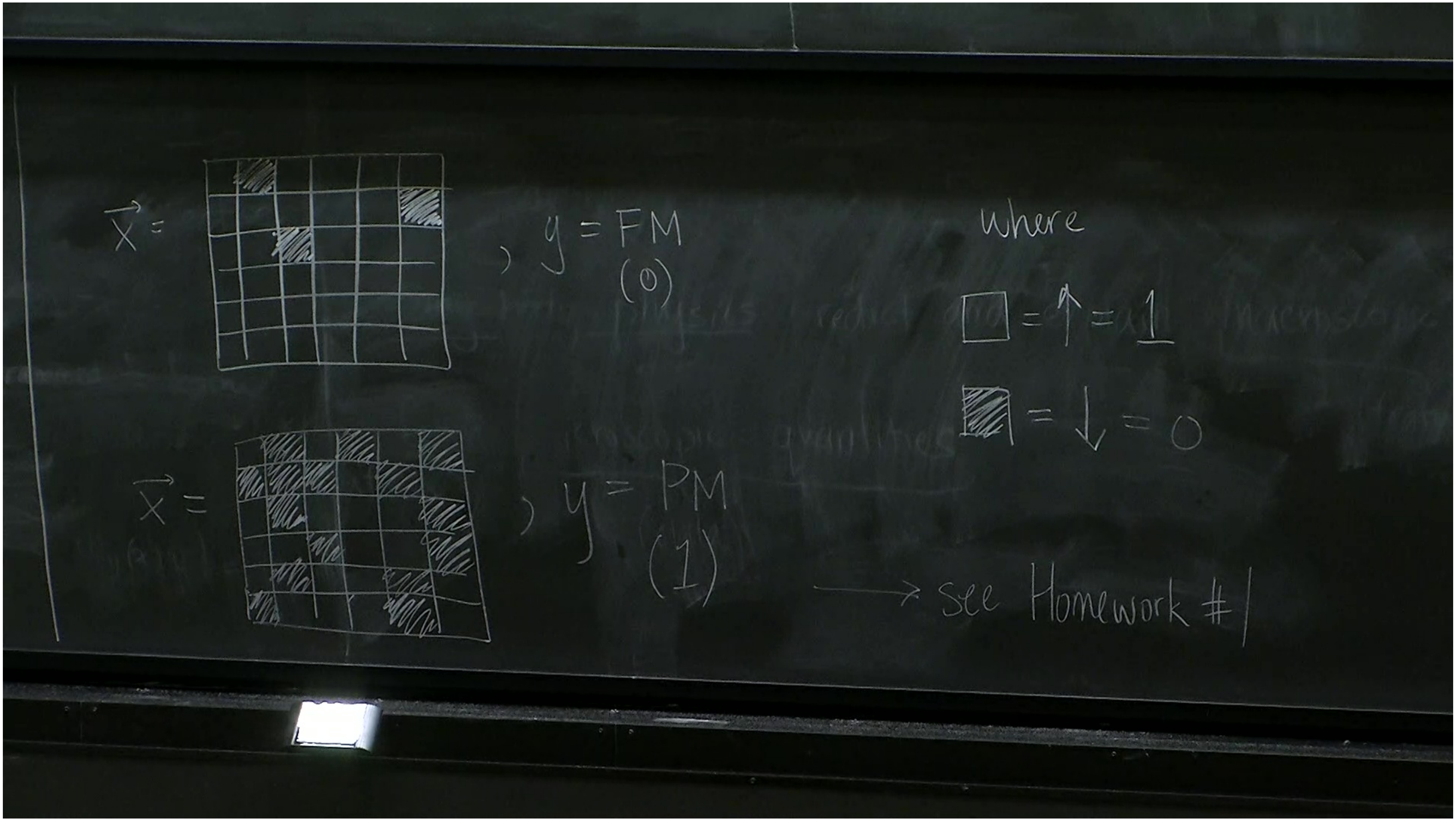
Recall: In 2D, this model undergoes a phase transition when $T/J = T_c/J \approx 2.269$

- For $T < T_c$: ferromagnetic (FM)
- For $T > T_c$: paramagnetic (PM)


Ising phase classifier
Dataset $\mathcal{D} = \{(\vec{x}, y)\}$
spin configuration
$(s_1, s_2, \ldots, s_N)$
(N = # of spins)
FM or PM
(0 or 1)



x =
, y = FM
(0)
x =
, y = PM
(1)
where
= ↑ = 1
= ↓


x⃗ =
, y = FM
(0)
where
= ↑ = 1
= ↓ = 0
x⃗ =
, y = PM
(1)
→ see Homework #1