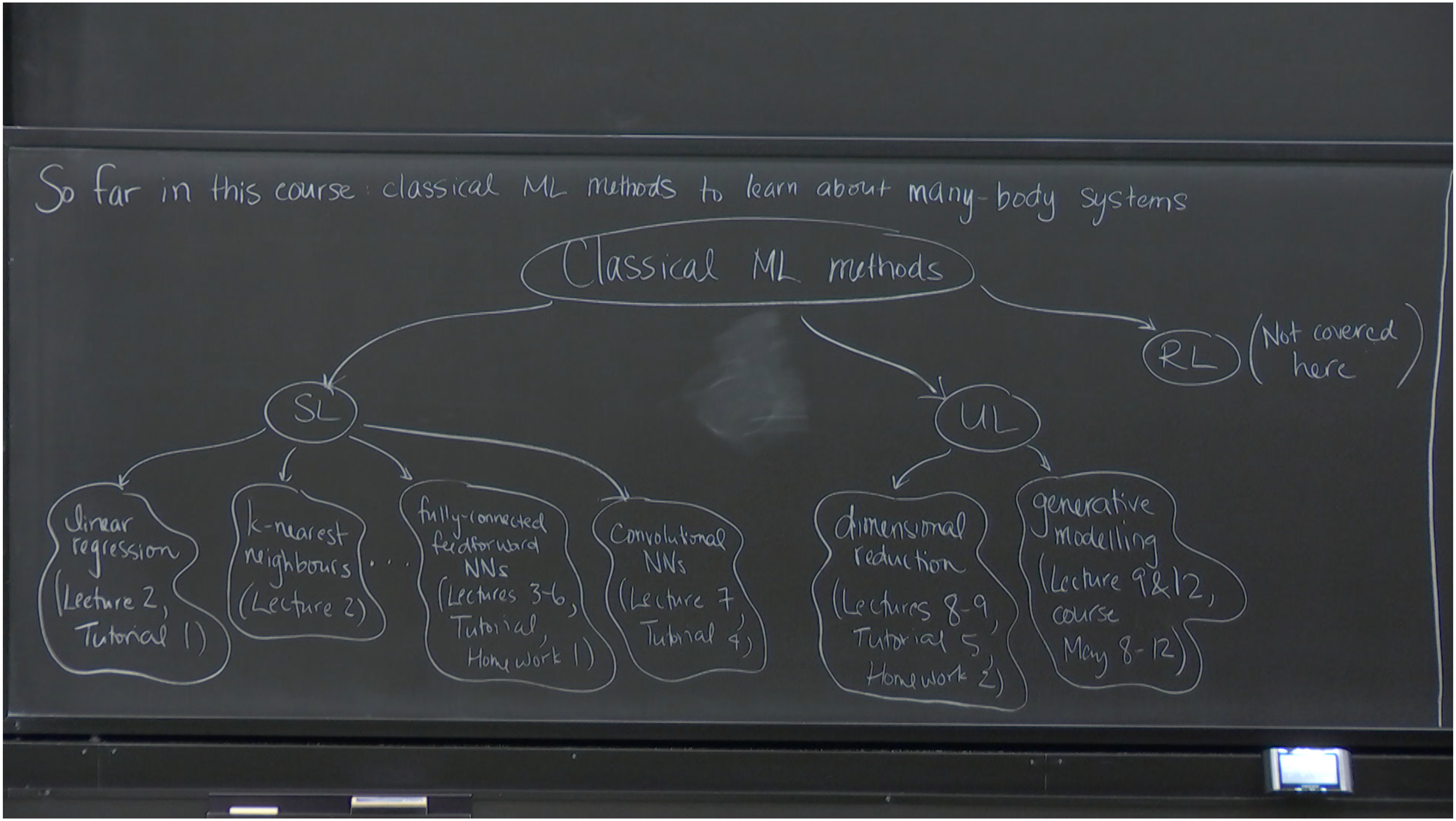
Title: Machine Learning Lecture - 230323

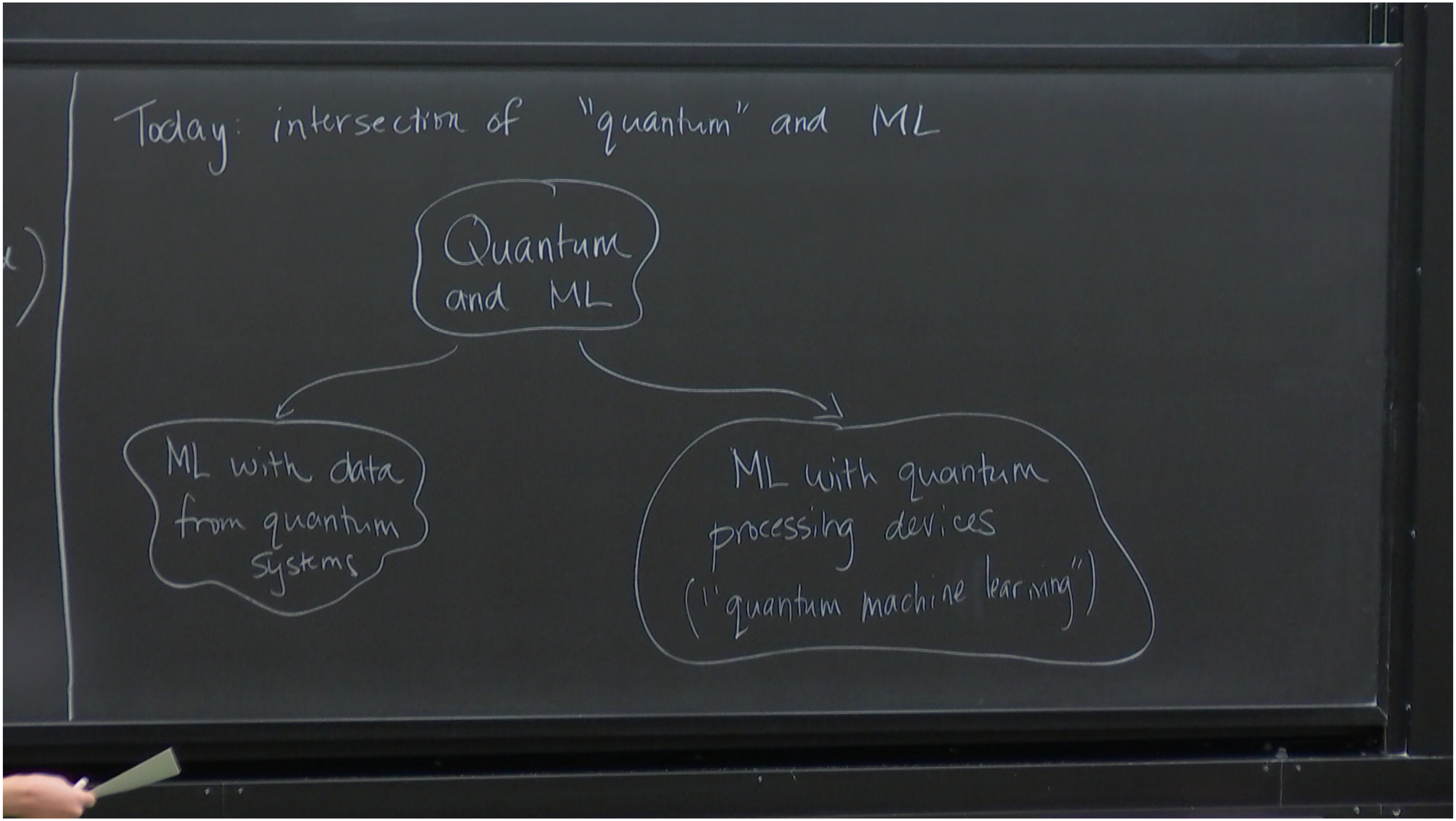
Speakers: Lauren Hayward

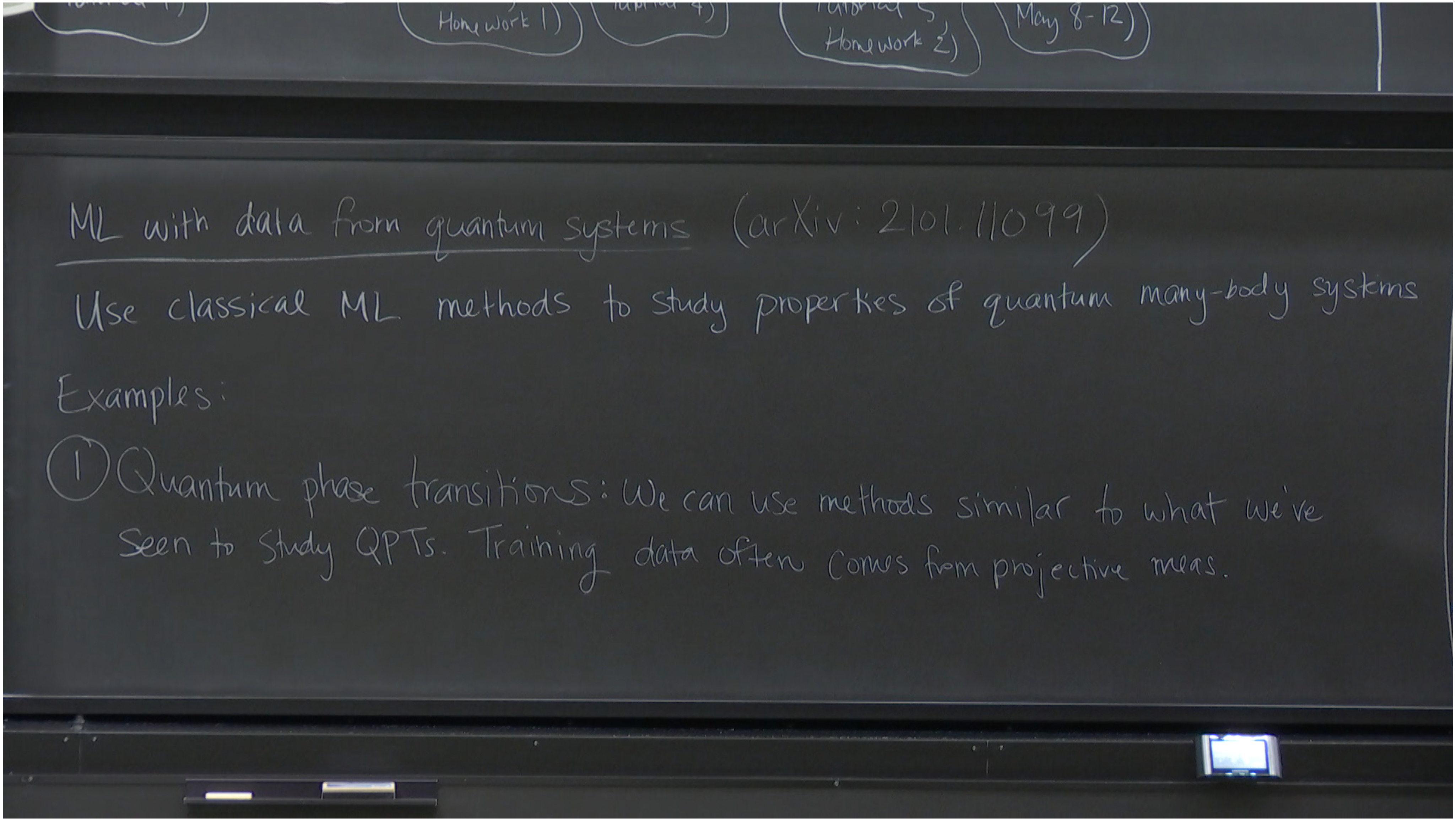
Collection: Machine Learning for Many-Body Physics (2022/2023)

Date: March 23, 2023 - 9:00 AM

URL: https://pirsa.org/23030035


So far in this course: classical ML methods to learn about many-body systems
Classical ML methods
SL
UL
RL
(Not covered here)
linear regression (Lecture 2, Tutorial 1)
k-nearest neighbours (Lecture 2)
. . .
fully-connected feedforward NNs (Lectures 3-6, Tutorial, Homework 1)
Convolutional NNs (Lecture 7, Tutorial 4)
dimensional reduction (Lectures 8-9, Tutorial 5, Homework 2)
generative modelling (Lecture 9 & 12, course May 8-12)


Today: intersection of "quantum" and ML
Quantum and ML
ML with data from quantum systems
ML with quantum processing devices ("quantum machine learning")


Homework 1)
Homework 2)
May 8-12)
ML with data from quantum systems (arXiv: 2101.11099)
Use classical ML methods to study properties of quantum many-body systems
Examples:
① Quantum phase transitions: We can use methods similar to what we've
seen to study QPTs. Training data often comes from projective meas.

stems

② Quantum state tomography

Goal: Given (limited) meas. data, reconstruct the corresponding (unknown) quantum state

Generative modelling with NN architectures (eg: RBMs) can be used to produce an efficient NN repr. of the q. state

(3) Finding approx. ground states

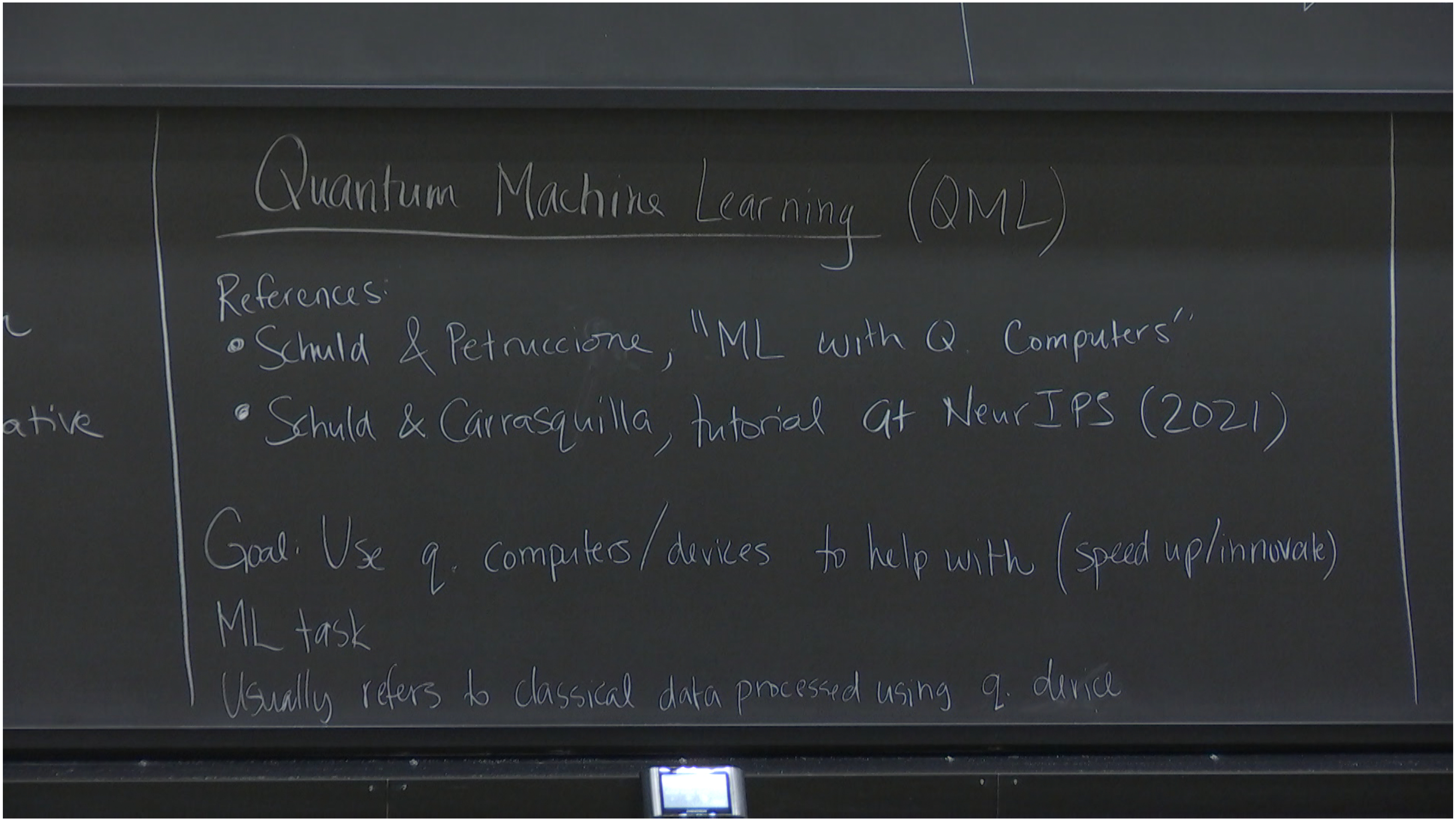
Given a Hamiltonian of quantum many-body system, train a generative model to approx. the g.s.

(eg: Mohamed's work w̄ RNNs)

Quantum

References:

- Schuld & Pet


Quantum Machine Learning (QML)
References:
• Schuld & Petruccione, "ML with Q. Computers"
• Schuld & Carrasquilla, tutorial at NeurIPS (2021)
Goal: Use q. computers/devices to help with (speed up/innovate) ML task
Usually refers to classical data processed using q. device

ML)

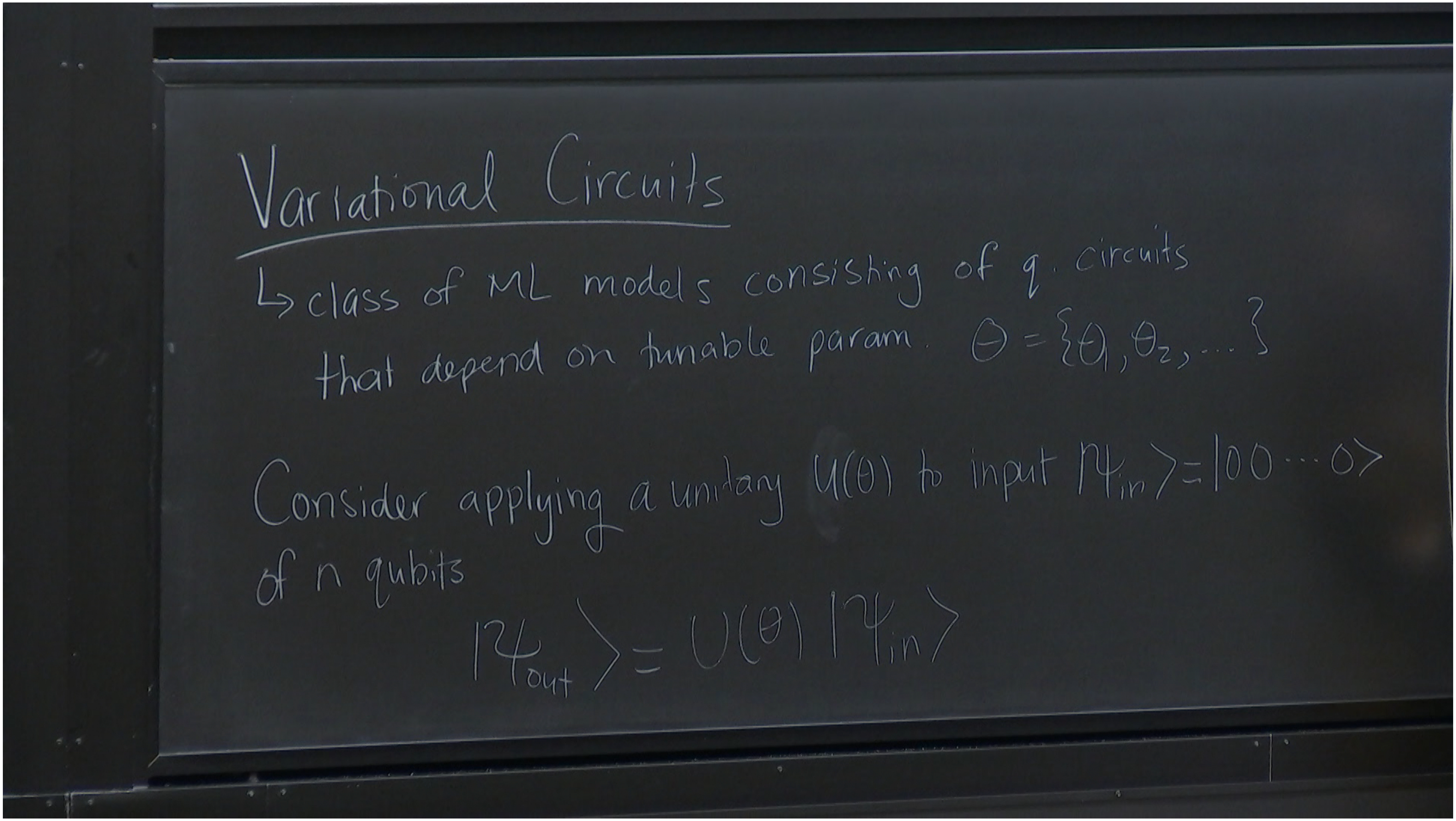
Computers"

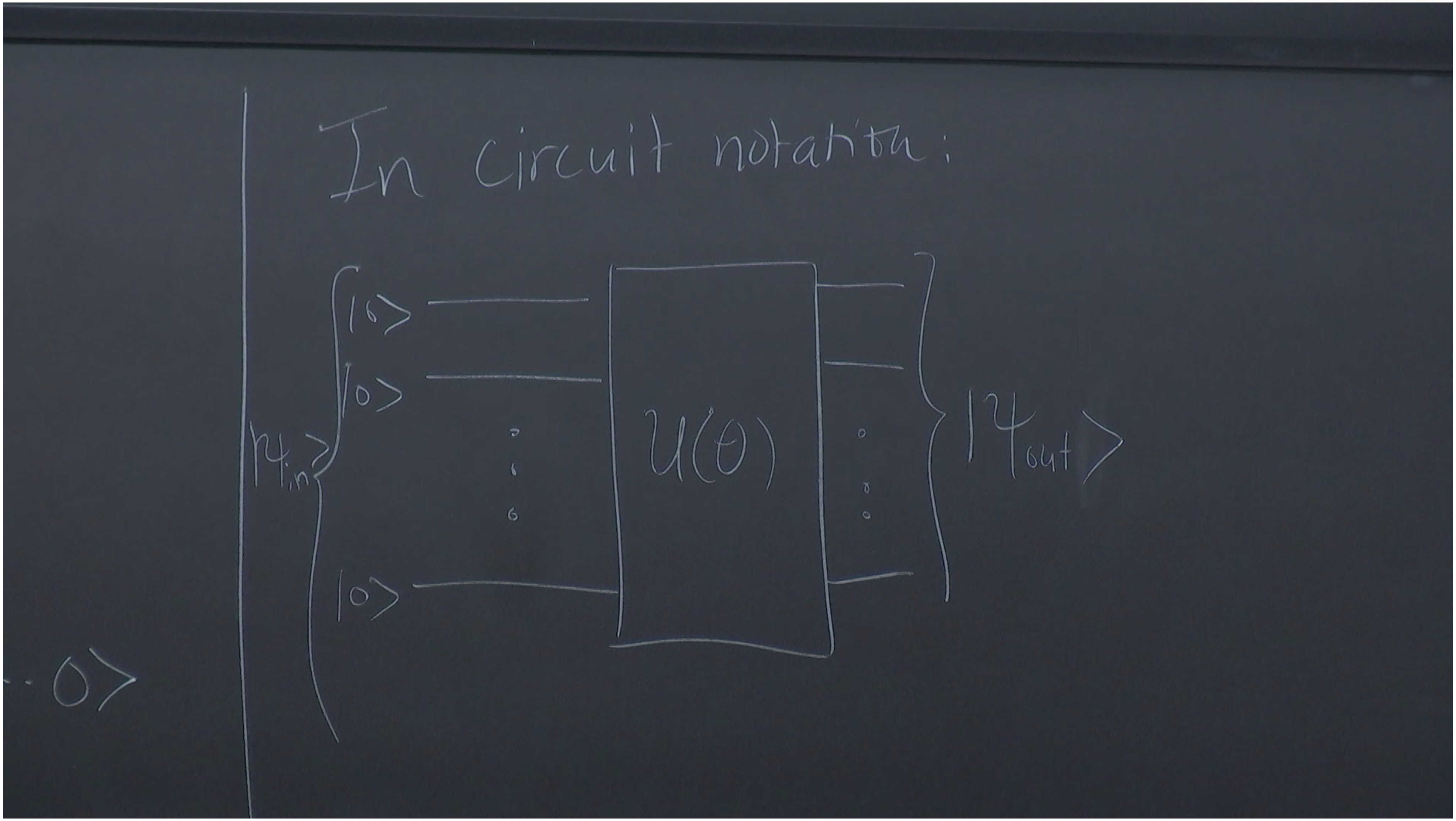
NeurIPS (2021)

with (speed up/innovate)

q. device

Here, we will focus on the goal of using q. devices (including "near-term" devices) as ML models known as "variational circuits"


Variational Circuits
↳ class of ML models consisting of q. circuits
that depend on tunable param. $\Theta = \{\theta_1, \theta_2, \dots\}$
Consider applying a unitary $U(\theta)$ to input $|\psi_{in}\rangle = |00\cdots0\rangle$
of n qubits
$|\psi_{out}\rangle = U(\theta)|\psi_{in}\rangle$

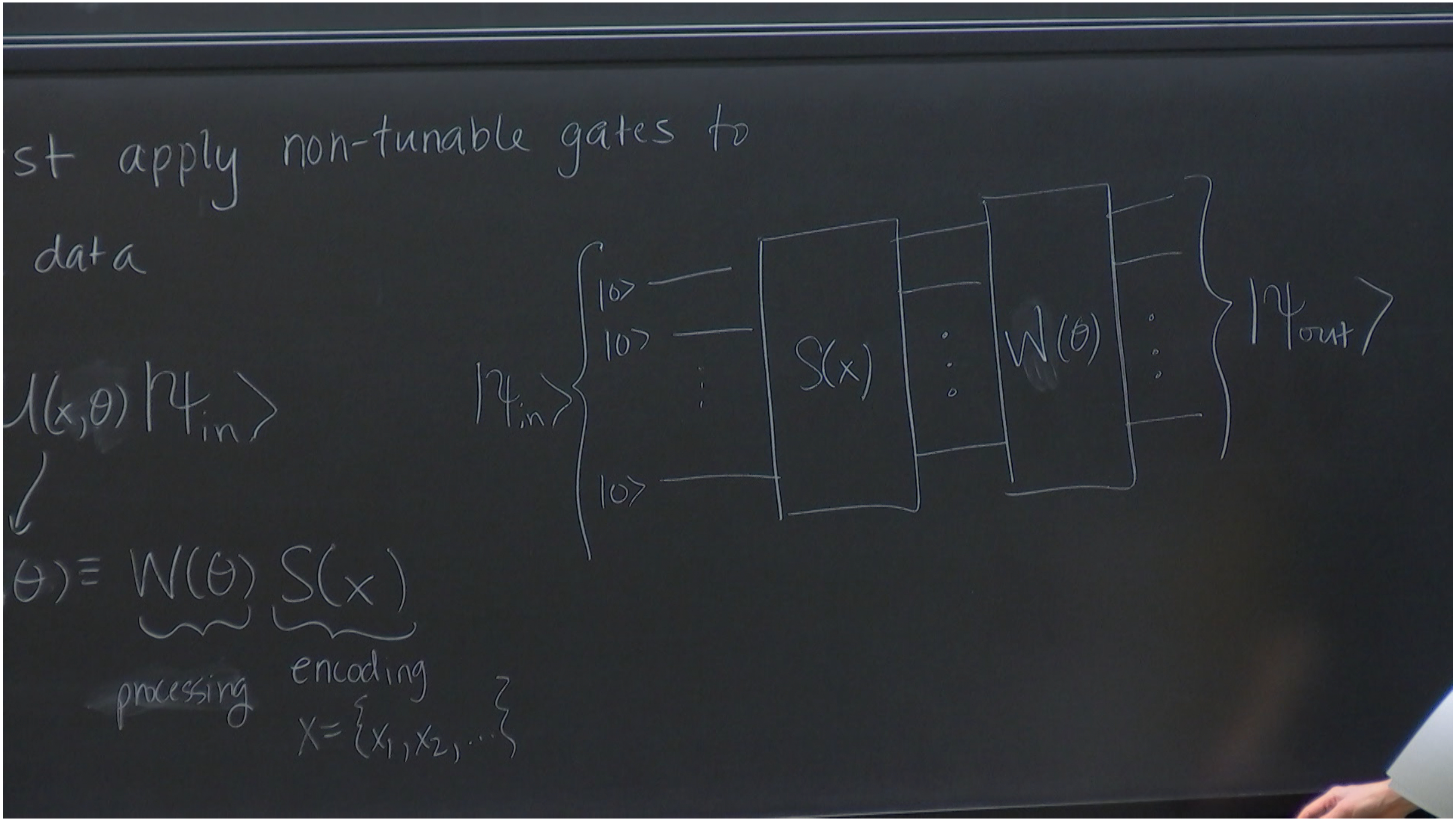

In circuit notation:
|0⟩
|0⟩
|0⟩
$U(\theta)$
$|\psi_{in}\rangle$
$|\psi_{out}\rangle$
0⟩

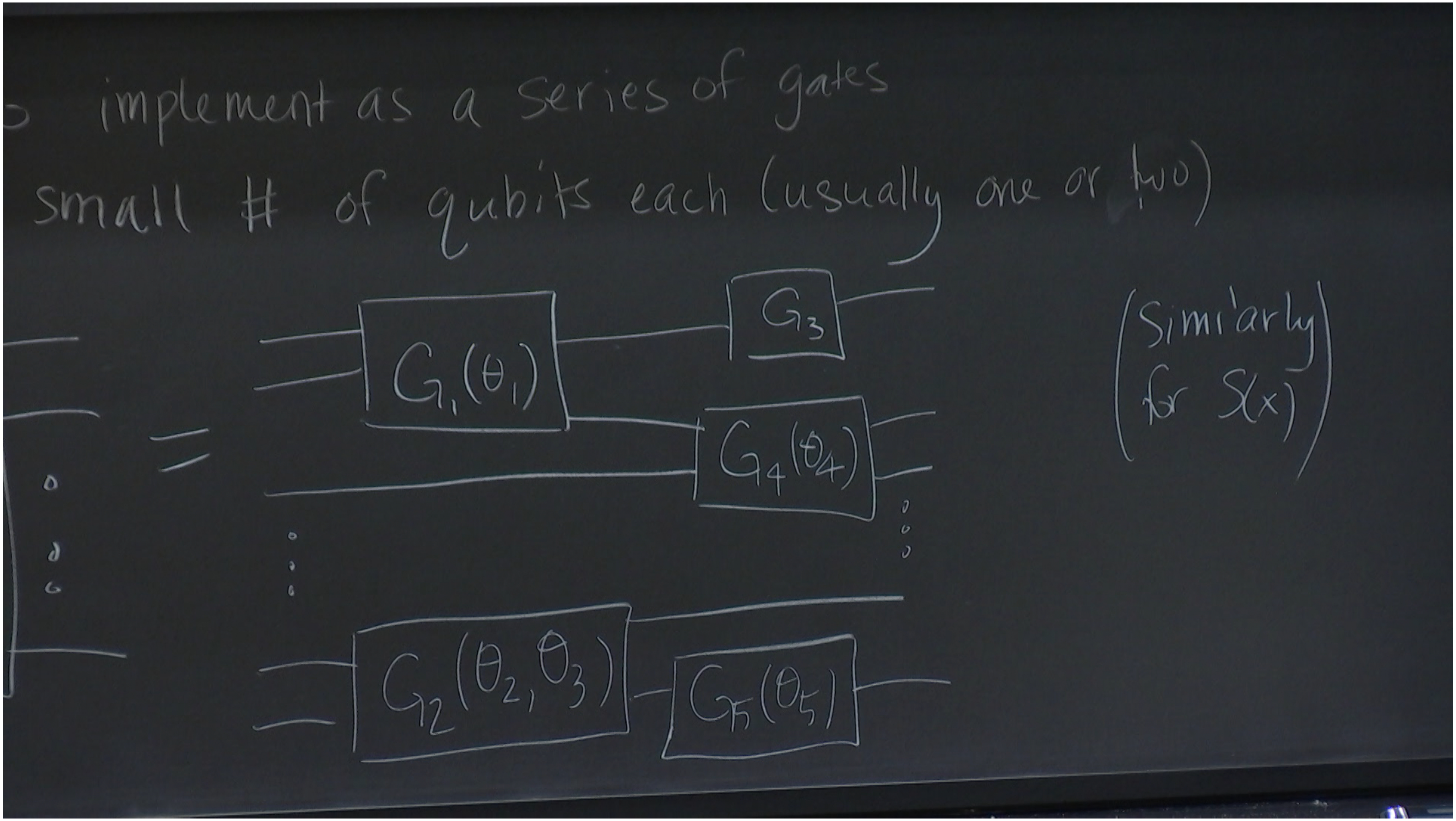
We can first apply non-tunable gates to encode the data

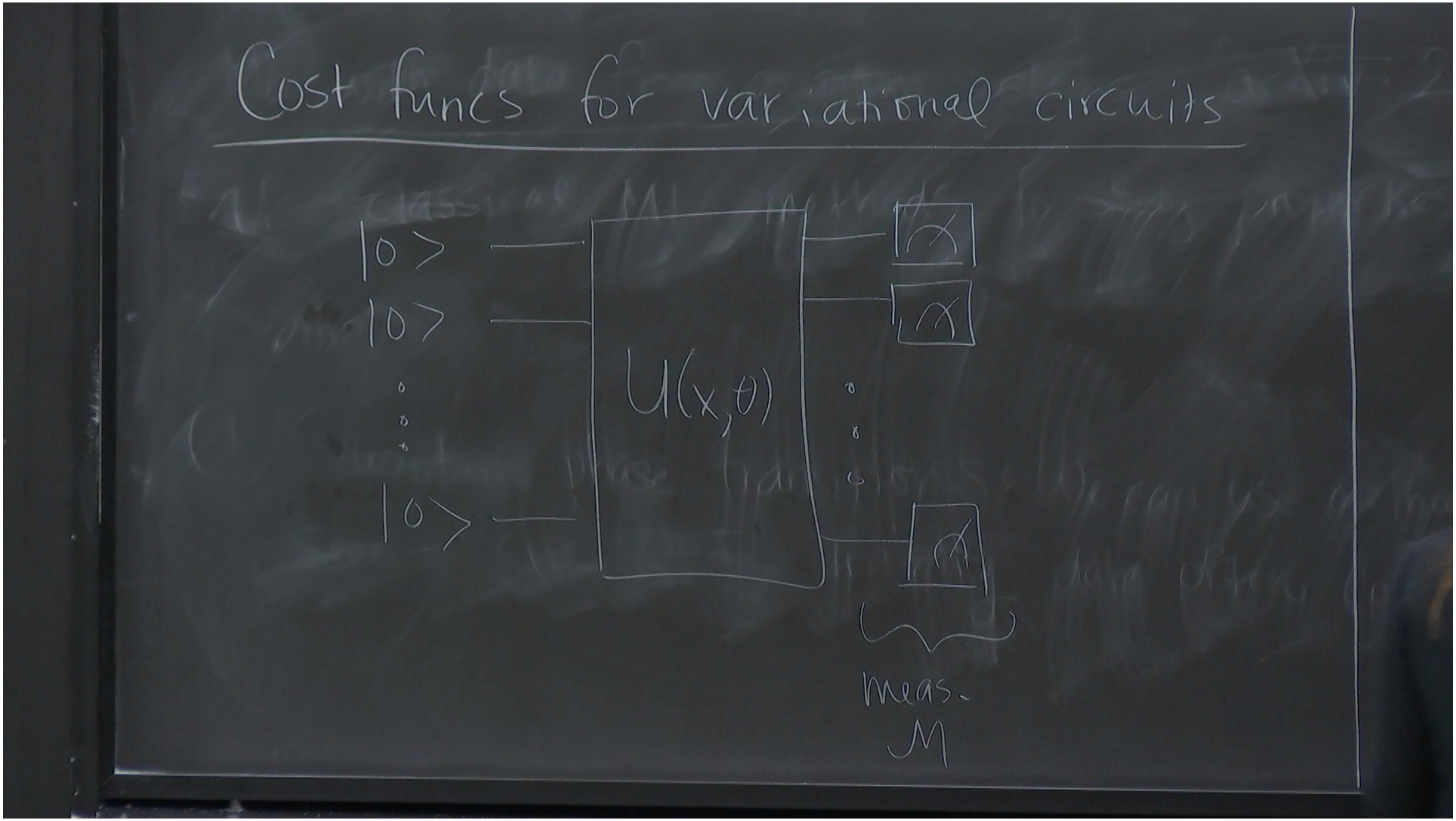
$$|\psi_{out}\rangle = U(x,\theta)|\psi_{in}\rangle$$

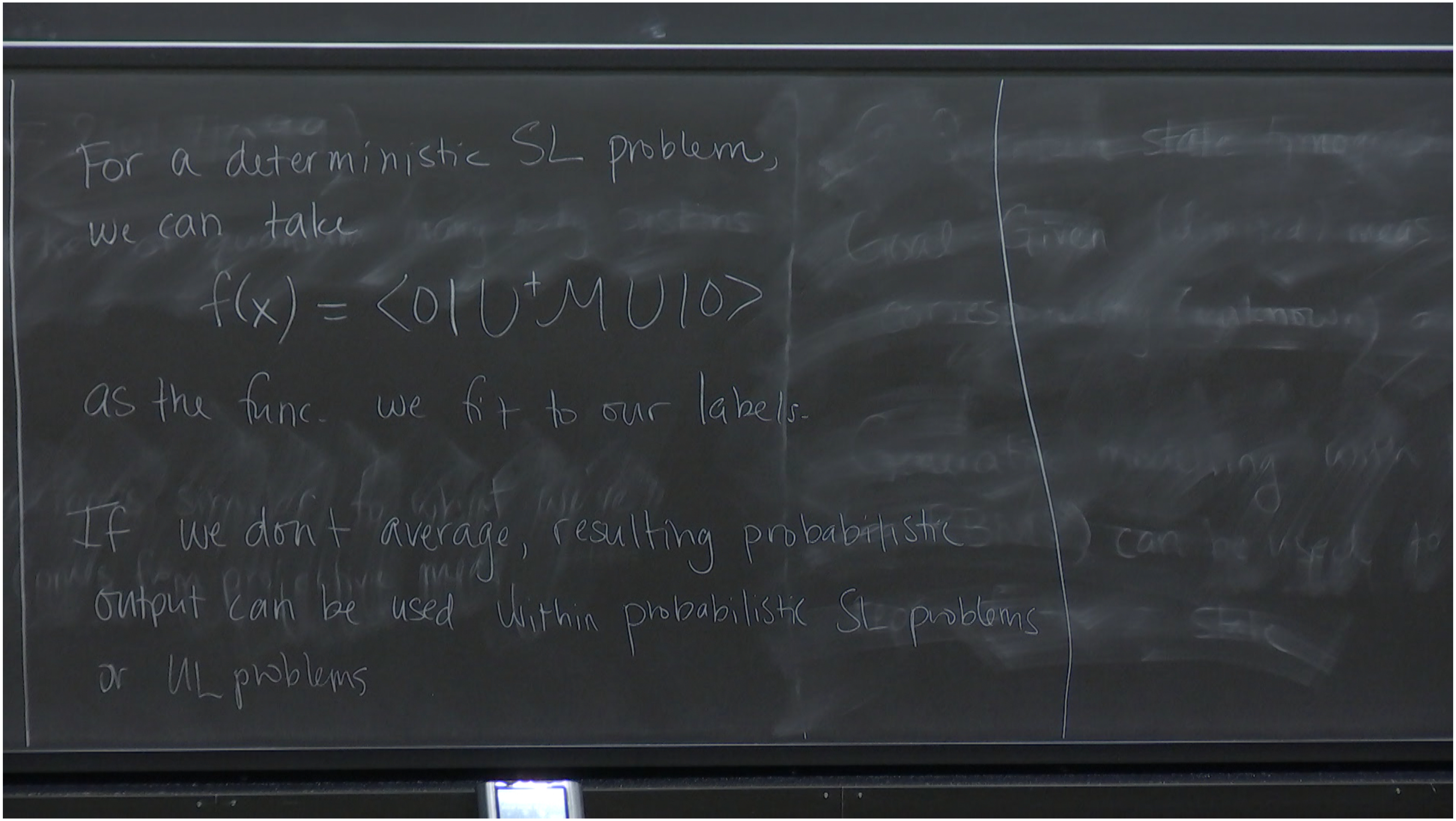
$$U(x,\theta) = W(\theta)\underbrace{S(x)}_{\text{encoding}}$$

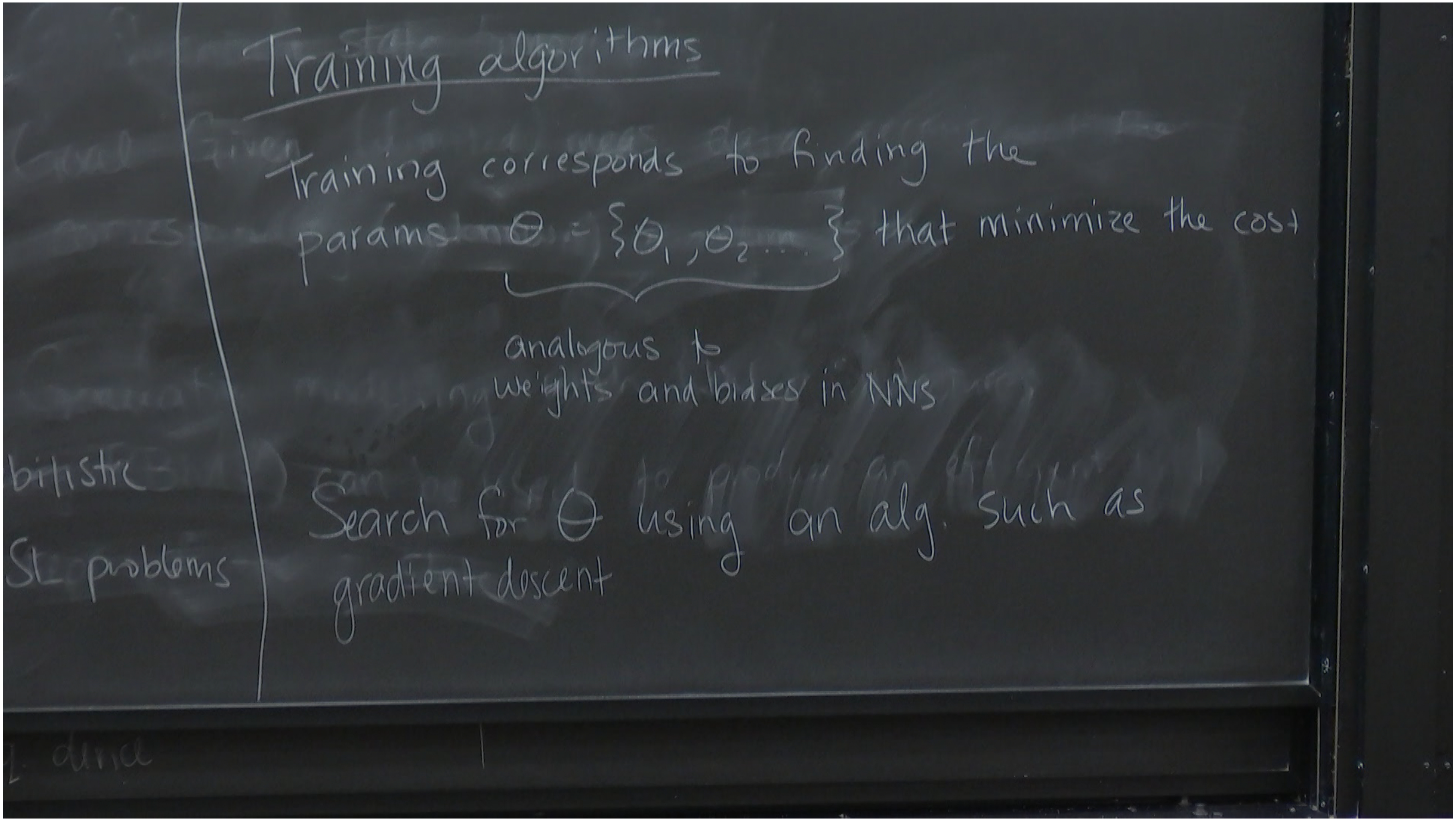
$x = \{x_1, x_2, \ldots\}$


st apply non-tunable gates to
data
U(x,θ)|ψin>
(θ) ≡ W(θ) S(x)
processing
encoding
x={x1,x2,...}
|ψin>
|0>
|0>
|0>
S(x)
W(θ)
|ψout>


implement as a series of gates
small # of qubits each (usually one or two)
$G_1(\theta_1)$
G_3
$G_4(\theta_4)$
$G_2(\theta_2, \theta_3)$
$G_5(\theta_5)$
(Similarly for S(x))


Cost funcs for variational circuits
|0>
|0>
|0>
U(x,θ)
meas.
M

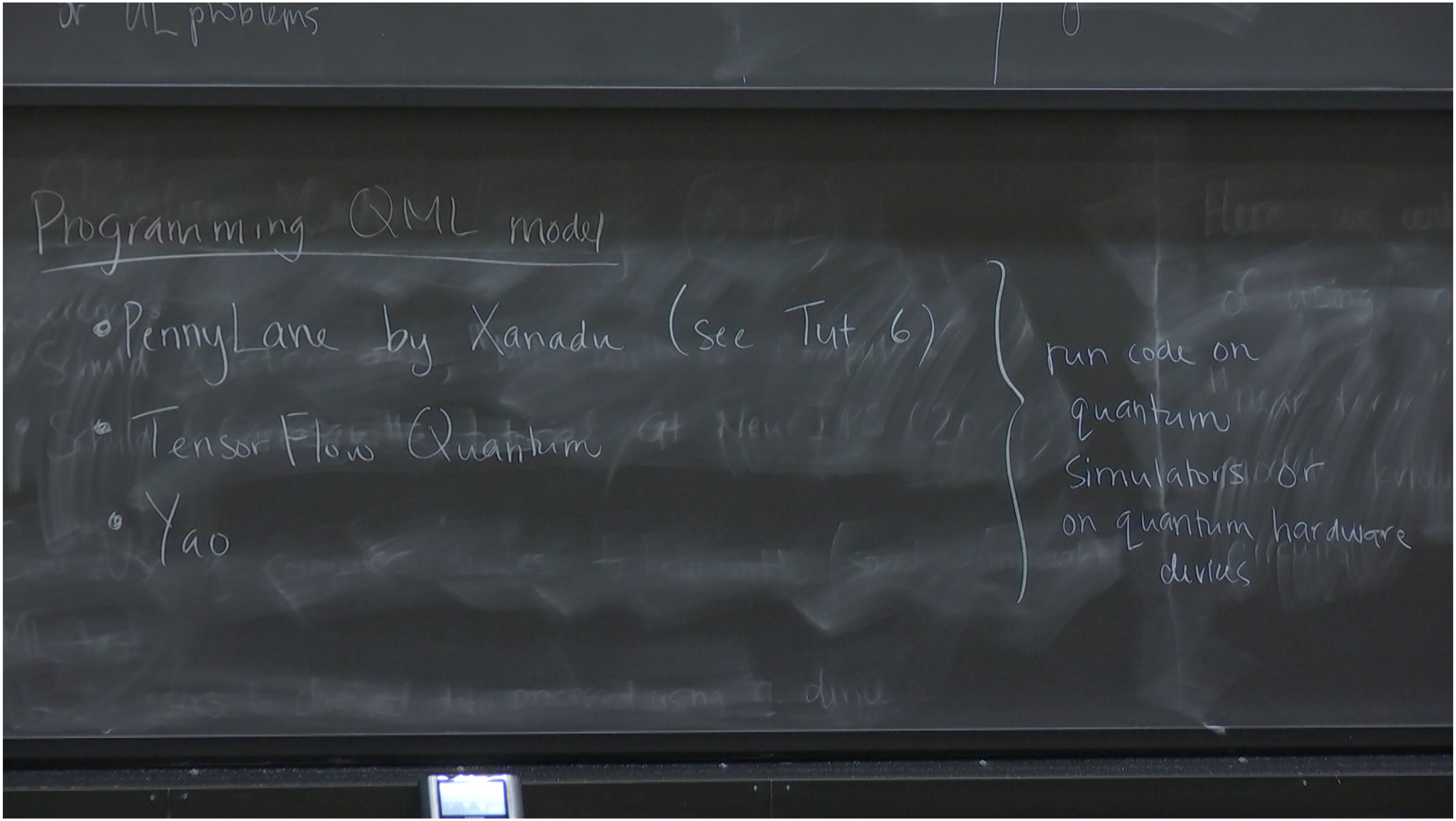

For a deterministic SL problem,
we can take
$f(x) = \langle 0 | U^\dagger M U | 0 \rangle$
as the func. we fit to our labels.
If we don't average, resulting probabilistic
output can be used within probabilistic SL problems
or UL problems

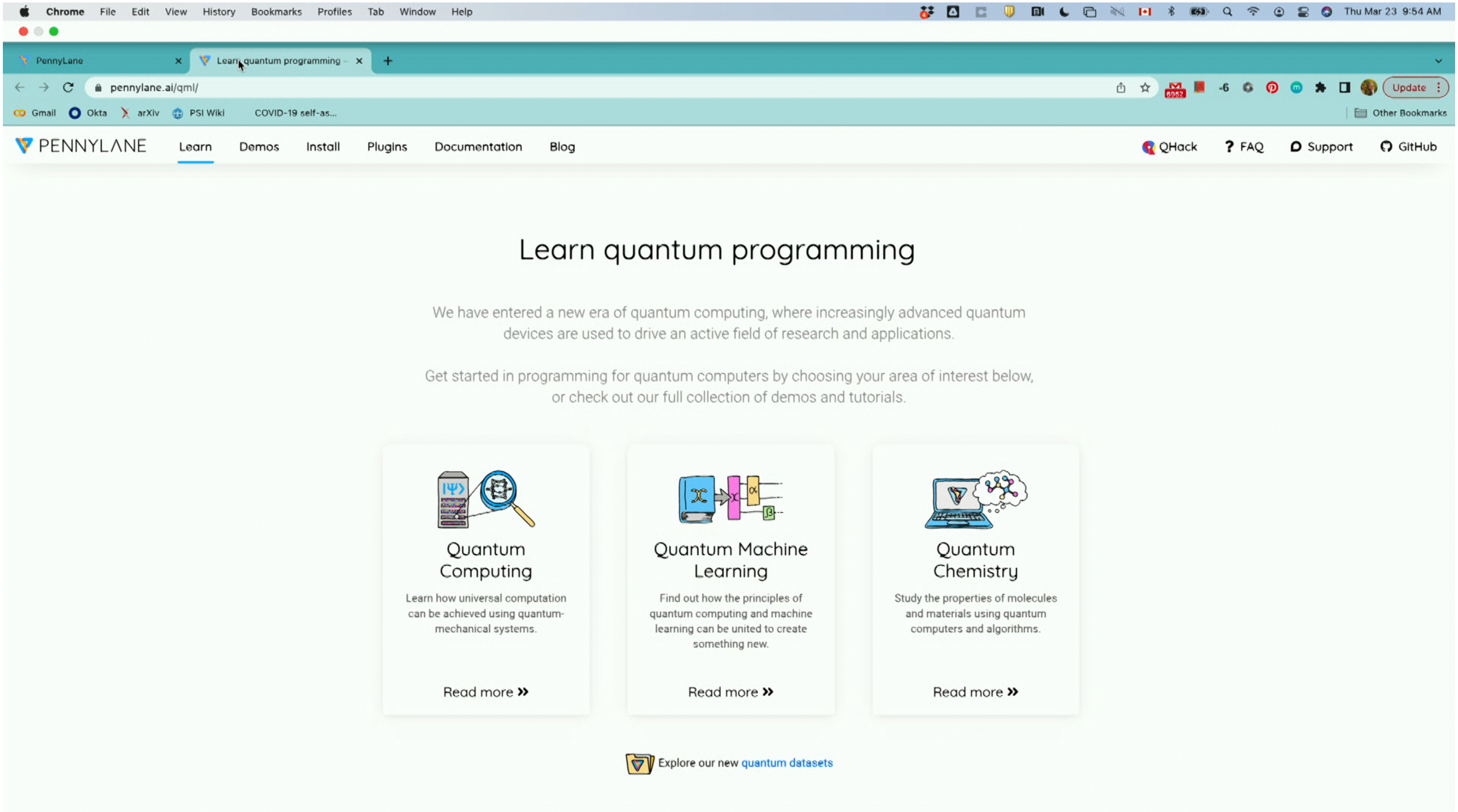

Training algorithms
Training corresponds to finding the
params Θ = {Θ1, Θ2 ...} that minimize the cost
analogous to
weights and biases in NNs
Search for Θ using an alg. such as
gradient descent
SL problems

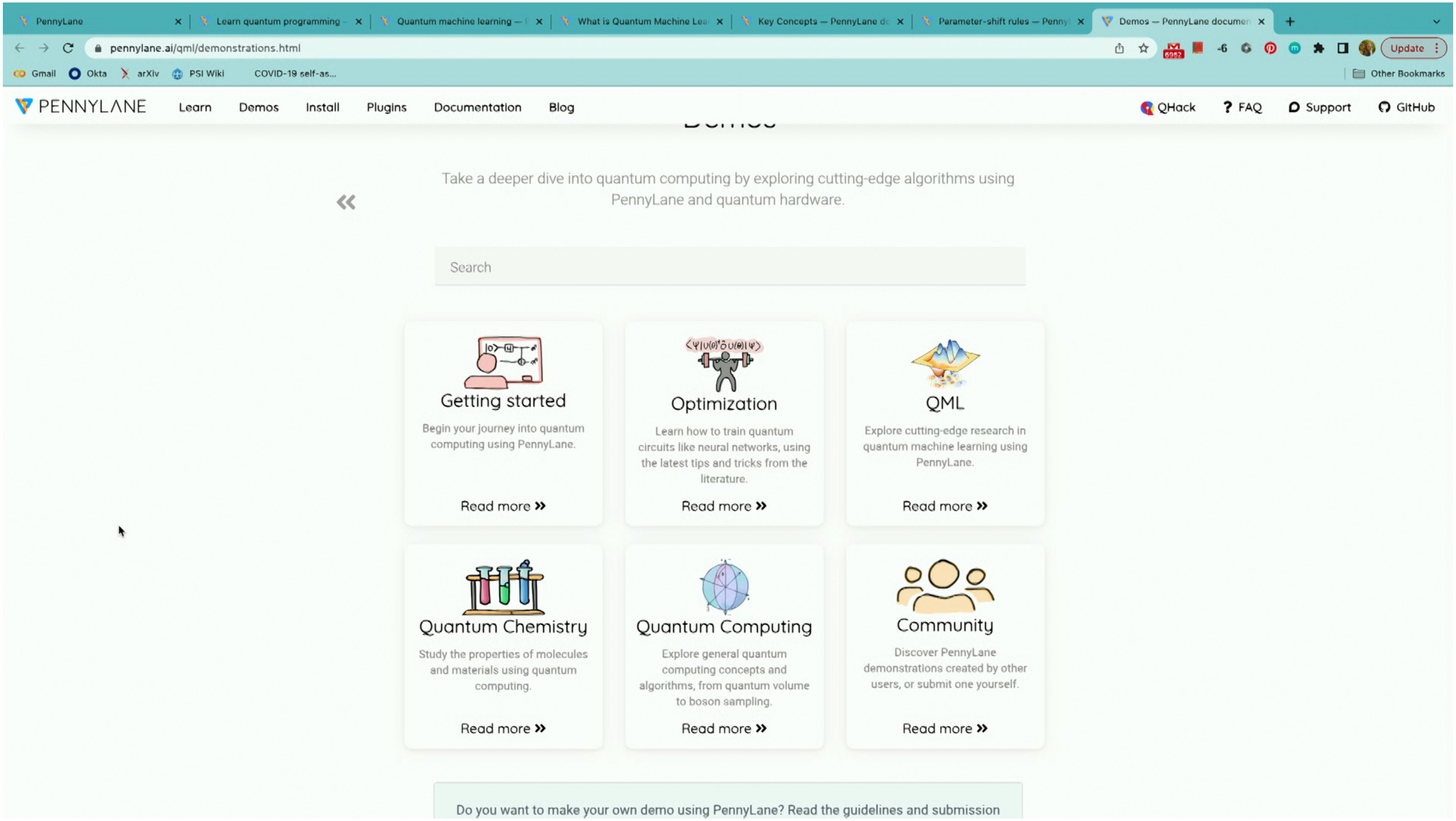
To compute the gradients:

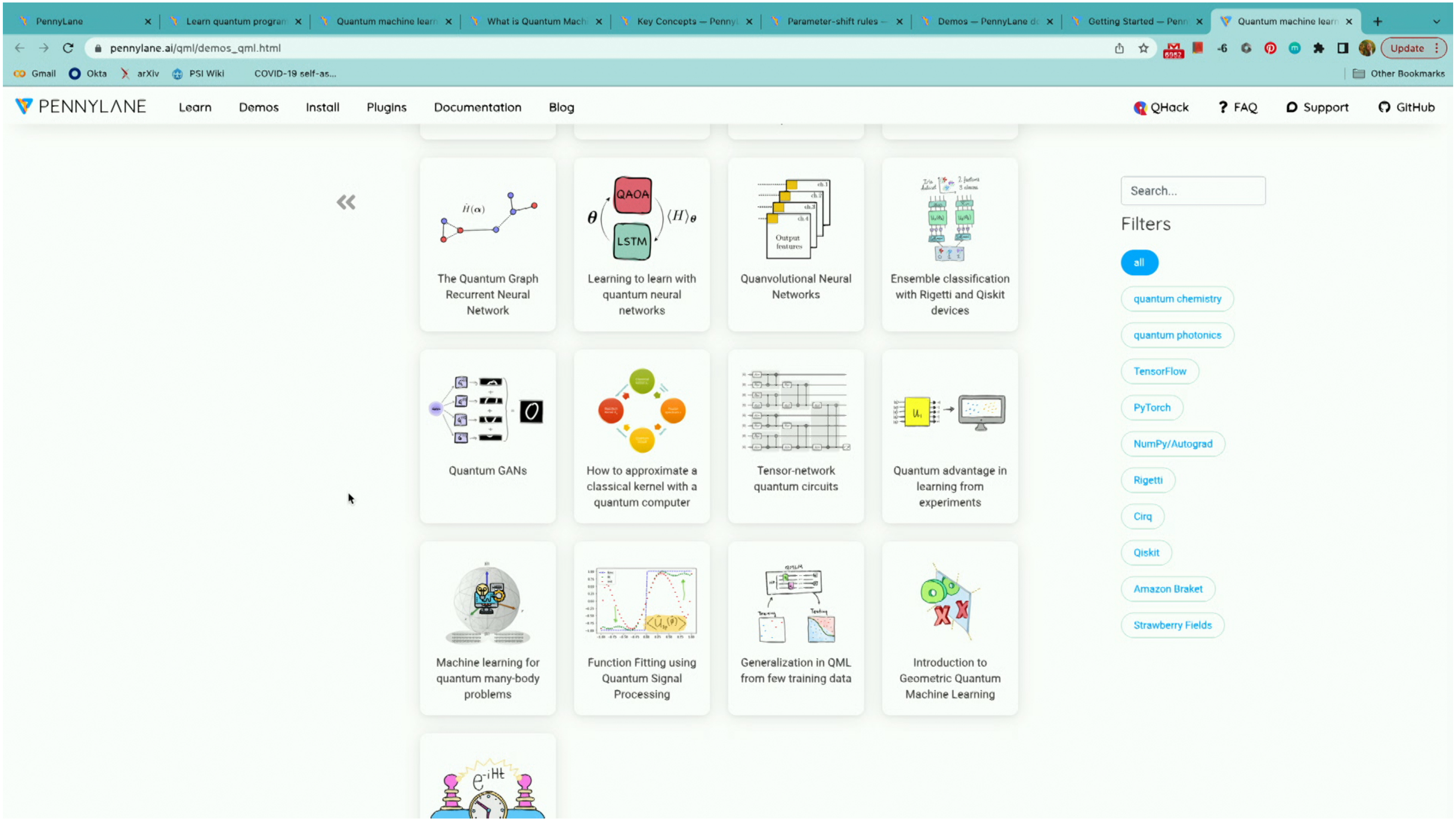
- use finite-diff. methods or "parameter shift rules"
- be aware of "barren plateaus": regions in param. space where all gradients are close to zero

Active area of research!


Programming QML model
• PennyLane by Xanadu (see Tut. 6)
• TensorFlow Quantum
• Yao
run code on quantum simulators or on quantum hardware devices


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu Mar 23 9:54 AM
PennyLane
Learn quantum programming
pennylane.ai/qml/
Gmail Okta arXiv PSI Wiki COVID-19 self-as...
Other Bookmarks
PENNYLANE
Learn Demos Install Plugins Documentation Blog
QHack FAQ Support GitHub
Learn quantum programming
We have entered a new era of quantum computing, where increasingly advanced quantum devices are used to drive an active field of research and applications.
Get started in programming for quantum computers by choosing your area of interest below, or check out our full collection of demos and tutorials.
Quantum Computing
Learn how universal computation can be achieved using quantum-mechanical systems.
Read more »
Quantum Machine Learning
Find out how the principles of quantum computing and machine learning can be united to create something new.
Read more »
Quantum Chemistry
Study the properties of molecules and materials using quantum computers and algorithms.
Read more »
Explore our new quantum datasets


PennyLane
Learn quantum programming
Quantum machine learning
What is Quantum Machine Lea
Key Concepts — PennyLane d
Parameter-shift rules — Penny
Demos — PennyLane documen
pennylane.ai/qml/demonstrations.html
Update
Gmail
Okta
arXiv
PSI Wiki
COVID-19 self-as...
Other Bookmarks
PENNYLANE
Learn
Demos
Install
Plugins
Documentation
Blog
QHack
FAQ
Support
GitHub
Take a deeper dive into quantum computing by exploring cutting-edge algorithms using PennyLane and quantum hardware.
Search
Getting started
Begin your journey into quantum computing using PennyLane.
Read more
Optimization
Learn how to train quantum circuits like neural networks, using the latest tips and tricks from the literature.
Read more
QML
Explore cutting-edge research in quantum machine learning using PennyLane.
Read more
Quantum Chemistry
Study the properties of molecules and materials using quantum computing.
Read more
Quantum Computing
Explore general quantum computing concepts and algorithms, from quantum volume to boson sampling.
Read more
Community
Discover PennyLane demonstrations created by other users, or submit one yourself.
Read more
Do you want to make your own demo using PennyLane? Read the guidelines and submission


pennylane.ai/qml/demos_qml.html
PENNYLANE
Learn
Demos
Install
Plugins
Documentation
Blog
QHack
FAQ
Support
GitHub
The Quantum Graph Recurrent Neural Network
Learning to learn with quantum neural networks
Quanvolutional Neural Networks
Ensemble classification with Rigetti and Qiskit devices
Quantum GANs
How to approximate a classical kernel with a quantum computer
Tensor-network quantum circuits
Quantum advantage in learning from experiments
Machine learning for quantum many-body problems
Function Fitting using Quantum Signal Processing
Generalization in QML from few training data
Introduction to Geometric Quantum Machine Learning
Search...
Filters
all
quantum chemistry
quantum photonics
TensorFlow
PyTorch
NumPy/Autograd
Rigetti
Cirq
Qiskit
Amazon Braket
Strawberry Fields