SIAM QIS WORKSHOP 2024

Quantum Machine Learning

An Introduction and Perspective

Alexander M. Dalzell

AWS Center for Quantum Computing

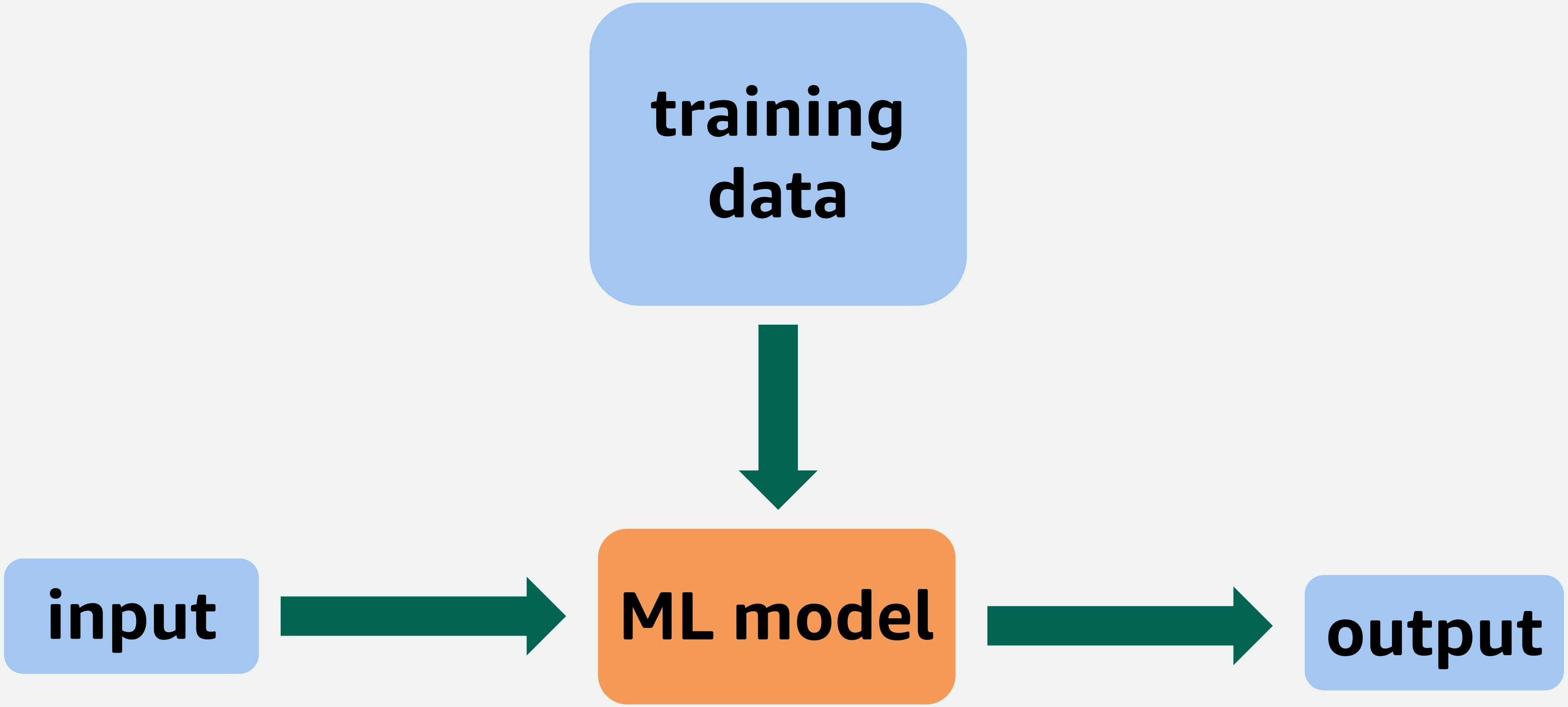
Introduction … and, a disclaimer

- QML brings together many areas: ML / AI, theoretical & applied CS, physics, quantum info science, hardware engineering, **applied math**, etc.
- My viewpoint on QML is shaped by:
  - my background in physics / theoretical CS
  - my research preference for intermediate-to-far term applications for fault-tolerant quantum computers

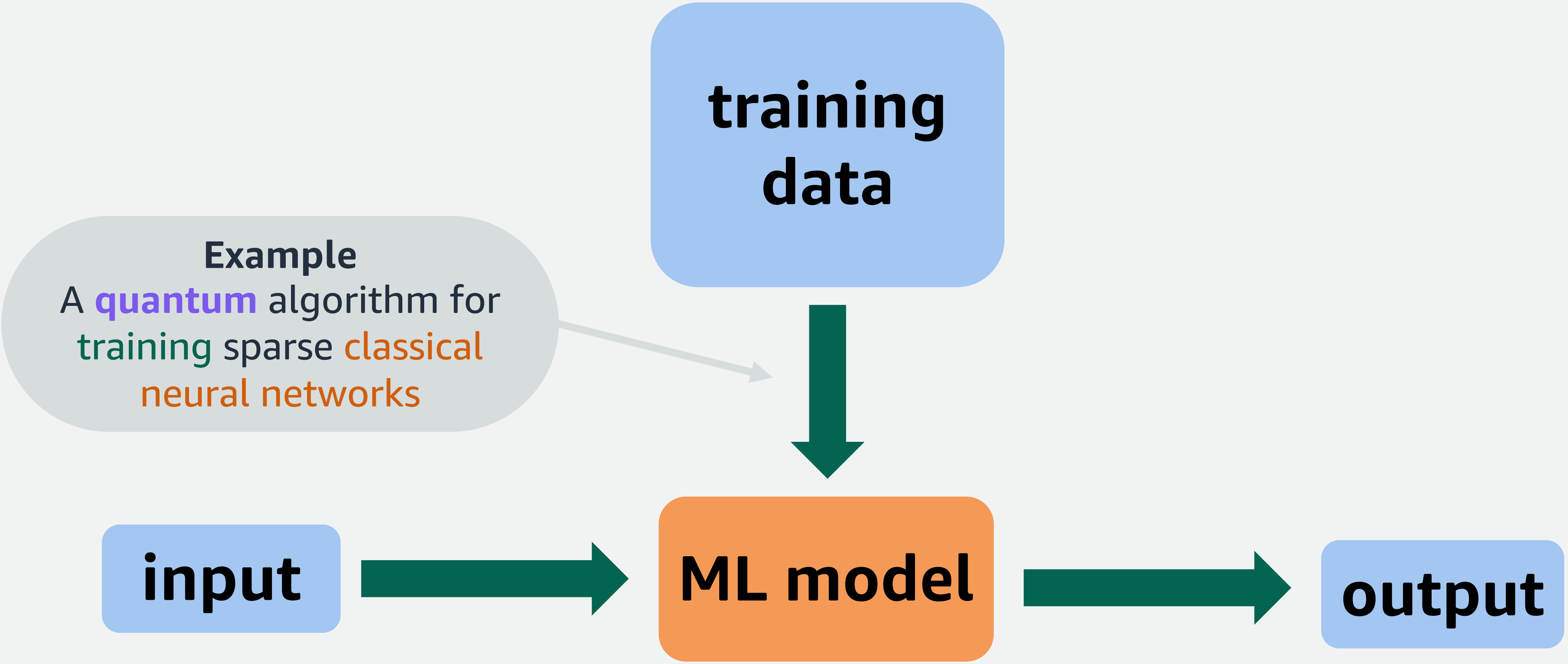
Goals and key questions

- Why might we hope quantum computing will be good at ML in the first place?
- What are the biggest outstanding technical challenges in QML (and how can Applied Math help?)
- What is the outlook of QML as an application area of quantum computing?

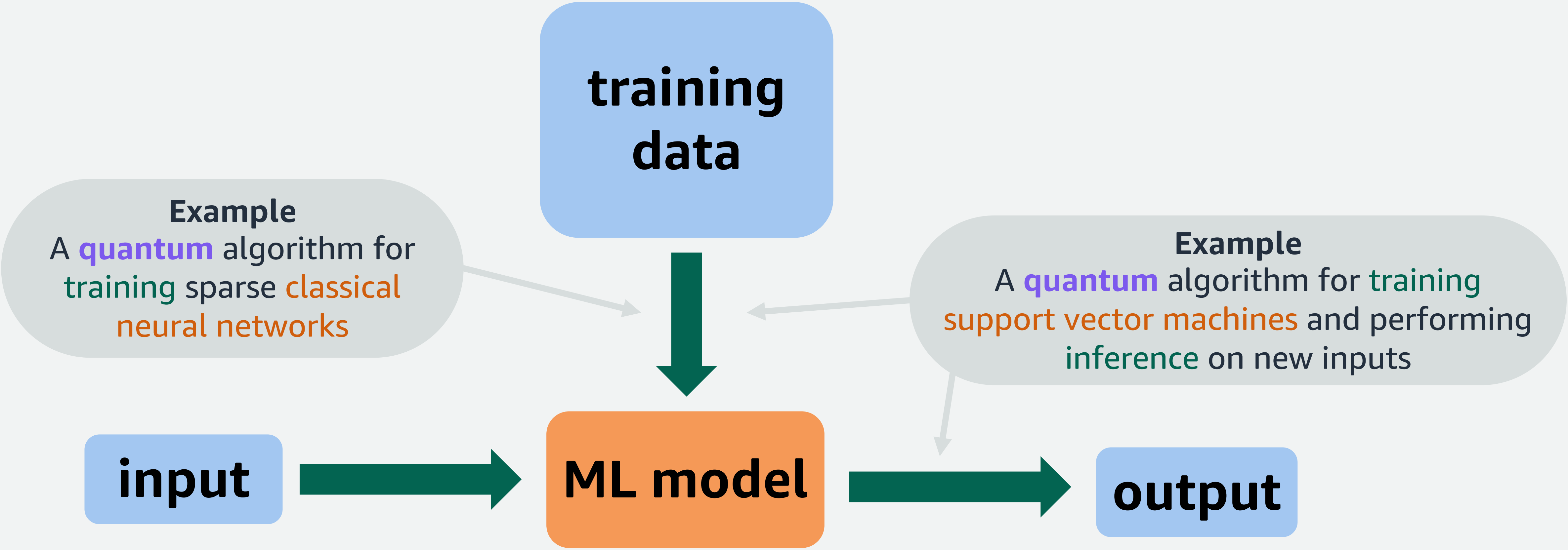
Flavors of quantum machine learning



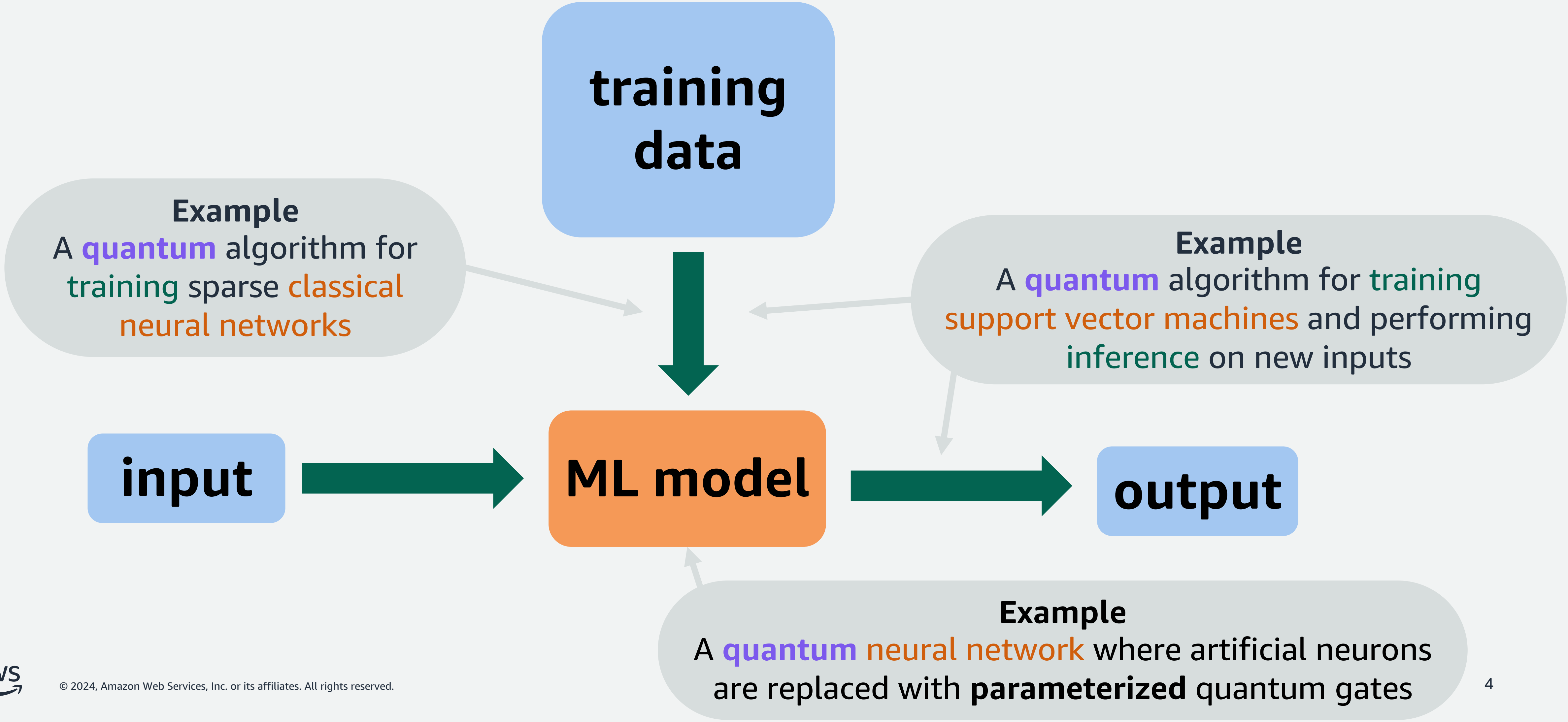
Flavors of quantum machine learning



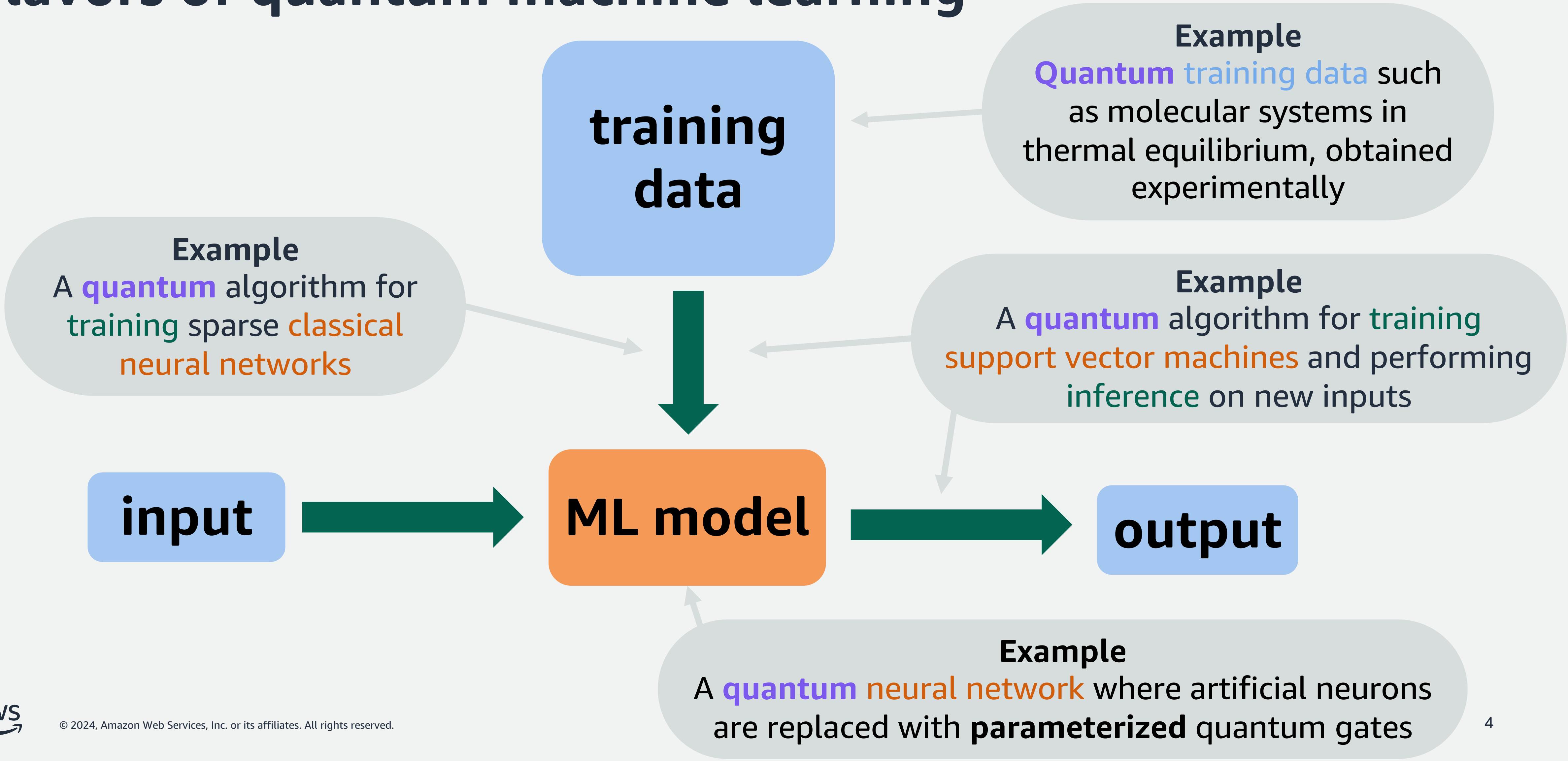
Flavors of quantum machine learning



Flavors of quantum machine learning



Flavors of quantum machine learning



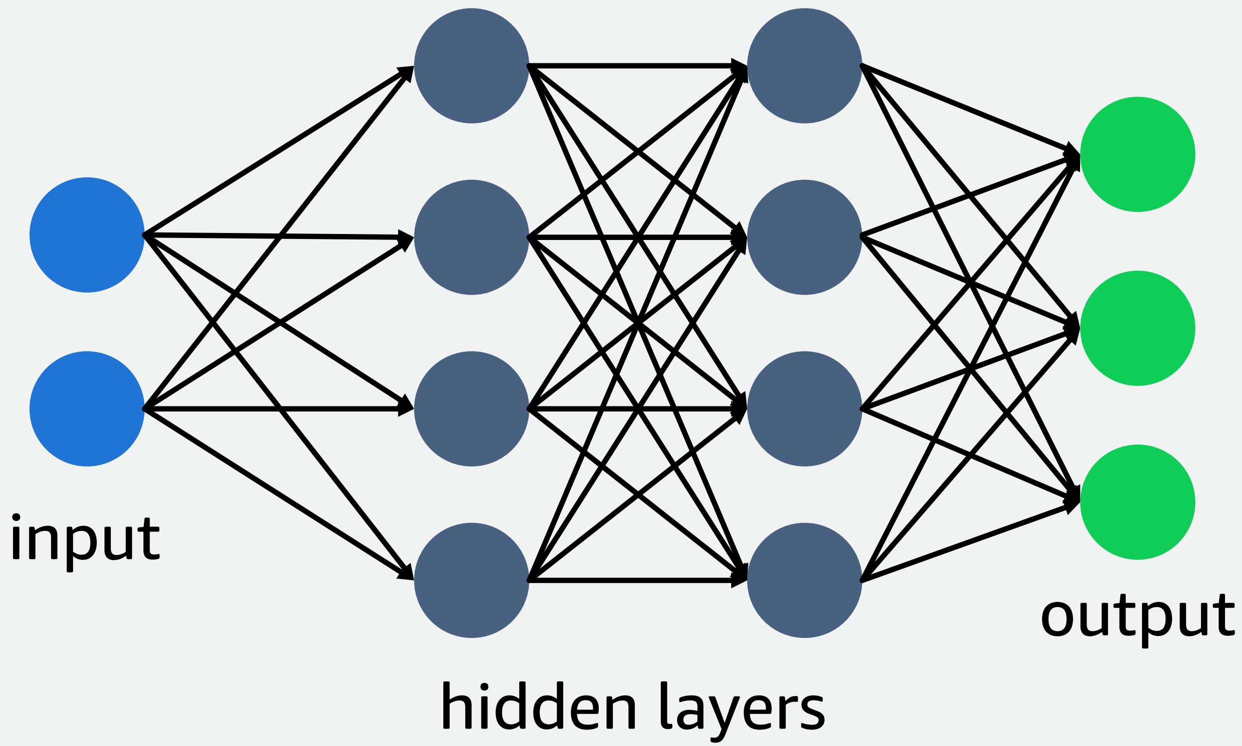
Example: Quantum neural networks / variational quantum algorithms

Key idea: entangled quantum states can capture nonclassical correlations

- Quantum states live in a high-dimensional vector space, not directly simulable classically
- Interference and entanglement allow quantum information to be processed in fundamentally nonclassical way
- New tool to try on big data problems

Example: Quantum neural networks (QNNs)

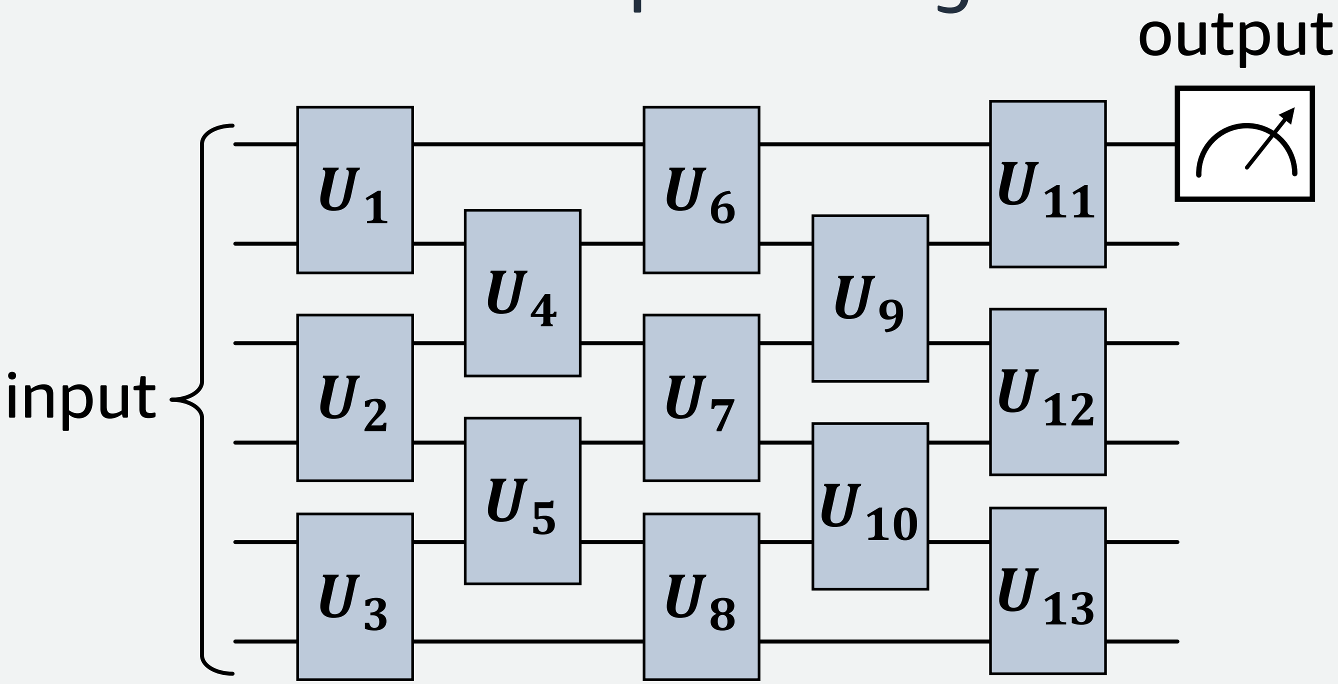
But first, classical neural networks:



- Computes functions from inputs to outputs
- Tunable weights, trained by optimizing a loss function
- Training occurs via (stochastic) gradient descent, with "backpropagation"
- Heuristic

Example: Quantum neural networks (QNNs)

QNNs replace artificial neurons with quantum gates



Parameterized quantum gates

- Computes functions from inputs to outputs
- Tunable gate parameters trained by gradient descent
- Heuristic

QNN caveats

- While somewhat NISQ-friendly, QNNs **cannot be scaled indefinitely without quantum error correction**

QNN caveats

- While somewhat NISQ-friendly, QNNs **cannot be scaled indefinitely without quantum error correction**
- **Barren plateaus –** due to exponentially large Hilbert space, gradients of loss function can be exponentially small
  - Recent unification of barren plateau phenomenon in language of Lie algebras and their subalgebras [Larocca et al. 2022] [Larocca et al., 2024]
- **No quantum analogue of backpropagation** – for a model with $O(M)$ parameters, computing function requires $O(M)$ work but computing gradient of function requires at least $O(M^{3/2})$ work

[Abbas et al. 2023]

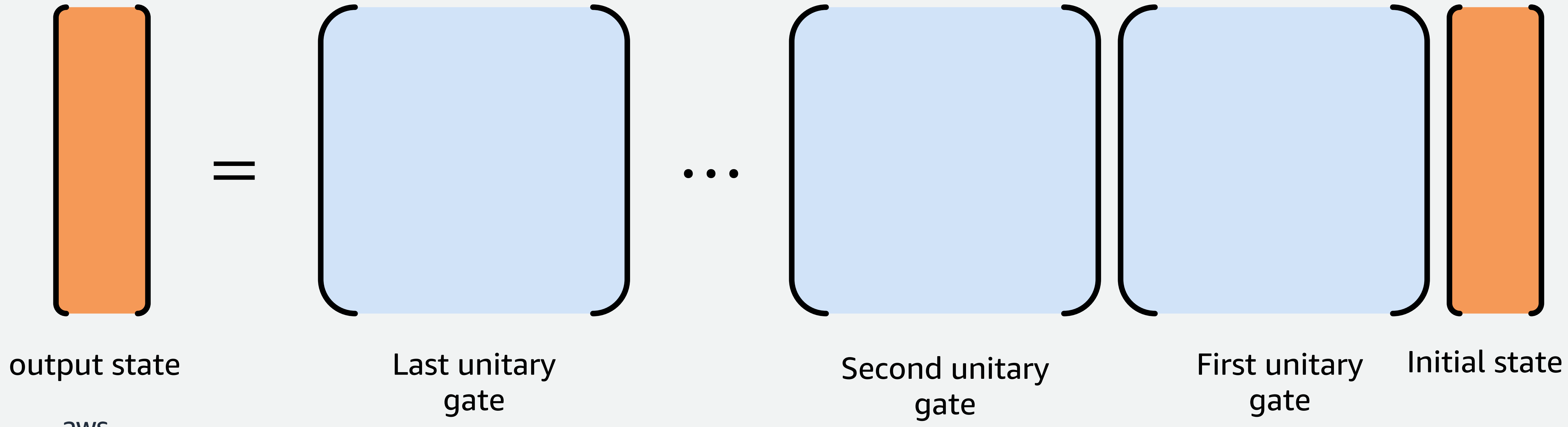
QNN caveats

- While somewhat NISQ-friendly, QNNs **cannot be scaled indefinitely without quantum error correction**
- **Barren plateaus –** due to exponentially large Hilbert space, gradients of loss function can be exponentially small
  - Recent unification of barren plateau phenomenon in language of Lie algebras and their subalgebras [Larocca et al. 2022] [Larocca et al., 2024]
- **No quantum analogue of backpropagation** – for a model with $O(M)$ parameters, computing function requires $O(M)$ work but computing gradient of function requires at least $O(M^{3/2})$ work [Abbas et al. 2023]
- **Classical ML is too good!**

Example: QML via quantum linear algebra

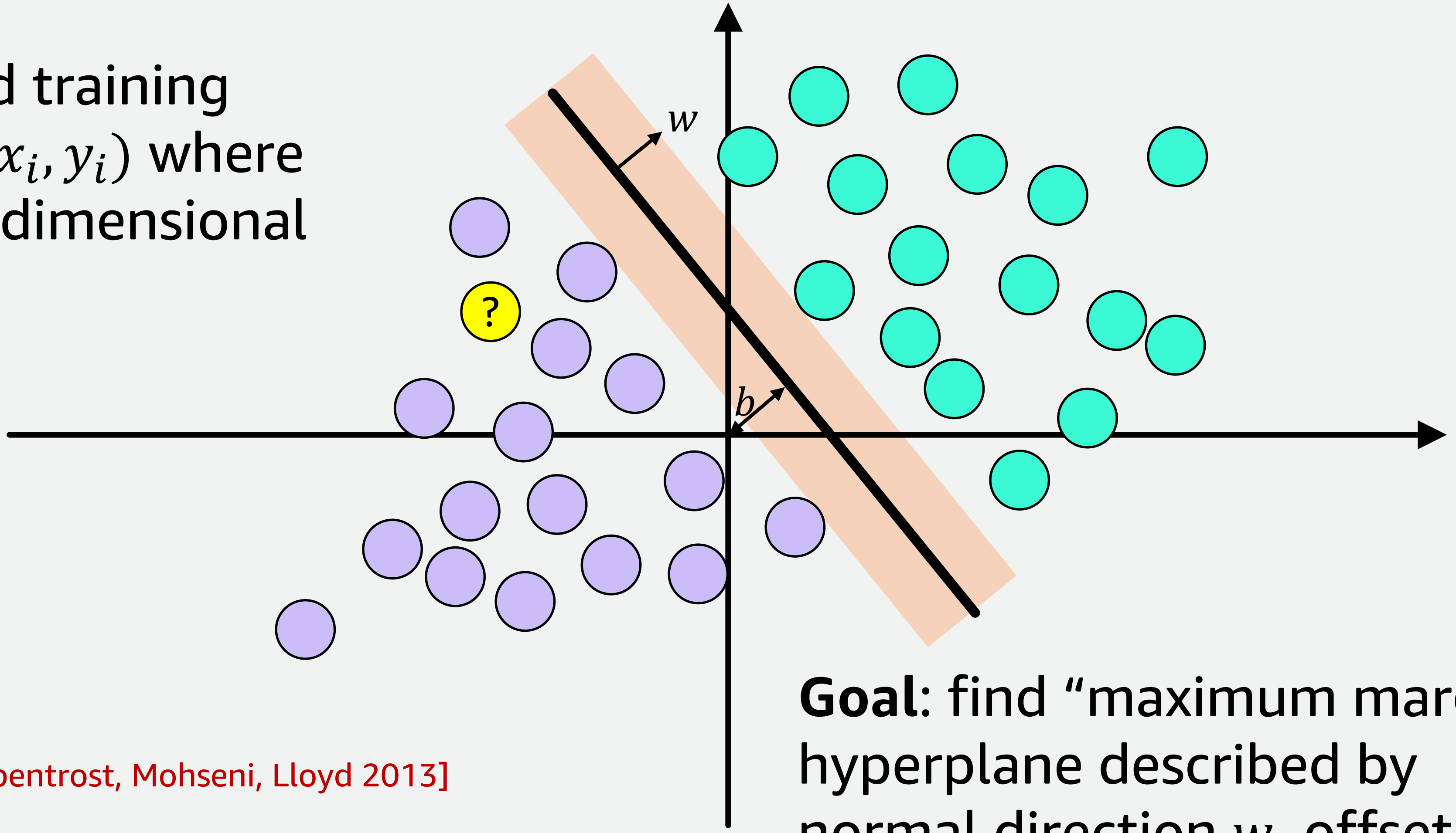
Key idea: quantum computing and ML are both high-dimensional linear algebra

A quantum algorithm on $\log(n)$ qubits is a sequence of sparse matrix-vector multiplications in n-dimensional vector space



Example: Support vector machine

M labelled training samples (x_i, y_i) where x_i is an N dimensional vector



Goal: find "maximum margin" hyperplane described by normal direction w, offset b

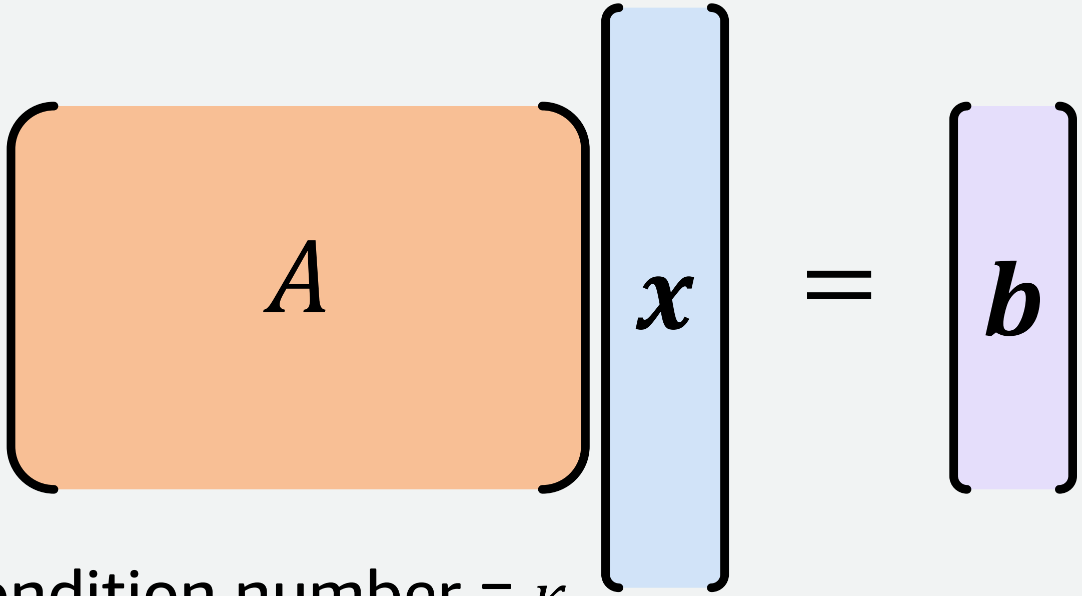
[See Rebentrost, Mohseni, Lloyd 2013]

Other examples of QML problems with linear algebra

- Recommendation systems [Kerenidis, Prakash, 2017]
- Principal component analysis [Lloyd, Mohseni, Rebentrost, 2014]
- Supervised cluster assignment [Lloyd, Mohseni, Rebentrost, 2013]
- Gaussian process regression [Zhao, Fitzsimons, Fitzsimons, 2019]

Problem often reduces to linear system of equations

A and b depend on the training and input data



condition number = κ

Often, in QML applications one also assumes A is low rank, or close to low rank

HHL algorithm can prepare quantum state encoding linear system solution in logarithmic time

Solution vector

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix}$$

Quantum state

$$|\boldsymbol{x}\rangle = \frac{1}{\|\boldsymbol{x}\|} \sum_{i=0}^{n-1} x_i |\boldsymbol{i}\rangle$$

[Harrow, Hassidim, Lloyd, 2009]

HHL algorithm (2009)
Prepares the state $|\boldsymbol{x}\rangle$ in time $\kappa^2 \text{ polylog}(n)$

Later improved to $\kappa \text{ polylog}(n)$
See, e.g. [Ambainis 2010] [Costa et al. 2021]

HHL algorithm can prepare quantum state encoding linear system solution in logarithmic time

Solution vector

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix}$$

Quantum state

$$|\boldsymbol{x}\rangle = \frac{1}{\|\boldsymbol{x}\|} \sum_{i=0}^{n-1} x_i |\boldsymbol{i}\rangle$$

[Harrow, Hassidim, Lloyd, 2009]

HHL algorithm (2009)
Prepares the state $|\boldsymbol{x}\rangle$ in time $\kappa^2 \text{ polylog}(n)$

Later improved to $\kappa \text{ polylog}(n)$
See, e.g. [Ambainis 2010] [Costa et al. 2021]

Compare to classical iterative methods

- Gaussian elimination $O(n^{2.37})$
- Conjugate gradient method $O(\sqrt{\kappa}\, n)$ for psd sparse matrices
- Randomized Kaczmarz method $O(\kappa^2 n)$ for low-rank matrices

<u>Exponential speedup?!?!</u>

Caveat #1: Output problem

- Need to read out useful information from state

$$|\boldsymbol{x}\rangle = \frac{1}{\|\boldsymbol{x}\|} \sum_{i=0}^{n-1} x_i |\boldsymbol{i}\rangle$$

Measuring this state yields outcome i with probability $\frac{x_i^2}{\|\boldsymbol{x}\|^2}$

Caveat #1: Output problem

- Need to read out useful information from state

$$|\boldsymbol{x}\rangle = \frac{1}{\|\boldsymbol{x}\|}\sum_{i=0}^{n-1} x_i|\boldsymbol{i}\rangle$$

Measuring this state yields outcome i with probability $\frac{x_i{}^2}{\|\boldsymbol{x}\|^2}$

- Learning entire state costs $O(n)$ copies, negating exponential speedup
- Can read out one quantity to error ε at multiplicative overhead of $O(1/\varepsilon)$

Caveat #1: Output problem

- Need to read out useful information from state

$$|\boldsymbol{x}\rangle = \frac{1}{\|\boldsymbol{x}\|}\sum_{i=0}^{n-1} x_i|\boldsymbol{i}\rangle$$

Measuring this state yields outcome i with probability $\frac{{x_i}^2}{\|\boldsymbol{x}\|^2}$

- Learning entire state costs $O(n)$ copies, negating exponential speedup
- Can read out one quantity to error ε at multiplicative overhead of $O(1/\varepsilon)$
- End-to-end problem needs to rely on a small number of quantities, and not require high precision
- Example: SVMs – new vector can be classified by reading out 1 number

Caveat #2: Input problem

- How is it possible that the algorithm has runtime $\mathrm{polylog}(n)$ when the data takes $O(n)$ space to even write down?

Caveat #2: Input problem

- How is it possible that the algorithm has runtime $\mathrm{polylog}(n)$ when the data takes $O(n)$ space to even write down?
- Answer: parallelism, via assumption of quantum RAM

Quantum RAM allows data to be accessed in superposition

$$\sum_{i=0}^{n-1} \alpha_i |i\rangle \mapsto \sum_{i=0}^{n-1} \alpha_i |i\rangle |f(i)\rangle$$

Many QML algorithms assume this operation can be done at cost $\mathrm{polylog}(n)$

Index i	**Data $f(i)$**
000	0
001	1
010	1
011	1
100	0
101	1
110	0
111	0

Caveat #2: Input problem (cont'd)

[See Jaques, Rattew, 2023]

- Assumption of $\mathrm{polylog}(n)$-cost QRAM is **controversial**
  - Assumption roughly holds for classical RAM
  - QRAM not perfectly compatible with quantum error correction
  - No compelling hardware proposal for large-scale physical QRAM

- Without assumption of cheap QRAM, exponential speedup is gone

Caveat #3: “dequantization” of QML reduces available quantum speedup in many cases

- One should compare QML algorithms to classical algorithms under analogous input assumptions
- “Sample-and-query” access model for classical algorithms is analogue of QRAM
  - Given dataset represented by a vector $\boldsymbol{x} \in \mathbb{R}^n$, one can **query** entries x_i of $\boldsymbol{x}$, or **sample** an entry with probability $\frac{{x_i}^2}{\|\boldsymbol{x}\|^2}$

Caveat #3: "dequantization" of QML reduces available quantum speedup in many cases

- One should compare QML algorithms to classical algorithms under analogous input assumptions
- "Sample-and-query" access model for classical algorithms is analogue of QRAM
  - Given dataset represented by a vector $\boldsymbol{x} \in \mathbb{R}^n$, one can **query** entries x_i of $\boldsymbol{x}$, or **sample** an entry with probability $\frac{{x_i}^2}{\|\boldsymbol{x}\|^2}$
- 2018: Quantum recommendation systems algorithm "dequantized" via classical algorithm with $\mathrm{poly}\left(\frac{\kappa}{\varepsilon}\right)\mathrm{polylog}(n)$ total cost [Tang, 2018]
- Also dequantized: Quantum Principal Component Analysis, Support Vector Machines, Nearest Centroid Classification, HHL for low-rank matrices

Other topics not covered

- Topological data analysis [Berry et al. 2024] [McArdle, Gilyén, Berta, 2022]
- Learning theory (e.g. PAC learning) [Arunachalam, de Wolf, 2017]
- Energy-based models (e.g. quantum Boltzmann machines) [Amin et al. 2017] [Schuld, Petruccione 2021]
- Tensor PCA [Hastings, 2020]
- Training sparse classical neural networks via quantum algorithms for nonlinear differential equations [Liu et al., 2023]
- Learning with quantum data [Chen, Cotler, Huang, Li, 2022]

Technical opportunities for applied math in QML

- **Heuristic algorithms** – how to gather evidence with limited empirical data?
- **End-to-end problems** – how to connect the capabilities of quantum computers with real-world problems that aren't served by classical ML?
- **More creative solutions to input-output problems**

Applications where input is small and calculation is hard offer clearer path to quantum advantage

	Input/training data size	Available quantum speedup	Relative confidence in speedup
Machine learning e.g., training support vector machines	Big e.g., large database of classified images	Small / Medium / Unknown	Low
Simulation e.g., computing energies of chemical systems	Small e.g., locations of nuclei in molecule	Medium / Large	Medium
Optimization e.g., finding an optimal route	Small / Medium e.g., locations of destinations along route	Medium / Unknown	Medium
Cryptanalysis e.g., breaking RSA	Small e.g., 2048-bit integer	Large	High

} More feasible than ML in the intermediate term

Conceptual outlook and next steps

- QML needs new ideas to circumvent known caveats and expected scaling issues
- The energy in quantum computing is moving away from NISQ and toward fault-tolerant (FT) quantum computing
- What can we learn about QML from early FT devices?
- What "quantum data" problems are interesting in science and industry, and can we solve them?

Some more references

- **General**: https://arxiv.org/pdf/1707.08561
- **On classification of different QML tasks**: Fig. 1 of https://arxiv.org/pdf/2303.09491
- **Quantum algorithm for training sparse classical neural networks**: https://www.nature.com/articles/s41467-023-43957-x
- **Quantum neural networks**: https://arxiv.org/pdf/2303.09491
- **Quantum algorithms for support vector machines**: https://arxiv.org/abs/1307.0471
- **General**: Sec. 9 of https://arxiv.org/pdf/2310.03011

Notes after presentation

- Thank you to attendees who pointed out mistake in conjugate gradient complexity (it has been fixed in this version)
- I have added more references