

Day 2 Recap

- Two qubit gates
 - CNOT, SWAP
- No cloning
- Superdense coding
- Three qubit gates
 - Controlled CNOT, Controlled SWAP
- Teleportation
- A simple QA with two qubits: Deutsch Algorithm

- Deutsch-Jozsa algorithm
- A very short comment on Bernstein-Vazirani Algorithm

- Skipped: Simon's algorithm, Quantum Fourier Transformation, Shor's algorithm and Grover's algorithm

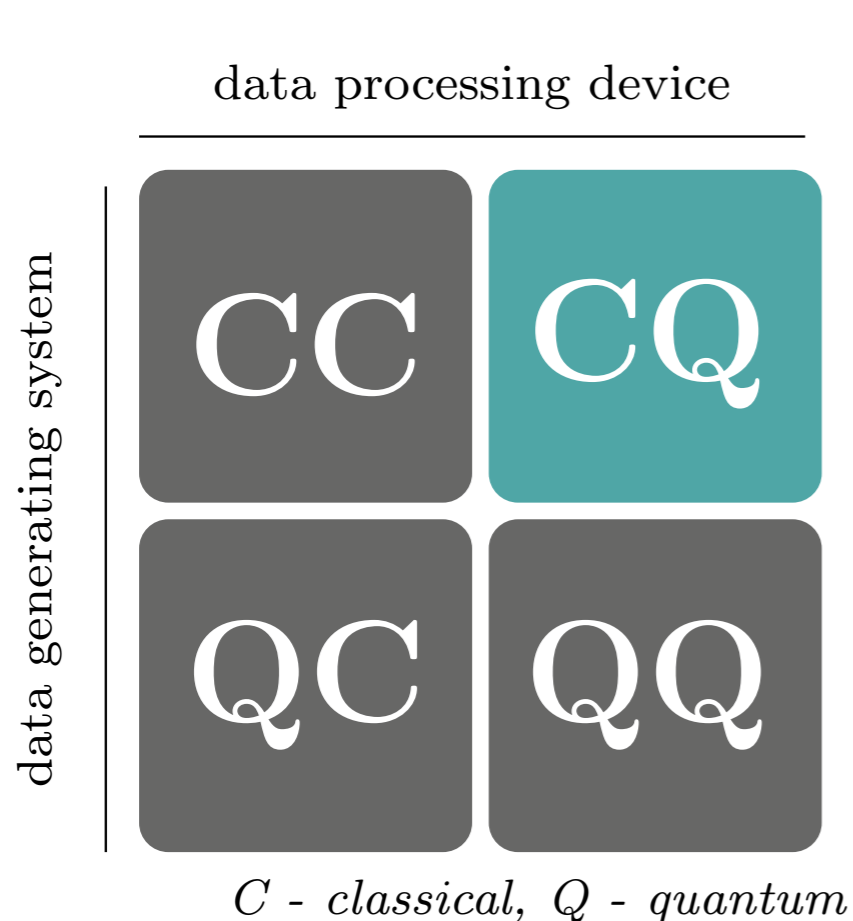
Day 3 Plan

- Day 3:
 - Distance-based classifier
 - Quantum optimization and adiabatic theorem
 - QAOA and ADAPT-QAOA
- Day 4: 2pm - 4pm
- Day 5: 1pm - 3:30pm
 - Feedback-based Algorithm Quantum Optimization (FALQON)
 - Data re-uploading for a universal quantum classifier
 - Error correction

 - Bernstein-Vazirani Algorithm and Simon's algorithm
 - Quantum Fourier Transformation and Phase estimation
 - Shor's algorithm, Grover's algorithm

Quantum Machine Learning

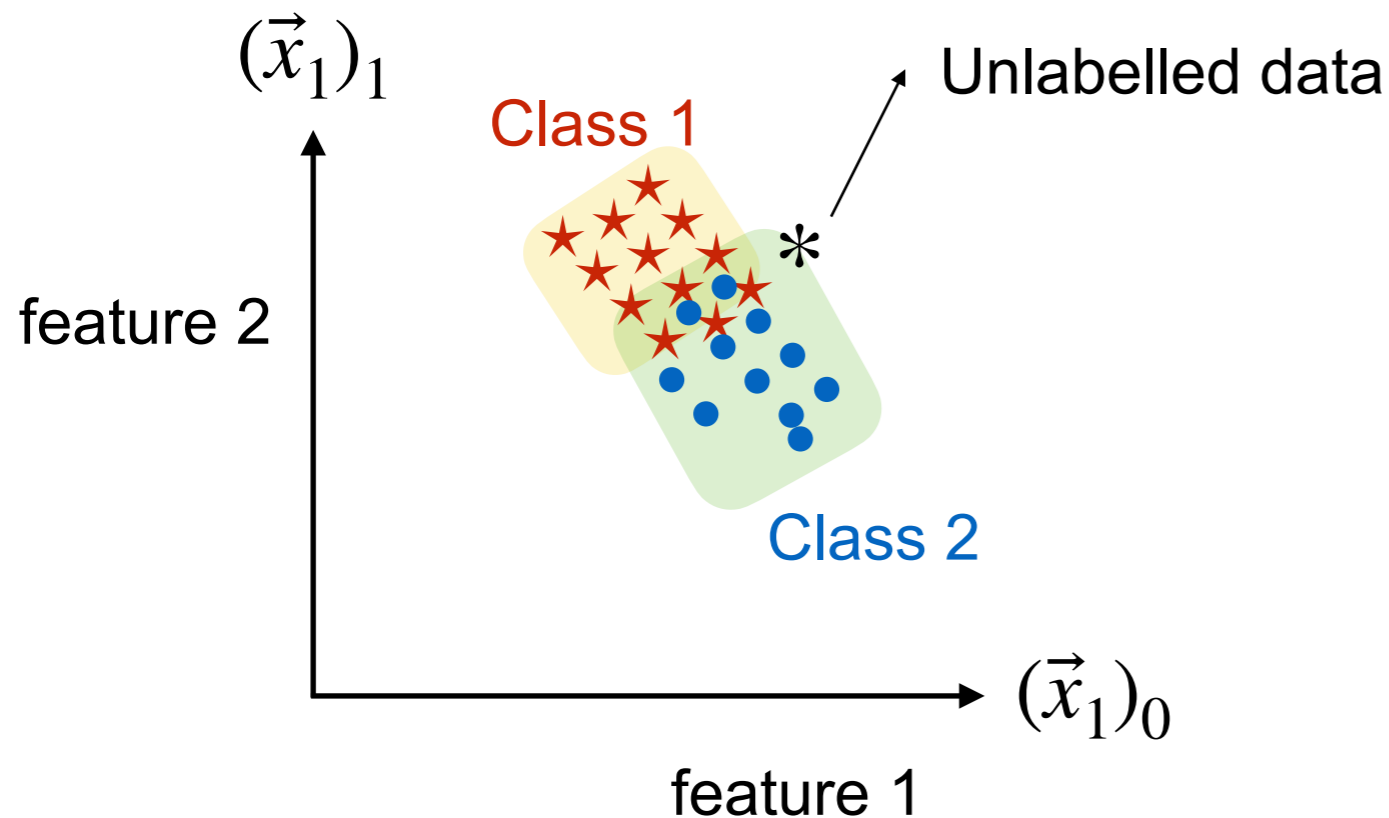
- Artificial Intelligence: Statistical prediction
- Machine Learning: Learn from data
- Quantum Machine Learning: Learn from data with quantum algorithms
 - Subdiscipline of quantum computing and quantum information science



- CC: classical data being processed classically
- QC: how machine learning can help with quantum computing
- CQ: classical data fed into quantum computer for analysis (quantum machine learning)
- QQ: quantum data being processed by quantum computer (ex: Quantum simulation)

Distance-based classifier

- A distance-based classifier with a quantum interference circuit: arXiv:1703:10793 (supervised binary classification)



training data set

$$D = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_M, y_M)\}$$

$$\vec{x}_m \in \mathbb{R}^N \quad y_m \in \{-1, +1\}$$

$$m = 1, 2, \dots, M$$

M = the number of data

N = the number of features

$$\vec{\tilde{x}}_m \in \mathbb{R}^N : \text{unlabelled data}$$

→ Find the label $\tilde{y} \in \{-1, 1\}$

Classical Kernel Method

- Kernel methods: kNN (k-nearest neighborhood), KDE (kernel density estimation), SVM (support vector machine), Gaussian processes
 - Nearest neighborhood method: a new input data is given the same label as the data point closest to it → k-nearest neighborhood (kNN)
 - Closeness = distance measure
 - (ex) Euclidean distance $|\vec{x} - \vec{x}_m|^2$

$$\tilde{y} = \text{sign} \left[\sum_{m=1}^M y_m \left(1 - \frac{1}{4M} |\vec{x} - \vec{x}_m|^2 \right) \right]$$

- include all data but weigh influence of each data toward the decision by the weight $\kappa(\vec{x}, \vec{x}_m)$

$$\tilde{y} = \text{sign} \left[\sum_{m=1}^M w_m y_m \kappa(\vec{x}, \vec{x}_m) \right]$$

weight

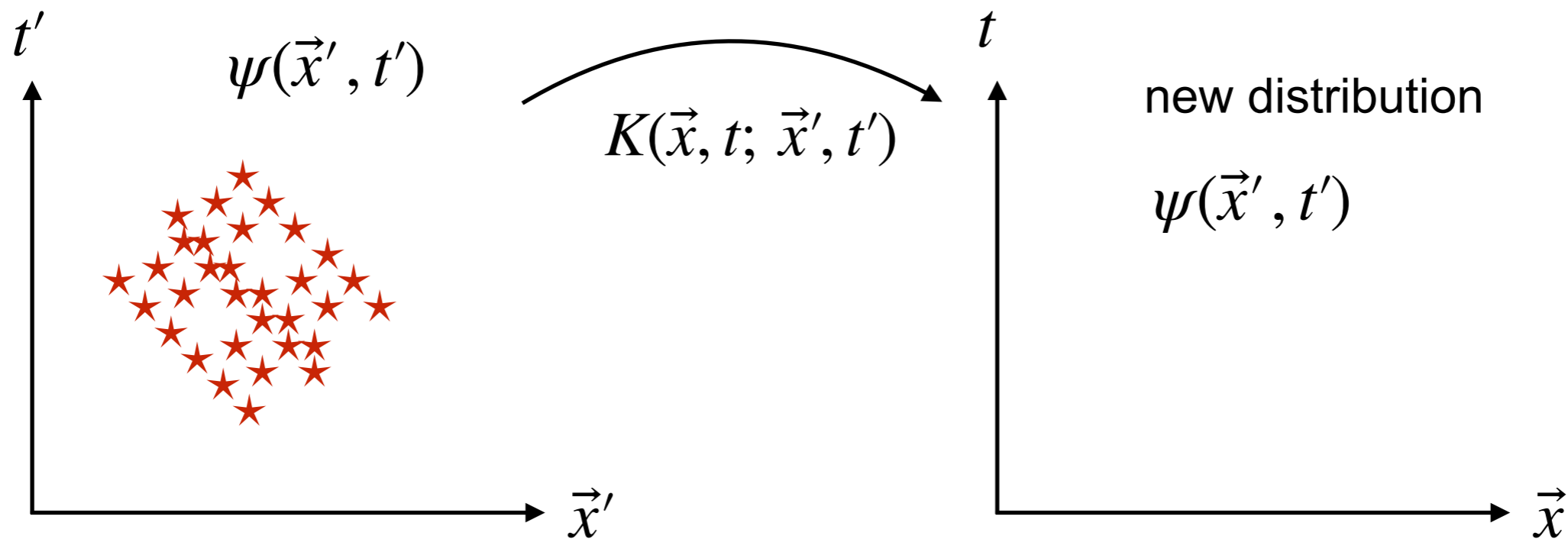
Kernel

Label ± 1 for \vec{x}_m

Classical Kernel Method

$$\psi(\vec{x}, t) = \int K(\vec{x}, t; \vec{x}', t') \psi(\vec{x}', t') d^3\vec{x}'$$

$K(\vec{x}, t; \vec{x}', t')$: kernel, Green's function, or propagator contains the probability of particle propagation between (\vec{x}, t) and (\vec{x}', t')



Distance-based classifier

- Choose $w_m = 1$ for all equally important data

$$\kappa(\vec{x}, \vec{x}_m) = 1 - \frac{1}{4M} |\vec{x} - \vec{x}_m|^2$$

Close data (small distance) are weighted more importantly.

- (1) Encode input data (features) into the amplitude of a quantum system (amplitude encoding). For classical vector $\vec{x} \in \mathbb{R}^N$, ($N = 2^n$) Assume $x^T x = \vec{x} \cdot \vec{x} = 1$ (normalized to 1)

$N = 2^n$: number of features

$$|\psi_X\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$$

i : index in the computational basis

Dimension of Hilbert space $\approx O(\log N)$

ancilla qubit is entangled with third register

(2) initial state: $|D\rangle = \frac{1}{\sqrt{2M}} \sum_{m=1}^M |m\rangle \left(|0\rangle |\psi_{\vec{x}}\rangle + |1\rangle |\psi_{x_m}\rangle \right) |y_m\rangle$

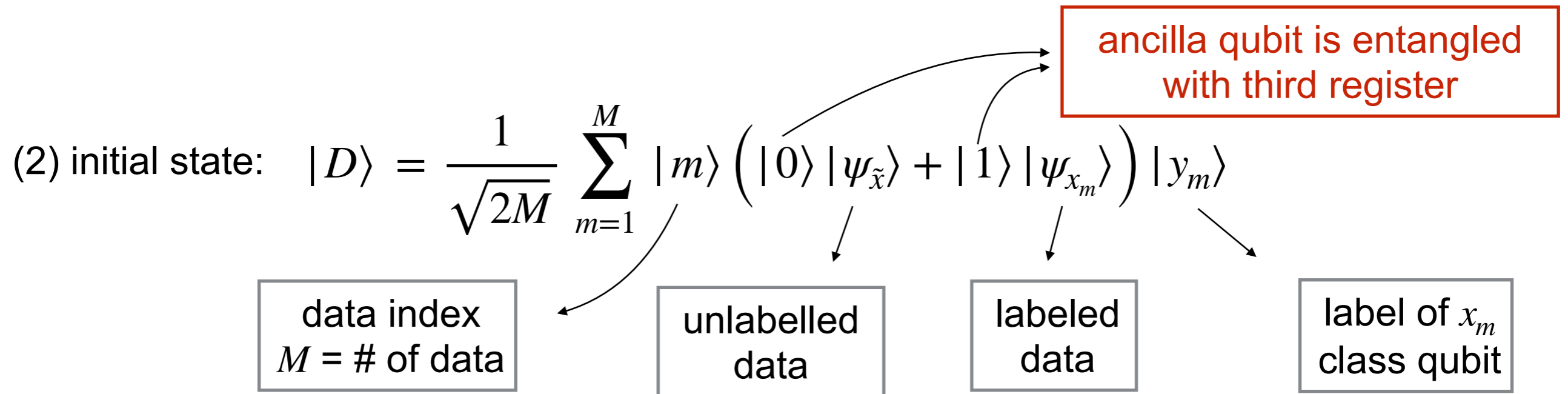
data index
 $M = \#$ of data

unlabelled data

labeled data

label of x_m
class qubit

Distance-based classifier



$$|\psi_{x_m}\rangle = \sum_{i=0}^{N-1} x_m^i |i\rangle \quad \text{encoding of } m\text{-th training data (labeled)}$$

$$|\psi_{\tilde{x}}\rangle = \sum_{i=0}^{N-1} \tilde{x}^i |i\rangle \quad \text{encoding of new data (unlabeled)}$$

$$|y_m\rangle = \begin{cases} |0\rangle, & \text{if } y_m = -1 \\ |1\rangle, & \text{if } y_m = +1 \end{cases}$$

$|D\rangle$ contains all training data as well as M copies of new inputs.

Distance-based classifier

(3) Apply Hadamard gate on the ancilla (second) qubit. $|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$$|D\rangle = \frac{1}{\sqrt{2M}} \sum_{m=1}^M |m\rangle \left(|0\rangle |\psi_{\tilde{x}}\rangle + |1\rangle |\psi_{x_m}\rangle \right) |y_m\rangle \quad |1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$\downarrow$$

$$|D'\rangle = \frac{1}{2\sqrt{M}} \sum_{m=1}^M |m\rangle \left(|0\rangle |\psi_{\tilde{x}+x_m}\rangle + |1\rangle |\psi_{\tilde{x}-x_m}\rangle \right) |y_m\rangle$$

$$|\psi_{\tilde{x}\pm x_m}\rangle = |\psi_{\tilde{x}}\rangle \pm |\psi_{x_m}\rangle = \sum_{i=0}^{M-1} (\tilde{x}^i \pm x_m^i) |i\rangle$$

(4) Conditional measurement selecting the branch with ancilla state $|0\rangle$.
Likely to succeed if the collective Euclidean distance b/w \tilde{x} and training data set is small. For standard data, $p \geq 0.5$.

Probability is $p = \frac{1}{4M} \sum_m |\vec{\tilde{x}} + \vec{x}_m|^2$

$$|D''\rangle = \frac{1}{2\sqrt{Mp}} \sum_{m=1}^M \sum_{i=0}^{N-1} |m\rangle (\tilde{x}^i + x_m^i) |i\rangle |y_m\rangle$$

Distance-based classifier

(5) Probability of measuring the class qubit $|y_m\rangle = |0\rangle$

$$|D''\rangle = \frac{1}{2\sqrt{Mp}} \sum_{m=1}^M \sum_{i=0}^{N-1} |m\rangle (\tilde{x}^i + x_m^i) |i\rangle |y_m\rangle$$

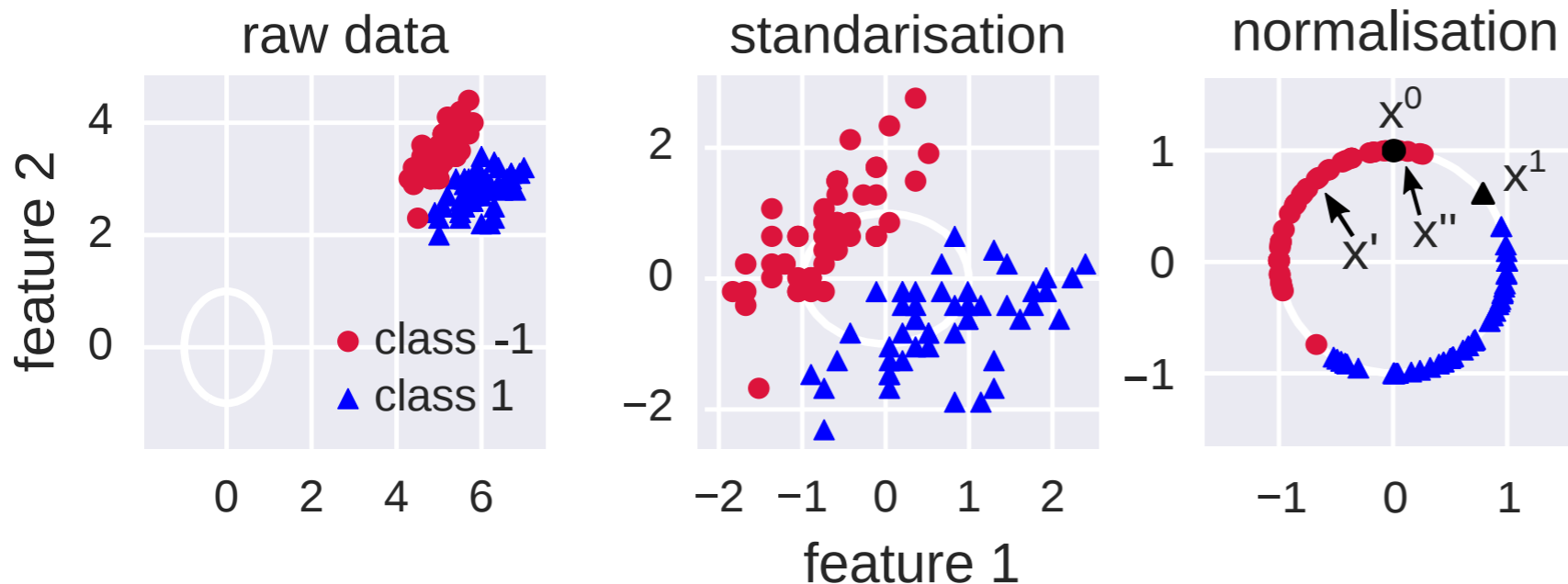
$$P(\tilde{y} = 0) = \frac{1}{4Mp} \sum_{y_m=0, m=1}^M |\vec{\tilde{x}} + \vec{x}_m|^2 = 1 - \frac{1}{4Mp} \sum_{y_m=0, m=1}^M |\vec{\tilde{x}} - \vec{x}_m|^2$$

Class 1

using normalization condition

→ choosing the class with the higher probability gives result of kernel method.

The # of measurement needed to estimate $P(\tilde{y} = 0)$ to error ϵ with a reasonably high confidence interval grows with $O(\epsilon^{-1})$.



- [arXiv:1703:10793](https://arxiv.org/abs/1703.10793)
used Iris data

Distance-based classifier

Example: square distance classifier
Kaggle Titanic dataset

Quantum Optimization

- Optimization problems are everywhere: math, science, business, finance etc
 - In general, time-consuming.
 - In many cases, can not be solved in polynomial time.
 - Need approximation algorithms: find approximation of the best solution rather than the best solution (time complexity is reduced).
- Two classes
 - Continuous optimization
 - Discrete optimization: combinatorial optimization
 - Quadratic Unconstrained Binary Optimization (QUBO)
- Apply quantum algorithms to solve optimization problem
 - (1) **Gate model**: use universal gates (Pauli's), problem-independent.
 - (2) Non-gate model (**quantum annealer**): relies on adiabatic theorem to find a minimum energy of Hamiltonian corresponding to the minimum value of some cost function.

Quadratic Unconstrained Binary Optimization (QUBO)

- QUBO: combinatorial optimization problem with a wide range of applications from finance to ML (partitioning, graph coloring, task allocation, max-sat, max-cut etc)

$f : \mathbb{Z}_2^n \longrightarrow \mathbb{R}$ Quadratic polynomial over binary variable

$$f(x) = \sum_{i=1}^n \sum_{j=1}^i q_{ij} x_i x_j + \sum_{i=1}^n h_i x_i$$

$x_i \in \mathbb{Z}_2 = \{0,1\}, \quad h_i, q_{ij} \in \mathbb{R}$
 $x = x_n x_{n-1} \cdots x_2 x_1$
(binary strings of n-bits)

- Find a binary vector x^* which minimizes f

$$x^* = \operatorname{argmin}_{x \in \mathbb{Z}_2^n} f(x)$$

- In matrix notation, $f(x) = x^T Q x$, where $Q \in \mathbb{R}^{n \times n}$

Quadratic Unconstrained Binary Optimization (QUBO)

- In matrix notation, $f(x) = x^T Q x$, where $Q \in \mathbb{R}^{n \times n}$

$$f(x) = -2x_1 - 3x_2 + 8x_3 + 4x_4 + 4x_1x_2 + 5x_1x_3 + 6x_2x_3 + 10x_3x_4$$

$$= (x_1 \ x_2 \ x_3 \ x_4) \begin{pmatrix} -2 & 2 & 5/2 & 0 \\ 2 & -3 & 3 & 0 \\ 5/2 & 3 & 8 & 5 \\ 0 & 0 & 5 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = x^T Q x \quad \begin{array}{l} x_i = x_i^2 \\ x_i \in Z_2 = \{0, 1\} \end{array}$$

- QUBO:
 - NP hard problem
 - Quadratic function might have several local minima
 - Close connection to Ising model

P vs NP

- In Theoretical Computer Science, the two most basic classes of problems are P and NP.
- **P** includes all problems that **can be solved efficiently**.
 - For example: add two numbers. The formal definition of "efficiently" is in **time that's polynomial in the input's size**.
- **NP** (nondeterministic polynomial (time)) includes all problems that given a solution, one can efficiently verify that the solution is correct.
 - An example is the following problem: given a bunch of numbers, can they be split into two groups such that the sum of one group is the same as the other. Clearly, if one is given a solution (two groups of numbers), it's simple to verify that the sum is the same. (This is a partitioning problem).
- **What's unknown is whether problems such as the one above have an efficient algorithm for finding the solution**. This is the (in)famous (unsolved) **P = NP problem**, and the common conjecture is that no such algorithm exists.
- Now, NP hard problems are such problems that were shown that if they can be efficiently solved (which, as mentioned, is believed to not be the case), then each and every problem in NP (each and every problem whose results can be efficiently verified) can be efficiently solved. In other words, if you're up to showing that $P=NP$, you might want to take a stand at any of those NP-hard problems since they are "equivalent" in some way to all others.

Ising Model

- Mathematical model for ferromagnetism in statistical mechanics.
- The energy of spin configuration for a given lattice is given by the following classical Hamiltonian

$$E(s) = - \sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i \quad s = \{s_i\}, \quad s_i \in \{-1,1\}$$

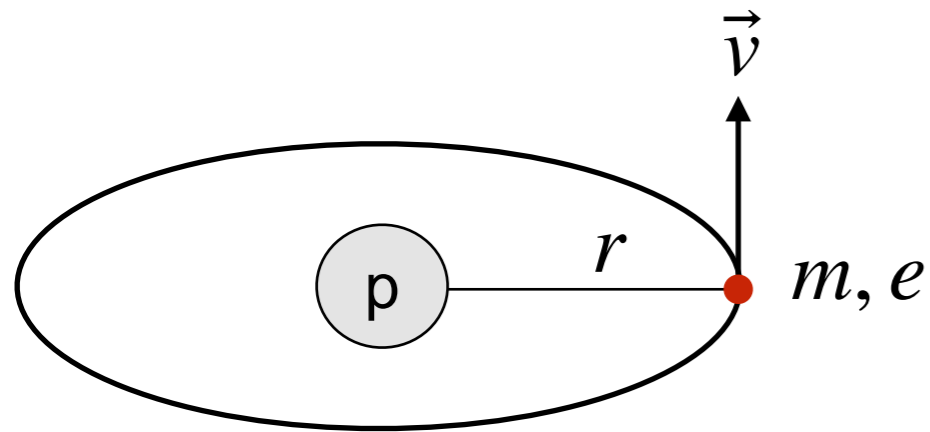
- J_{ij} is called an interaction, spin-spin coupling, and h_i is an external magnetic field, interacting with spin s_i .
- The configuration probability is given by the Boltzmann distribution

$$P(s) = \frac{e^{-\beta H(s)}}{\sum_s e^{-\beta H(s)}}, \quad \beta = \frac{1}{k_B T}$$

- Quantum Ising model:

$$H = - \sum_{i,j} J_{ij} \sigma_i^z \sigma_j^z - \sum_i h_i \sigma_i^z$$

Spin-Spin Interaction



Current loop produced by an electron in circular orbit

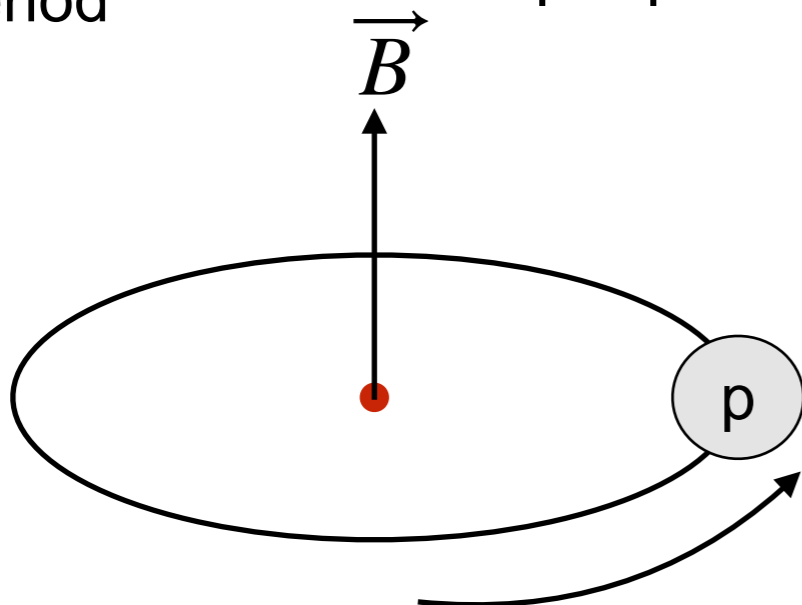
$$I = \frac{e}{T} = e \frac{v}{2\pi r} \quad v = \frac{2\pi r}{T}$$

$$\mu = IA = \frac{ev}{2\pi r} \pi r^2 = \frac{1}{2} evr \quad \longrightarrow \quad \frac{\mu}{L} = \frac{e}{2m}$$

$$\vec{L} = \vec{r} \times \vec{p} \longrightarrow L = mvr$$

μ : magnetic moment
 L : angular momentum
 p : momentum
 m : mass
 v : velocity
 I : current
 T : period

$$\frac{|\vec{\mu}|}{|\vec{L}|} = g = \frac{e}{2m} = \text{gyro magnetic ratio} \quad \vec{\mu} = \frac{e}{2m} (\vec{L} + g \vec{S})$$

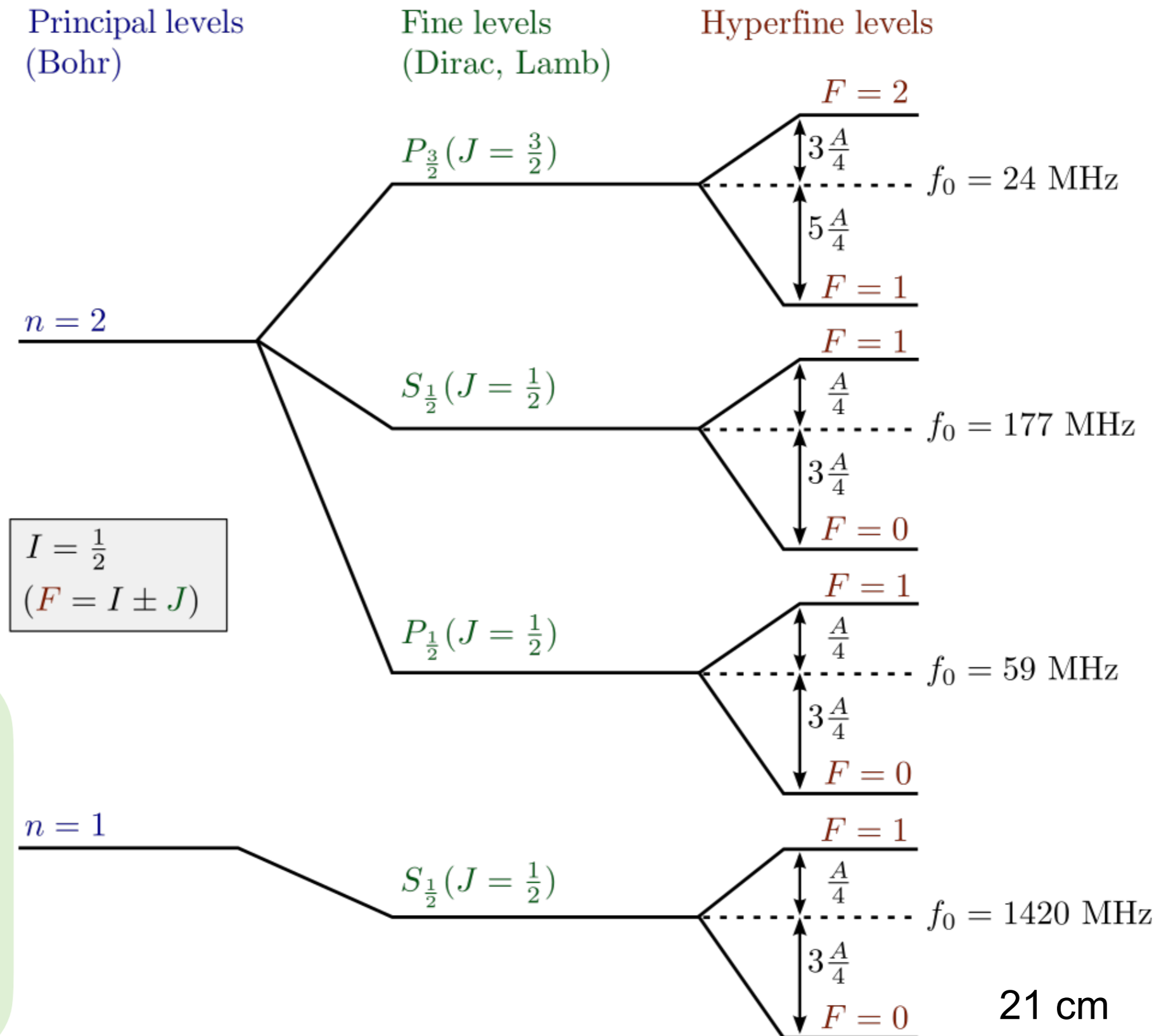


$$H = -\vec{\mu}_e \cdot \vec{B}_p \sim \vec{S}_e \cdot \vec{L}_p$$

(spin – orbital momentum coupling)

$$H \sim -\vec{S}_e \cdot \vec{S}_p \text{ (spin – spin coupling)}$$

Hydrogen Hyperfine Structure

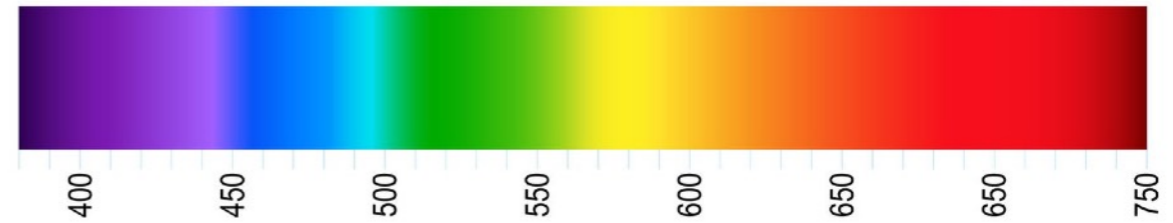


MRI,
solar magnetic field,
Bird migration,
Zeeman effect,
Nuclear and particle
models

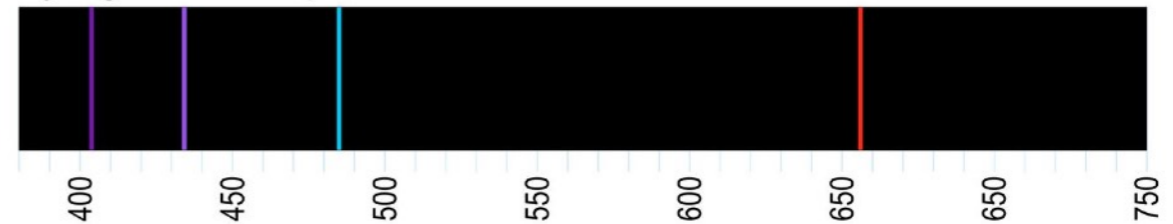
Hydrogen Hyperfine Structure

MRI,
solar magnetic field,
Bird migration,
Zeeman effect,
Nuclear and particle models

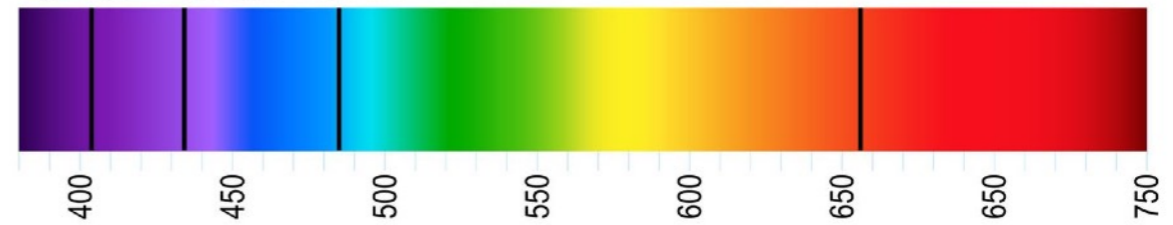
Continuous spectrum



Hydrogen Emission spectrum

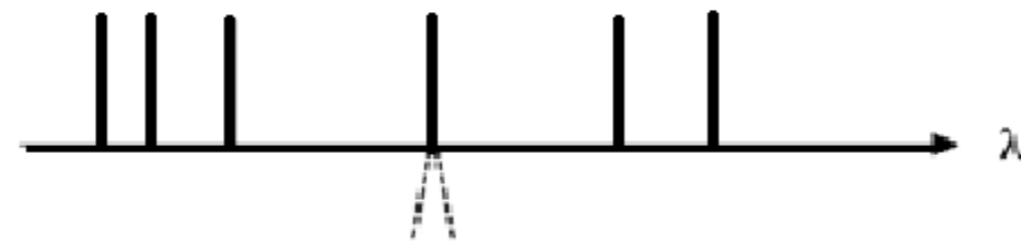


Hydrogen Absorption spectrum



UV visible IR

Gross Structure



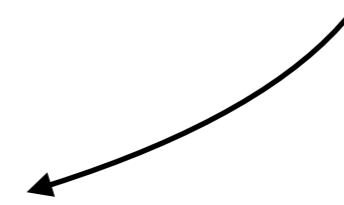
Fine Structure



Hyperfine Structure

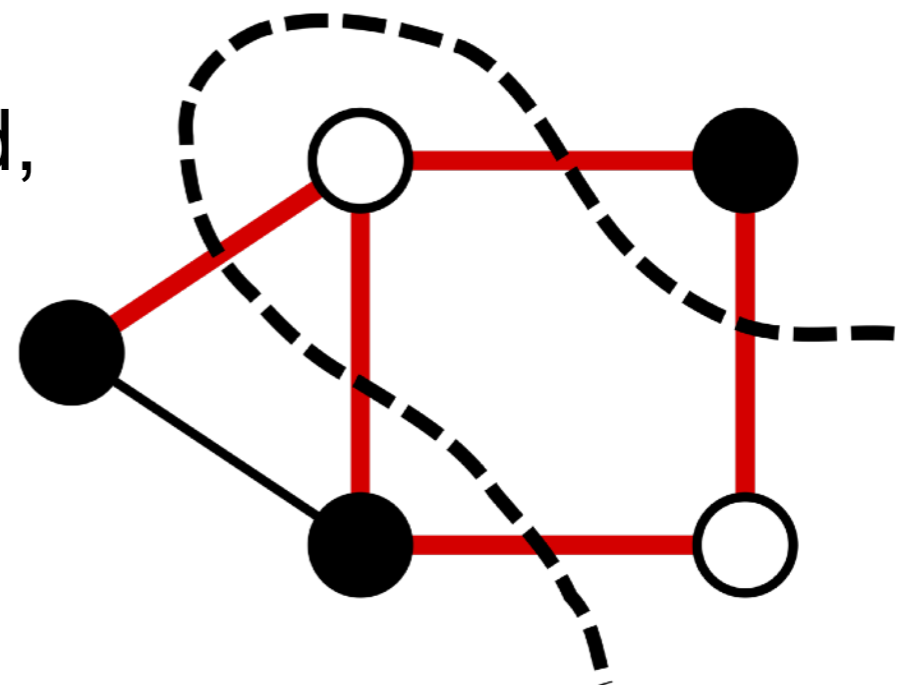


Spectrum



QUBO example: Max-cut Problem

- Max-Cut is the NP-hard problem of finding a partition of the graph's vertices into two distinct sets that maximizes the number of edges between the two sets.
- Undirected Graph: $G = (V, E)$
 - V : set of nodes, and E : set of edges
- Partition vertices into two complementary sets such that the number of edges between the two sets is as large as possible.
- As the Max-Cut Problem is NP-hard, no polynomial-time algorithms for Max-Cut in general graphs are known.



QUBO example: Max-cut Problem

- The cost function to be maximized:

$$C(x) = \sum_{(i,j) \in E} (x_i + x_j - 2x_i x_j) \quad \text{where } x_i \in \{0,1\}$$

$x_i + x_j - 2x_i x_j = 1$, if x_i and x_j belong in different sets.

$x_i + x_j - 2x_i x_j = 0$, if x_i and x_j belong in the same set.

$$s_i \in Z_2 = \{-1,1\}$$

- Introducing $x_i = \frac{s_i + 1}{2}$, the cost function can be rewritten

$$C(s) = \frac{1}{2} \sum_{(i,j) \in E} (1 - s_i s_j) \longrightarrow C(s) = \frac{1}{2} \sum_{(i,j) \in E} (1 - \sigma_i^z \sigma_j^z) \quad \begin{array}{l} (i,j) : \text{the edge index} \\ i : \text{vertex index} \end{array}$$

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{array}{l} \sigma^z |0\rangle = +1 |0\rangle \\ \sigma^z |1\rangle = -1 |1\rangle \end{array}$$

σ_i^z : Pauli's Z matrix acting on the i^{th} vertex

σ_j^z : Pauli's Z matrix acting on the j^{th} vertex

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Matrices = linear operators = observables

Eigenvalues = what are actually measured in experiments

Adiabatic Theorem

- **Adiabatic theorem:** A physical system remains in its instantaneous eigenstate, if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum. (Max Born and Vladimir Folk 1928)
- Under a slowly changing Hamiltonian $H(t)$ with instantaneous eigenstate $|n(t)\rangle$ and the corresponding energy $E(t)$, a quantum system evolves from initial state $|\psi(0)\rangle = \sum_n c_n(0) |n(0)\rangle$ to final state $|\psi(t)\rangle = \sum_n c_n(t) |n(t)\rangle$ where $c_n(t) = c_n(0) e^{i\theta_n(t)} e^{i\gamma_n(t)}$ with the dynamical phase $\theta_n(t) = -\frac{1}{\hbar} \int_0^t E_n(t') dt'$ and geometrical phase $\gamma(t) = i \int_0^t \langle n(t') | \dot{n}(t') \rangle dt'$
- **Adiabatic approximation:** the rate of change of Hamiltonian $\dot{H}(t)$ is small and there is finite gap $E_m(t) - E_n(t) \neq 0$ between energies for $m \neq n \rightarrow$

$$\langle n(t') | \dot{n}(t') \rangle = -\frac{\langle m(t) | \dot{H}(t) | n(t) \rangle}{E_m(t) - E_n(t)} \rightarrow 0$$
- $|c_n(t)|^2 = |c_n(0)|^2$ so if the system begins in an eigenstate of $H(0)$, it remains in an eigenstate of $H(t)$ during the evolution with a change of phase only.

Adiabatic Theorem

$$H(t) |n(t)\rangle = E_n(t) |n(t)\rangle$$

$|n(t)\rangle$: is eigenstates of Hamiltonian, basis

$$|\psi(t)\rangle = \sum_n c_n(t) |n(t)\rangle$$

satisfies time-dependent Schrödinger equation

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

$$\frac{d}{dt} : \quad \dot{H}(t) |n(t)\rangle + H(t) |\dot{n}(t)\rangle = \dot{E}_n(t) |n(t)\rangle + E_n(t) |\dot{n}(t)\rangle$$

Assume $m \neq n$ and perform inner product with $|m(t)\rangle$:

$$H(t) |m(t)\rangle = E_m(t) |m(t)\rangle$$

$$\langle m(t) | n(t) \rangle = \delta_{mn}$$

$$\langle m(t) | \dot{H}(t) | n(t) \rangle + \langle m(t) | H(t) | \dot{n}(t) \rangle = \dot{E}_n(t) \langle m(t) | n(t) \rangle + E_n(t) \langle m(t) | \dot{n}(t) \rangle$$

$$\langle m(t) | \dot{H}(t) | n(t) \rangle + E_m(t) \langle m(t) | \dot{n}(t) \rangle = E_n(t) \langle m(t) | \dot{n}(t) \rangle \rightarrow \langle m(t) | \dot{n}(t) \rangle = - \frac{\langle m(t) | \dot{H}(t) | n(t) \rangle}{E_m(t) - E_n(t)}$$

Adiabatic approximation: the rate of change in Hamiltonian $\dot{H}(t)$ is small and there is finite gap $E_m(t) - E_n(t) \neq 0$ between energies $\rightarrow \langle m(t) | \dot{n}(t) \rangle \approx 0$.

Adiabatic Theorem

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H(t) |\psi(t)\rangle \quad \rightarrow \quad i\hbar \sum_n \dot{c}_n(t) |n(t)\rangle + c_n(t) |\dot{n}(t)\rangle = \sum_n E_n(t) c_n(t) |n(t)\rangle$$

$$|\psi(t)\rangle = \sum_n c_n(t) |n(t)\rangle$$

Inner product with $|m(t)\rangle$: $\langle m(t) | \left[i\hbar \sum_n \dot{c}_n(t) |n(t)\rangle + c_n(t) |\dot{n}(t)\rangle = \sum_n E_n(t) c_n(t) |n(t)\rangle \right]$

Using $\langle m(t) | n(t) \rangle = \delta_{mn}$, we obtain $i\hbar \dot{c}_m(t) + i\hbar \sum_n c_n(t) \langle m(t) | \dot{n}(t) \rangle = c_m(t) E_m(t)$

In the adiabatic limit,
 $\langle m(t) | \dot{n}(t) \rangle \approx 0$ for $m \neq n$

$$i\hbar \dot{c}_m(t) + i\hbar c_m(t) \langle m(t) | \dot{m}(t) \rangle = c_m(t) E_m(t)$$

$$i \dot{c}_m(t) = \left(\frac{E_m(t)}{\hbar} - i \langle m(t) | \dot{m}(t) \rangle \right) c_m(t) \quad \rightarrow \quad \dot{c}_m(t) = i \left(-\frac{E_m(t)}{\hbar} + i \langle m(t) | \dot{m}(t) \rangle \right) c_m(t)$$

$$\frac{d}{dt} \ln c_m(t) = \frac{1}{c_m(t)} \frac{dc_m(t)}{dt} = \frac{\dot{c}_m(t)}{c_m(t)} = -\frac{i}{\hbar} E_m(t) + i \langle m(t) | \dot{m}(t) \rangle$$

$$c_m(t) = c_m(0) e^{i\theta_m(t)} e^{i\gamma_m(t)} \quad \theta_m(t) = -\frac{1}{\hbar} \int_0^t E_m(t') dt' \quad \gamma(t) = i \int_0^t \langle m(t') | \dot{m}(t') \rangle dt'$$

dynamical phase,
real, function of E

geometrical phase,
pure imaginary

Adiabatic Theorem

$$c_m(t) = c_m(0)e^{i\theta_m(t)}e^{i\gamma_m(t)}$$

$$\theta_m(t) = -\frac{1}{\hbar} \int_0^t E_m(t') dt'$$

$$\gamma(t) = i \int_0^t \langle m(t') | \dot{m}(t') \rangle dt'$$

dynamical phase,
real, function of E

geometrical phase,
pure imaginary,
Has something to do with
direction in the Hilbert space

$$\begin{aligned} 0 &= \frac{d}{dt} \langle m(t) | m(t) \rangle = \langle \dot{m}(t) | m(t) \rangle + \langle m(t) | \dot{m}(t) \rangle \\ &= \langle m(t) | \dot{m}(t) \rangle^* + \langle m(t) | \dot{m}(t) \rangle \\ &= 2 \operatorname{Re} \langle m(t) | \dot{m}(t) \rangle \quad \rightarrow \quad \gamma_m(t) : \text{ pure imaginary} \end{aligned}$$

Adiabatic Theorem

- Schrodinger equation: $i\hbar \frac{d\psi(t)}{dt} = H(t) \psi(t)$
- Instantaneous eigenstate: $H(t) \psi_n(t) = E_n(t) \psi_n(t)$
- Initial condition: $\psi(t = 0) = \psi_0$
- If evolution is slow enough, $\psi(t) \approx e^{i\theta(t)} \psi_0$

Born and Folk 1928

$$\psi(t) = U(t) \psi(0)$$

$$U_I(t) = \sum_{q=0}^{\infty} (-i)^q \int_0^t dt_q \cdots \int_0^{t_2} dt_1 H_I(t_q) \cdots H_I(t_1)$$

Quantum Annealing

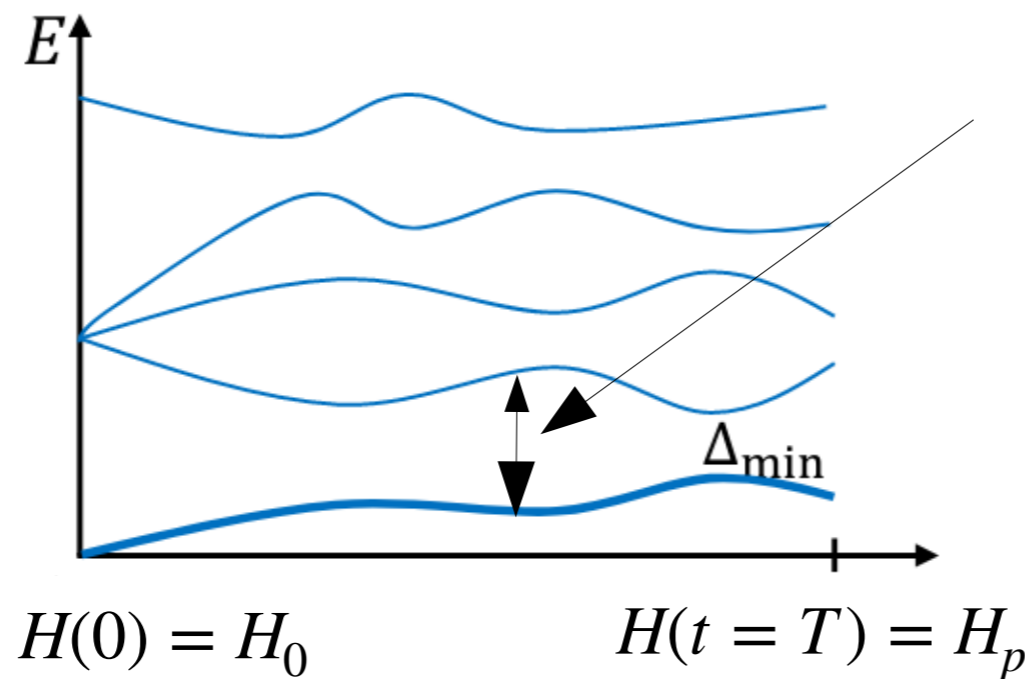
- H_p is the problem Hamiltonian whose ground state encodes the solution to the optimization problem
- H_0 is the initial Hamiltonian whose ground state is easy to prepare.
- Prepare a quantum system to be in the ground state of H_0 and evolve the system using the following time-dependent Hamiltonian,

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_p$$

- The system will remain to its ground state at all times, which means for $t=T$, the system will be in the ground state of H_p , our problem Hamiltonian.
- D-wave has built Quantum Annealing that solves optimization problem by transferring the original optimization to a hardware, that allows nearest neighbor interaction of qubits.
- If the energy gap b/w the ground state and 1st excited state is small, T must be very large \rightarrow computationally difficult.

Limitation of Quantum Annealing

- Performance of quantum annealing are governed by the size of the gap.



$$\Delta t \gg \max_{0 \leq s \leq 1} \left| \frac{\langle 1(s) | \frac{dH(s)}{ds} | 0(s) \rangle}{\Delta(s)^2} \right|$$

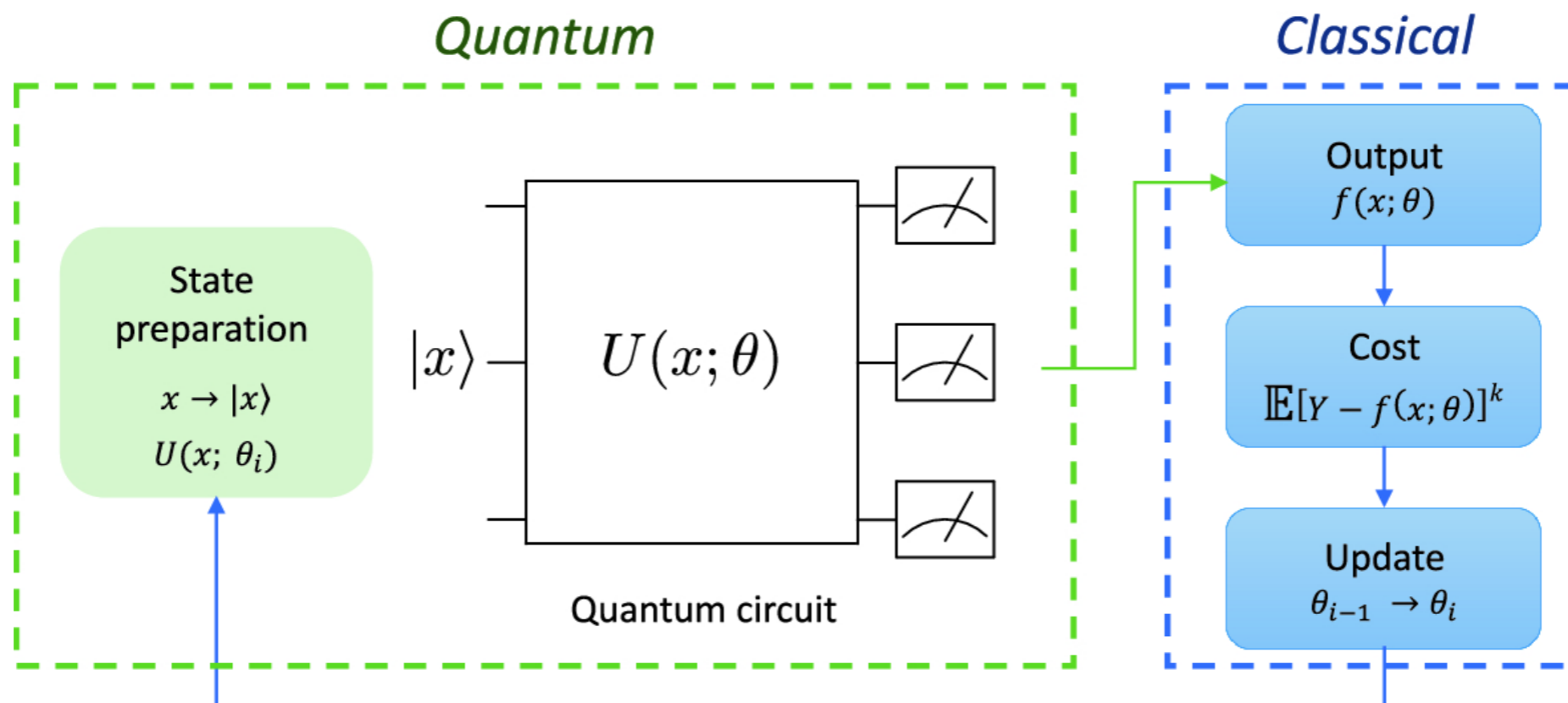
$$T = \Delta s, \quad 0 \leq s \leq 1$$

- Performance is poor, when eigenvalues are degenerate.

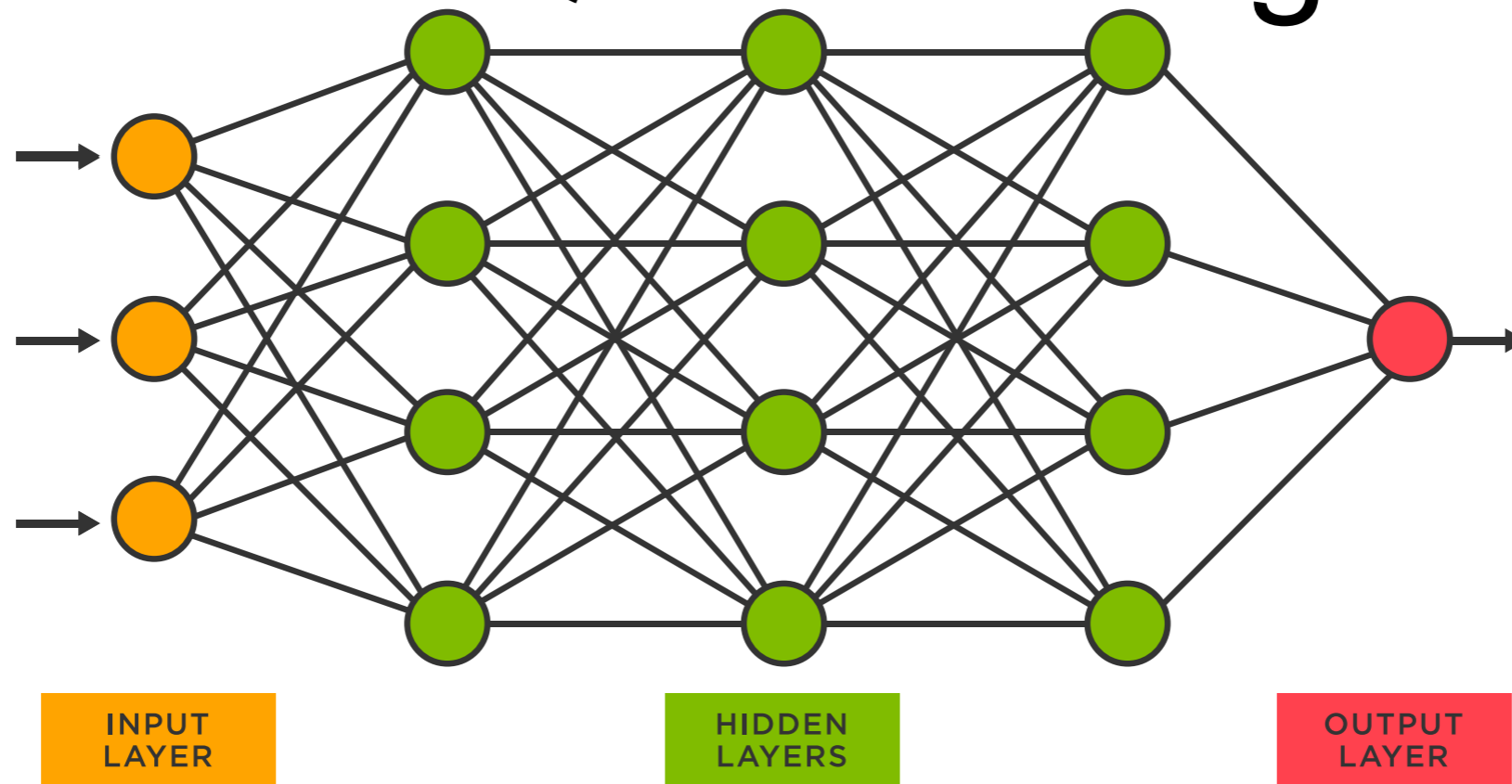
Variational Quantum Algorithms

- Hybrid quantum-classical model is suggested to circumvent the issue of going slow with quantum annealer as well as implementing Hamiltonian in the available hardware.
- Quantum: parameterize wave function
- Classical: minimize/maximize the expectation value of H in the problem.

$$E(\vec{\theta}) = \langle \psi(\vec{\theta}) | H | \psi(\vec{\theta}) \rangle$$

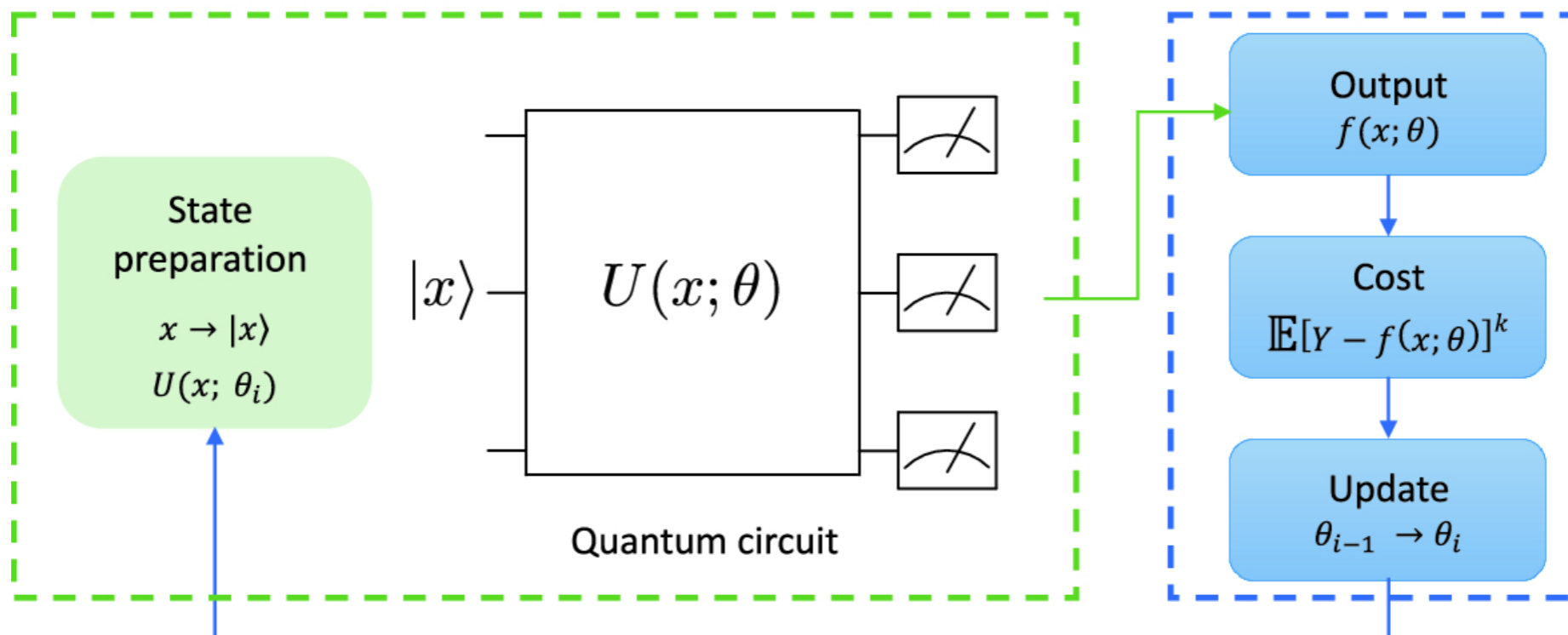


Variational Quantum Algorithms



Quantum

Classical



Variational Quantum Algorithms

- 2016: first cloud-based quantum computer became available.
- Current state-of-the-art device size ranges from 50 to 100 qubits which allows one to achieve ‘quantum supremacy’: outperforming the best classical supercomputer, for certain contrived mathematical tasks.
 - Sycamore (53 qubits, corresponding to a computational state-space of dimension $2^{53} \approx 10^{16}$): 200 seconds vs 10,000 years for sampling the output of a pseudo-random quantum circuit.
- Variational Quantum Algorithms (VQAs) have emerged as the leading strategy to obtain quantum advantage on NISQ (Noisy Intermediate-Scale Quantum) devices. Accounting for all of the constraints imposed by NISQ computers with a single strategy requires an optimization-based or learning-based approach, precisely what VQAs use.
- VQAs are arguably the quantum analog of highly successful machine-learning methods, such as neural networks.
- VQAs leverage the toolbox of classical optimization, since VQAs use parametrized quantum circuits to be run on the quantum computer, and then outsource the parameter optimization to a classical optimizer. This approach has the added advantage of keeping the quantum circuit depth shallow and hence mitigating noise, in contrast to quantum algorithms developed for the fault-tolerant era.

Quantum Approximate Optimization Algorithm (QAOA)

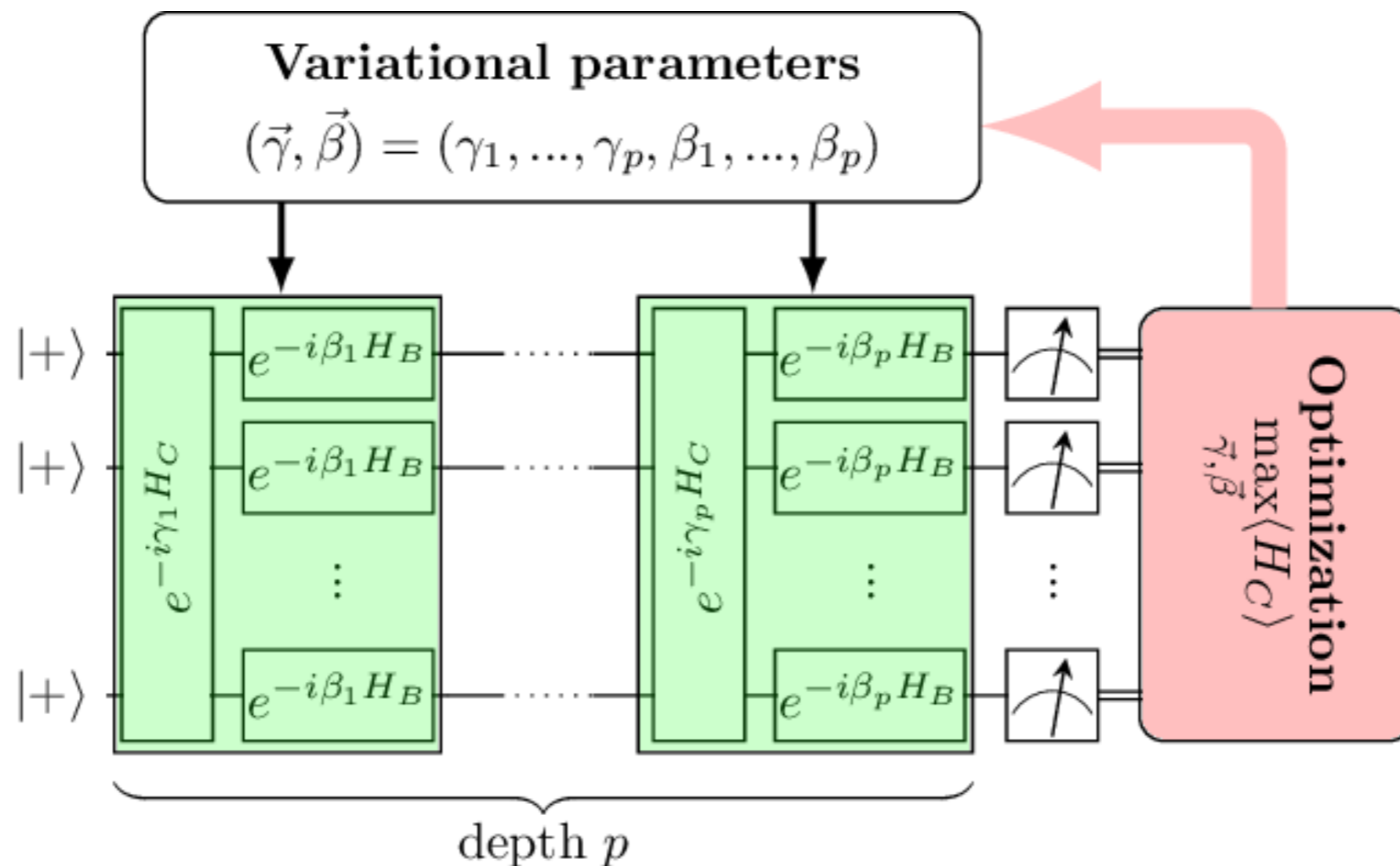
1411.4028 E. Farhi, J. Goldstone, S. Gutmann

- **Abstract:** We introduce a quantum algorithm that produces approximate solutions for combinatorial optimization problems. The algorithm depends on a positive integer p and the quality of the approximation improves as p is increased. The quantum circuit that implements the algorithm consists of unitary gates whose locality is at most the locality of the objective function whose optimum is sought. The depth of the circuit grows linearly with p times (at worst) the number of constraints. If p is fixed, that is, independent of the input size, the algorithm makes use of efficient classical preprocessing. If p grows with the input size a different strategy is proposed. We study the algorithm as applied to MaxCut on regular graphs and analyze its performance on 2-regular and 3-regular graphs for fixed p . For $p = 1$, on 3-regular graphs the quantum algorithm always finds a cut that is at least 0.6924 times the size of the optimal cut.

Quantum Approximate Optimization Algorithm (QAOA)

Farhi et al 2014

- Hybrid quantum algorithm: contains a parameterized quantum circuit which depends on variational parameters.
- Use classical computer to optimize the output of the quantum circuit.
- Consider the Ising model for illustration.



Quantum Approximate Optimization Algorithm (QAOA)

Farhi et al 2014

$$H_P = C(s) = \frac{1}{2} \sum_{(i,j) \in E} \left(1 - \sigma_i^z \sigma_j^z \right) : \text{Problem Hamiltonian}$$

(i, j) : the edge index

i : vertex index

$$H_M = B = \sum_j \sigma_j^X : \text{Mixer Hamiltonian}$$

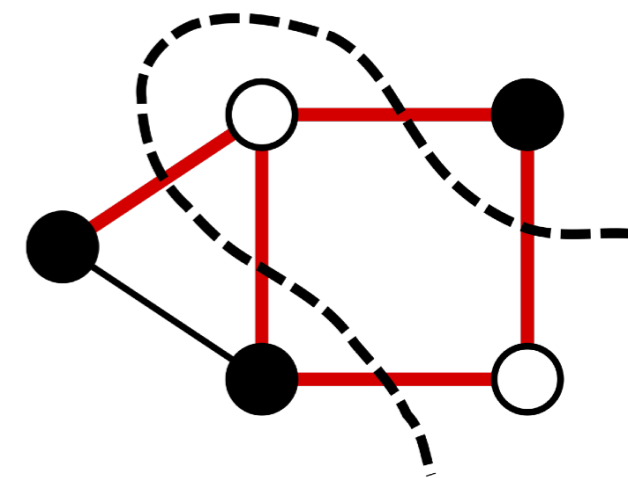
Full Hamiltonian: $H(t) = \left(1 - \frac{t}{T} \right) H_M + \frac{t}{T} H_P$

$$|\psi\rangle = \exp \left[-i \int_0^t H(t') dt' \right] |\psi_0\rangle = \exp \left[-i \sum_{j=1}^p H(j\Delta t) \Delta t \right] |\psi_0\rangle$$

$$\approx \prod_{j=1}^p \exp \left[-i\Delta t \left[\left(1 - \frac{j\Delta t}{T} \right) H_M + \frac{j\Delta t}{T} H_P \right] \right] |\psi_0\rangle$$

$$\approx \prod_{j=1}^p \exp \left[-i\Delta t \left(1 - \frac{j\Delta t}{T} \right) H_M \right] \exp \left[-i\Delta t \frac{j\Delta t}{T} H_P \right] |\psi_0\rangle$$

$$= \prod_{j=1}^p \underbrace{\exp \left[-i\beta_j H_M \right]}_{U(H_M, \beta_j)} \underbrace{\exp \left[-i\gamma_j H_P \right]}_{U(H_P, \gamma_j)} |\psi_0\rangle \quad \Rightarrow \quad |\gamma, \beta\rangle = \prod_{j=1}^p U(H_M, \beta_j) U(H_P, \gamma_j) |+\rangle^{\otimes n}$$



Undirected Graph: $G = (V, E)$
 V : set of nodes
 E : set of edges

Works in the adiabatic limit or $p \rightarrow \infty$

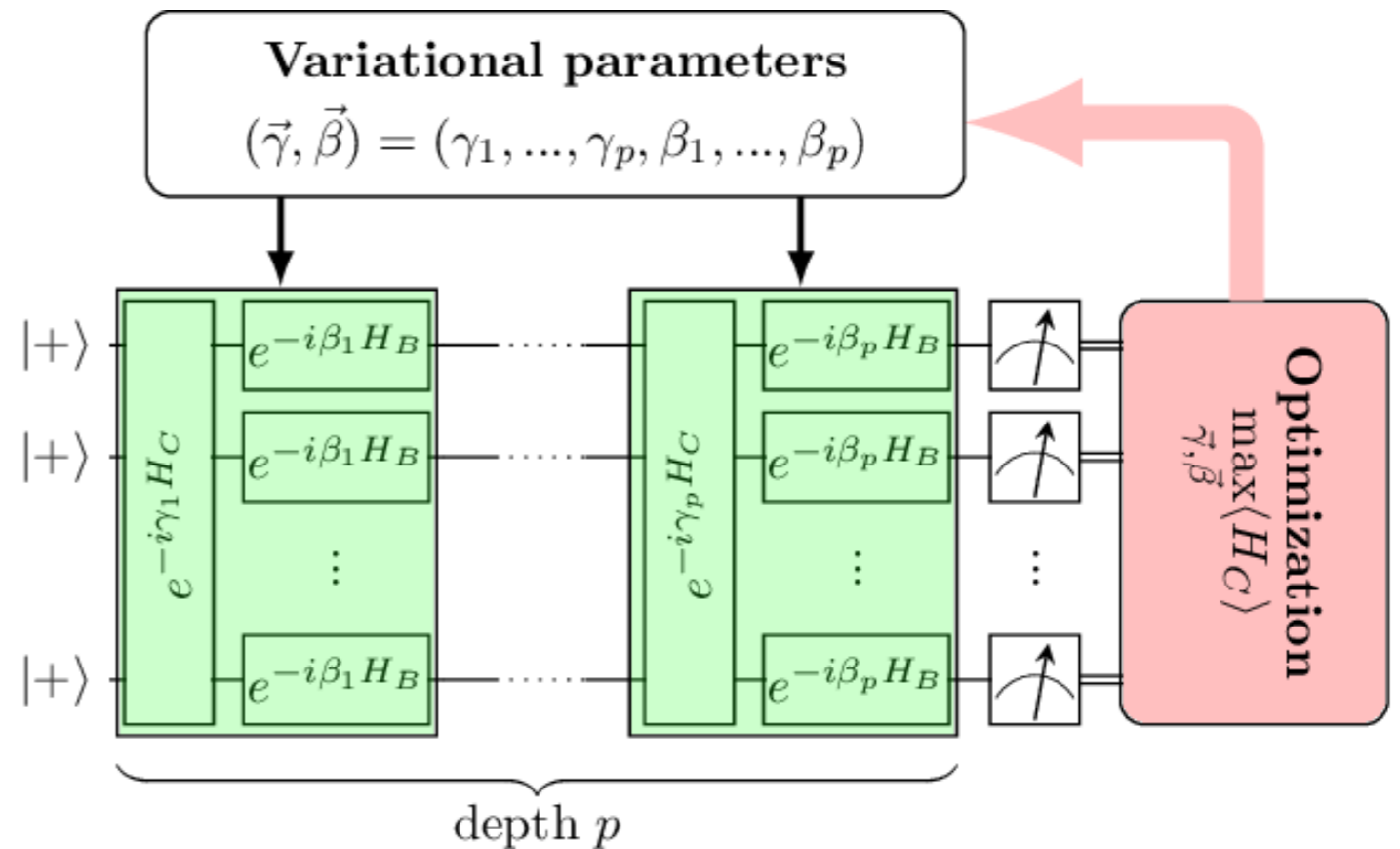
Quantum Approximate Optimization Algorithm (QAOA)

$$H(t) = \left(1 - \frac{t}{T}\right) H_M + \frac{t}{T} H_P$$

$H_M = B$: mixer Hamiltonian

$H_P = C$: problem Hamiltonian

$$|\gamma, \beta\rangle = \prod_{j=1}^p U(H_M, \beta_j) U(H_P, \gamma_j) |+\rangle^{\otimes n}$$



$$U(H_P, \gamma_i) = \exp \left[-i\gamma_i \sum_{j,k \in E} \sigma_j^z \sigma_k^z \right] = \prod_{j,k \in E} \text{CNOT}_{j,k} R_z^k(2\gamma_i) \text{CNOT}_{j,k}$$

$$U(H_M, \beta_j) = \exp \left[-i\beta_j \sum_{i=1}^n \sigma_i^X \right]$$

$$\left. \begin{array}{c} |j\rangle \\ |k\rangle \end{array} \right\} \left. \begin{array}{c} \text{CNOT}_{j,k} \\ R_z^k(2\gamma_i) \\ \text{CNOT}_{j,k} \end{array} \right\} \exp \left[-i(-1)^{j+k} \gamma_i \right] |jk\rangle$$

$$\begin{aligned} &= \prod_{i=1}^n e^{-i\beta_j \sigma_i^X} \\ &= \prod_{i=1}^n R_x^i(2\beta_j) \end{aligned}$$

$$F_p(\gamma, \beta) = \langle \gamma, \beta | H_P | \gamma, \beta \rangle$$

Quantum Approximate Optimization Algorithm (QAOA)

Farhi et al 2014

$$C(s) = \frac{1}{2} \sum_{(i,j) \in E} (1 - \sigma_i^z \sigma_j^z), \quad B = \sum_j \sigma_j^X$$

(i, j) : the edge index
 i : vertex index

$$\sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

σ_i^z : Pauli's Z matrix acting on the i^{th} vertex

σ_j^z : Pauli's Z matrix acting on the j^{th} vertex

$$\sigma^z |0\rangle = +1 |0\rangle \quad \sigma^z |1\rangle = -1 |1\rangle \quad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$U(C, \gamma) = e^{-i\gamma C} = \prod_{(i,j) \in E} e^{-i\gamma C_{ij}}, \quad U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^n e^{-i\beta B_j}$$

$$|\psi(\vec{\gamma}, \vec{\beta})\rangle = \left[\prod_{i=1}^p U(B, \beta_i) U(C, \gamma_i) \right] H^{\otimes n} |0\rangle$$

$$e^{-i\beta \sigma_j^X} = \cos \beta - i \sigma_j^X \sin \beta$$

$$= U(B, \beta_p) U(C, \gamma_p) \cdots U(B, \beta_1) U(C, \gamma_1) \frac{1}{\sqrt{2}^n} \sum_{i=1}^{2^n-1} |i\rangle$$

2p angles (parameters): $\vec{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_p)$, and $\vec{\beta} = (\beta_1, \beta_2, \dots, \beta_p)$

Goal is to find minimum/maximum over angles:

$$M_p = \max_{\vec{\gamma}, \vec{\beta}} \langle \psi(\vec{\gamma}, \vec{\beta}) | C | \psi(\vec{\gamma}, \vec{\beta}) \rangle$$

Quantum Approximate Optimization Algorithm (QAOA)

- How do $U(C, \gamma)$ and $U(B, \beta)$ operate on $|\psi\rangle$?

$$\begin{aligned}
 U(C, \gamma) H^{\otimes n} |0 \dots 0\rangle &= e^{-i\gamma C} H^{\otimes n} |0 \dots 0\rangle = \exp\left[-i\gamma \frac{1}{2} \sum_{(i,j) \in E} (1 - \sigma_i^Z \sigma_j^Z)\right] H^{\otimes n} |0 \dots 0\rangle \\
 &= \prod_{(i,j) \in E} \exp\left[-i\gamma \frac{1}{2} (1 - \sigma_i^Z \sigma_j^Z)\right] H^{\otimes n} |0 \dots 0\rangle
 \end{aligned}$$

$$\exp\left[-i\gamma \frac{1}{2} (1 - \sigma_i^Z \sigma_j^Z)\right] H^{\otimes n} |0 \dots 0\rangle = \exp\left(-i\frac{\gamma}{2}\right) \exp\left(+i\frac{\gamma}{2} \sigma_i^Z \sigma_j^Z\right) H^{\otimes n} |0 \dots 0\rangle$$

Four different possibilities:

$$\begin{aligned}
 \exp\left(+i\frac{\gamma}{2} \sigma_i^Z \sigma_j^Z\right) |\dots 0 \dots 0 \dots\rangle &= \exp\left(+i\frac{\gamma}{2} 1 \cdot 1\right) |\dots 0 \dots 0 \dots\rangle \\
 \exp\left(+i\frac{\gamma}{2} \sigma_i^Z \sigma_j^Z\right) |\dots 0 \dots 1 \dots\rangle &= \exp\left(-i\frac{\gamma}{2} 1 \cdot 1\right) |\dots 0 \dots 1 \dots\rangle \\
 \exp\left(+i\frac{\gamma}{2} \sigma_i^Z \sigma_j^Z\right) |\dots 1 \dots 0 \dots\rangle &= \exp\left(-i\frac{\gamma}{2} 1 \cdot 1\right) |\dots 1 \dots 0 \dots\rangle \\
 \exp\left(+i\frac{\gamma}{2} \sigma_i^Z \sigma_j^Z\right) |\dots 1 \dots 1 \dots\rangle &= \exp\left(+i\frac{\gamma}{2} 1 \cdot 1\right) |\dots 1 \dots 1 \dots\rangle
 \end{aligned}$$

Quantum Approximate Optimization Algorithm (QAOA)

- If bits i and j are the same, $\exp\left[-i\gamma\frac{1}{2}\left(1 - \sigma_i^Z\sigma_j^Z\right)\right]|\dots\rangle = +1|\dots\rangle$
- If bits i and j are different, $\exp\left[-i\gamma\frac{1}{2}\left(1 - \sigma_i^Z\sigma_j^Z\right)\right]|\dots\rangle = e^{-i\gamma}|\dots\rangle$

- ➔ • If bits i and j are different, rotate the output state around z axis by an angle γ .

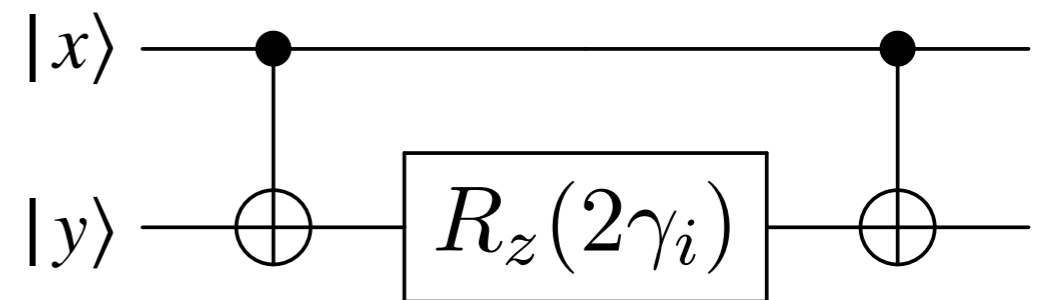
$$\sigma_Z|a\rangle = (-1)^a|a\rangle \quad R_Z(\theta) = \exp\left(-i\frac{\theta}{2}\sigma_Z\right) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{+i\theta/2} \end{pmatrix}$$

In circuit:

$$\text{CNOT}|xy\rangle = |xx \oplus y\rangle$$

$$I \otimes R_z(2\gamma)|xx \oplus y\rangle = \exp\left(-i\gamma(-1)^{x \oplus y}\right)|xx \oplus y\rangle$$

$$\text{For } U(B, \beta) = e^{-i\beta B} = \prod_{j=1}^n e^{-i\beta\sigma_j^X} = \prod_{j=1}^n R_x^j(2\beta)$$

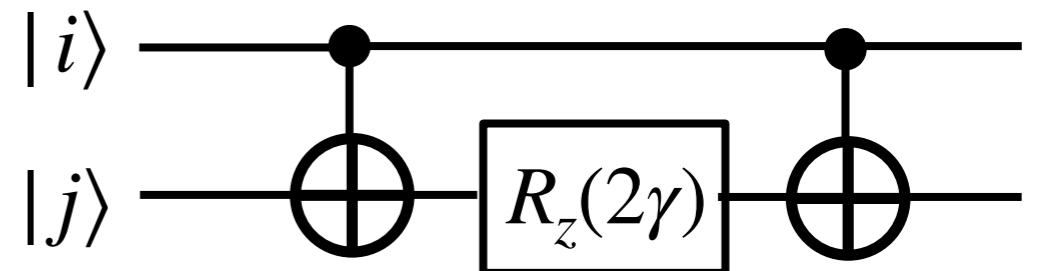


- Rotation of all n -qubits about x -axis with angle 2β

Quantum Approximate Optimization Algorithm (QAOA)

$$|\psi(\vec{\gamma}, \vec{\beta})\rangle = U(B, \beta_p) U(C, \gamma_p) \cdots U(B, \beta_1) U(C, \gamma_1) \frac{1}{\sqrt{2^n}} \sum_{i=1}^{2^n-1} |i\rangle$$

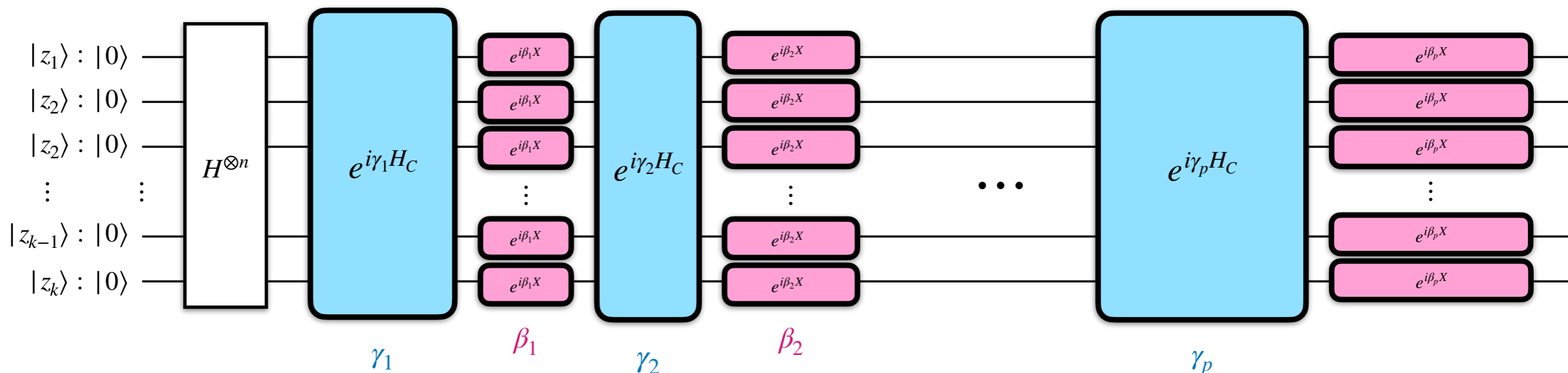
$$|\psi(\vec{\gamma}, \vec{\beta})\rangle = \left[\prod_{i=1}^p U(B, \beta_i) U(C, \gamma_i) \right] H^{\otimes n} |0\rangle$$



$$e^{-i\beta\sigma_j^X} = \cos\beta - i\sin\beta\sigma_j^X = R_x^j(2\beta)$$

Rotate qubit j around x-axis by 2β

$$C(s) = \frac{1}{2} \sum_{(i,j) \in E} \left(1 - \sigma_i^z \sigma_j^z \right), \quad B = \sum_j \sigma_j^X$$



Quantum Approximate Optimization Algorithm (QAOA)

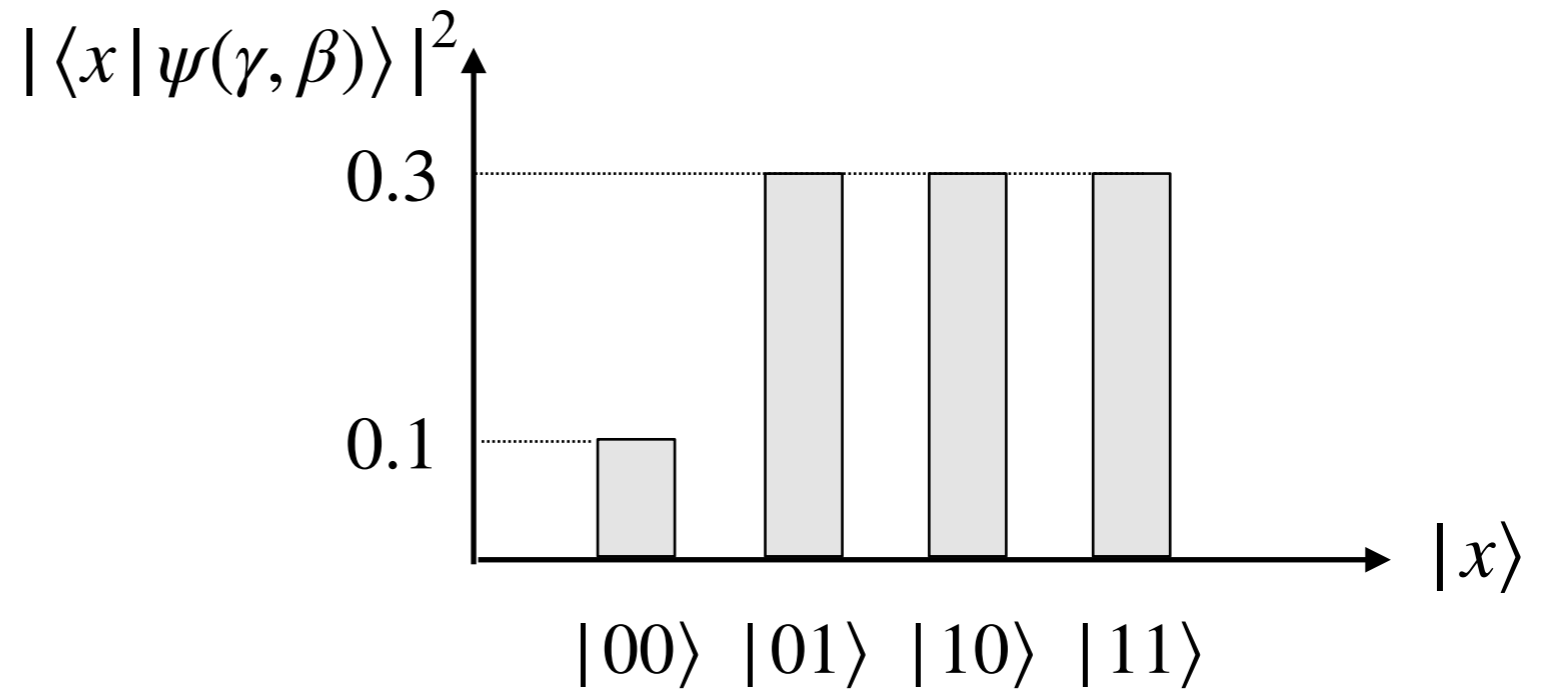
$$|\psi(\vec{\gamma}, \vec{\beta})\rangle = U(B, \beta_p) U(C, \gamma_p) \cdots U(B, \beta_1) U(C, \gamma_1) \frac{1}{\sqrt{2^n}} \sum_{i=1}^{2^n-1} |i\rangle$$

For $n = 2$ and $p = 1$, $|\psi(\vec{\gamma}, \vec{\beta})\rangle = \delta_0(\gamma, \beta) |00\rangle + \delta_1(\gamma, \beta) |01\rangle + \delta_2(\gamma, \beta) |10\rangle + \delta_3(\gamma, \beta) |11\rangle$

$$C = \frac{1}{2} \sum_{(i,j) \in E} (1 - \sigma_i^z \sigma_j^z)$$

$$C = \sum_{x \in \{0,1\}^{\otimes n}} C(x) |x\rangle \langle x|$$

$$|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=1}^{2^n-1} |i\rangle$$



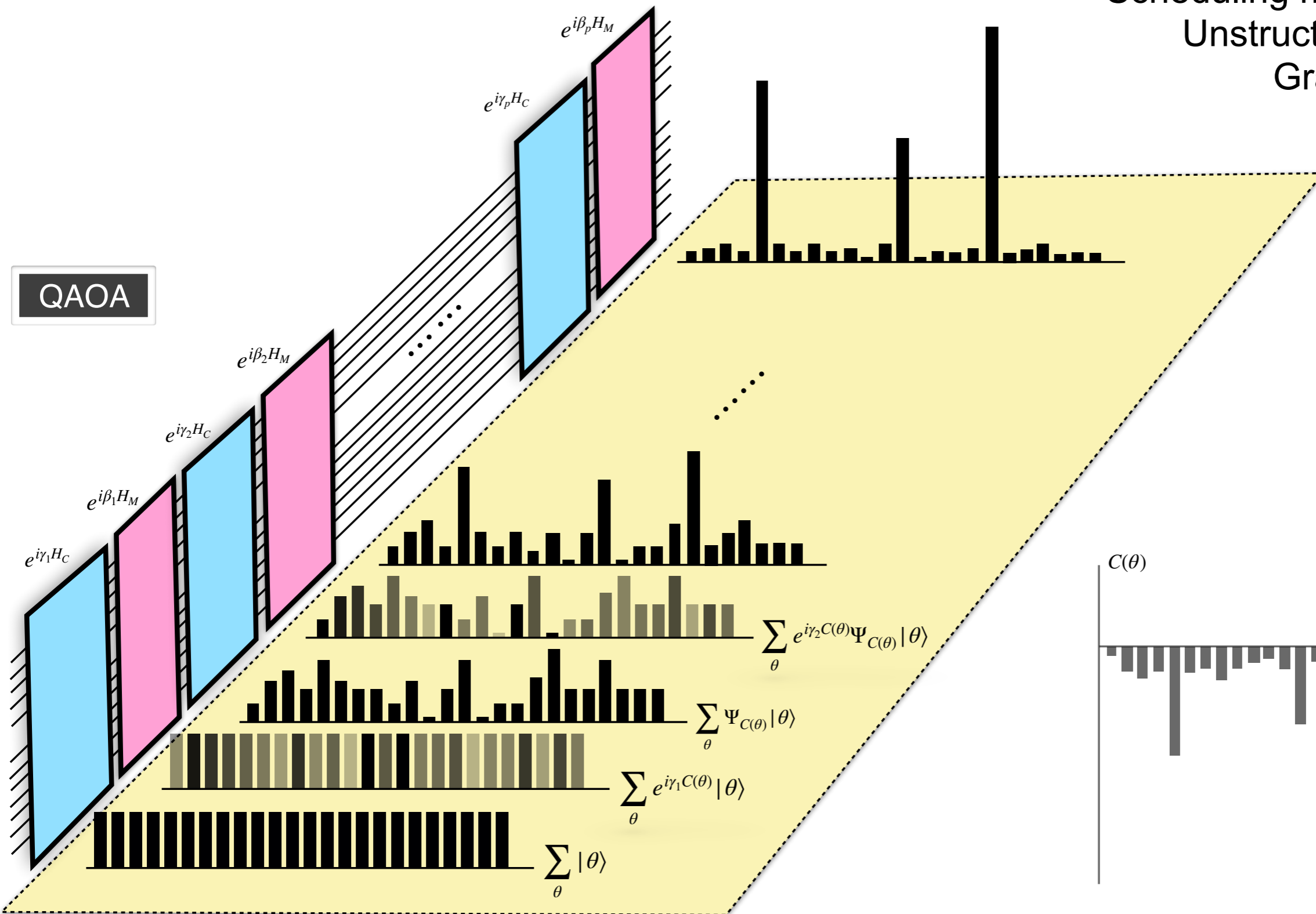
$$F_p(\gamma, \beta) = \max_{\gamma, \beta} \langle \psi(\gamma, \beta) | C | \psi(\gamma, \beta) \rangle = \sum_{x \in \{0,1\}^{\otimes n}} C(x) \left| \langle x | \psi(\gamma, \beta) \rangle \right|^2$$

- Measure of how good the approximation is to actual best value of the cost function

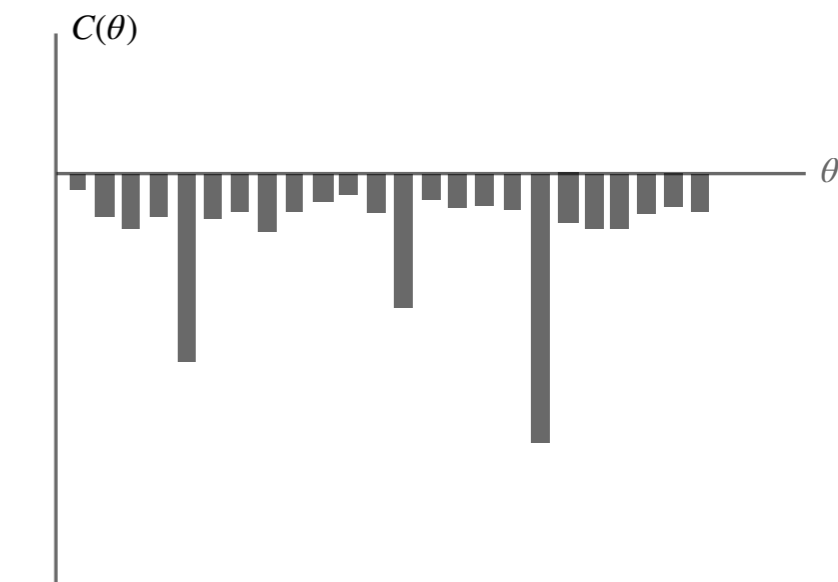
$$\alpha = \frac{F_p(\vec{\gamma}, \vec{\beta})}{C_{max}}$$

QAOA

- Maximum Likelihood detection
- Traveling salesman problem
- Scheduling management
- Unstructured search
- Graph coloring
- Max-cut



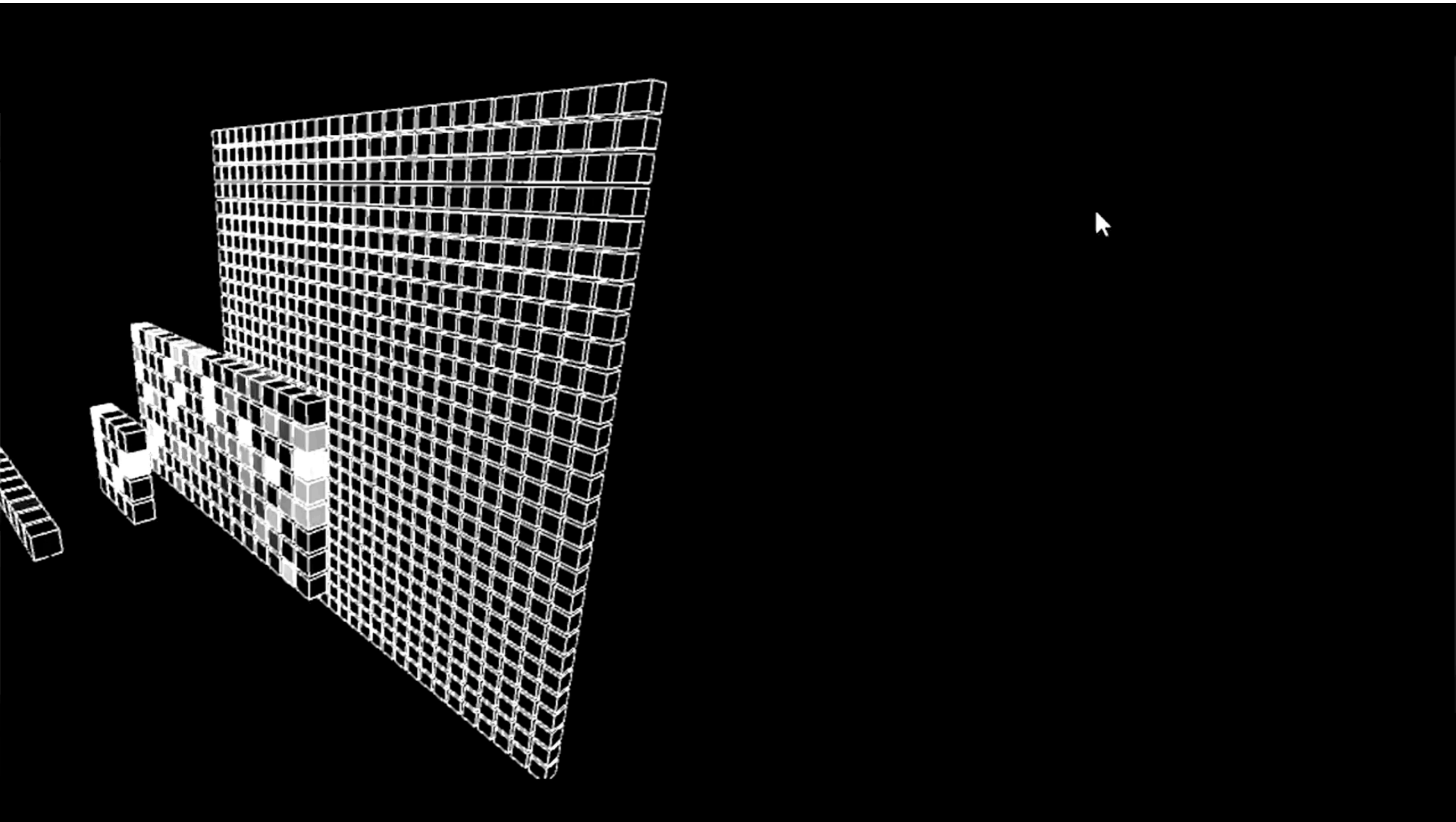
QAOA



Example: Max Cut

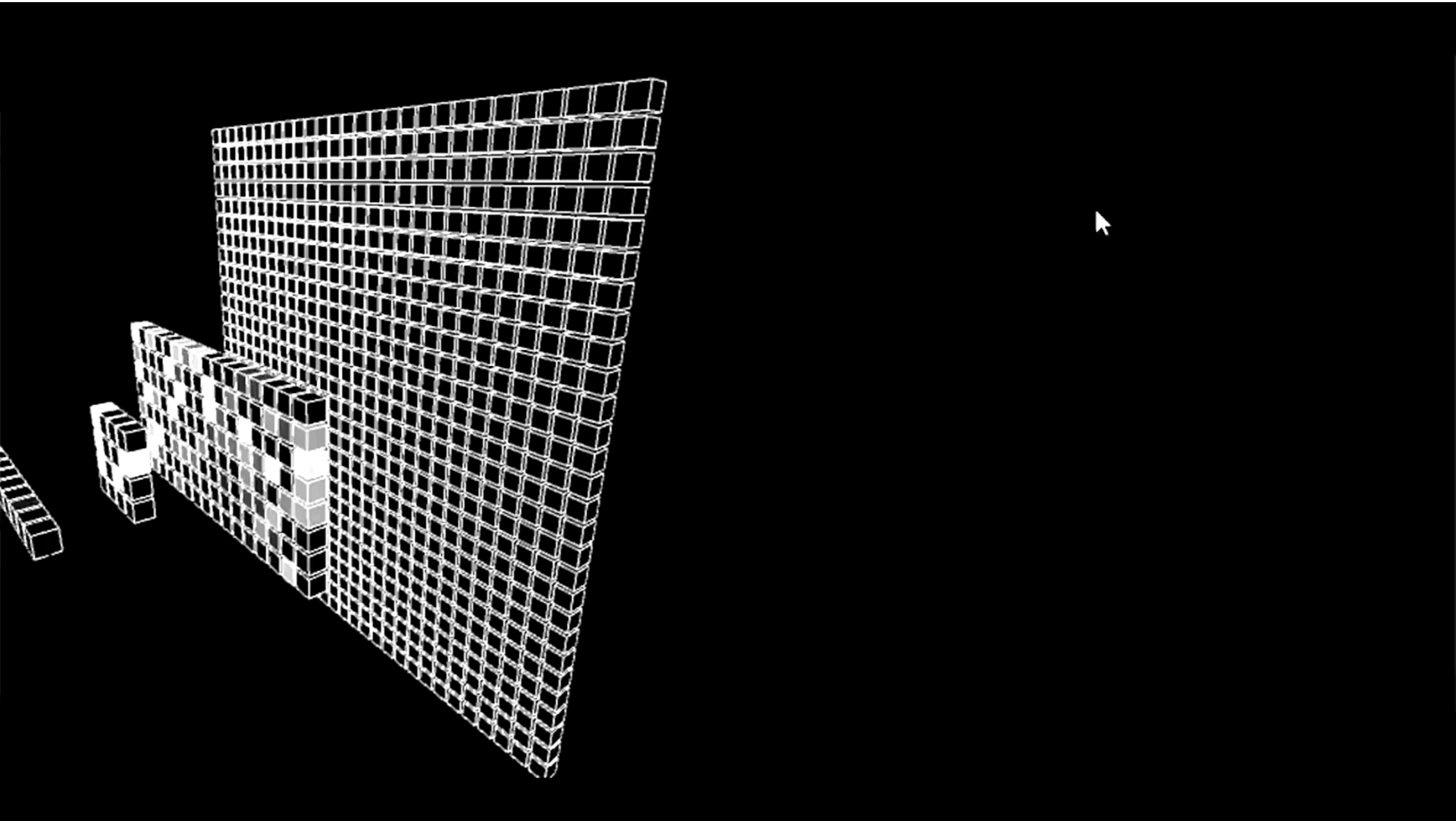
Object identification

Taken from Vecanoi (Youtube Educational channel about AI)



Object identification

Taken from Vecanoi (Youtube Educational channel about AI)



QAOA summary

- One can solve the optimization problems on a quantum computer by initializing the quantum device in the ground state of a hamiltonian that is easy to prepare and adiabatically tuning H into the problem Hamiltonian.
- In a digital quantum computer, this translates into a Trotterized version of the adiabatic evolution operator. In the limit of an infinite product, this Trotterized form becomes exact.
- QAOA is a hybrid quantum-classical variational algorithm with a finite order version of the evolution operator.
- Many experimental and theoretical studies, suggesting QAOA may provide a significant quantum advantage over classical algorithms, and that it is computationally universal.

Limitations and potential issues with QAOA

- The performance improves with the number of alternating layers in the Ansatz, which is limited by coherences times in existing and near-term quantum processors.
- More layers implies more variational parameters (challenging for classical optimizers) [19].
- Short-depth ansatz is not really the digitized version of the adiabatic problem but rather an adhoc ansatz, which does not guarantee to perform optimally [20].
- Fixed form of standard QAOA is not optimal but no systematic approach for finding a better ansatz.
- ADAPT-QAOA converges faster, reducing the required number of CNOT gates and optimization parameters.
- Connection to concept of shortcuts to adiabaticity.
- Inspired by ADAPT-VQE [28,29] (Refs in 2005.10258).