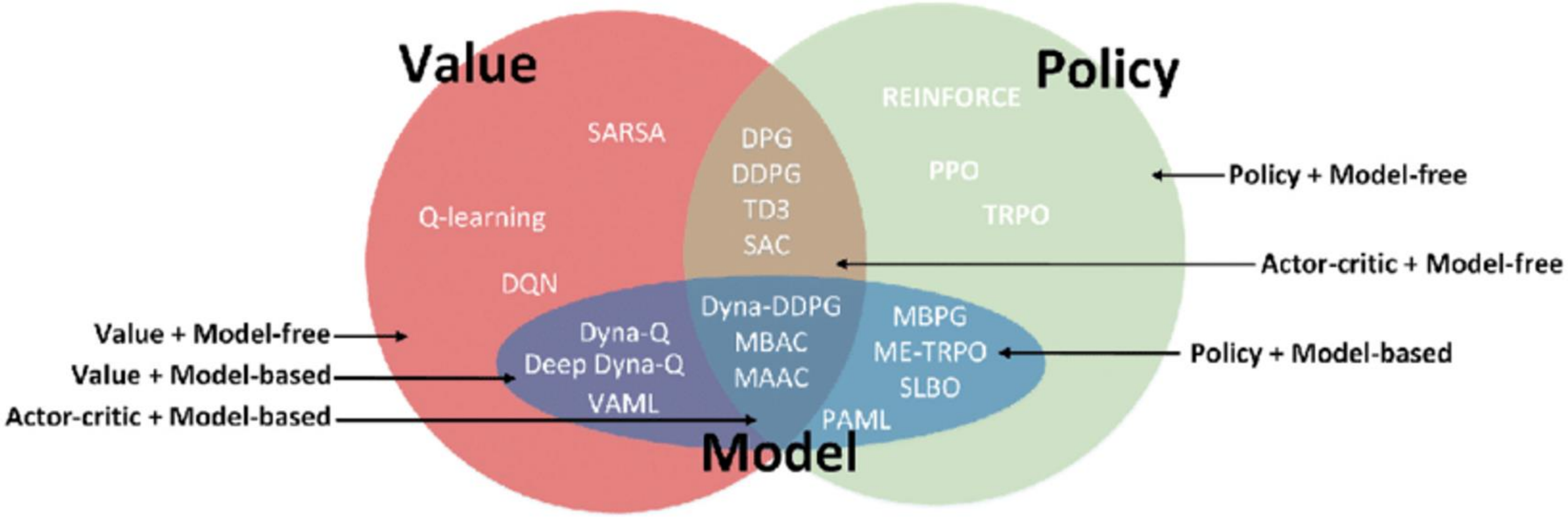
Value-based reinforcement learning for quantum control

Popular approaches in reinforcement learning



For example,

- **Q-learning** is a value-based method: it learns optimal value functions
- **Policy Gradient**, such as TRPO: it learns optimal policy, works well for continuous or high-dimensional actions

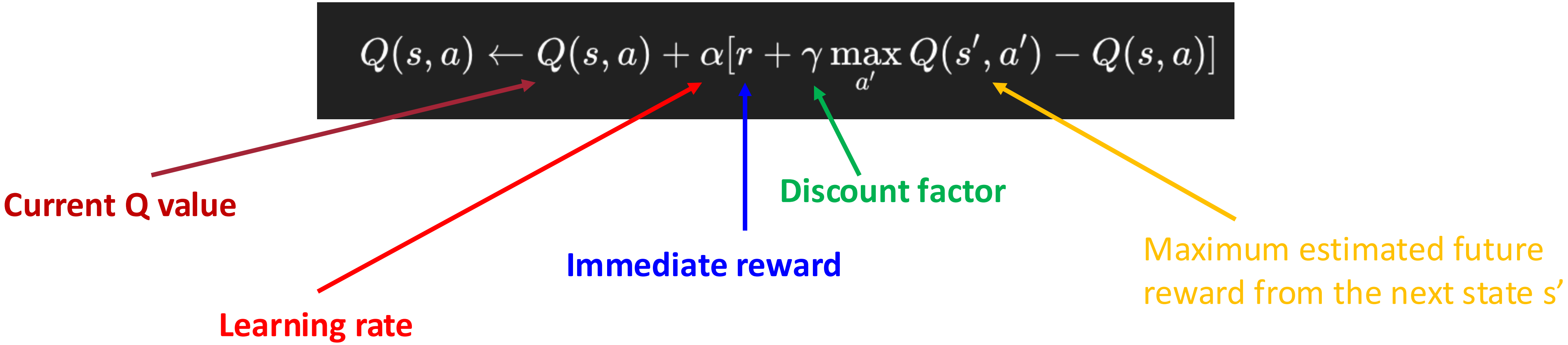
Q-learning overview

What is Q-learning?

- A **model-free**, **value-based** reinforcement learning
- It learns an **action-value function Q(s, a)** to evaluate the expected return from taking action *a* in a state *s* and following the best policy afterward.

- **No model of the environment needed** – just observe state transition and rewards

- **Bellman update equation** (core idea of Q-learning):



$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

Current Q value

Discount factor

Immediate reward

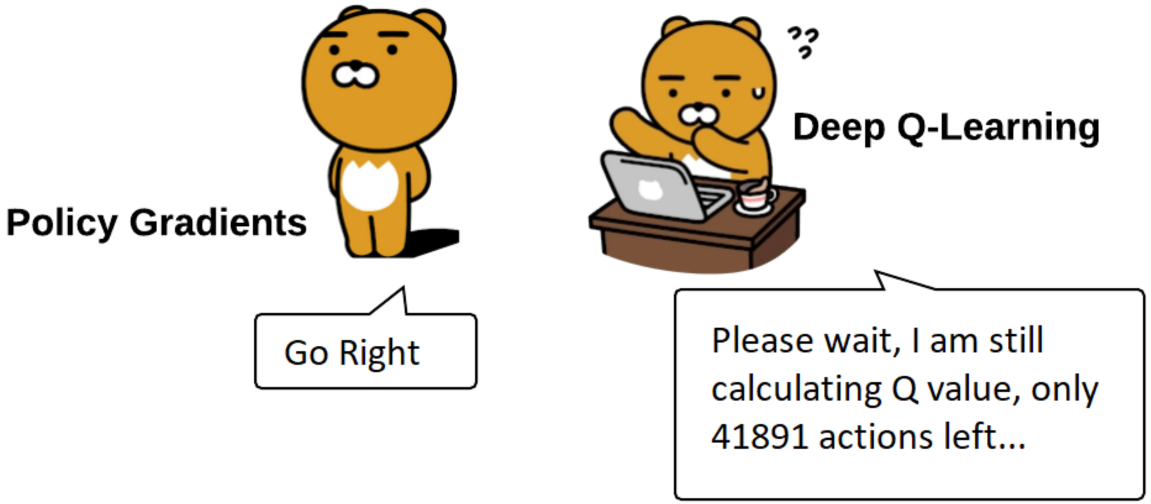
Maximum estimated future reward from the next state s’

Learning rate

Q-learning vs. Policy Gradient

Q-learning

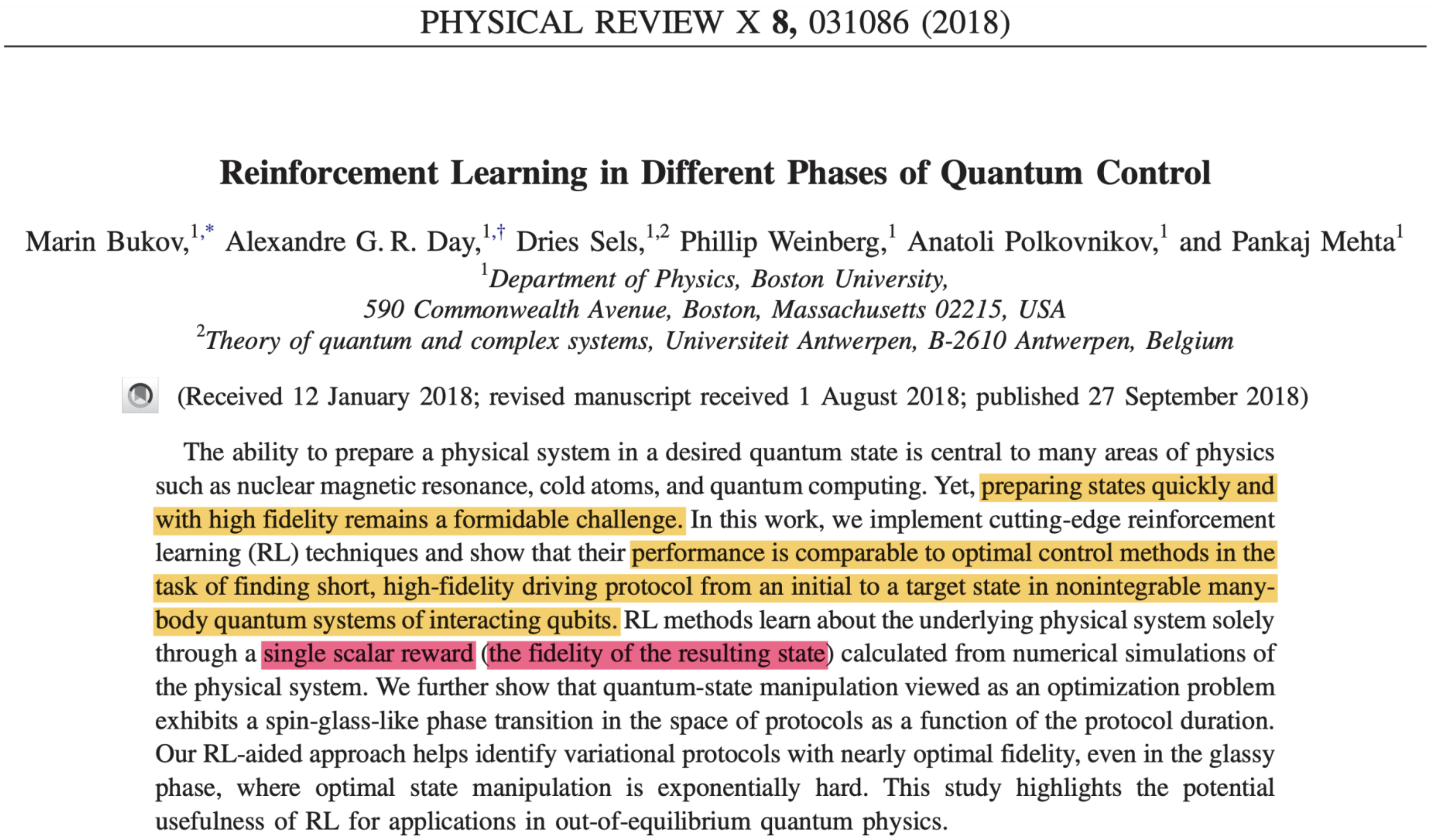
- Learns a value function
- Policy is implicit $\pi(s) = \arg max_a Q(s,a)$
- Good for discrete action spaces


Policy Gradients
Go Right
Deep Q-Learning
Please wait, I am still calculating Q value, only 41891 actions left...

Policy gradient

- Directly learn a parameterized policy
- Better for continuous or high-dimensional action spaces

Application of Q-learning for Quantum Control



PHYSICAL REVIEW X **8,** 031086 (2018)

Reinforcement Learning in Different Phases of Quantum Control

Marin Bukov,[1,*] Alexandre G. R. Day,[1,†] Dries Sels,[1,2] Phillip Weinberg,[1] Anatoli Polkovnikov,[1] and Pankaj Mehta[1]

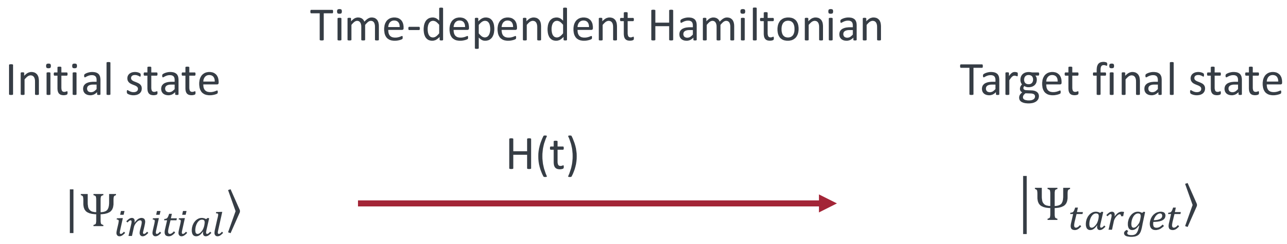
[1]*Department of Physics, Boston University,*
590 Commonwealth Avenue, Boston, Massachusetts 02215, USA
[2]*Theory of quantum and complex systems, Universiteit Antwerpen, B-2610 Antwerpen, Belgium*

(Received 12 January 2018; revised manuscript received 1 August 2018; published 27 September 2018)

The ability to prepare a physical system in a desired quantum state is central to many areas of physics such as nuclear magnetic resonance, cold atoms, and quantum computing. Yet, preparing states quickly and with high fidelity remains a formidable challenge. In this work, we implement cutting-edge reinforcement learning (RL) techniques and show that their performance is comparable to optimal control methods in the task of finding short, high-fidelity driving protocol from an initial to a target state in nonintegrable many-body quantum systems of interacting qubits. RL methods learn about the underlying physical system solely through a single scalar reward (the fidelity of the resulting state) calculated from numerical simulations of the physical system. We further show that quantum-state manipulation viewed as an optimization problem exhibits a spin-glass-like phase transition in the space of protocols as a function of the protocol duration. Our RL-aided approach helps identify variational protocols with nearly optimal fidelity, even in the glassy phase, where optimal state manipulation is exponentially hard. This study highlights the potential usefulness of RL for applications in out-of-equilibrium quantum physics.

Task

Teach a computer agent to find driving protocols to prepare a quantum state



Time-dependent Hamiltonian

Initial state

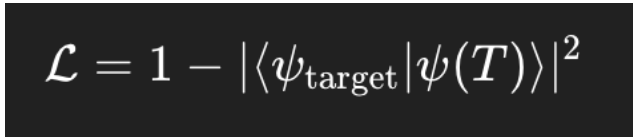
Target final state

$$|\Psi_{initial}\rangle \xrightarrow{H(t)} |\Psi_{target}\rangle$$

Gradient Descent

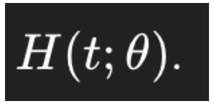
A natural method for quantum control is gradient descent

Cost function (or fidelity loss)



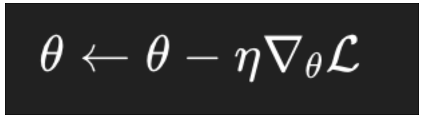
$$\mathcal{L} = 1 - |\langle \psi_{\text{target}} | \psi(T) \rangle|^2$$

The parameterized Hamiltonian is



$$H(t;\theta).$$

Gradient descent updates the control parameter to minimize the loss function



$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$$

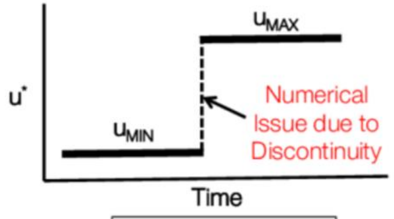
Why Gradient Descent can Fail?

- **Non-convex landscape**: the fidelity landscape is typically highly non-convex, with many local minima, saddle points,, and flat regions;

- **Sparse reward**: When the initial guess is poor, the fidelity may be very low and gradients nearly vanish – leading to stagnation (updates to the control parameter have negligible effect)

- **Bang-bang optimality**: in many quantum control tasks, optimal controls are not smooth (e.g., abrupt on/off pulses), which are hard to capture using smooth gradient updates.

- **Hard constraints**: control fields may have discrete values, etc., that may make gradient-based updates ill-suited.



Bang-Bang Control

Example: single-qubit quantum control

A two-level system

$$H[h_x(t)] = -S^z - h_x(t)S^x,$$

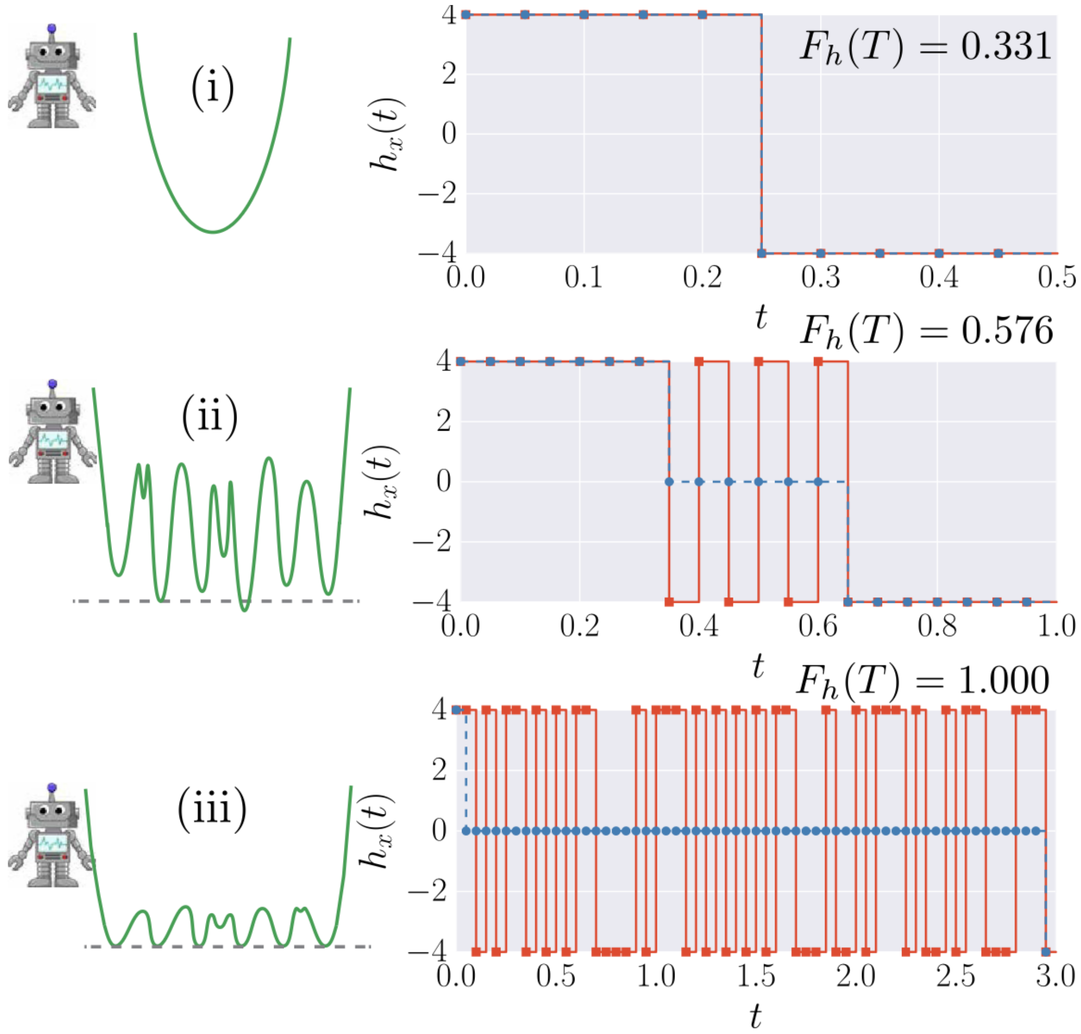
- Integrable, non-interacting system
- **Initial state**: ground state of H with $h_x = -2$
- **Target state**: ground state of H with $h_x = +2$
- There exists an analytical solution to solve for the optimal protocol.
- For many nonintegrable many-body systems, there are no analytical solution!

Illustration of three different driving protocols

The agents searches for a piecewise-constant protocols of duration T, by choosing a drive strength $h_x(t)$ at each time $t = j \ * \ \delta t$, with $j = \{0, 1, \dots, \frac{T}{\delta t}\}$.

We restrict to field $h_x(t) \in [-4, +4]$, as we do not have access to infinite control fields;

Also, restrict the RL algorithm to the bang-bang protocols.



Left: Schematic illustration of the fidelity loss landscape

Right: The optimal bang-bang protocol found by the RL agent

Reinforcement learning

State: all tuples of $[t, h_x(t)]$ of time t and the corresponding magnetic field $h_x(t)$

Action: all jumps δh_x in the protocol $h_x(t)$

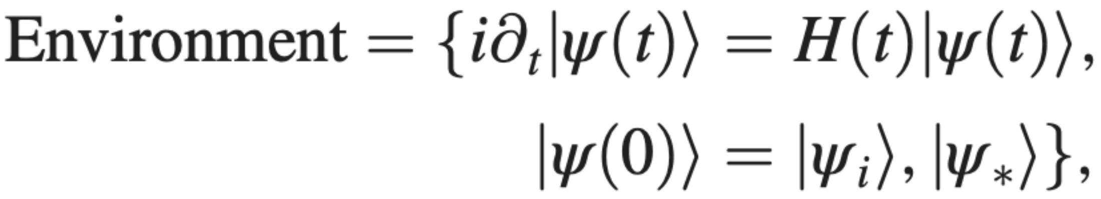
$$\mathcal{S} = \{s = [t, h_x(t)]\}, \quad \mathcal{A} = \{a = \delta h_x\}, \quad \mathcal{R} = \{r \in [0,1]\}.$$

Reward: real numbers in the interval [0, 1]

$$r(t) = \begin{cases} 0, & \text{if } t < T, \\ F_h(T) = |\langle \psi_* | \psi(T) \rangle|^2, & \text{if } t = T. \end{cases}$$

Reinforcement Learning

Environment: The Schrodinger initial value problem together with the target state



$$\text{Environment} = \{i\partial_t|\psi(t)\rangle = H(t)|\psi(t)\rangle,\\ |\psi(0)\rangle = |\psi_i\rangle, |\psi_*\rangle\},$$

RL Algorithm

Start with the initial RL state

$$s_0 = (t = 0, h_x = -4)$$

Take the action

$$\delta h_x = 8$$

Go to the next RL state

$$s_1 = (\delta t, +4).$$

Due to interaction with the environment, the initial state is evolved according to

$$|\psi(\delta t)\rangle = e^{-iH[h_x=4]\delta t}|\psi_i\rangle$$

After each step, compute the reward

$$r(t) = \begin{cases} 0, & \text{if } t < T, \\ F_h(T) = |\langle\psi_*|\psi(T)\rangle|^2, & \text{if } t = T. \end{cases}$$

Markov decision process

State-action-reward chain

$$s_0 \rightarrow a_0 \rightarrow r_0 \rightarrow s_1 \rightarrow a_1 \rightarrow r_1 \rightarrow s_2 \rightarrow, \ldots, \rightarrow s_{N_T}.$$

Policy $\pi(a|s)$: the probability of taking action a from the state s

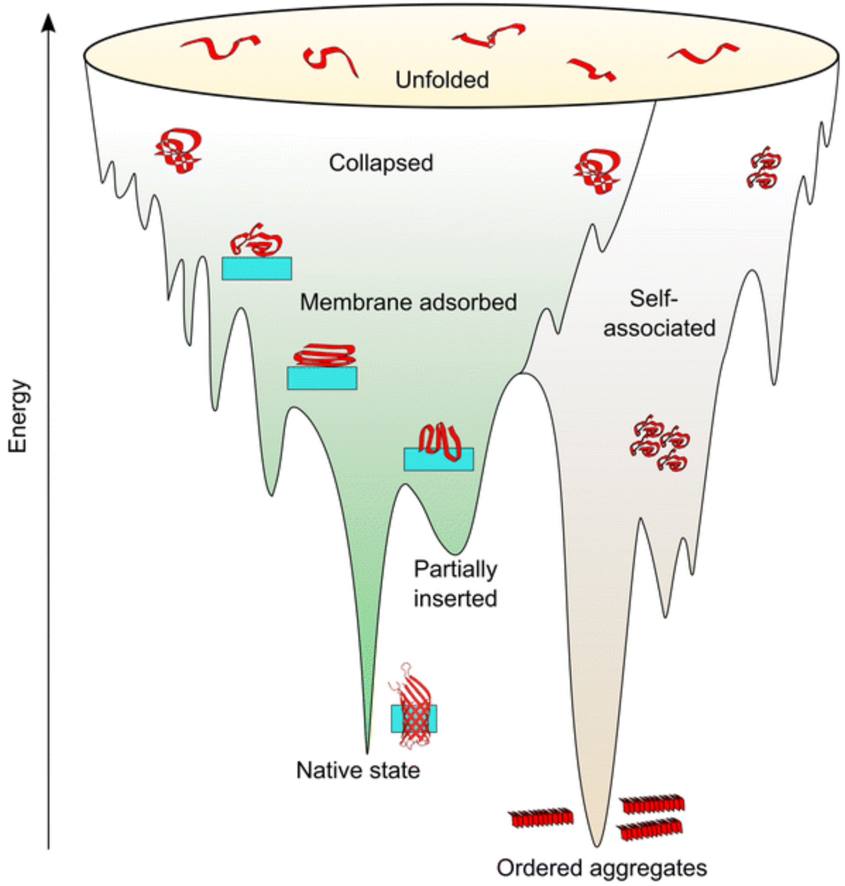
- ➢ Policy gradient: to optimize the policy;
- ➢ Q-learning: optimize the Q(s, a) function – the expected total return $R = \sum_{i=0}^{N_T} r_i$

Quantum optimization

- One of the most promising applications of quantum computing

- Applications range from logistics, machine learning, finance, biology, to materials science.

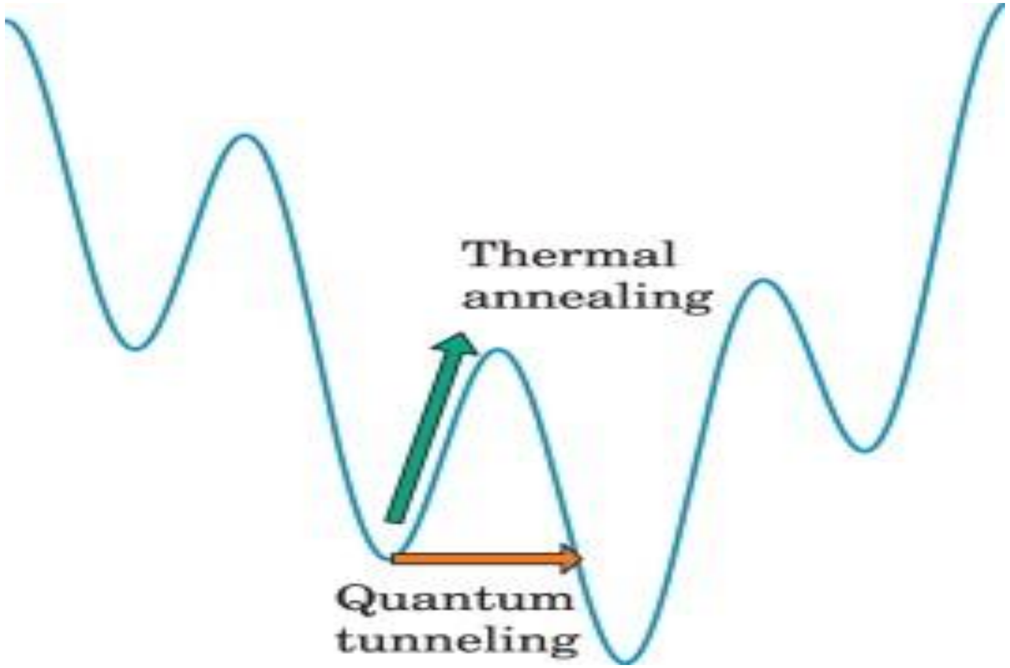
- Search solution space more efficiently, by using phenomena like superposition, tunneling, entanglement

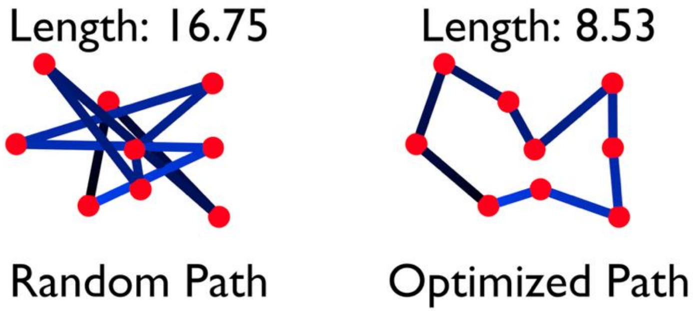


Rugged Energy Landscape

Quantum annealing

- Traveling salesman problem
- D-wave quantum computer
- Quantum annealing vs. Thermal annealing



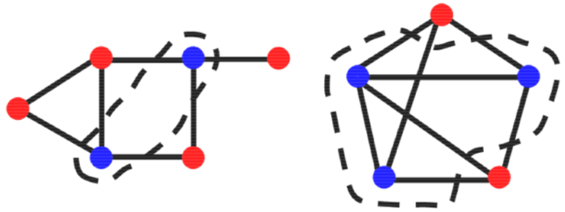


Ising formulation of the combinatorial optimization problems

- Many optimization problems can be recast into a famous model in statistical mechanics: Ising model
- MaxCut problem: given a graph G(V, E), partition vertices into two sets to maximize the number of edges between sets
- Binary variables: Assign each vertex i a spin $s_i \in \{-1, +1\}$ (up or down)
- An edge (i, j) contributes to the cut if $s_i \neq s_j$
- Maximizing the number of edges cut is equivalent to minimizing the energy of the Ising model: $H = -\sum_{(i,j)\in E} \frac{1-s_i s_j}{2}$

➢ When $s_i \neq s_j$, the energy contribution is -1;

➢ When $s_i = s_j$, the energy contribution is 0;



We want to have many edges in the cut, thus, the energy shall be minimized