Optimization problems on quantum computers

Lecture 1

Quantum optimization algorithms inspired by physics

Yassine Hamoudi

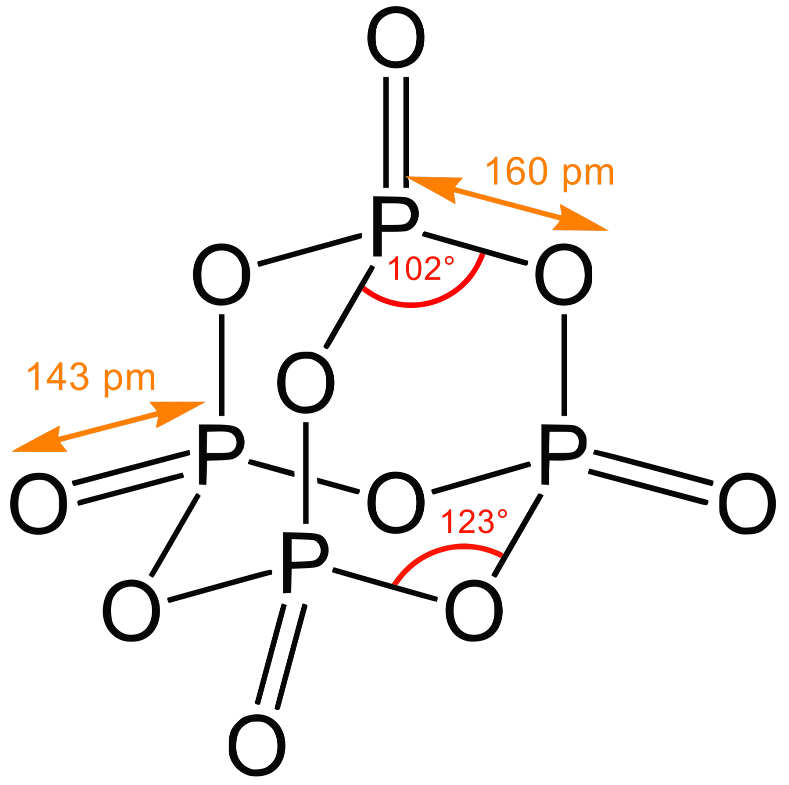
Materials: https://yassine-hamoudi.github.io/cemracs2025/

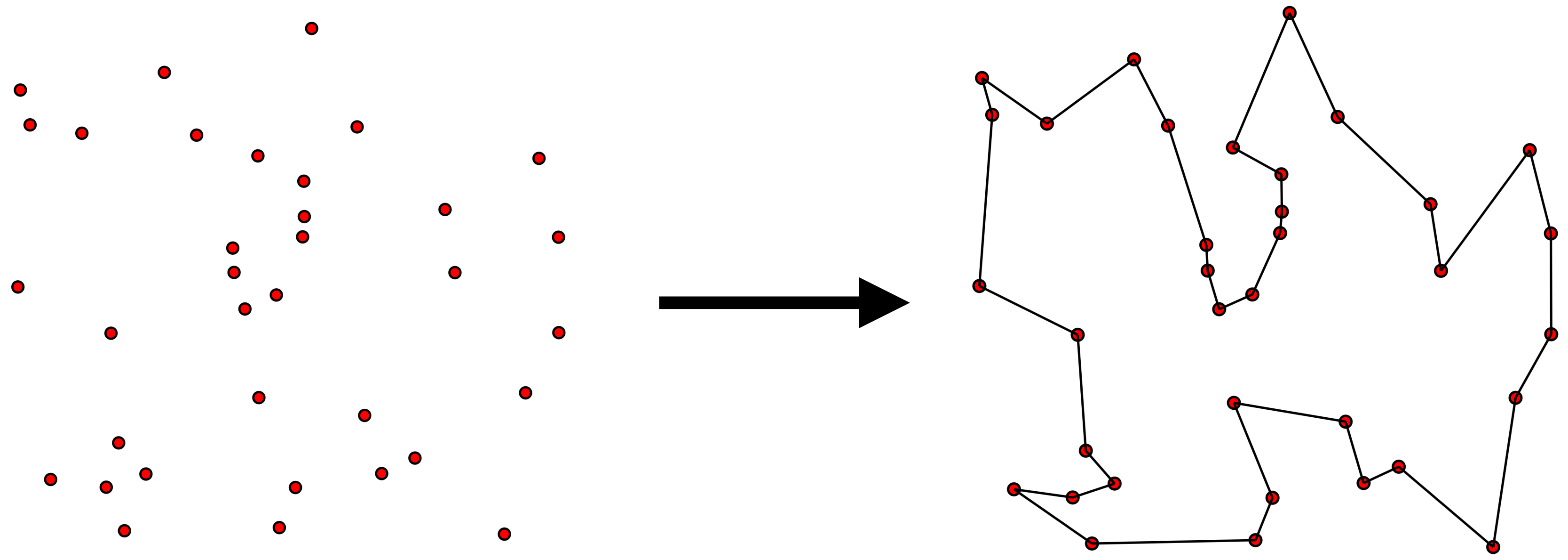
Optimization is about finding elements that minimize a given objective

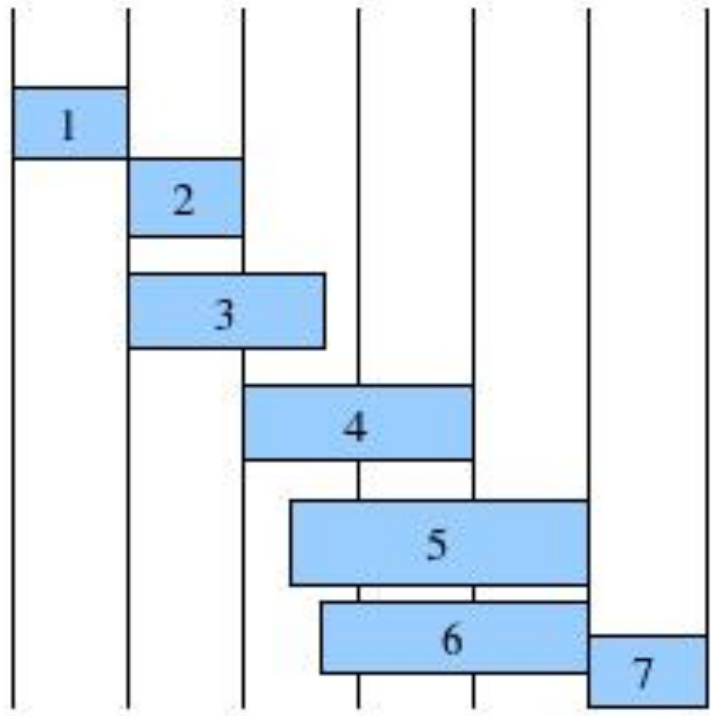
Stable configuration of a molecule

Exploration of a graph

Scheduling of concurrent tasks







Hackathon

Project: ERC EMC2
Quantum algorithms for ground state computation

Project: L'Oréal
Genome assembly from DNA fragments

Project: IFPEN
Route planning problems

Project: RTE
Maintenance of electrical installations

Project: La Poste
Supply of empty containers

Optimization is about finding elements that minimize a given objective

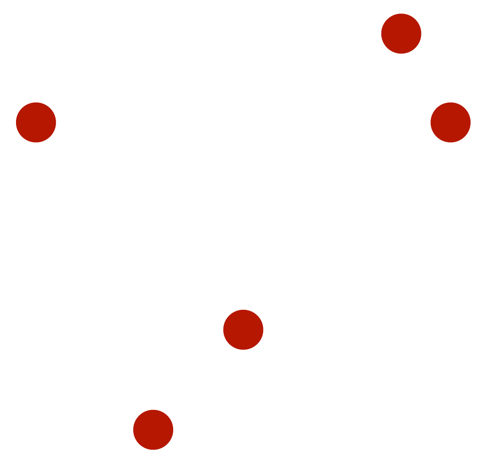
$$\min_{x\in\Omega} f(x)$$

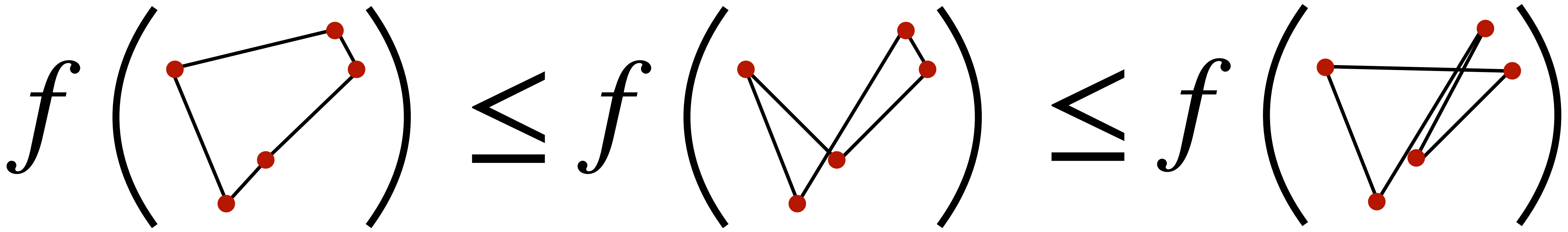
Value of element x

Objective function

Search space of the problem

Example: shortest path that connects all points





$f =$ length of a given path $\quad\Omega =$ set of all valid paths

Focus of this course

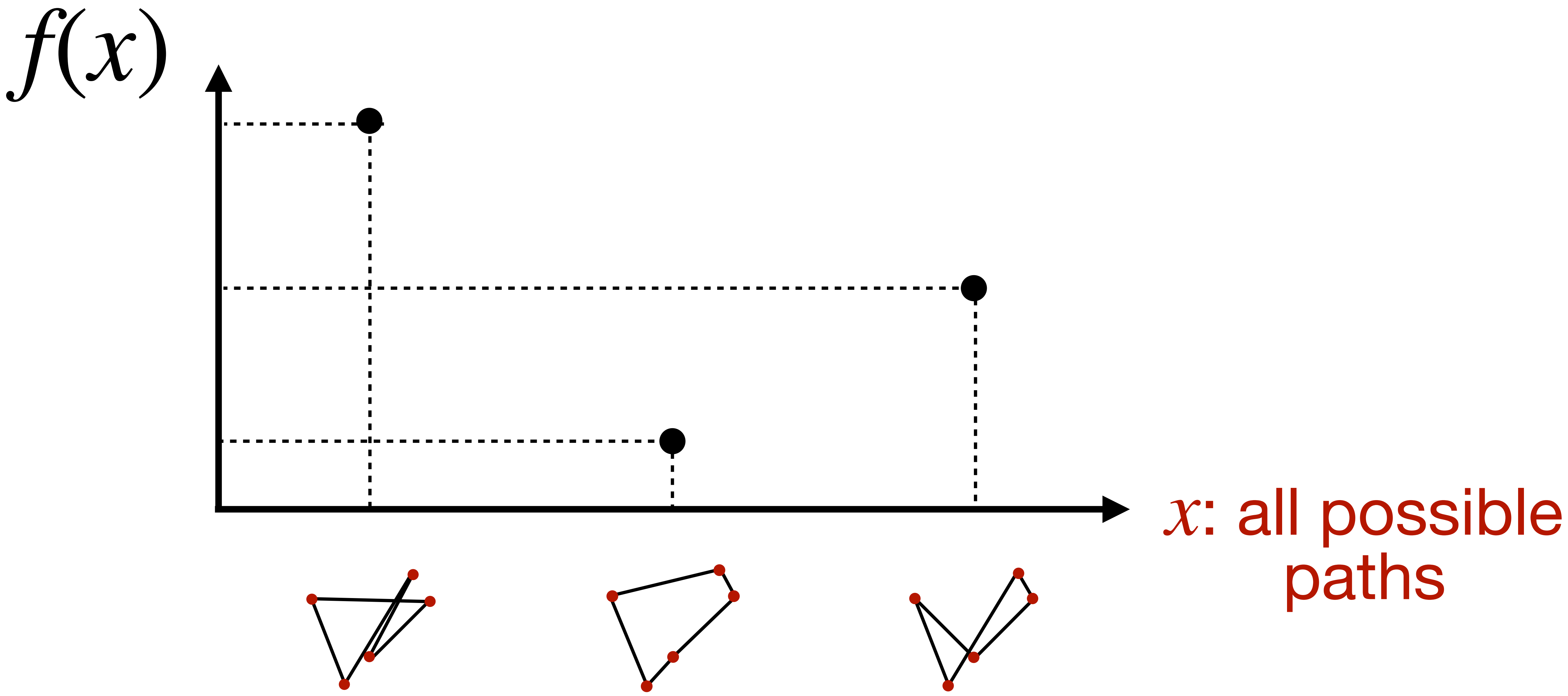
How quantum computers may help in solving optimization problems?

- New types of algorithms based on the capabilities of quantum computers
- Example of optimization problems solved by such quantum algorithms
- Benefits and limitations compared to other optimization methods

Terminology

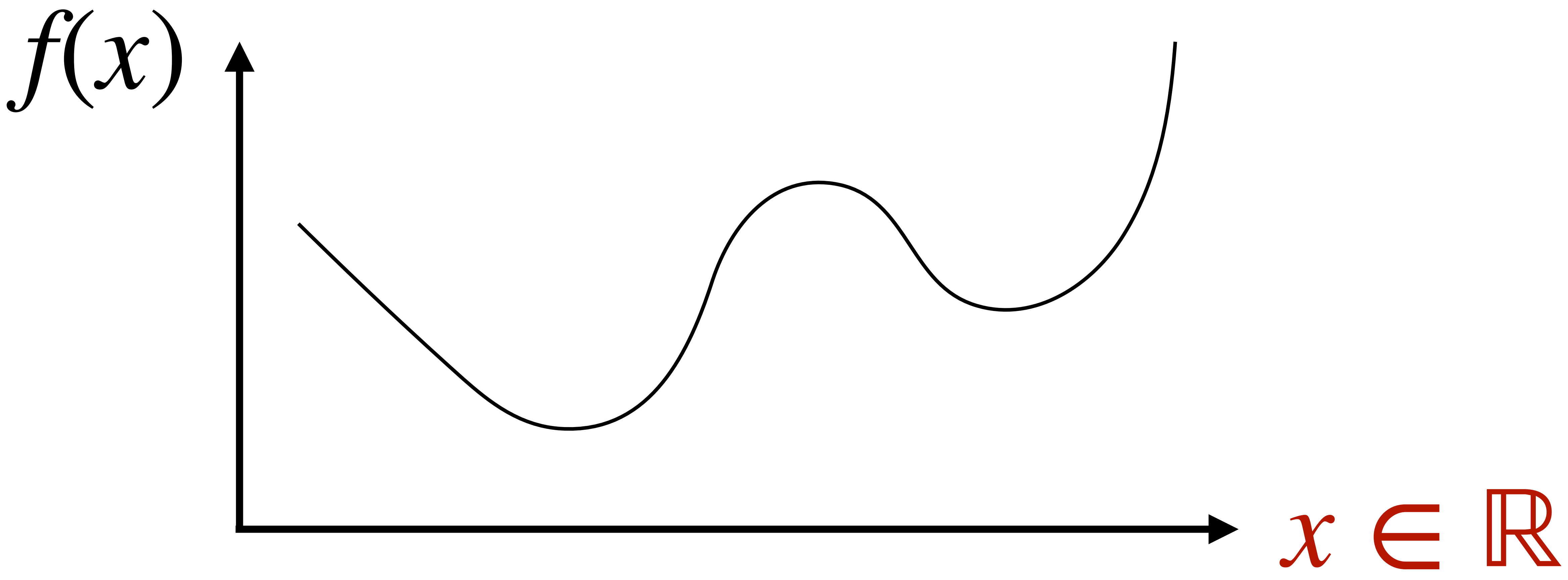
Discrete optimization

Objective function supported over a discrete set of values



Continuous optimization

Objective function supported over a continuous set of values



Terminology

Exact solution

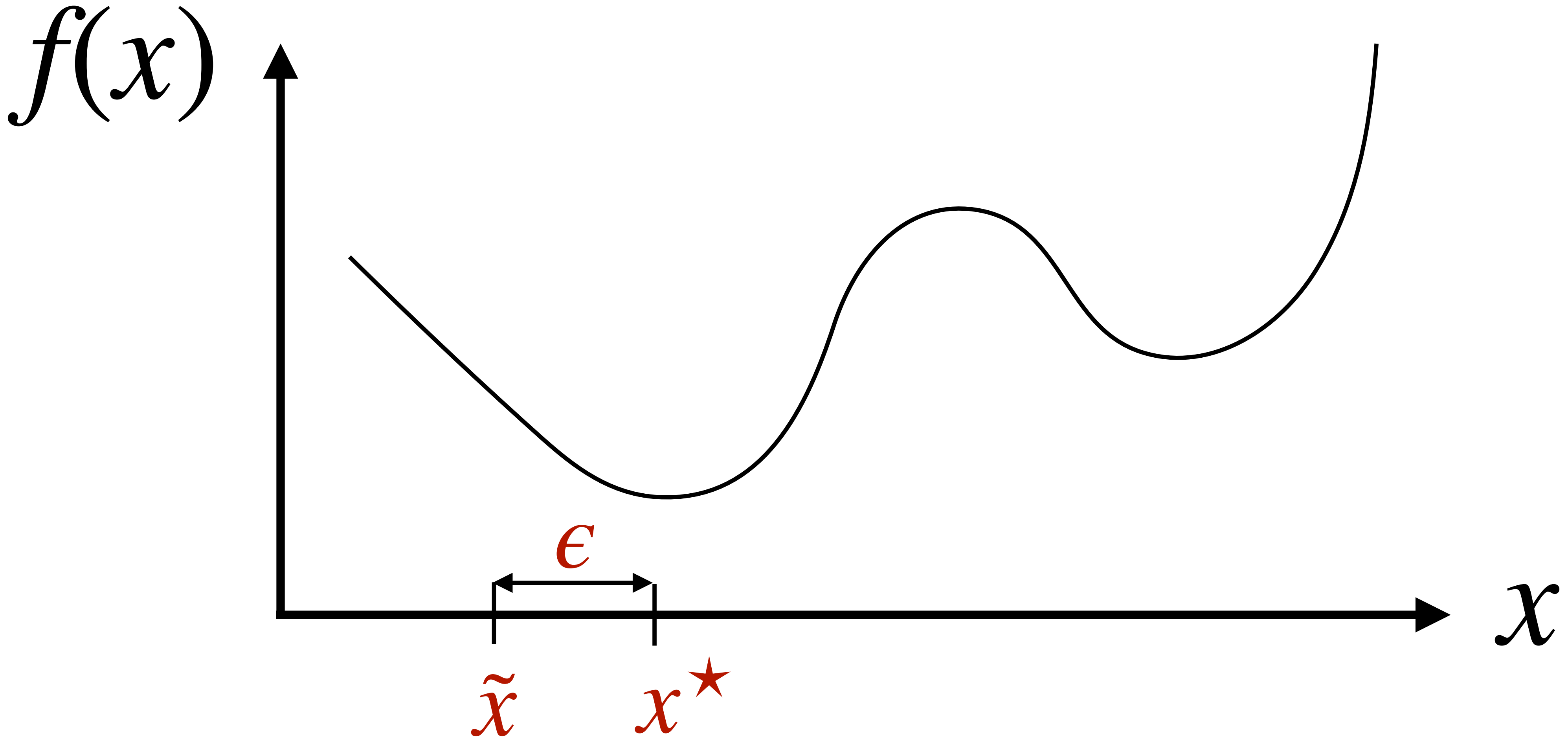
Find the minimum-value solution
(or one of them if there are many)

$$\forall x, f(x^\star) \leq f(x)$$

Approximate solution

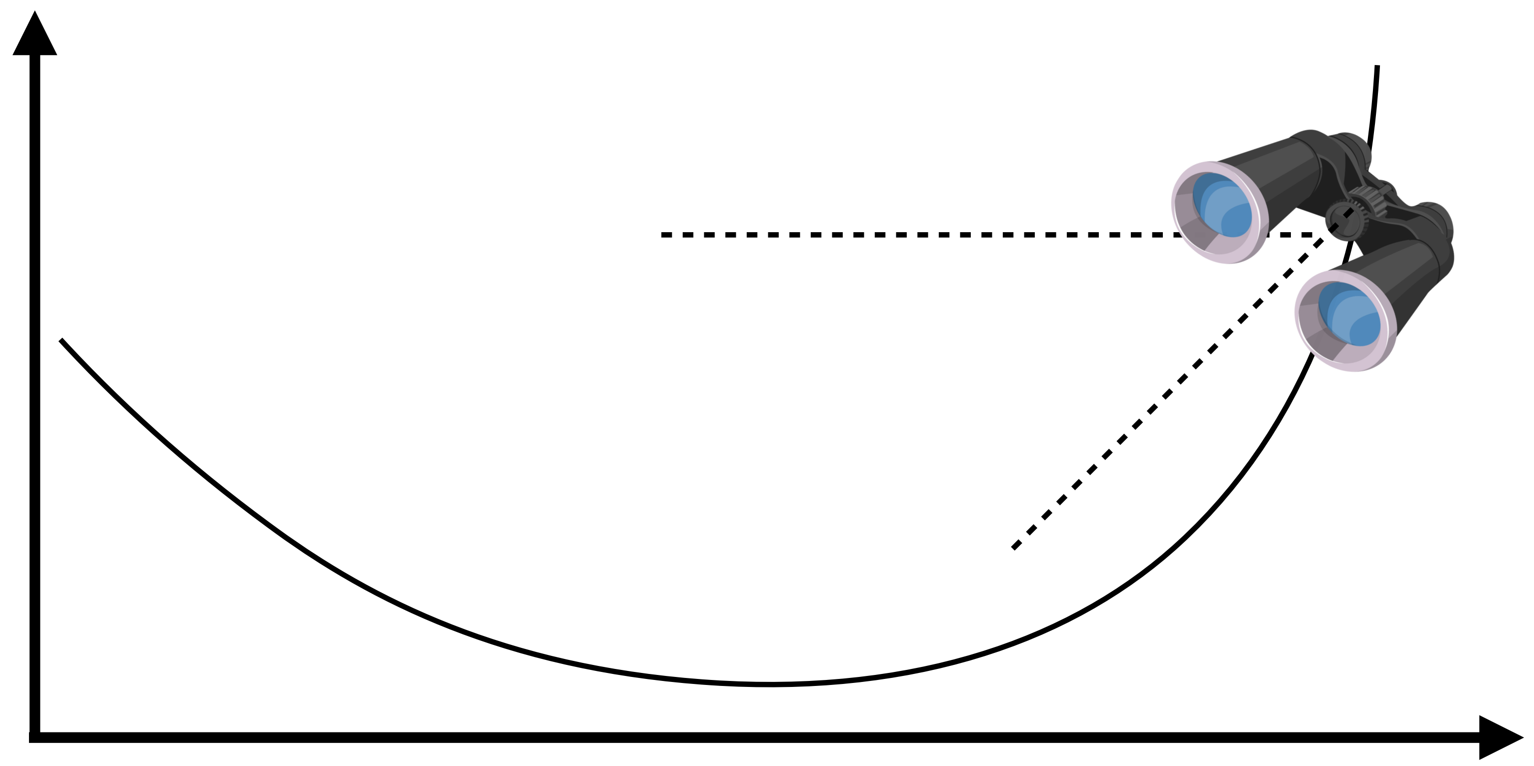
Find a solution whose value is
not too far from the minimum

$$\forall x, f(\tilde{x}) \leq f(x^\star) + \epsilon$$



Terminology

Convex function

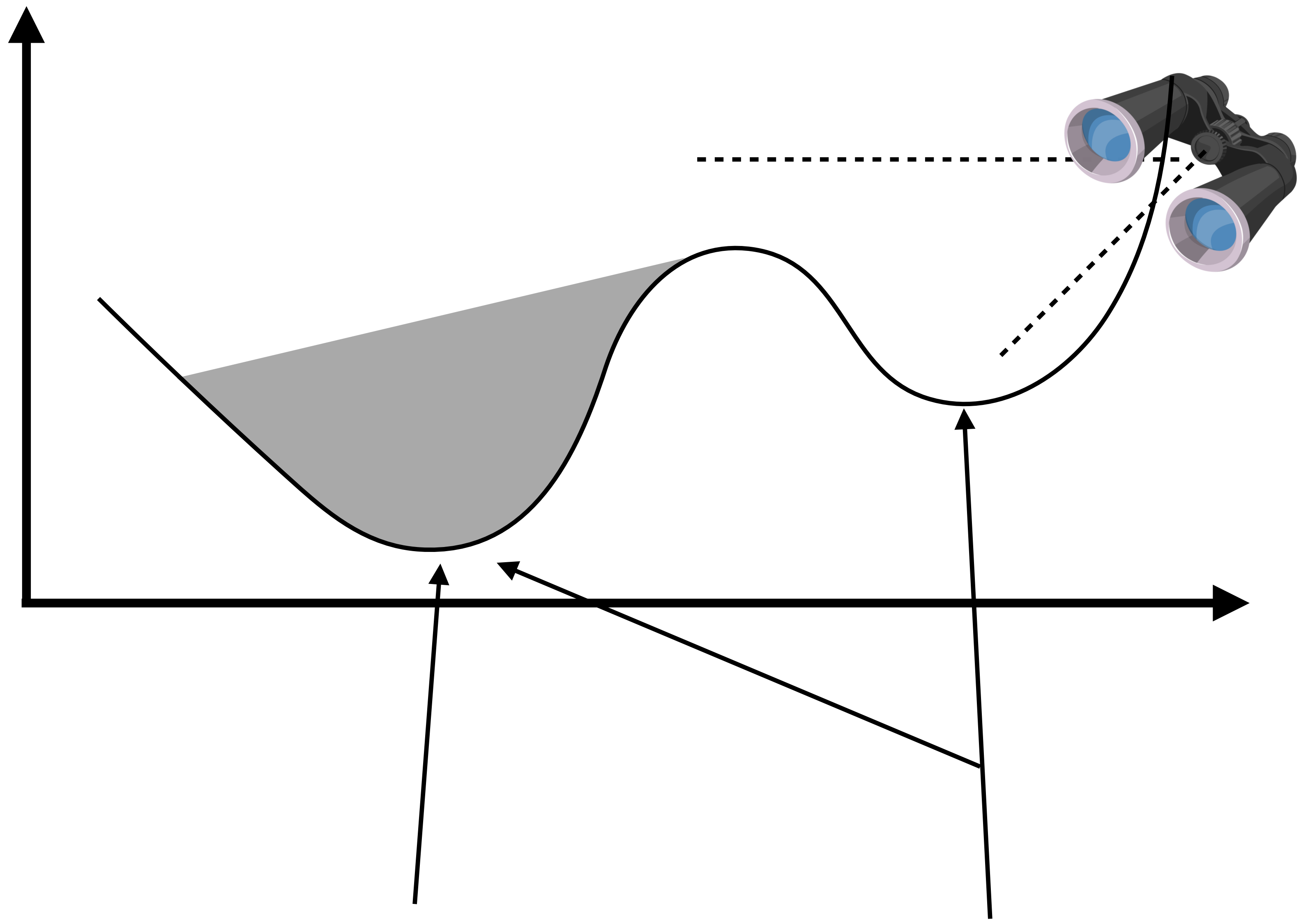


The entire graph is “visible” from any point

$$\forall t \in (0,1),\ f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

Global minima = Local minima

Non-convex function



Global minima ≠ Local minima

(in the context of continuous optimization) optimistic

Great goal of quantum optimizers:

Find relevant optimization problems that can be solved much faster than with any classical algorithm

We are not there yet:

- Quantum optimization often lacks theoretical guarantees, or makes contrived assumptions
- Quantum advantages vanish under implementation overhead (ex: quantum error correction)
- New classical optimization methods fight back (ex: deep learning)

… but the field is emerging and first-gen quantum computers accelerate its development

Quantum optimization algorithms

Lecture 1 (Physics-inspired)

Exact algorithms

- Quantum Phase Estimation (QPE)
- Quantum Adiabatic Algorithm (QAA)

Variational quantum algorithms

- Variational Quantum Eigensolver (VQE)
- Quantum Approximate Optimization Algorithm (QAOA)

Lecture 2 (Oracle-based)

Grover-type algorithms

- Quantum Minimum Finding
- Minimum Spanning Tree

Gradient computation

Monte-Carlo algorithms

- Linear programming
- Escaping Saddle Points

Optimization as a physics problem

The Hamiltonian operator

Linear operator encoding the possible energy levels of a system

$$H \in \mathbb{C}^{2^n \times 2^n}$$

Hermitian $H^\dagger = H$

Eigendecomposition: $H = \sum_{i=1}^{2^n} \lambda_i | v_i \rangle$

Energy levels: real eigenval. $\lambda_1 \leq \lambda_2 \leq \ldots$

Stationary states: eigenvec. $H | v_i \rangle = \lambda_i | v_i \rangle$

Smallest eigenvalue λ_1 (lowest energy) and corresponding eigenvector $| v_1 \rangle$ (ground state) characterize the most stable configuration of the system

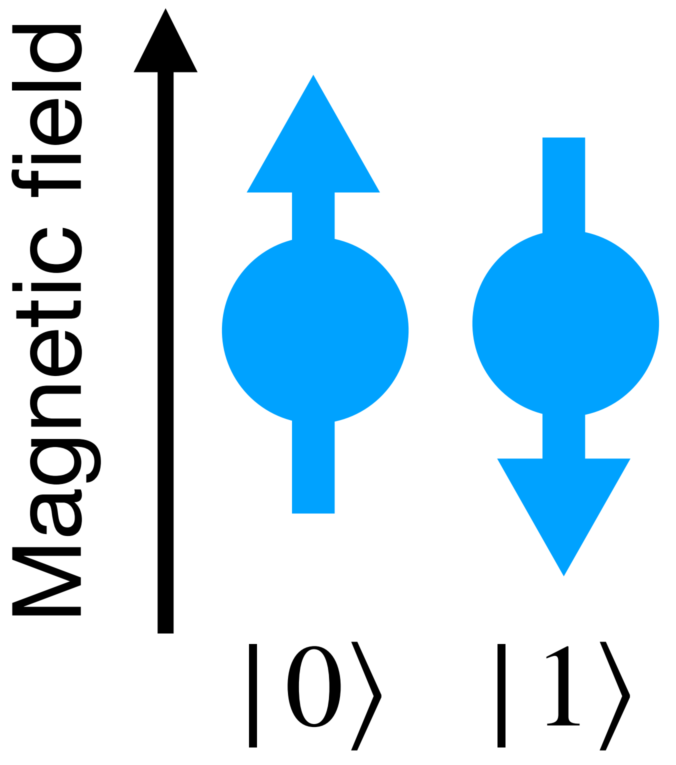
The Hamiltonian operator

Linear operator encoding the possible energy levels of a system

$$H \in \mathbb{C}^{2^n \times 2^n}$$

Hermitian $H^\dagger = H$

Example: A qubit in a magnetic field of angular frequency ω



$$H = \begin{pmatrix} \frac{\hbar\omega}{2} & 0 \\ 0 & -\frac{\hbar\omega}{2} \end{pmatrix}$$

Eigenvectors: $|0\rangle$ and $|1\rangle$ (ground state)

Eigenvalues: $\frac{\hbar\omega}{2}$ $-\frac{\hbar\omega}{2}$
(energies)

$\hbar$: Dirac constant

The Hamiltonian operator

Ground state computation is an optimization problem:

$$\lambda_1 = \| H \,|\, v_1 \rangle \| = \min_{|v\rangle \in \mathbb{C}^{2^n} : \|v\| = 1} \underbrace{\| H \,|\, v \rangle \|}_{\text{objective function} \sim f(x)}$$

optimized element ~ x

How hard is it to optimize?

- Diagonalization is infeasible in general (matrix of exponential size)
- Optimization landscape is highly complicated (non-convex)
- … requires making additional assumptions on H (examples: next slide)

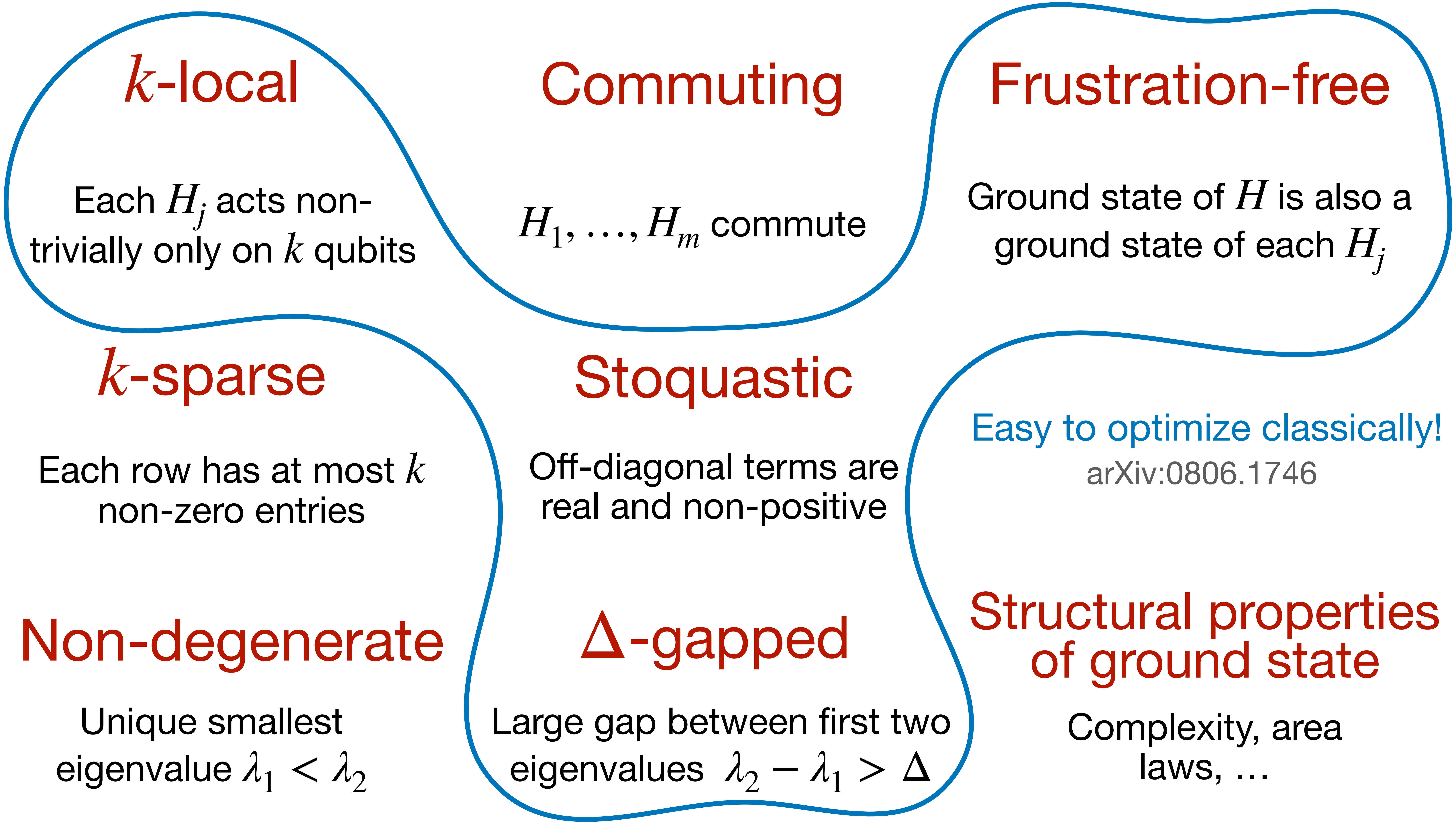
Examples of more friendly Hamiltonians

Representation

$$H = \sum_{j=1}^{m} H_j$$

$$H = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{12^n} \\ H_{21} & & & \vdots \\ \vdots & & & \\ H_{2^n1} & & \cdots & H_{2^n2^n} \end{pmatrix}$$

$$H = \sum_{i=1}^{2^n} \lambda_i |v_i\rangle$$



k-local

Each H_j acts non-trivially only on k qubits

Commuting

$H_1, \ldots, H_m$ commute

Frustration-free

Ground state of H is also a ground state of each H_j

k-sparse

Each row has at most k non-zero entries

Stoquastic

Off-diagonal terms are real and non-positive

Easy to optimize classically!

arXiv:0806.1746

Non-degenerate

Unique smallest eigenvalue $\lambda_1 < \lambda_2$

Δ-gapped

Large gap between first two eigenvalues $\lambda_2 - \lambda_1 > \Delta$

Structural properties of ground state

Complexity, area laws, …

The Hamiltonian operator

Ground state computation is an optimization problem:

$$\lambda_1 = \| H | v_1 \rangle \| = \min_{|v\rangle \in \mathbb{C}^{2^n} : \|v\| = 1} \| H | v \rangle \|$$

How useful/general is it?

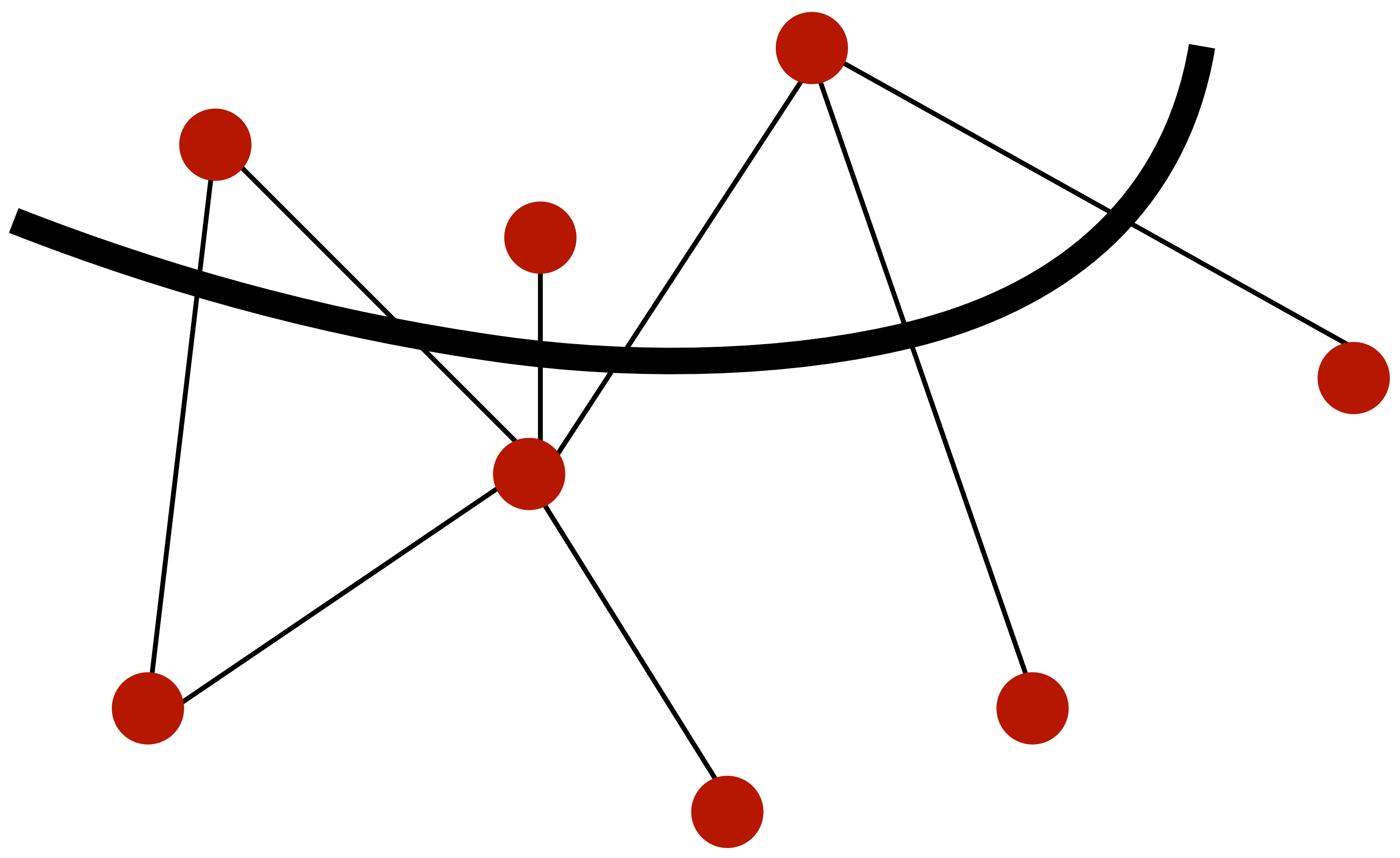
- Ground state reveals physical properties (electronic configurations, phases of matter…) exploited in quantum chemistry, condensed matter physics, etc.
- Lots of other optimization problems can be reduced to it

  Example (next slide): combinatorial optimization via QUBO Hamiltonian

Example: The “QUBO” Hamiltonian

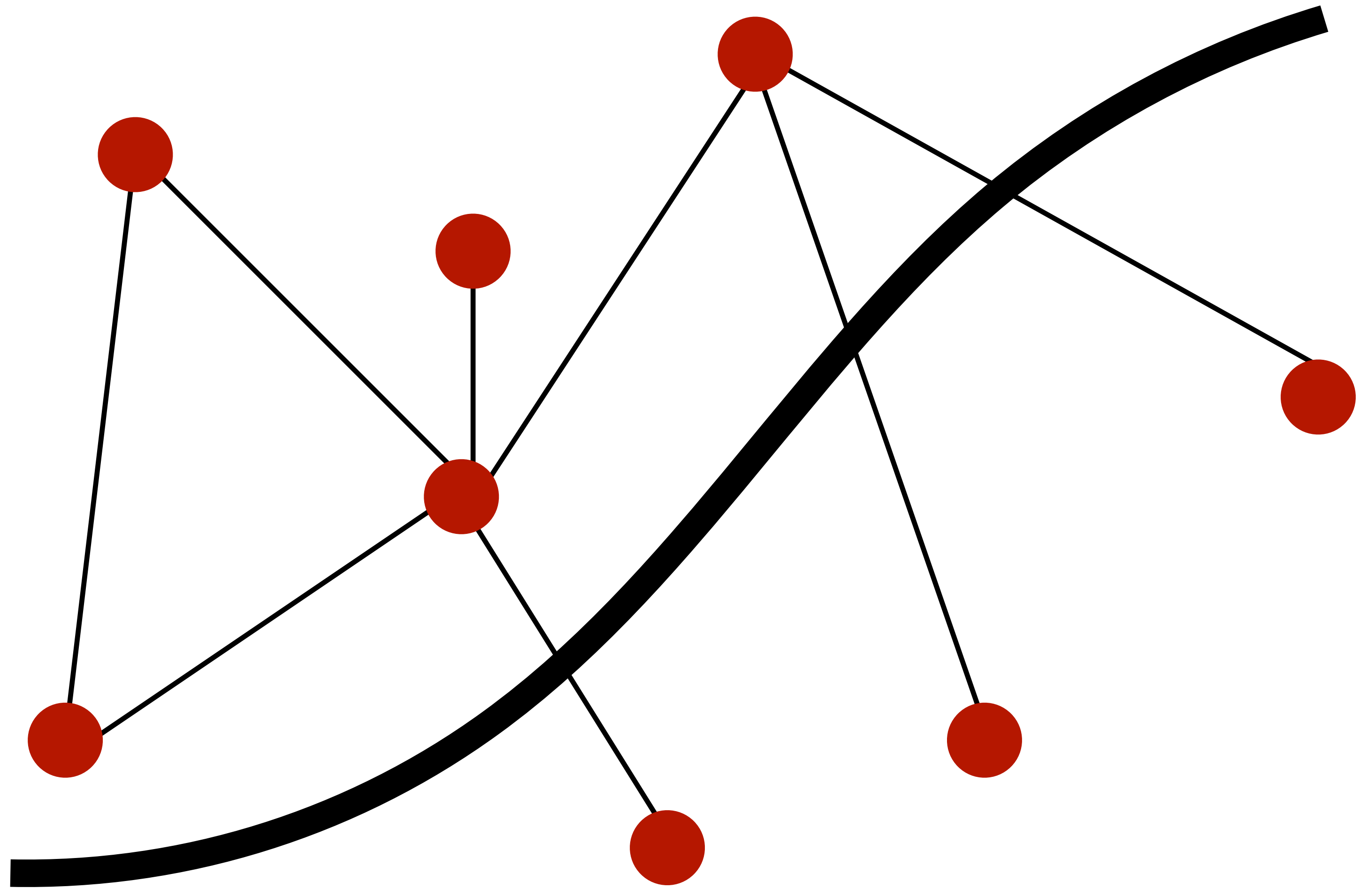
Max-Cut problem

Partition vertices of a graph into two parts that maximize number of edges between the two



Cut of value 6

$\geq$

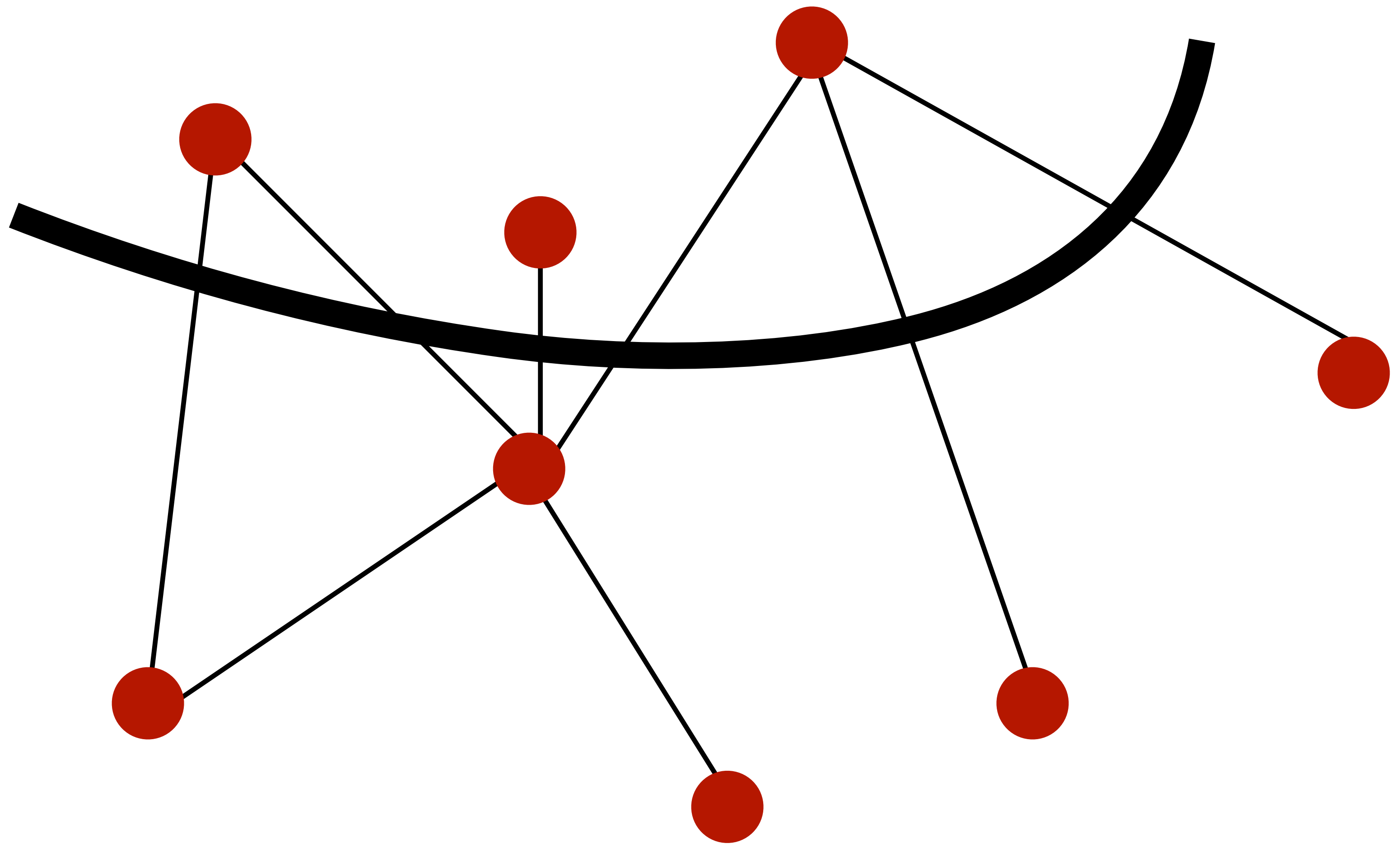


Cut of value 3

Example: The “QUBO” Hamiltonian

Max-Cut problem

Partition vertices of a graph into two parts that maximize number of edges between the two



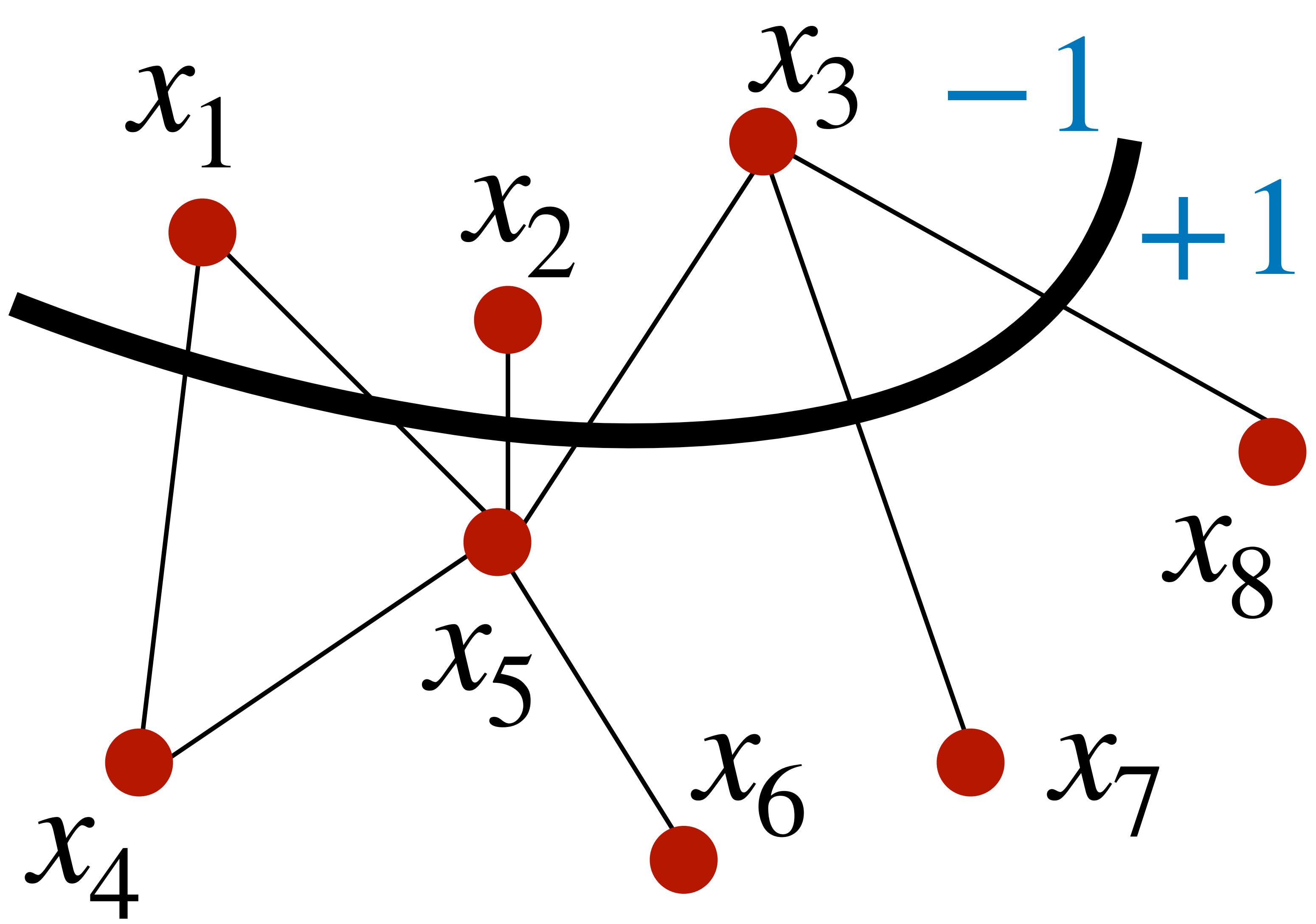
Cut of value 6

- Fundamental problem in discrete optimization
- NP-Hard
- Famous classical approximation algo. (Goemans-Williamson)

Example: The “QUBO” Hamiltonian

Max-Cut problem

Partition vertices of a graph into two parts that maximize number of edges between the two



Cut of value 6

How to encode it into a Hamiltonian?

$$\text{Max-Cut} = \min_{x\in\{-1,1\}^n} \underbrace{\sum_{\{i,j\}\in\text{Edge}} -\frac{1}{4}(x_i - x_j)^2}_{f(x)}$$

= ground state of

$$H = \begin{array}{cc} & x \\ x & \begin{pmatrix} \ddots & & 0 \\ & f(x) & \\ 0 & & \ddots \end{pmatrix} \end{array} = \underbrace{\sum_{\{i,j\}\in\text{Edge}} -\frac{1}{4}(Z_i - Z_j)^2}_{\text{Ising Hamiltonian}}$$

Example: The "QUBO" Hamiltonian

What other polynomials can be converted into Ising Hamiltonians?

QUBO (Quadratic unconstrained binary optimization)

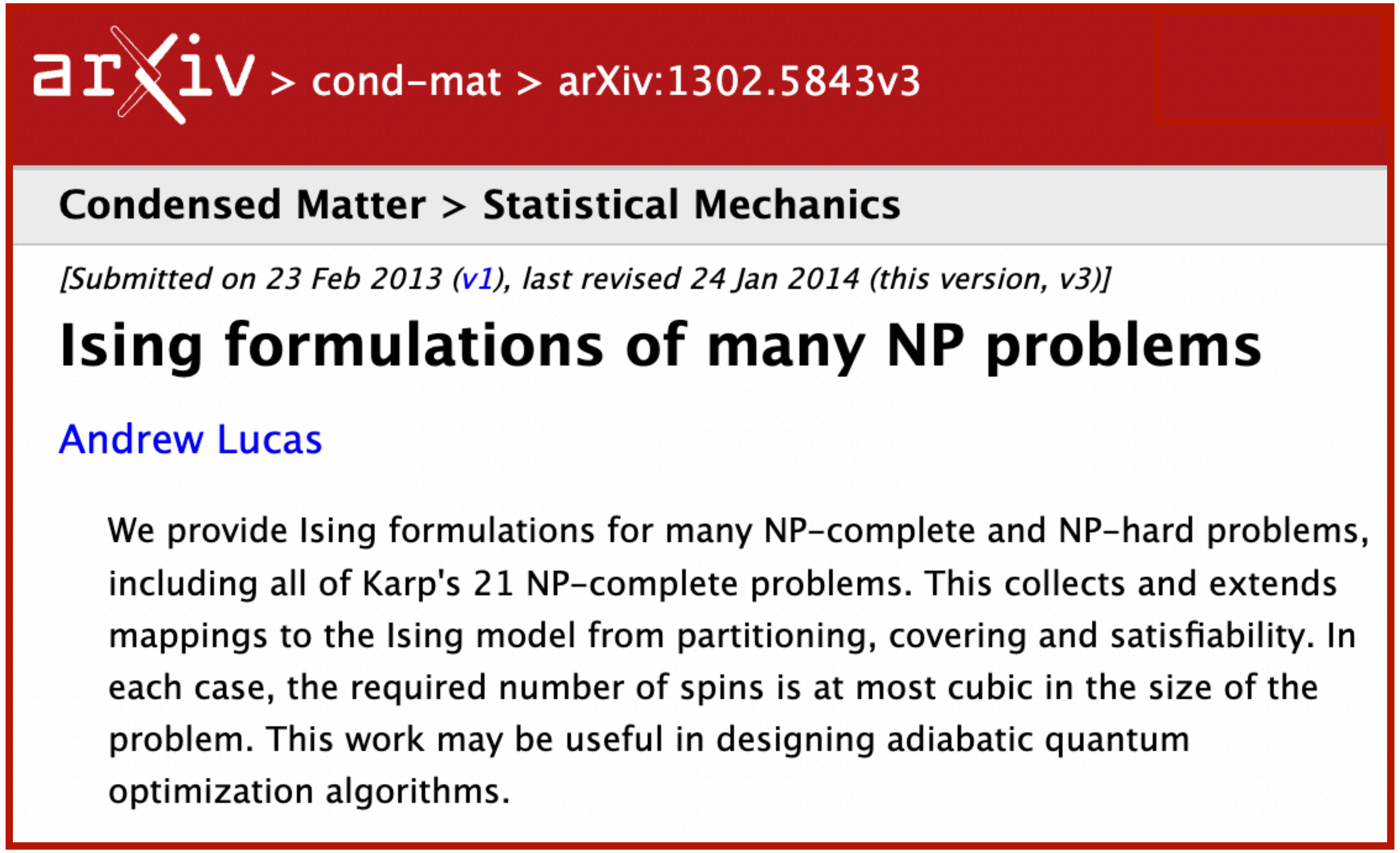
$$\min_{x \in \{0,1\}^n} x^\top Q x = \min_{x \in \{0,1\}^n} \sum_{i,j} Q_{i,j} x_i x_j \qquad Q \in \mathbb{R}^{n \times n} \text{ symmetric}$$

Exercise: show that substituting $x_i \mapsto (\mathrm{Id} - Z_i)/2$ yields a Hermitian matrix

Example: The "QUBO" Hamiltonian

QUBO encompasses a lot of optimization problems

… but no guarantee in general that "QUBO" Hamiltonian is nice:



arXiv > cond-mat > arXiv:1302.5843v3

Condensed Matter > Statistical Mechanics

[Submitted on 23 Feb 2013 (v1), last revised 24 Jan 2014 (this version, v3)]

Ising formulations of many NP problems

Andrew Lucas

We provide Ising formulations for many NP-complete and NP-hard problems, including all of Karp's 21 NP-complete problems. This collects and extends mappings to the Ising model from partitioning, covering and satisfiability. In each case, the required number of spins is at most cubic in the size of the problem. This work may be useful in designing adiabatic quantum optimization algorithms.

- small spectral gap
- long-range interactions
- lots an ancillae (higher dim.)
- …

The Hamiltonian operator

Ground state computation is an optimization problem:

$$\lambda_1 = \| H | v_1 \rangle \| = \min_{|v\rangle \in \mathbb{C}^{2^n} : \|v\| = 1} \| H | v \rangle \|$$

Why quantum computers may be useful here?

- They can solve the Schrödinger equation at a much lower cost Sim_H than classical algorithms (for “friendly” H)

$$i \frac{d | \psi(t) \rangle}{dt} = H | \psi(t) \rangle \xrightarrow{\text{Ham. simulation}} | \psi(1) \rangle = e^{-iH} \, | \psi(0) \rangle$$

- … not clear yet how simulation relates to optimization

  Example (next parts): Quantum Phase Estimation, Adiabatic algorithm

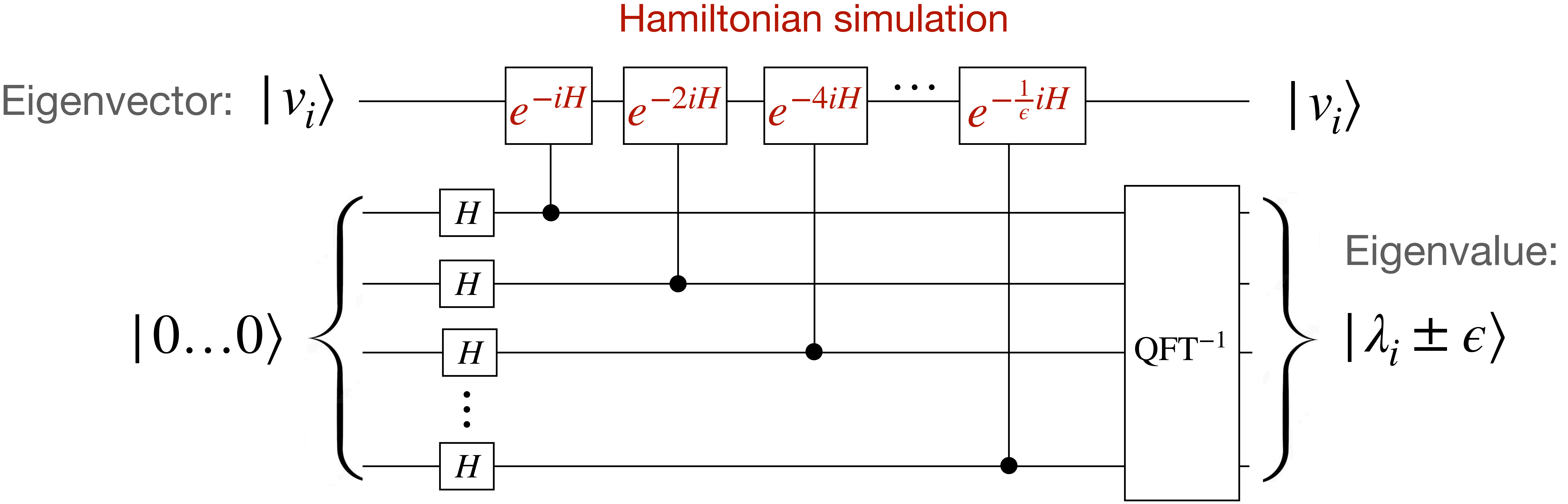
Exact Algorithms

Quantum optimization algorithms that provably return the exact ground state

Quantum Phase Estimation

Phase estimation circuit

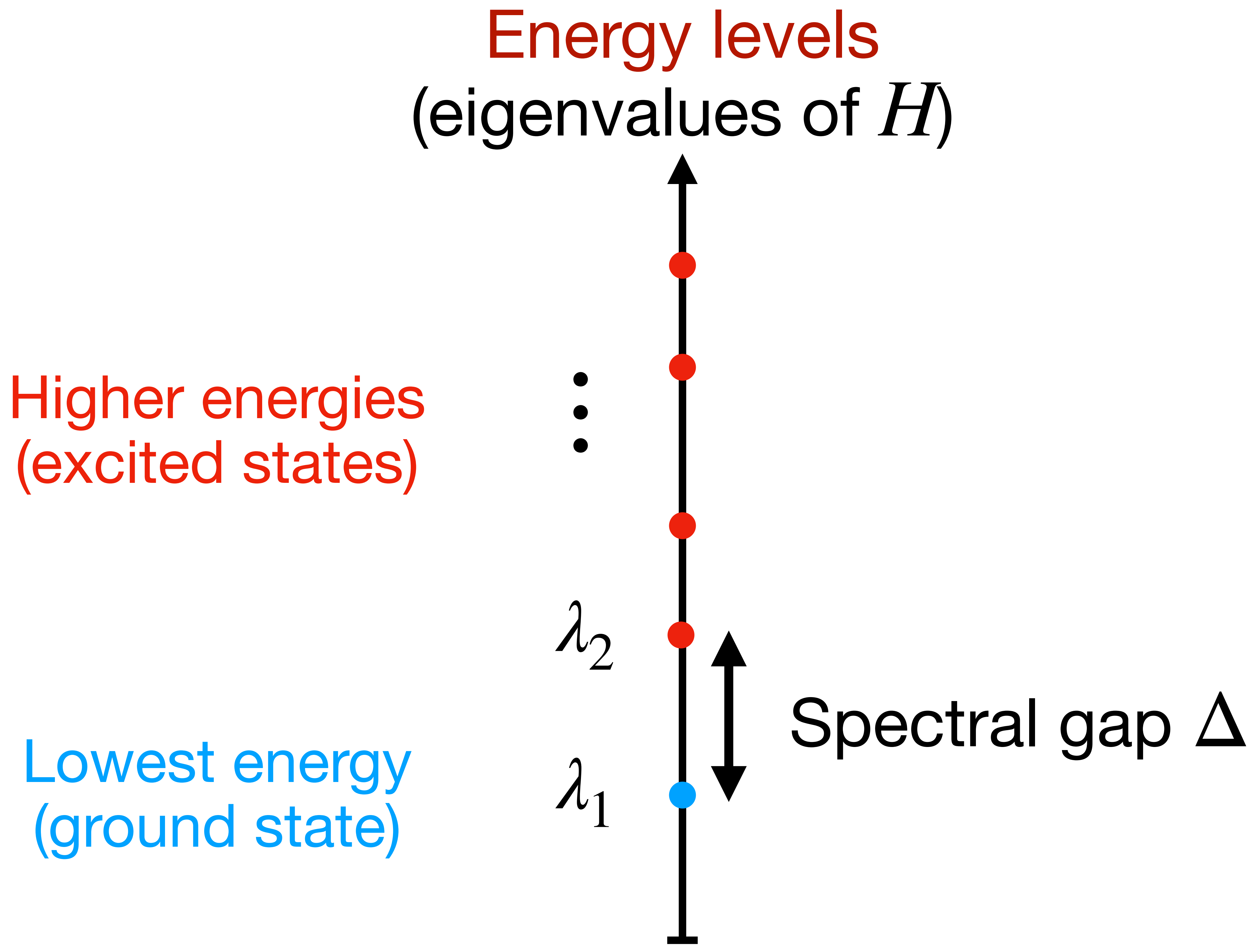
Allows computing energy levels of H



Cost: $\sim \epsilon^{-1} \times \mathrm{Sim}_H$

$\|H\| \leq 1$

Precision needed by Phase Estimation



Estimating the eigenvalues with precision $\epsilon < \Delta$ allows distinguishing ground state from excited states

Problem: eigenstates are not known a priori, what input should be provided to Phase estimation?

Warm-starts

= states with best-possible overlap with ground state

General purpose quantum optimizer

Warm-start
(decomposition into eigenbasis of H)

Quantum Phase Estimation

Amplitude amplification

$$|\psi(0)\rangle = \sum_i \alpha_i |v_i\rangle \longrightarrow \sum_i \alpha_i |v_i\rangle |\lambda_i \pm \Delta\rangle \longrightarrow |v_1\rangle |\lambda_1 \pm \Delta\rangle$$

Cost: $|\langle \psi(0) | v_1 \rangle|^{-1} \times \sim \Delta^{-1} \times \mathrm{Sim}_H$

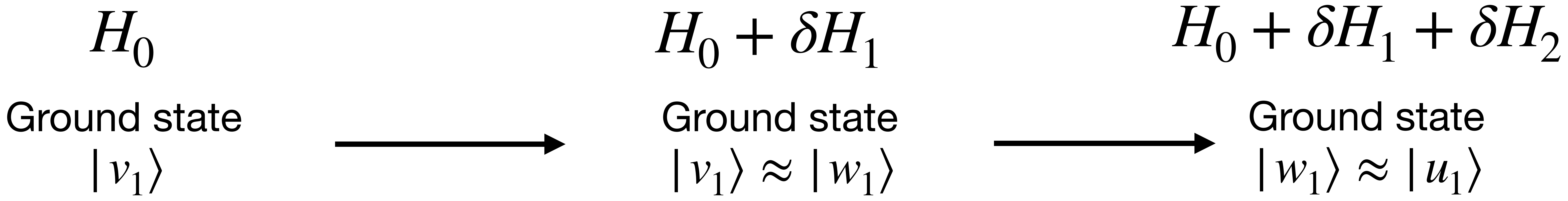
- ✓ Can provide the exact ground state
- ✓ Few assumptions required (H gapped and efficiently simulatable)
- ✗ Requires complicated quantum circuits (long coherence time, error correction…)
- ✗ Requires very good warm start (large overlap $\langle \psi(0) | v_1 \rangle$)

Next part: relax this requirement

Quantum Adiabatic Algorithm

Bootstrapping an approximate ground state

If we slightly perturb a Hamiltonian, its ground state should remain approximately the same.



H_0

Ground state $|v_1\rangle$

$H_0 + \delta H_1$

Ground state $|v_1\rangle \approx |w_1\rangle$

$H_0 + \delta H_1 + \delta H_2$

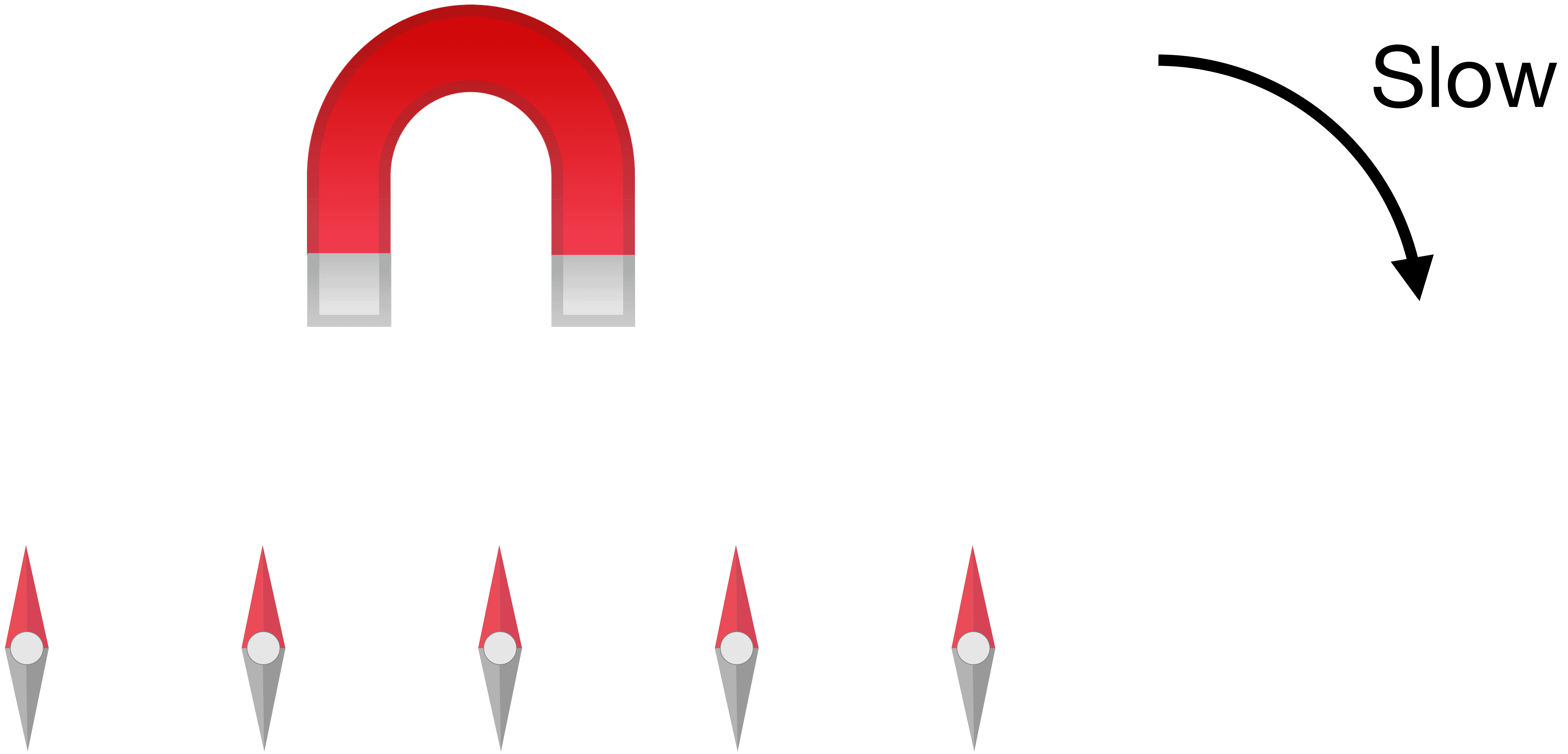
Ground state $|w_1\rangle \approx |u_1\rangle$

- Large overlap $|\langle v_1 | w_1 \rangle|$
- Use $|v_1\rangle$ as a warm-start to prepare $|w_1\rangle$

- Large overlap $|\langle w_1 | u_1 \rangle|$
- Use $|w_1\rangle$ as a warm-start to prepare $|u_1\rangle$

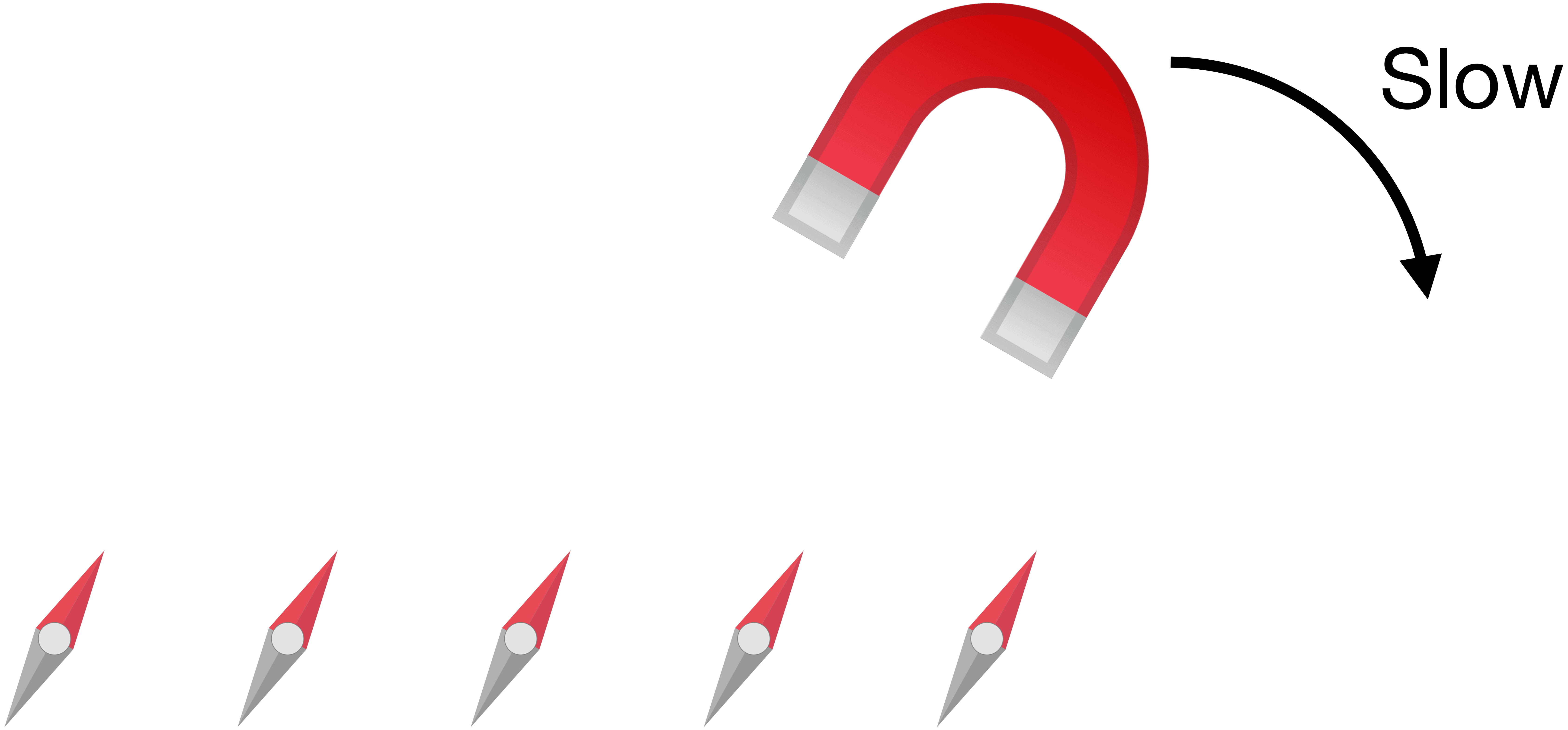
Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.



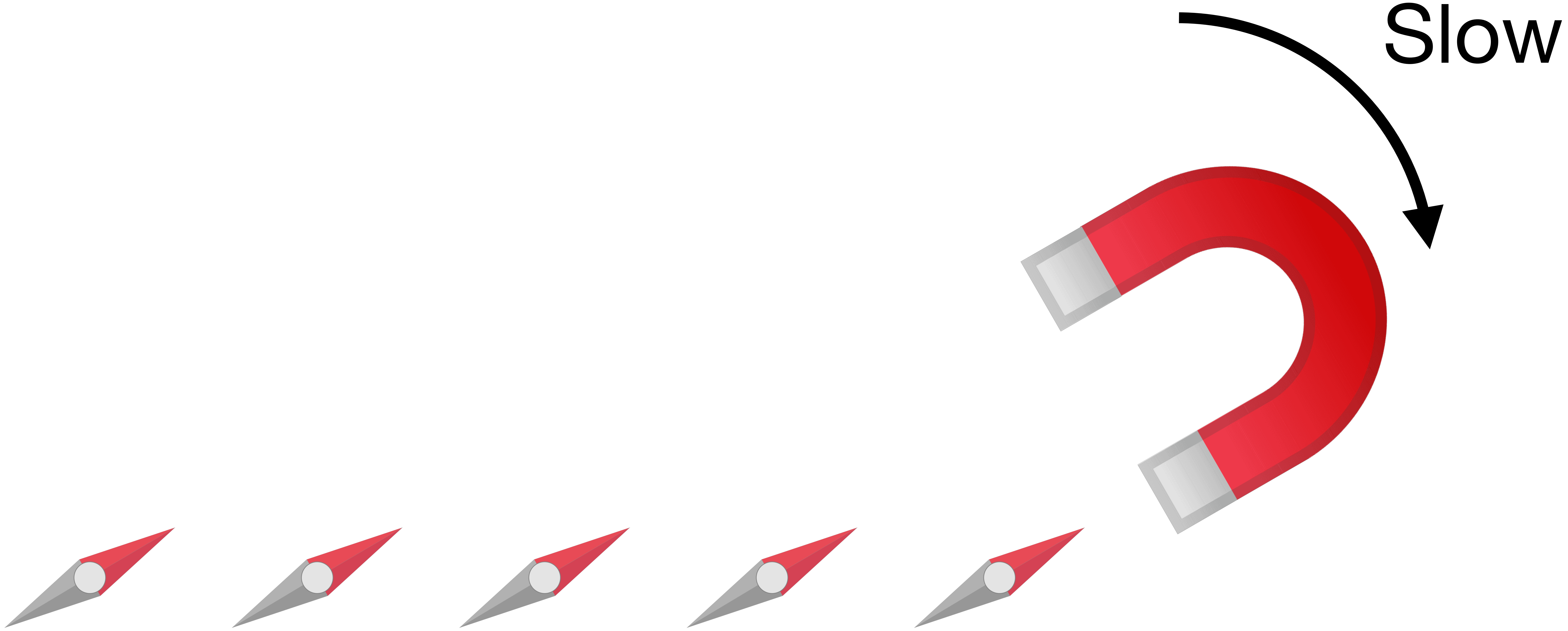
Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.



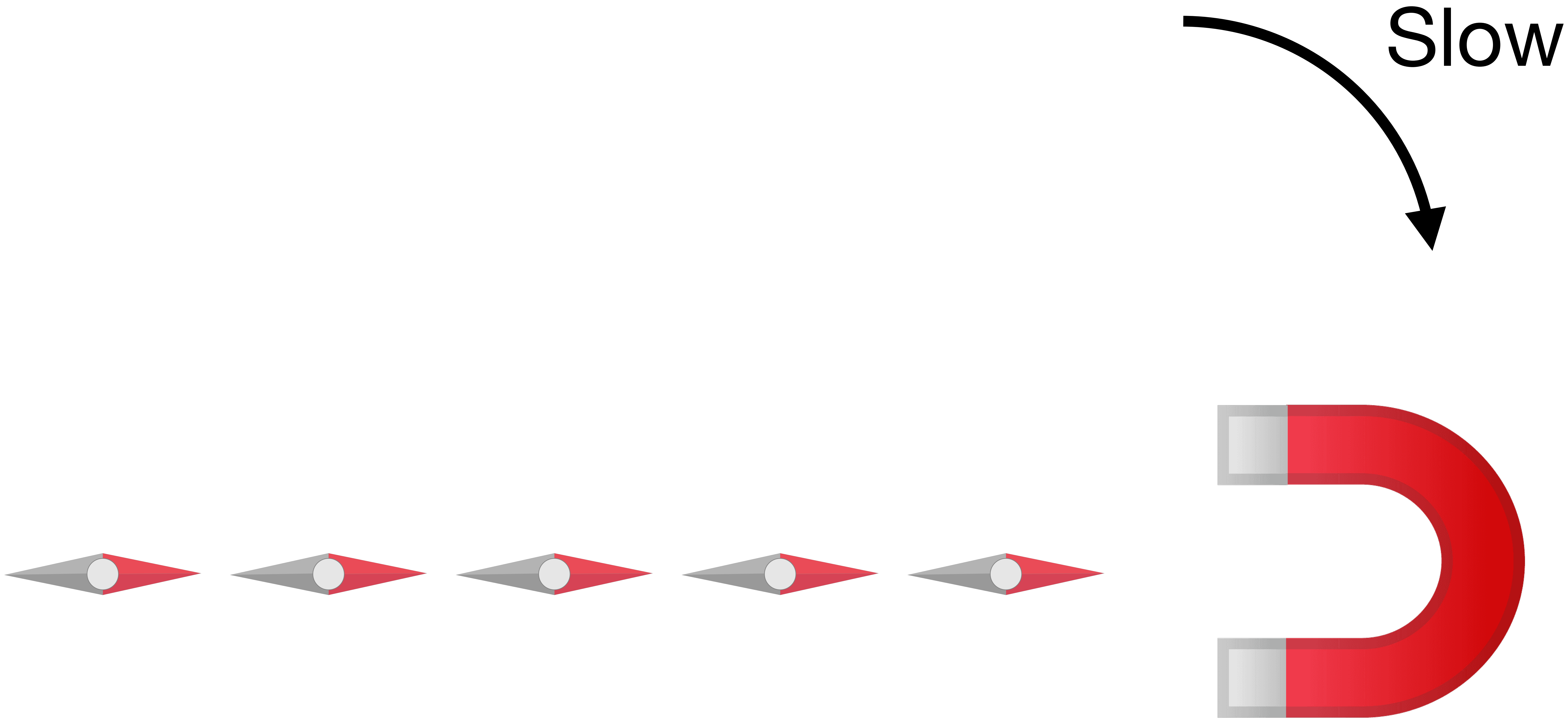
Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.



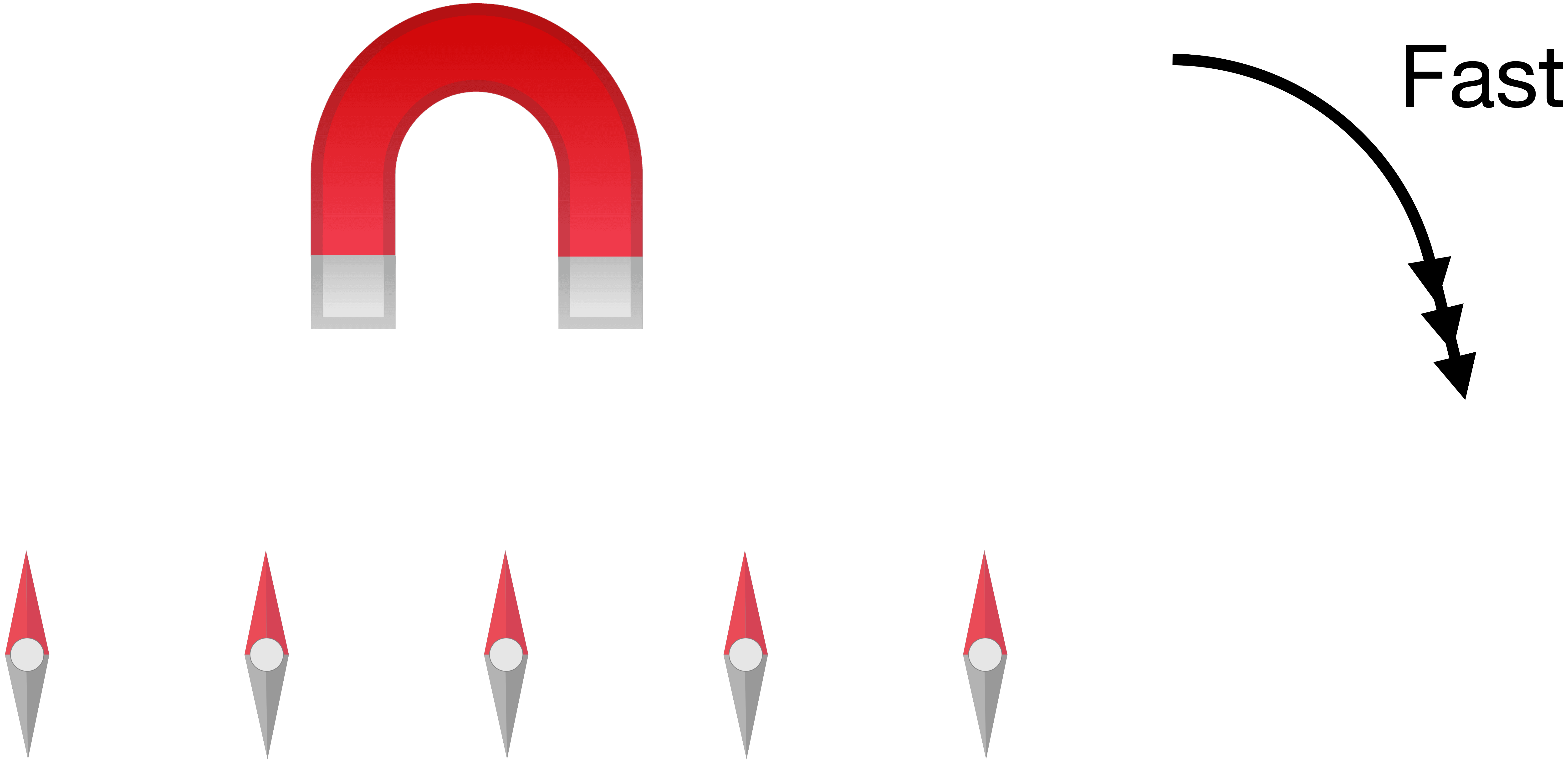
Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.



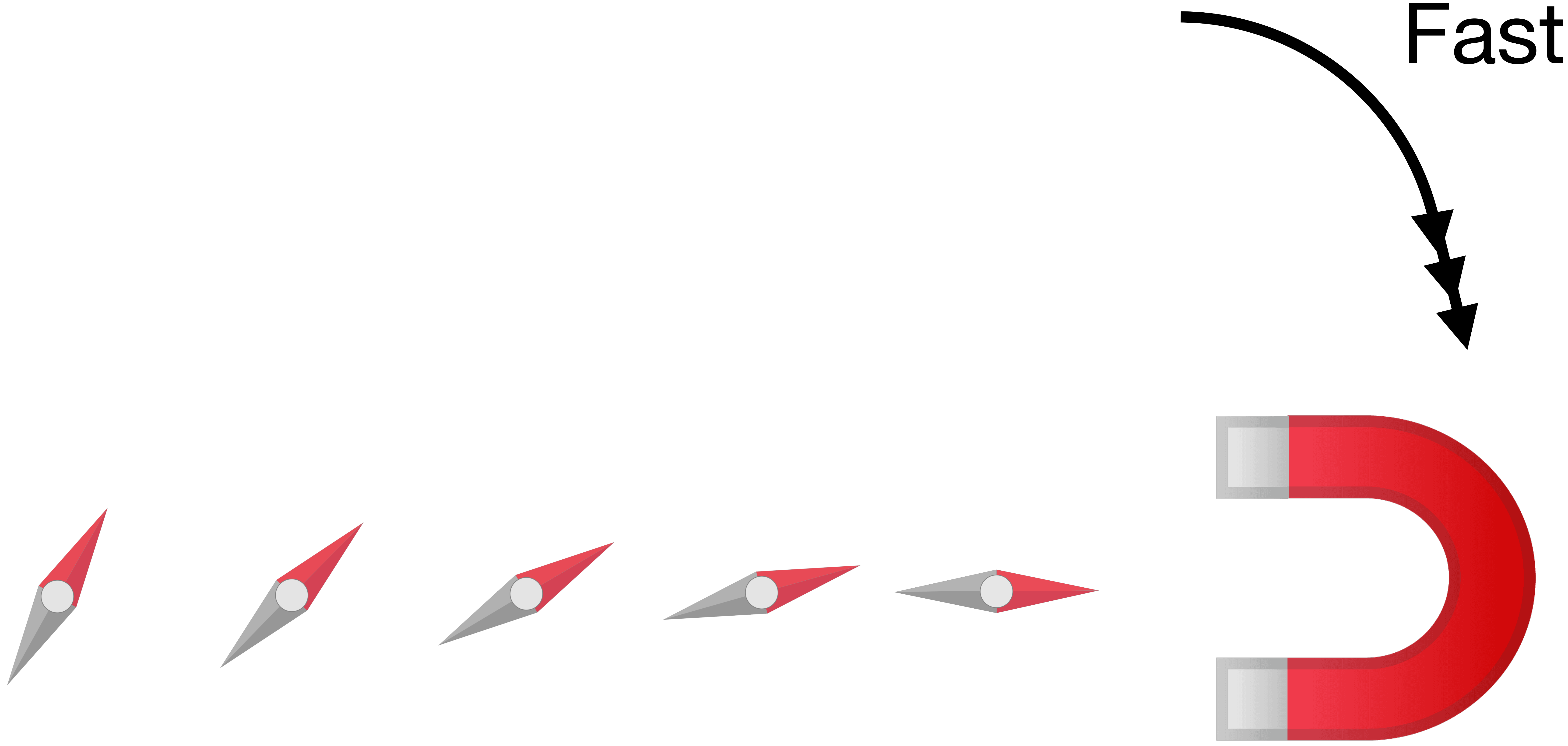
Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.



Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.



Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.

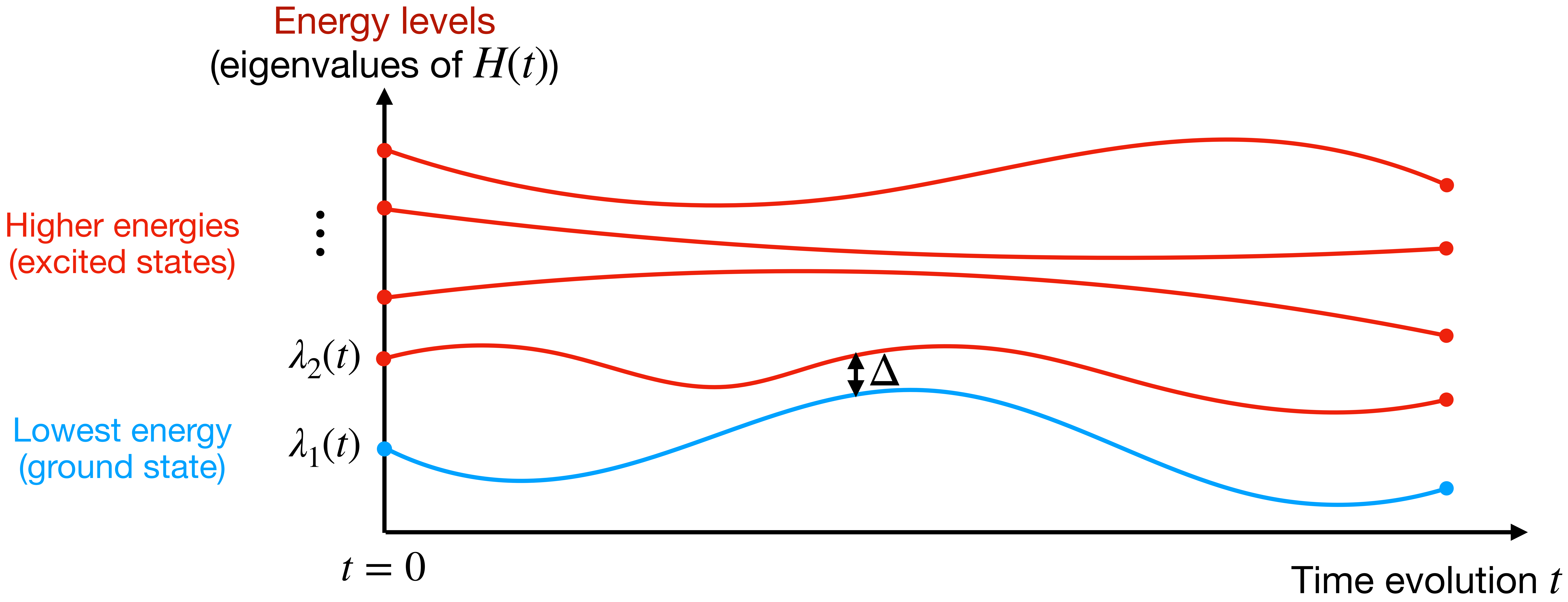
$$i\frac{d|\psi(t)\rangle}{dt} = H(vt)|\psi(t)\rangle$$

evolution slowed-down at speed $v \in (0,1]$

Time-dependent Hamiltonians:

- Eigenvalues and eigenvectors can also evolve over time
- Schrödinger equation has no analytical solution in general

Energy spectrum of time-evolving Hamiltonian



Assumption: no level crossings (nonzero energy gap Δ between ground and excited states throughout the evolution)

Adiabatic Theorem (simplified)

If a system is initialized in the ground state of a Hamiltonian $H(t)$ that evolves slowly over time, then it remains in the instantaneous ground state.

$$i\frac{d|\psi(t)\rangle}{dt} = H(vt)|\psi(t)\rangle$$

evolution slowed-down at speed $v \in (0,1]$

Minimum spectral gap: $\Delta = \min_{0\leq t\leq 1} \lambda_2(t) - \lambda_1(t)$

Cost: $\sim 1/v \times \mathrm{Sim}_H$

If $|\psi(0)\rangle$ is the ground state of $H(0)$ and $v \lesssim \Delta^2 \cdot \epsilon \cdot (\|\dot{H}(0)\| + \|\dot{H}(1)\| + \max_t \|\ddot{H}(t)\|^2 + \|\dot{H}(t)\|^3/\Delta)^{-1}$ then the solution $|\psi(1)\rangle$ to the Schrödinger eq. at $t = 1$ is ϵ-close to the ground state of $H(1)$

Applying the Adiabatic Theorem

H_1: target Hamiltonian whose ground state is sought (ex: QUBO Hamiltonian)

$$H_1 = -\sum_{i,i} (Z_i - Z_j)^2$$

H_0: starting Hamiltonian whose ground state is easy to prepare (ex: transverse field Hamiltonian)

$$H_0 = -\sum_i X_i$$

$H(t)$: interpolating Hamiltonian with $H(0) = H_0$ and $H(1) = H_1$ (ex: line interpolation)

$$H(t) = (1-t)\,H_0 + t\,H_1$$

Requirements:

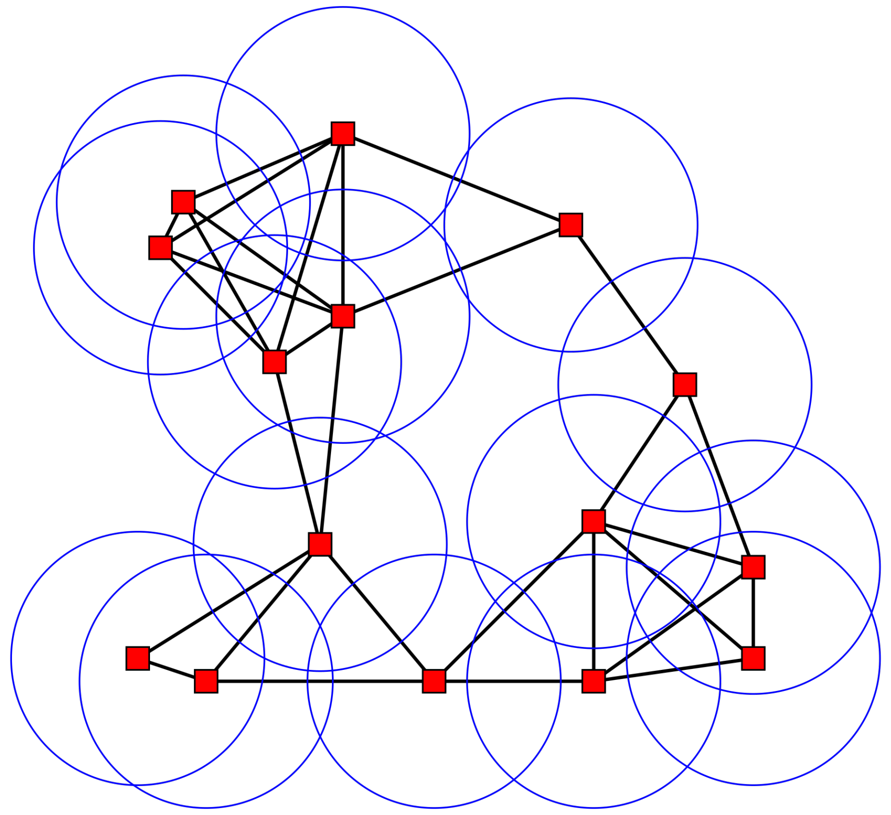
- H_0 and H_1 do not commute (or it leads to level crossings)
- Spectral gap Δ is large (allows fast evolution speed v)
- $H(t)$ can be simulated efficiently (depends on hardware restrictions)

Applying the Adiabatic Theorem

Caveat 1: spectral gap Δ is often exponentially small or unknown

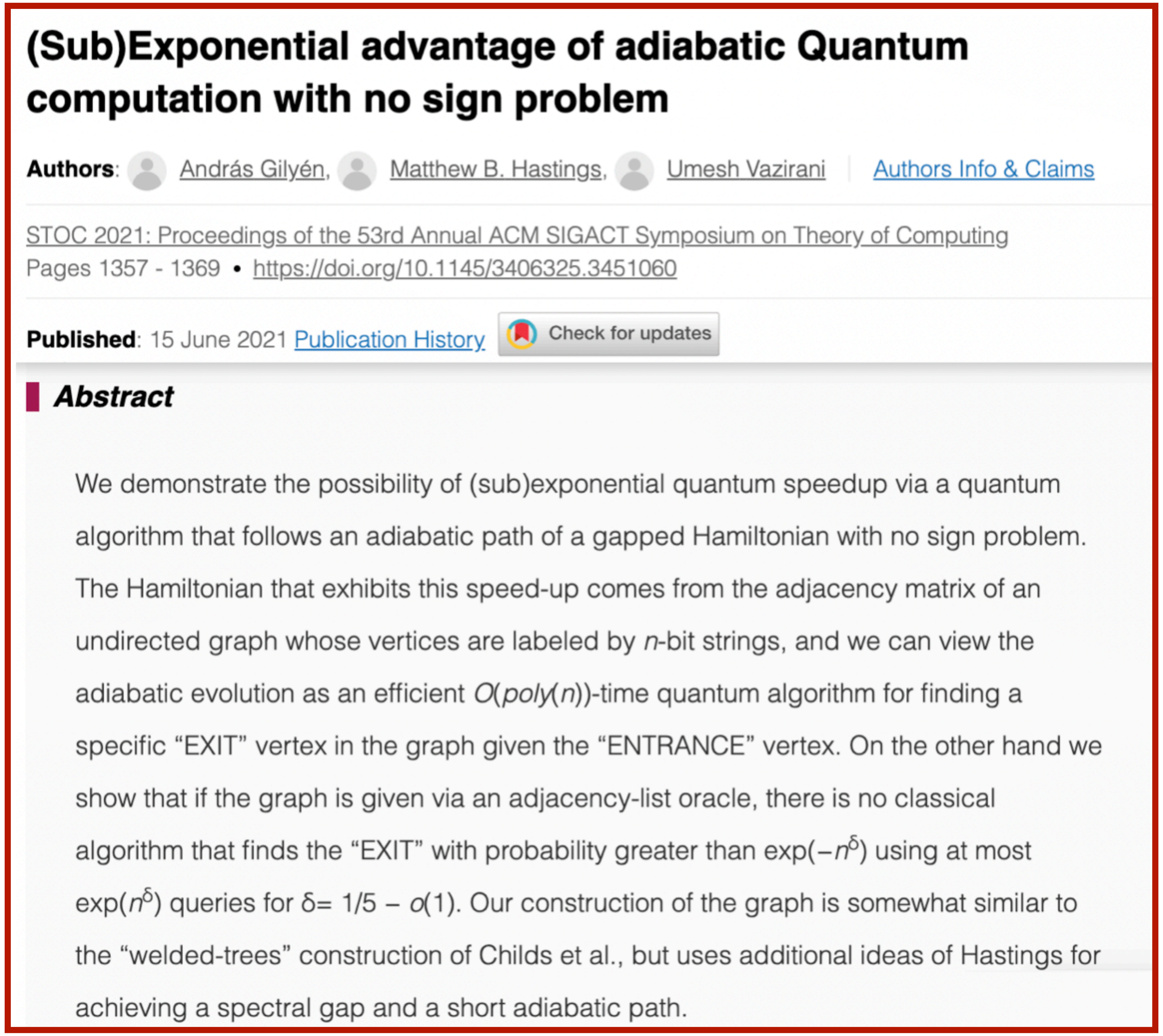
Caveat 2: choice of Hamiltonian is constrained by type of analog quantum computer

Example:
Rydberg atom arrays (as used by Pasqal or QuEra)
implement interactions corresponding to unit disk graphs



Adiabatic algorithm can also be run on digital, circuit-based computers
(discretized adiabatic evolution, randomized evolutions, …)

Caveat 3: requires sustained coherence throughout the entire runtime



(Sub)Exponential advantage of adiabatic Quantum computation with no sign problem

Authors: András Gilyén, Matthew B. Hastings, Umesh Vazirani | Authors Info & Claims

STOC 2021: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing
Pages 1357 - 1369 • https://doi.org/10.1145/3406325.3451060

Published: 15 June 2021 Publication History

Abstract

We demonstrate the possibility of (sub)exponential quantum speedup via a quantum algorithm that follows an adiabatic path of a gapped Hamiltonian with no sign problem. The Hamiltonian that exhibits this speed-up comes from the adjacency matrix of an undirected graph whose vertices are labeled by n-bit strings, and we can view the adiabatic evolution as an efficient $O(poly(n))$-time quantum algorithm for finding a specific "EXIT" vertex in the graph given the "ENTRANCE" vertex. On the other hand we show that if the graph is given via an adjacency-list oracle, there is no classical algorithm that finds the "EXIT" with probability greater than $\exp(-n^{\delta})$ using at most $\exp(n^{\delta})$ queries for $\delta = 1/5 - o(1)$. Our construction of the graph is somewhat similar to the "welded-trees" construction of Childs et al., but uses additional ideas of Hastings for achieving a spectral gap and a short adiabatic path.

There are Hamiltonians with properties similar to those in QUBO optimization (stoquastic), for which the adiabatic algorithm is provably much faster than any classical optimization algorithm

… but these Hamiltonians are somewhat artificial

Variational quantum algorithms

Reduce the search space to a smaller region than can be efficiently explored using a combination of classical + quantum computing

Variational Quantum Eigensolver (VQE)

The variational method

Ground state computation is an optimization problem:

$$\lambda_1 = \| H | v_1 \rangle \| = \min_{|v\rangle \in \mathbb{C}^{2^n} : \|v\| = 1} \| H | v \rangle \|$$

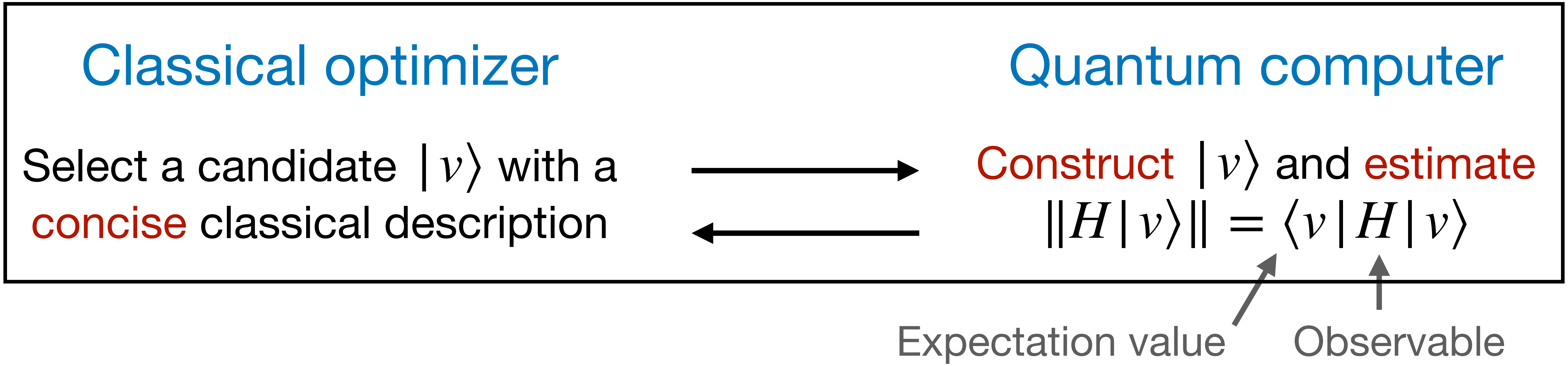
What if we use classical optimization solvers?

- Elements from the search space $| v \rangle \in \mathbb{C}^{2^n}$ can be difficult to store and manipulate on classical computers (exponential dimension)
- Objective function $\| H | v \rangle \|$ can be hard to evaluate

Variational method: hybrid quantum-classical optimizers

The variational method

The evaluation of the objective function is delegated to a quantum computer



Classical optimizer

Select a candidate $|v\rangle$ with a concise classical description

Quantum computer

Construct $|v\rangle$ and estimate $\| H | v \rangle \| = \langle v | H | v \rangle$

Expectation value

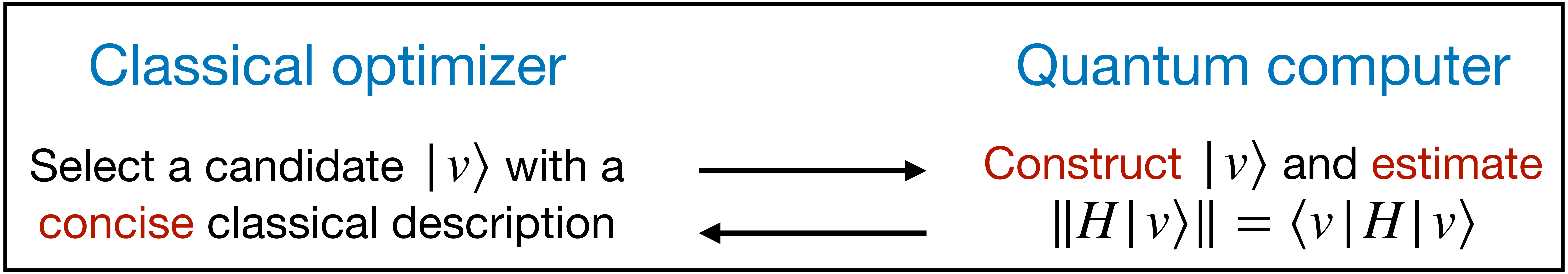
Observable

The classical optimizer refines its solution by making iterative calls to a quantum computer

The variational method

- Requires fewer quantum resources than full-fledged quantum algorithms
- A leading proposal for near-term quantum applications
- Tested in practice (ex: hybrid computing platform being installed at the French supercomputing center TGCC)
- Highly heuristic… but so is classical optimization on such problems

Variational Quantum Eigensolver



Classical optimizer

Select a candidate $|v\rangle$ with a concise classical description

Quantum computer

Construct $|v\rangle$ and estimate $\|H|v\rangle\| = \langle v|H|v\rangle$

How to represent the candidate states?

Classical parameters $(N \ll 2^n)$

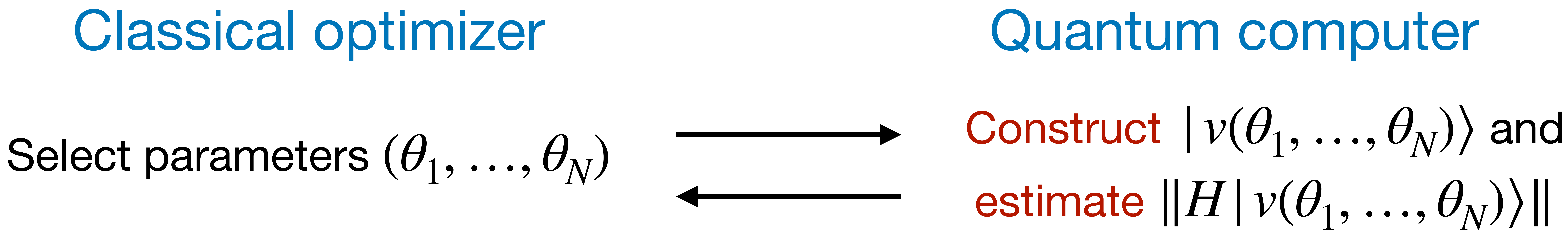
Quantum state

Ansatz: $(\theta_1, \ldots, \theta_N) \longmapsto |v(\theta_1, \ldots, \theta_N)\rangle$

Mapping should be easy to do on a quantum computer

Example (next part): the QAOA ansatz

Energy estimation



Classical optimizer

Quantum computer

Select parameters $(\theta_1, \ldots, \theta_N)$

Construct $|v(\theta_1, \ldots, \theta_N)\rangle$ and estimate $\| H | v(\theta_1, \ldots, \theta_N)\rangle \|$

The energy estimation $\| H | v\rangle \|$ could be performed using Quantum Phase Estimation, but this would make the algorithm unsuitable for near-term quantum architectures.

Instead, we typically rely on random measurement strategies based on prior knowledge about H (ex: if H is a QUBO Hamiltonian, we can average multiple measurements in the standard basis)

Barren Plateaus

The choice of the variational ansatz is crucial in making the method succeed

Highly expressive ansätze (i.e., those spanning a large region of the original search space) are more likely to cause the classical optimizer to get stuck in local minima

Phenomenon known as Barren Plateaus:

For a given observable H over n qubits, all but an exponentially small fraction of the quantum states $|v\rangle \in \mathbb{C}^{2^n}$ have energy exponentially close to the average: $\| H|v\rangle \| = \mathrm{Tr}(H)/2^n \pm 2^{-\Omega(n)}$.

(i.e., the optimization landscape is nearly flat over a large sub-region)

Quantum Approximate Optimization Algorithm (QAOA)

QAOA instantiates the Variational Quantum Eigensolver with an ansatz inspired by the Quantum Adiabatic Algorithm.

Trotterized adiabatic evolution

Recall: The adiabatic algorithm solves the Schrödinger equation $i\frac{d|\psi(t)\rangle}{dt} = H(t)|\psi(t)\rangle$.

Goal: Understand what the ground state $|\psi(1)\rangle$ looks like to derive an ansatz

Continuous-time Hamiltonian $H(t)$ can be approximated by a finite sequence:

$H(0), H(\delta), H(2\delta), \ldots, H(1)$ for a sufficiently small time step $\delta \ll 1$.

By solving the corresponding sequence of Schrödinger equations,

$$|\psi(1)\rangle \underset{\delta \to 0}{=} e^{-i\delta H(1)} \ldots e^{-i\delta H(2\delta)} e^{-i\delta H(\delta)} e^{-i\delta H(0)} |\psi(0)\rangle$$

Trotterized adiabatic evolution

$$|\psi(1)\rangle \underset{\delta \to 0}{=} e^{-i\delta H(1)} \dots e^{-i\delta H(2\delta)} e^{-i\delta H(\delta)} e^{-i\delta H(0)} \,|\psi(0)\rangle$$

Suppose $H(t)$ is given by the line interpolation $H(t) = (1-t)\,H_0 + t\,H_1$

Each unitary can be expanded as follows:

$$e^{-i\delta H(j\delta)} = e^{-i\delta(1-j\delta)H_0 - ij\delta^2 H_1} \neq \cancel{e^{-i\delta(1-j\delta)H_0} e^{-ij\delta^2 H_1}}$$

H_0, H_1 don’t commute

$$\underset{\delta' \to 0}{=} \left(e^{-i\delta'\delta(1-j\delta)H_0} e^{-ij\delta'\delta^2 H_1} \right)^{1/\delta'}$$

Trotter formulas

Trotterized adiabatic evolution

Putting everything together:

$$|\psi(1)\rangle \underset{\substack{\delta \to 0 \\ \delta' \to 0}}{=} \prod_{j=0}^{1/\delta} \left(e^{-i\delta'\delta(1-j\delta)H_0} e^{-ij\delta'\delta^2 H_1} \right)^{1/\delta'} |\psi(0)\rangle$$

Ground state is obtained by alternating small evolutions according to H_0 or H_1

For any $\epsilon \in (0,1)$, there exists a depth p and a sequence of angles $\theta_1, \theta_2, \ldots, \theta_{2p} \in [0,2\pi]$ such that,

$$|\psi(1)\rangle \approx_\epsilon e^{-i\theta_{2p}H_0} e^{-i\theta_{2p-1}H_1} \ldots e^{-i\theta_3 H_1} e^{-i\theta_2 H_0} e^{-i\theta_1 H_1} |\psi(0)\rangle$$

The QAOA ansatz

QAOA explores ansätze of the form:

$$|v(\theta_1, \dots, \theta_p)\rangle \;=\; e^{-i\theta_{2p}H_0} e^{-i\theta_{2p-1}H_1} \dots e^{-i\theta_3 H_1} e^{-i\theta_2 H_0} e^{-i\theta_1 H_1} |\psi(0)\rangle$$

where, typically, p is much smaller than what is required by Trotter approximation.

Advantage over adiabatic algorithm: low-depth quantum circuits

Search space with $p = 1$ already encompass some interesting states:

Non-trivial approximate solutions to QUBO Hamiltonians (ex: Max-Cut on 3-regular graphs)

arXiv:1411.4028

Measurement distributions that cannot be efficiently classically sampled from

arXiv:1602.07674

Significant speedups for certain Constraint Satisfaction Problems over best-known classical algorithms

arXiv:2411.04979

… but provable speedups remain elusive and QAOA ansatz is harder to analyze for $p > 1$

Quantum optimization algorithms

Lecture 1 (Physics-inspired)

Exact algorithms

- Quantum Phase Estimation (QPE)
- Quantum Adiabatic Algorithm (QAA)

Variational quantum algorithms

- Variational Quantum Eigensolver (VQE)
- Quantum Approximate Optimization Algorithm (QAOA)

Lecture 2 (Oracle-based)

Grover-type algorithms

- Quantum Minimum Finding
- Minimum Spanning Tree

Gradient computation

Monte-Carlo algorithms

- Linear programming
- Escaping Saddle Points