Quantum Optimization Algorithms – Part II

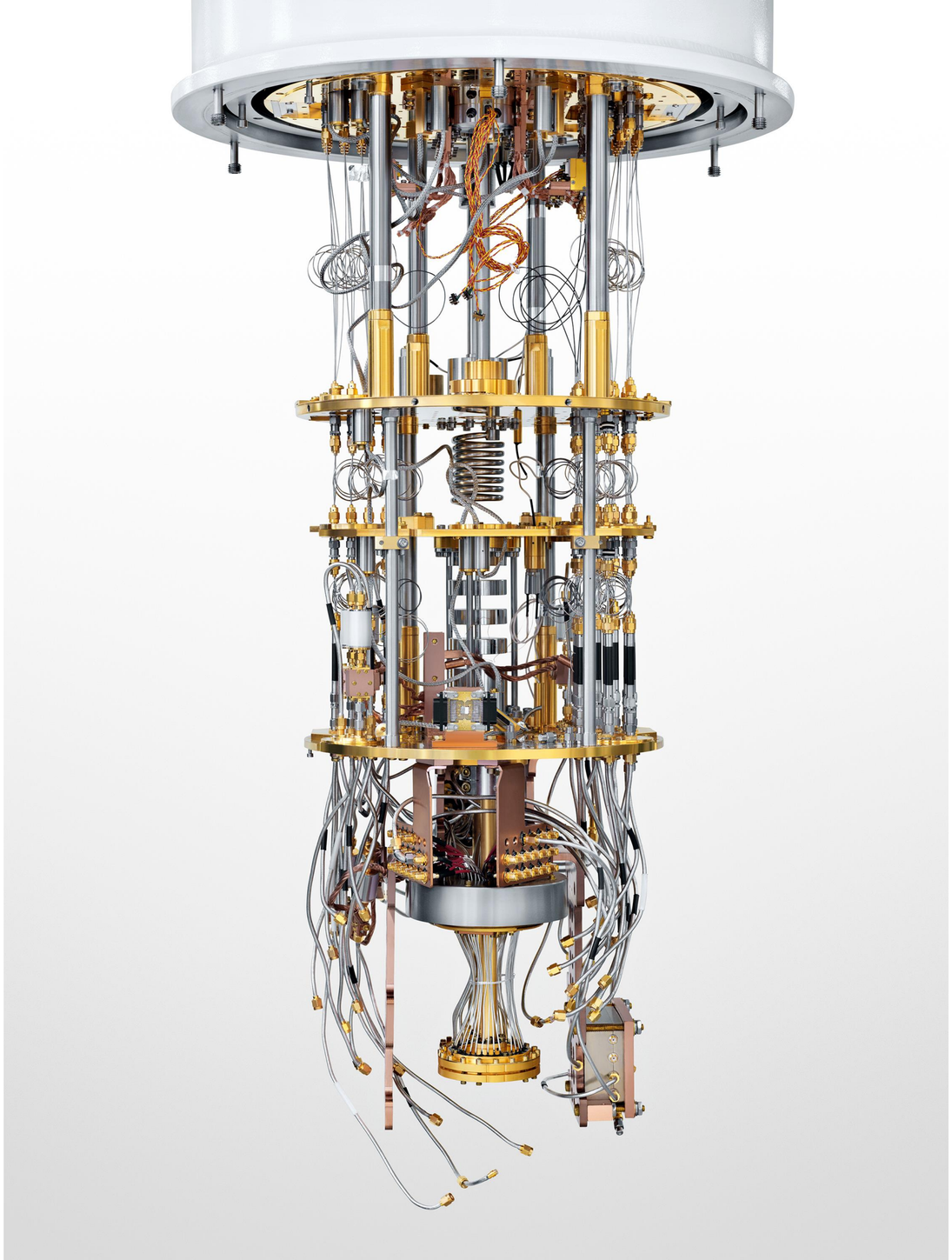
Alejandro Gómez Cadavid

Kipu Quantum GmbH

3rd Quantum Computing School ICTP-SAIFR

São Paulo, 2026

Quantum Optimization



Optimization problems

Finding the value of a set of decision variables that minimizes an objective function while satisfying a given set of constraints.

$$\min_{x \in \chi} \; f(x)$$
$$s.t. \quad g_i(x) \leq 0, \qquad i = 1, \dots, m$$
$$s.t. \quad h_j(x) = 0, \qquad j = 1, \dots, p$$

$\chi = \mathbb{R}^N$	Continuous
$\chi = \mathbb{Z}^N or \{0,1\}^N$	Integer / binary
$\chi = \mathbb{R}^{N_c} \times \mathbb{Z}^{N_d}$	Mixed-integer

Problem class, by the shape of f(x)

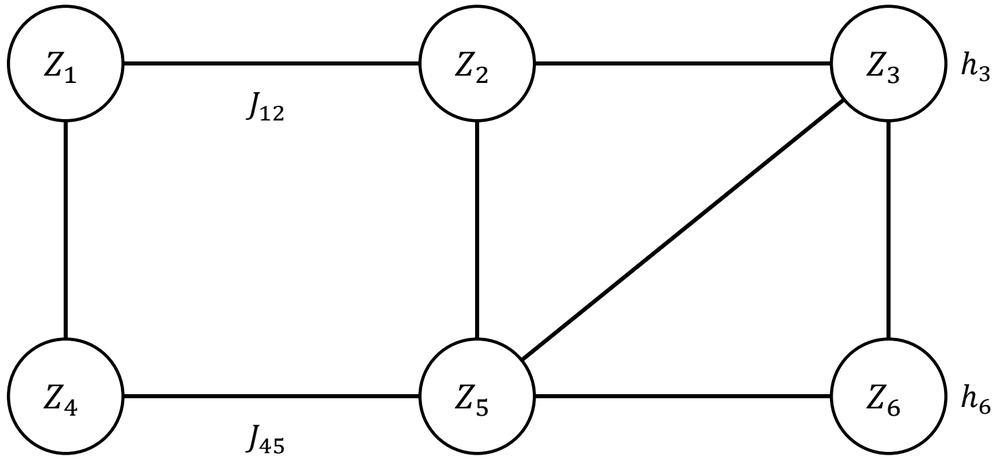
LP Linear program	QP Quadratic program	NLP Nonlinear program
$f(x) = c^T x$	$f(x) = \frac{1}{2} x^T Q x + c^T x$	$f(x)$ nonlinear
Linear f, linear constraints	Quadratic f, linear constraints	Nonlinear f, g or h

QUBO and HUBO: costs on graphs

A QUBO couples bits in pairs (a graph); a HUBO also couples three or more (a hypergraph).

QUBO: pairs only

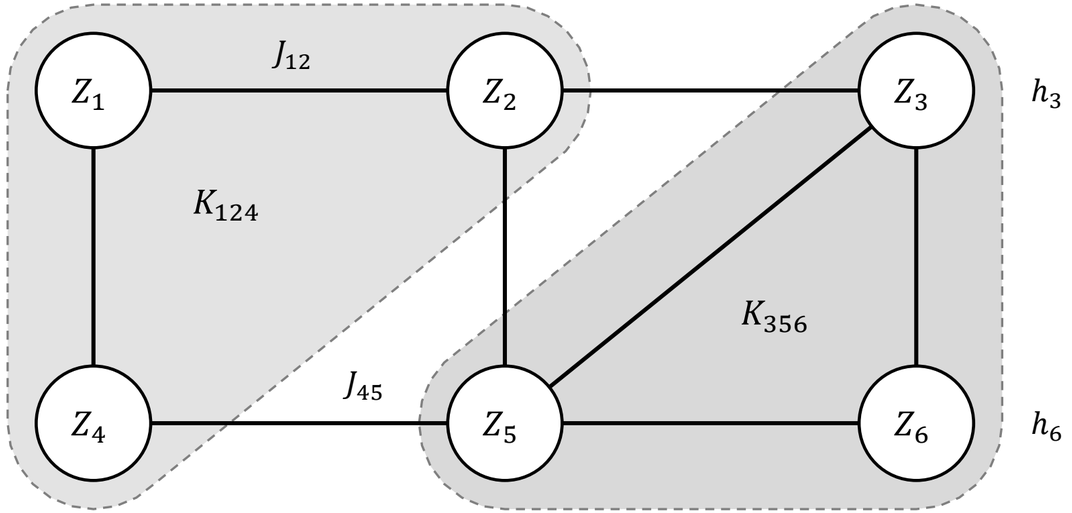
$$f(Z) = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j$$



Node: bit Z_i, weight h_i. Edge: pair term, weight J_{ij}.

HUBO: three or more bits

$$f(Z) = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j + \sum_{i<j<k} K_{ijk} Z_i Z_j Z_k + \cdots$$



Hyperedge: a term on three or more bits, weight K_{ijk}.

On qubits, every edge becomes a $Z_i Z_j$ interaction and every hyperedge a $Z_i Z_j Z_k$ interaction.

Following the instantaneous eigenstate

If the state stays in the instantaneous ground state, it ends in the solution of the optimization problem.

Interpolate from an easy Hamiltonian to the problem

$$H_0(\lambda) = (1-\lambda)H_i + \lambda H_P, \qquad \lambda(t/T):\ 0 \rightarrow 1$$

Start: $H_0(0) = H_i = -\sum_i X_i$

ground state easy to prepare

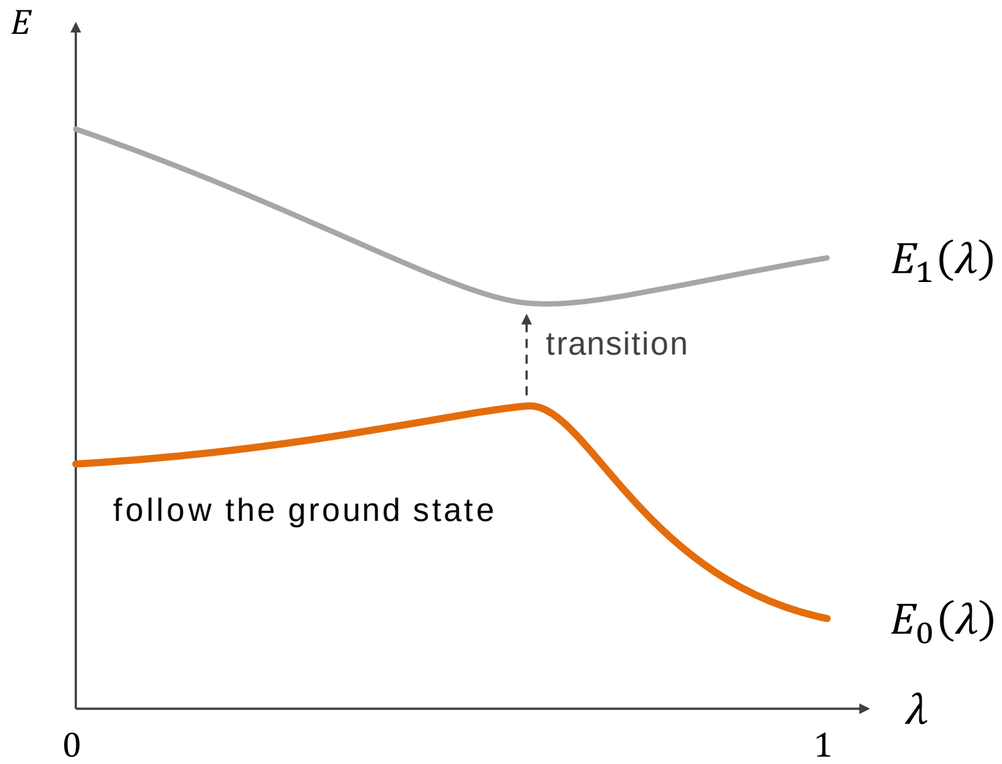
End: $H_0(1) = H_P = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j + \cdots$

ground state is the solution (previous lecture)

Two ways to avoid transitions

Adiabatic: drive slowly, so transitions have no time to build up

Counterdiabatic: add a term that cancels the transition term



Schrödinger equation in a moving frame

Avoiding transitions via slow driving.

$H_0(\lambda) = (1-\lambda)H_i + \lambda H_P$

$H_0(\lambda)|n(\lambda)\rangle = E_n(\lambda)|n(\lambda)\rangle$

$|n(\lambda)\rangle = U_0(\lambda)|n(0)\rangle$

Moving frame

Lab's frame

$i\hbar\partial_t|\tilde{\psi}(\lambda)\rangle = \big(\underbrace{\tilde{H}_0(\lambda)}_{\text{diagonal}} - \underbrace{\dot{\lambda}\,\tilde{A}_\lambda}_{\text{transitions}}\big)|\tilde{\psi}(\lambda)\rangle$

$A_\lambda = i\hbar(\partial_\lambda U_0)U_0^\dagger$ Adiabatic gauge potential

$i\hbar\partial_t|\psi_0(\lambda)\rangle = H_0(\lambda)|\psi_0(\lambda)\rangle$

$\langle n|A_\lambda|m\rangle = i\hbar\dfrac{\langle n|\partial_\lambda H_0|m\rangle}{E_m(\lambda) - E_n(\lambda)}$

Adiabatic condition

Avoiding transitions via slow driving.

$$i\hbar\partial_t|\tilde{\psi}(\lambda)\rangle = \big(\underbrace{\tilde{H}_0(\lambda)}_{\text{diagonal}} - \underbrace{\dot{\lambda}\,\tilde{A}_\lambda}_{\text{transitions}}\big)|\tilde{\psi}(\lambda)\rangle$$

$$A_\lambda = i\hbar(\partial_\lambda U_0)U_0^\dagger$$ Adiabatic gauge potential

$$\langle n|A_\lambda|m\rangle = i\hbar\frac{\langle n|\partial_\lambda H_0|m\rangle}{E_m(\lambda) - E_n(\lambda)}$$

$$i\hbar\partial_t|\tilde{\psi}(\lambda)\rangle = \tilde{H}_0(\lambda)|\tilde{\psi}(\lambda)\rangle$$

A coupling cannot mix two levels if it is much smaller than their splitting

$$\frac{\hbar}{T}\frac{d\lambda}{ds}\frac{|\langle n|\partial_\lambda H_0|m\rangle|}{E_m(\lambda) - E_n(\lambda)} \ll E_m(\lambda) - E_n(\lambda) \longrightarrow \boxed{T \gg \hbar \max_{\lambda, n\neq m}\frac{d\lambda}{ds}\frac{|\langle n(\lambda)|\partial_\lambda H_0|m(\lambda)\rangle|}{[E_n(\lambda) - E_m(\lambda)]^2}}$$

$$T \gg \frac{\hbar}{\Delta_{\min}^2}\max_\lambda\frac{d\lambda}{ds}|\langle 1(\lambda)|\partial_\lambda H_0|0(\lambda)\rangle|$$

Counterdiabatic driving

Avoiding transitions by counteracting the term responsible for transitions.

$$i\hbar\partial_t|\psi_0(\lambda)\rangle = H_0(\lambda)|\psi_0(\lambda)\rangle$$

Moving frame

$$i\hbar\partial_t|\tilde{\psi}(\lambda)\rangle = \big(\underbrace{\tilde{H}_0(\lambda)}_{\text{diagonal}} - \underbrace{\dot{\lambda}\tilde{A}_\lambda}_{\text{transitions}}\big)|\tilde{\psi}(\lambda)\rangle$$

$$H_0(\lambda) \to H_0(\lambda) + \dot{\lambda}A_\lambda$$

$$i\hbar\partial_t|\tilde{\psi}(\lambda)\rangle = \tilde{H}_0(\lambda)|\tilde{\psi}(\lambda)\rangle$$

$$i\hbar\partial_t|\psi_0(\lambda)\rangle = \big(H_0(\lambda) + \dot{\lambda}A_\lambda\big)|\psi_0(\lambda)\rangle$$

Moving frame

$$i\hbar\partial_t|\tilde{\psi}(\lambda)\rangle = \big(\underbrace{\tilde{H}_0(\lambda)}_{\text{diagonal}} - \underbrace{\cancel{\dot{\lambda}\tilde{A}_\lambda + \dot{\lambda}\tilde{A}_\lambda}}_{\text{transitions}}\big)|\tilde{\psi}(\lambda)\rangle$$

Counterdiabatic Hamiltonian

Avoiding transitions by counteracting the term responsible for transitions.

The total Hamiltonain including the counterdiabatic term is given by

$$H(\lambda) = H_0(\lambda) + \underbrace{\dot{\lambda} A_\lambda}_{H_{\mathrm{CD}}(\lambda)}$$

$$H_{\mathrm{CD}}(\lambda) = \dot{\lambda} A_\lambda = i\hbar\dot{\lambda} \sum_{m \neq n} \frac{|m(\lambda)\rangle\langle m(\lambda)|\partial_\lambda H_0|n(\lambda)\rangle\langle n(\lambda)|}{E_n(\lambda) - E_m(\lambda)}$$

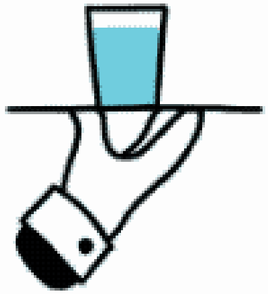
In the moving frame the transition term cancels, for any T: $i\hbar\partial_t \left|\tilde{\psi}(\lambda)\right\rangle = \tilde{H}_0(\lambda)\left|\tilde{\psi}(\lambda)\right\rangle$

Building it requires the eigenstates and gaps of $H_0(\lambda)$ along the whole path.

Berry, J. Phys. A 42, 365303 (2009)

Adiabatic

slow: the liquid stays level



Diabatic

fast: the liquid spills (transitions)

Counterdiabatic

fast + compensating tilt $\dot{\lambda}A_{\lambda}$

Defining the initial Hamiltonian

$$H_0(\lambda) = (1-\lambda)H_i + \lambda H_P$$

$$H(\lambda) = H_0(\lambda) + \dot{\lambda}A_\lambda$$

$$H_i = -\sum_i X_i$$

easy ground state



$|+\rangle^{\otimes N}$

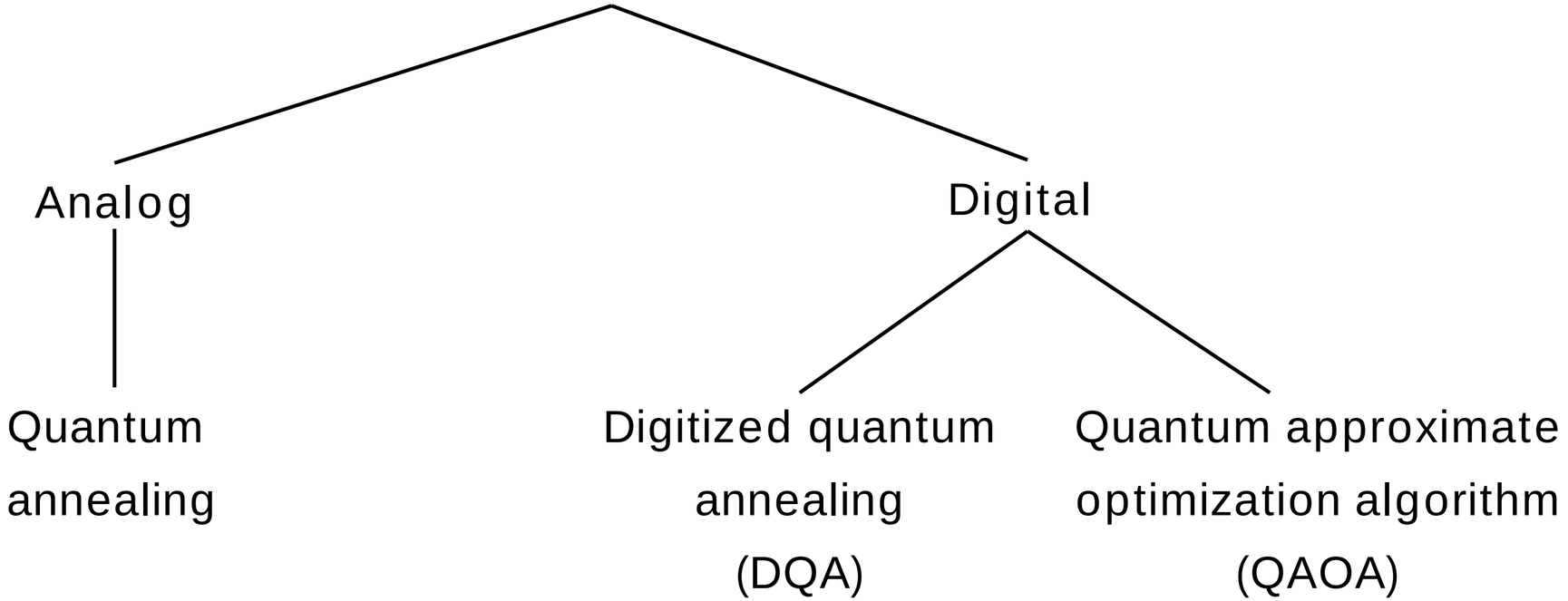
$$H_P = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j + \sum_{i<j<k} K_{ijk} Z_i Z_j Z_k + \cdots$$

ground state = solution

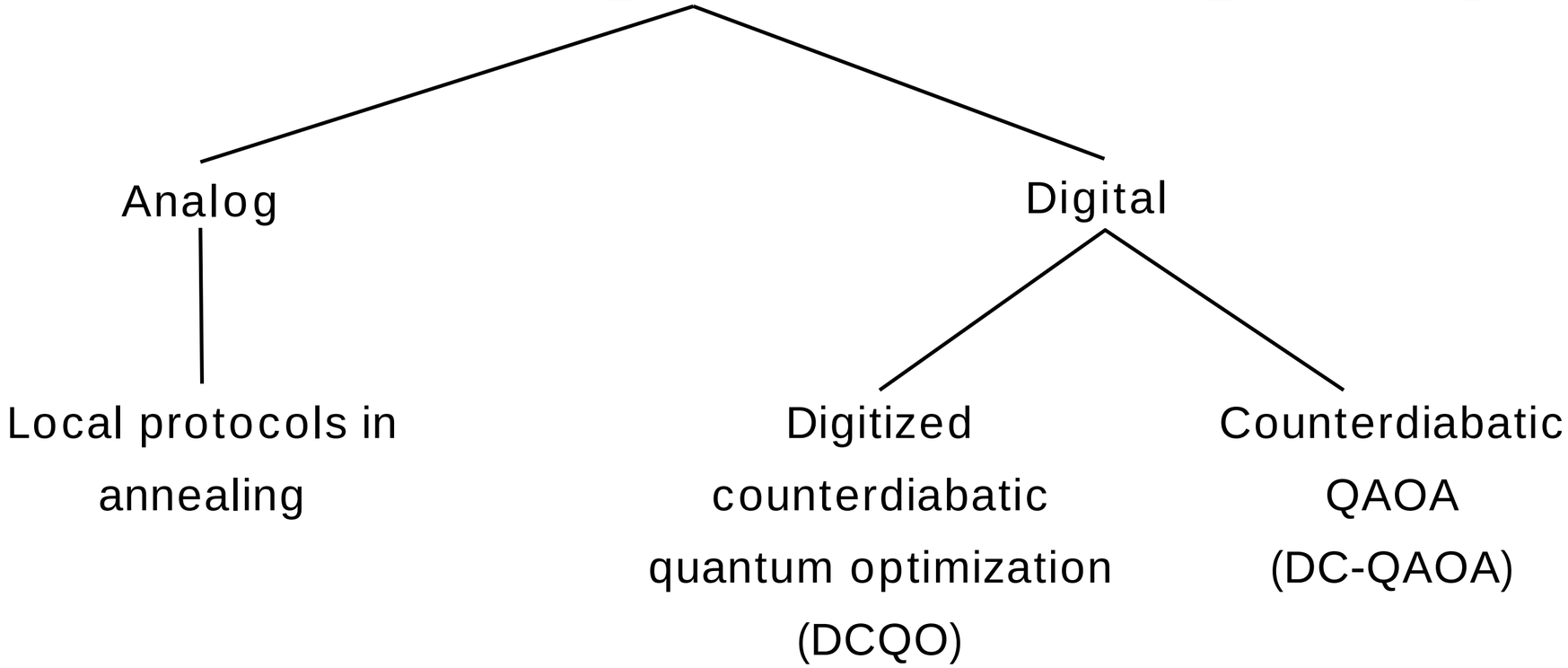


$|z_1, \cdots, z_N\rangle$

Adiabatic quantum computing



Counterdiabatic quantum computing



Digitized time evolutions

Translating a continuous time evolution into sequences of quantum gates.

$$U(T,0) = \mathcal{T}\exp\left(-i\int_0^T dt\, H(t)\right)$$

$$H(t) = \sum_{m=1}^{M} C_m(t) P_m$$

Digitization + Trotterization

$$T = n\,\Delta t$$

$$U(T,0) \to U_{dig}(0,T) = \prod_{j=1}^{n}\prod_{m=1}^{M} \exp(-i\,\Delta t\, C_m(j\Delta t)\, P_m)$$

1. Exact evolution
time-ordered, because $H(t)$ does not commute with itself at different times

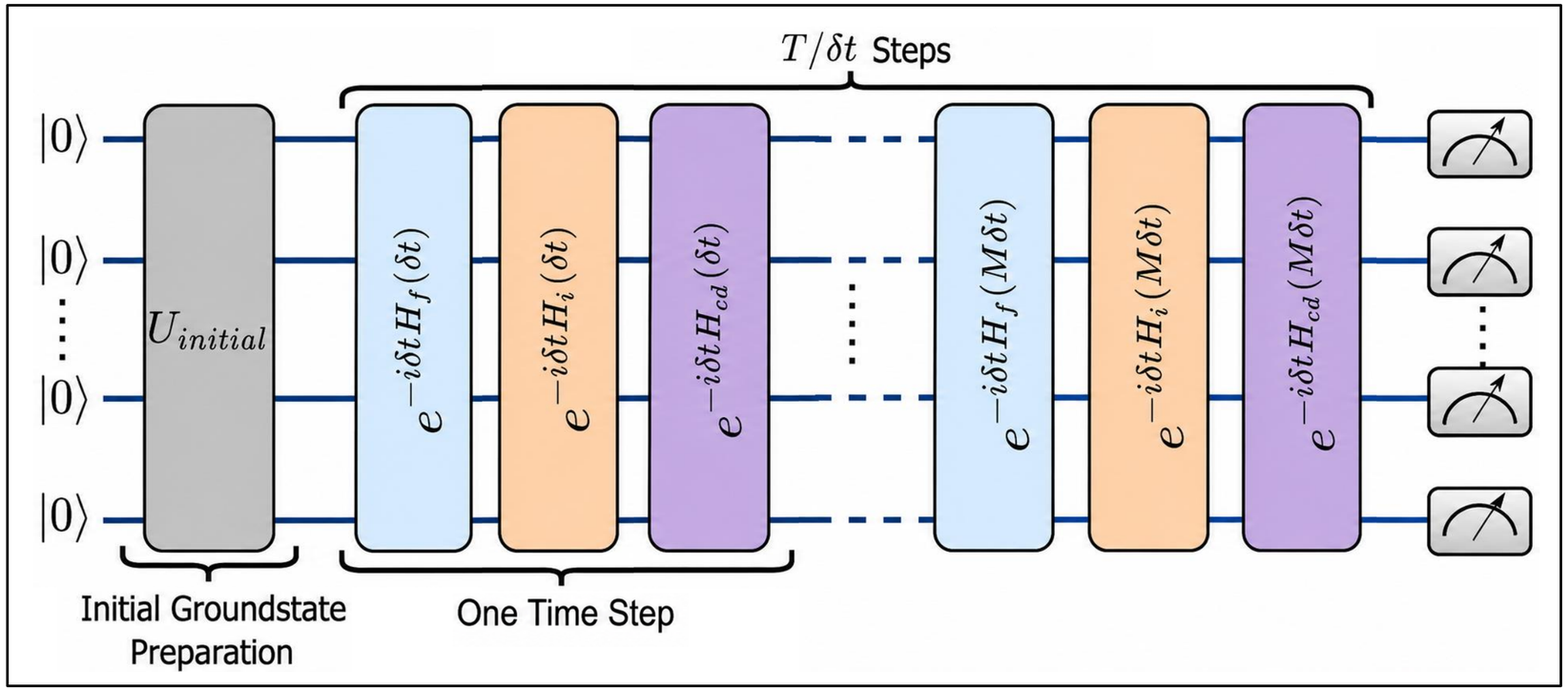
2. Pauli decomposition
expand in Pauli strings $P_m \in \{I, X, Y, Z\}^{\otimes N}$ with weights $C_m(t)$

3. Digitization
split T into n steps of length Δt; weights frozen at $t = j\Delta t$ in each step

4. Gates
each factor is one Pauli rotation, a few native gates; Trotter error $O(\Delta t^2)$ per step

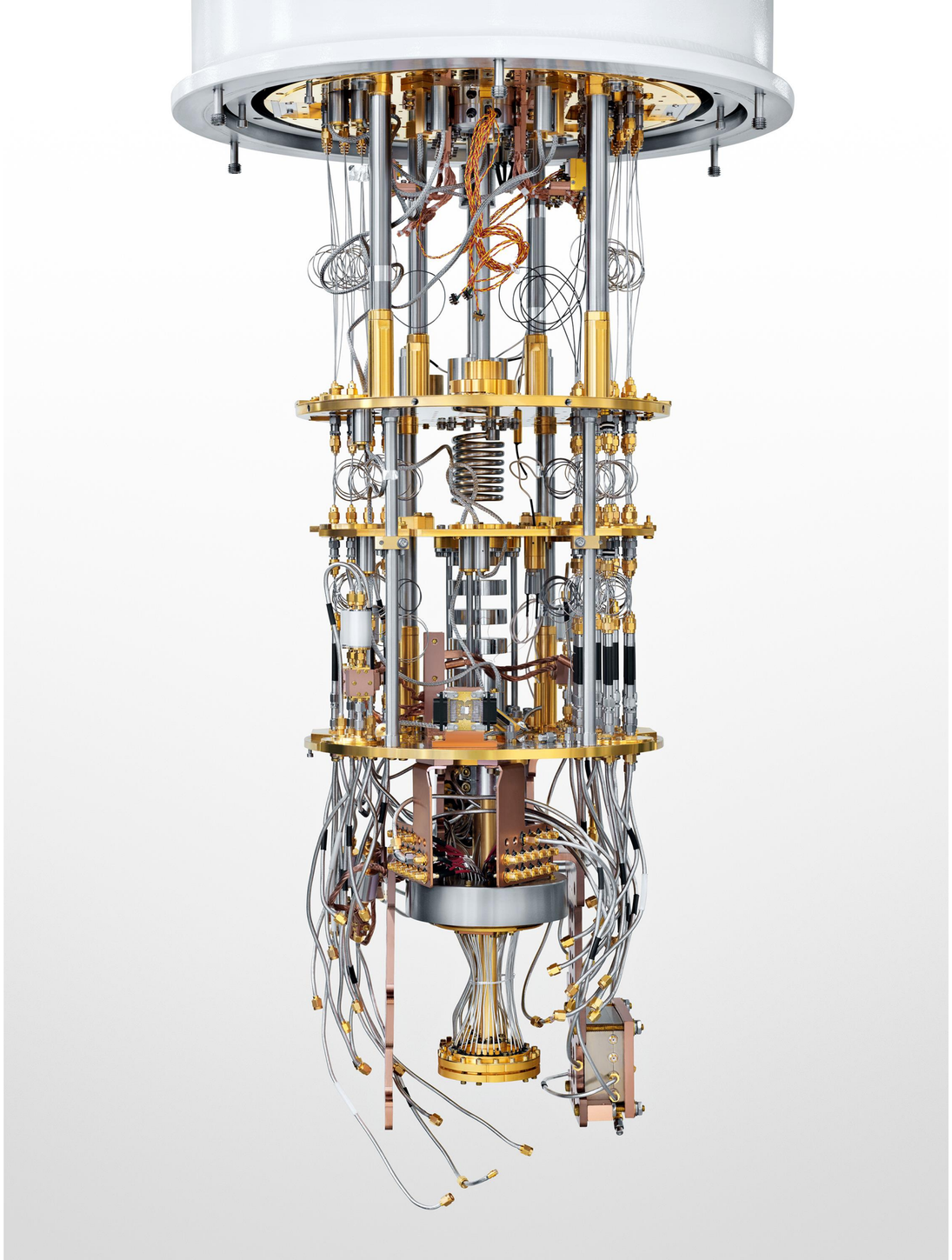
Digitized counterdiabatic circuit

Each time step applies one layer per term of the Hamiltonian.



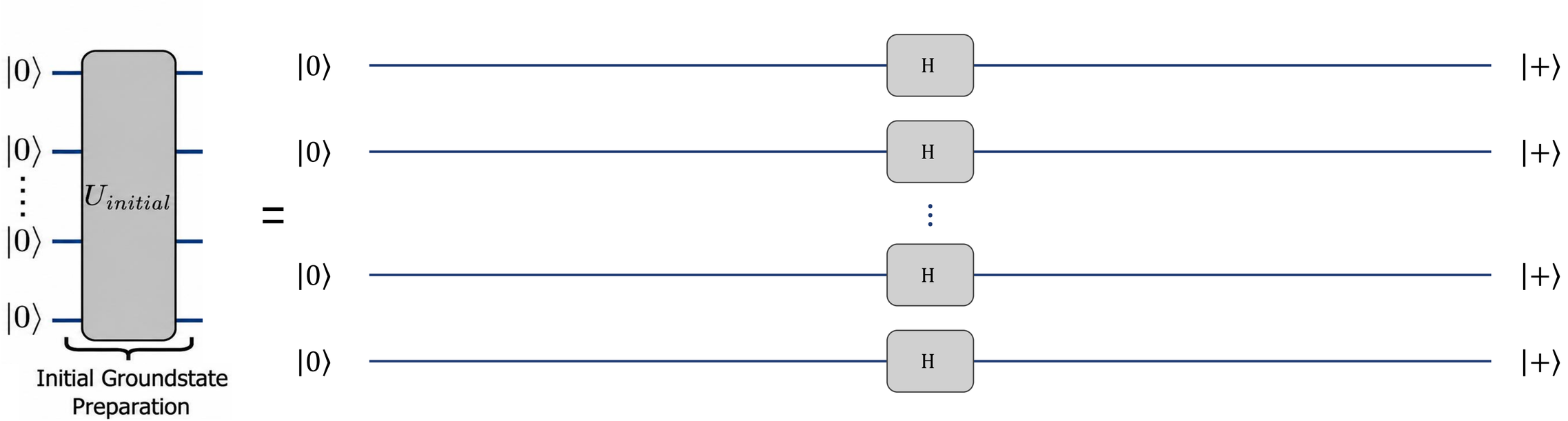
Hegade, et al. Phys. Rev. Applied 15, 024038 (2021)

DQA / QAOA



Initial state preparation

The first block of the circuit, applied once before the time steps.

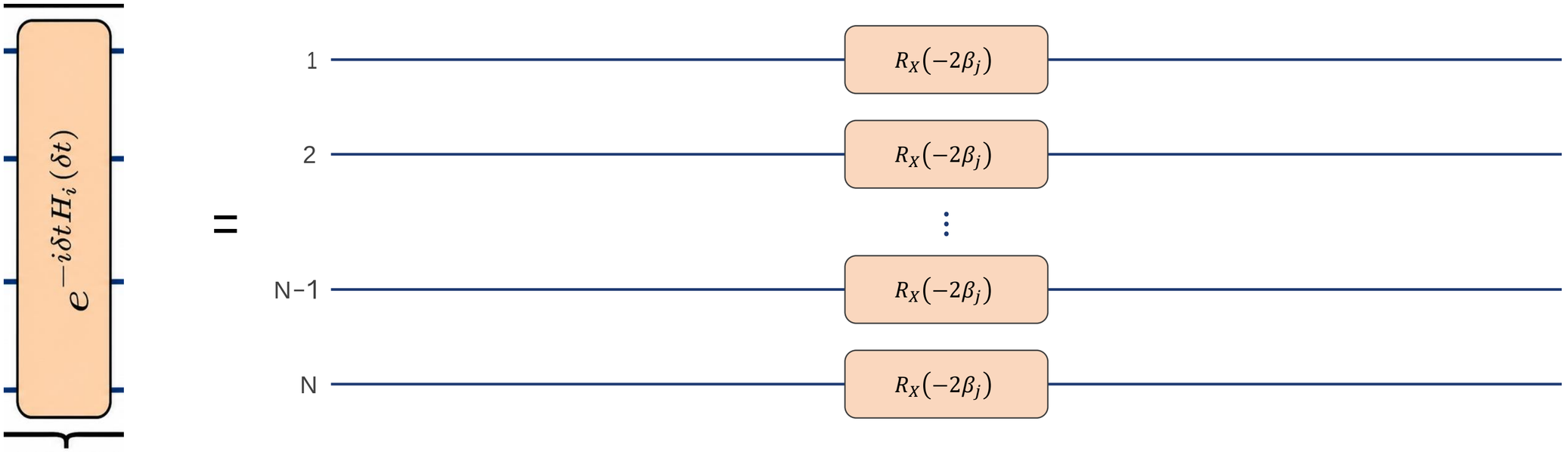


$$H_i = -\sum_i X_i \quad \Longrightarrow \quad |\psi(0)\rangle = |+\rangle^{\otimes N}$$

$$U_{\text{initial}} = \text{H}^{\otimes N}$$

Mixer layer

The initial Hamiltonian acts on each qubit separately: one X rotation per qubit.

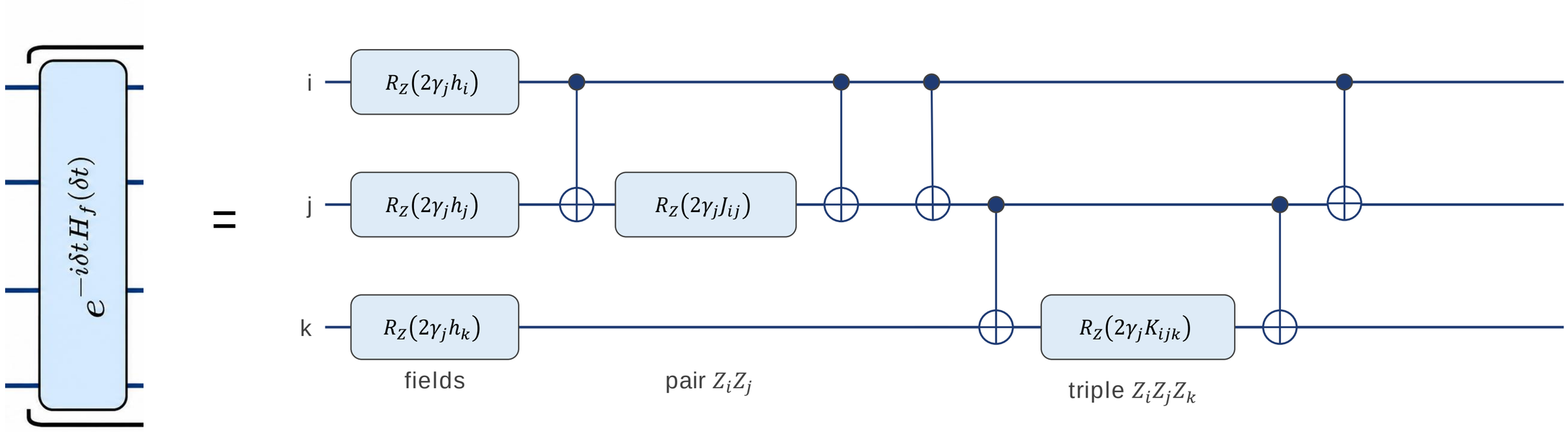


$$U_i(\beta_j) = e^{-i\beta_j H_i} = \prod_i R_X(-2\beta_j)$$

$$\beta_j = (1-\lambda_j)\Delta t/\hbar$$

Problem layer

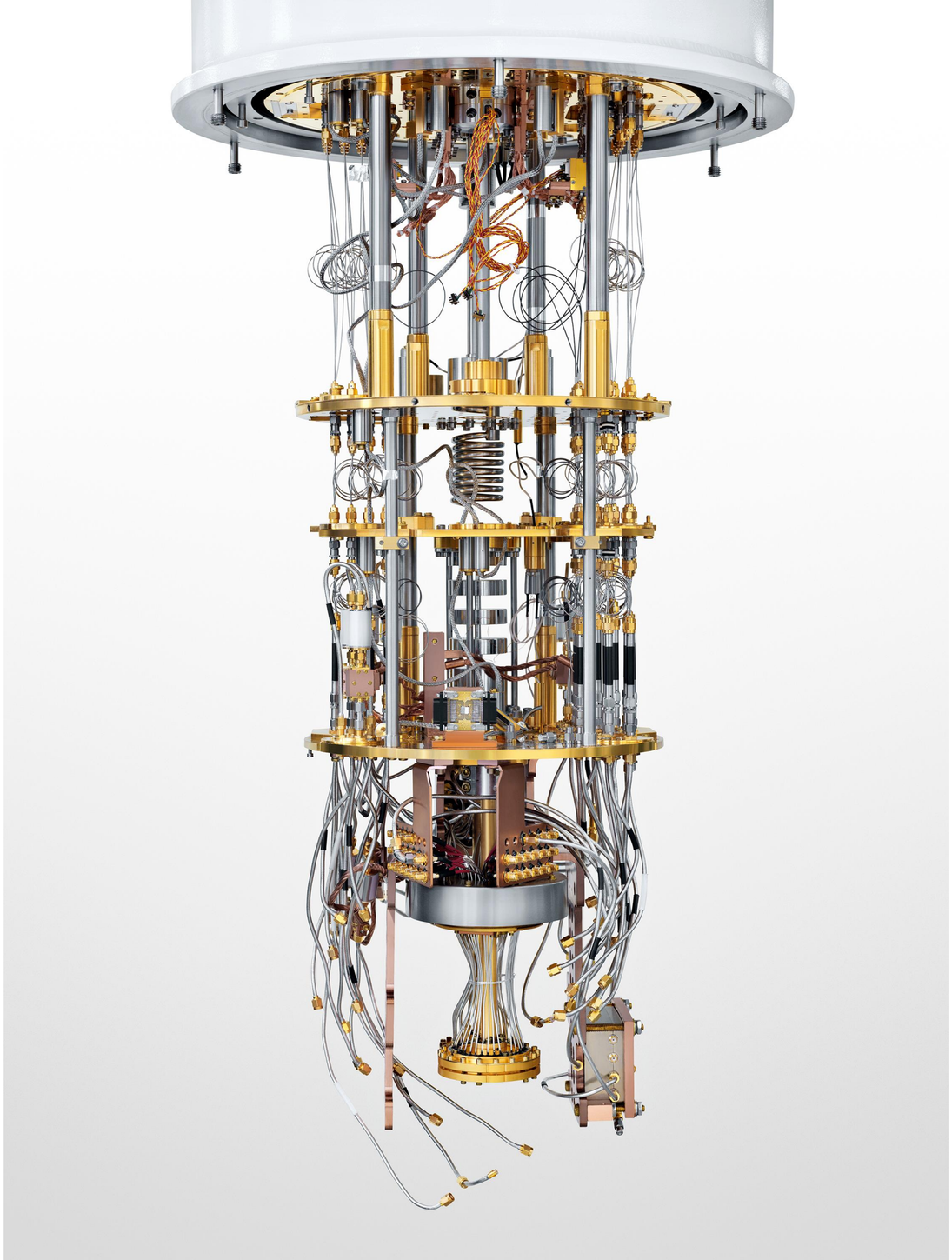
All terms of the problem Hamiltonian commute, so the block splits exactly into rotations.



$$U_f(\gamma_j) = e^{-i\gamma_j H_P} = \prod_i R_Z(2\gamma_j h_i) \prod_{i<j} R_{ZZ}(2\gamma_j J_{ij}) \prod_{i<j<k} R_{ZZZ}(2\gamma_j K_{ijk})$$

$$\gamma_j = \lambda_j \Delta t$$

DCQO



Nested commutator expansion

In the eigenbasis, each commutator multiplies by the energy gap

Exact (previous slides)

$$\langle m|A_\lambda|n\rangle = i\hbar \frac{\langle m|\partial_\lambda H_0|n\rangle}{E_n - E_m}$$

needs every eigenstate and gap along the path

Nested-commutator expansion

$$A_\lambda^{(\ell)} = i\hbar \sum_{k=1}^{\ell} \alpha_k \underbrace{[H_0, [H_0, \ldots [H_0, \partial_\lambda H_0]]]}_{2k-1 \text{ commutators}}$$

needs only H_0 and its derivative: no diagonalization

Matrix elements

$$\langle m|A_\lambda^{(\ell)}|n\rangle = i\hbar \left[\sum_{k=1}^{\ell} \alpha_k (E_m - E_n)^{2k-1}\right] \langle m|\partial_\lambda H_0|n\rangle$$

odd-polynomial fit to $-1/(E_m - E_n)$; $\ell \to \infty$ gives the exact A_λ

Sels & Polkovnikov, PNAS 114, E3909 (2017); Claeys et al., Phys. Rev. Lett. 123, 090602 (2019)

For $\ell = 1$

Keeping a single commutator gives few-body terms

Expansion (previous slide)

$$A_\lambda^{(\ell)} = i\hbar \sum_{k=1}^{\ell} \alpha_k \underbrace{[H_0, [H_0, \dots [H_0, \partial_\lambda H_0]]]}_{2k-1 \text{ commutators}}$$

Truncate at $\ell = 1$

$$A_\lambda^{(1)} = i\hbar\alpha_1 [H_0, \partial_\lambda H_0]$$

For the HUBO used throughout

$$H_P = \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j + \sum_{i<j<k} K_{ijk} Z_i Z_j Z_k$$

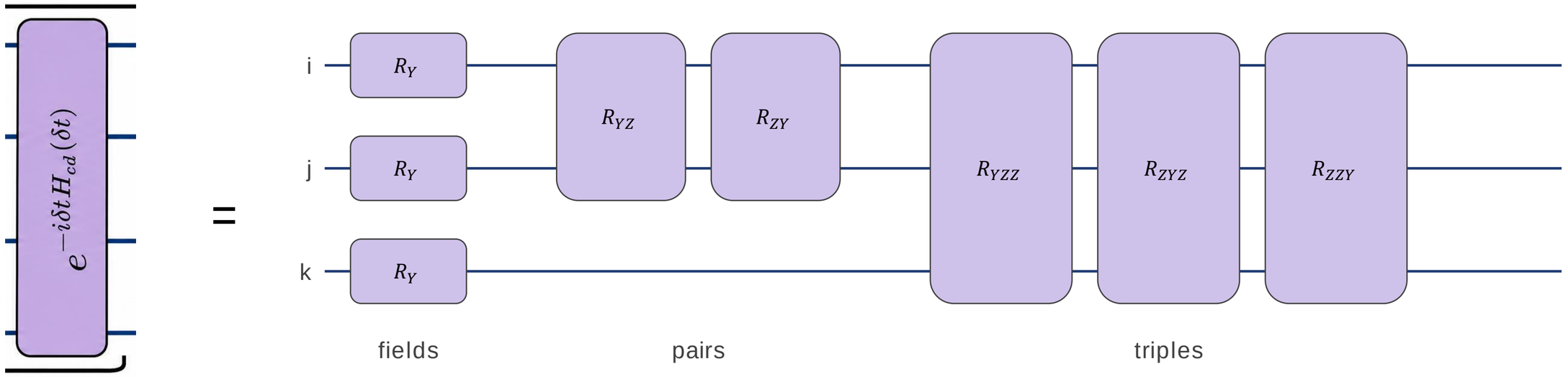
$$[H_0, \partial_\lambda H_0] = [H_i, H_P], \qquad [X_i, Z_i] = -2iY_i$$

$$A_\lambda^{(1)} = -2\hbar\alpha_1(\lambda) \left[\sum_i h_i Y_i + \sum_{i<j} J_{ij} \left(Y_i Z_j + Z_i Y_j \right) + \sum_{i<j<k} K_{ijk} \left(Y_i Z_j Z_k + Z_i Y_j Z_k + Z_i Z_j Y_k \right) \right]$$

Sels & Polkovnikov, PNAS 114, E3909 (2017); Claeys et al., Phys. Rev. Lett. 123, 090602 (2019)

Counterdiabatic layer

Each Pauli string of the gauge potential becomes one rotation.



$$H_{\mathrm{cd}} = \dot{\lambda} A_{\lambda}^{(1)} = -2\hbar\alpha_1\dot{\lambda}\left[\sum_i h_i Y_i + \sum_{i<j} J_{ij}\left(Y_i Z_j + Z_i Y_j\right) + \sum_{i<j<k} K_{ijk}\left(Y_i Z_j Z_k + Z_i Y_j Z_k + Z_i Z_j Y_k\right)\right]$$

$$U_{\mathrm{cd}}(\varphi_j) = \prod_i R_Y(-2\varphi_j h_i)\prod_{i<j} R_{YZ}(-2\varphi_j J_{ij})R_{ZY}(-2\varphi_j J_{ij})\prod_{i<j<k} R_{YZZ}(-2\varphi_j K_{ijk})R_{ZYZ}(-2\varphi_j K_{ijk})R_{ZZY}(-2\varphi_j K_{ijk})$$

$$\varphi_j = 2\alpha_1(\lambda_j)\dot{\lambda}_j \Delta t$$

Task: what does your DQA / QAOA circuit look like?

15 MINUTES · SAME PAIRS · AI-USAGE IS ENCOURAGED

Take the Ising model from last lecture and build one layer of its circuit

---

1 **Read off the terms**

Write H_P from your QUBO or HUBO: fields h_i, couplings J_{ij}, triples K_{ijk}, …

2 **Draw one layer**

One R_Z per field, R_{ZZ} per coupling, R_{ZZZ} per triple, then R_X on every qubit.

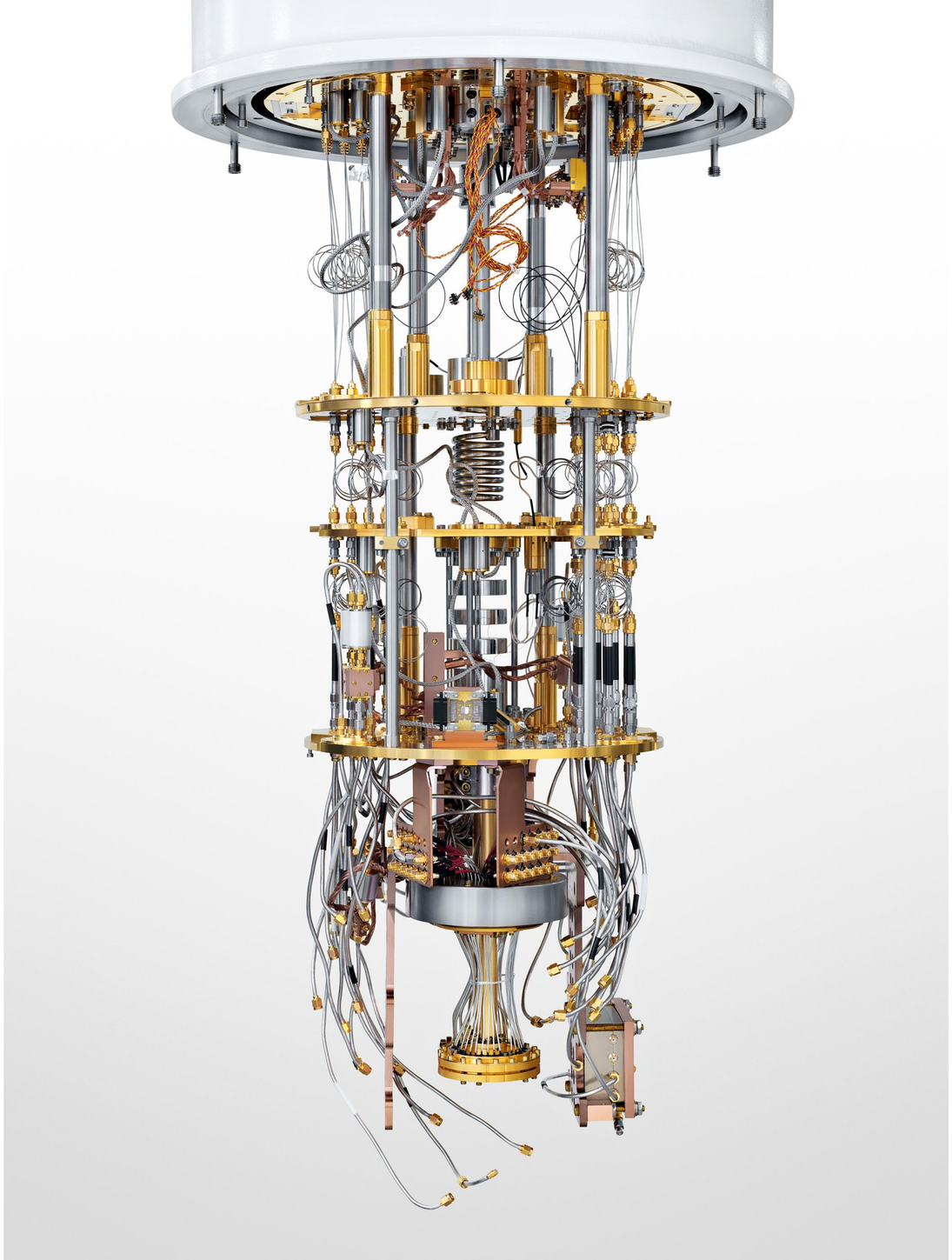
3 **Count the cost**

Qubits, two-qubit gates per layer and depth for p layers.

15:00

Try: “Write one QAOA layer for my Ising Hamiltonian and count its two qubit gates.”

End of Part II



Quantum Optimization Algorithms – Part III

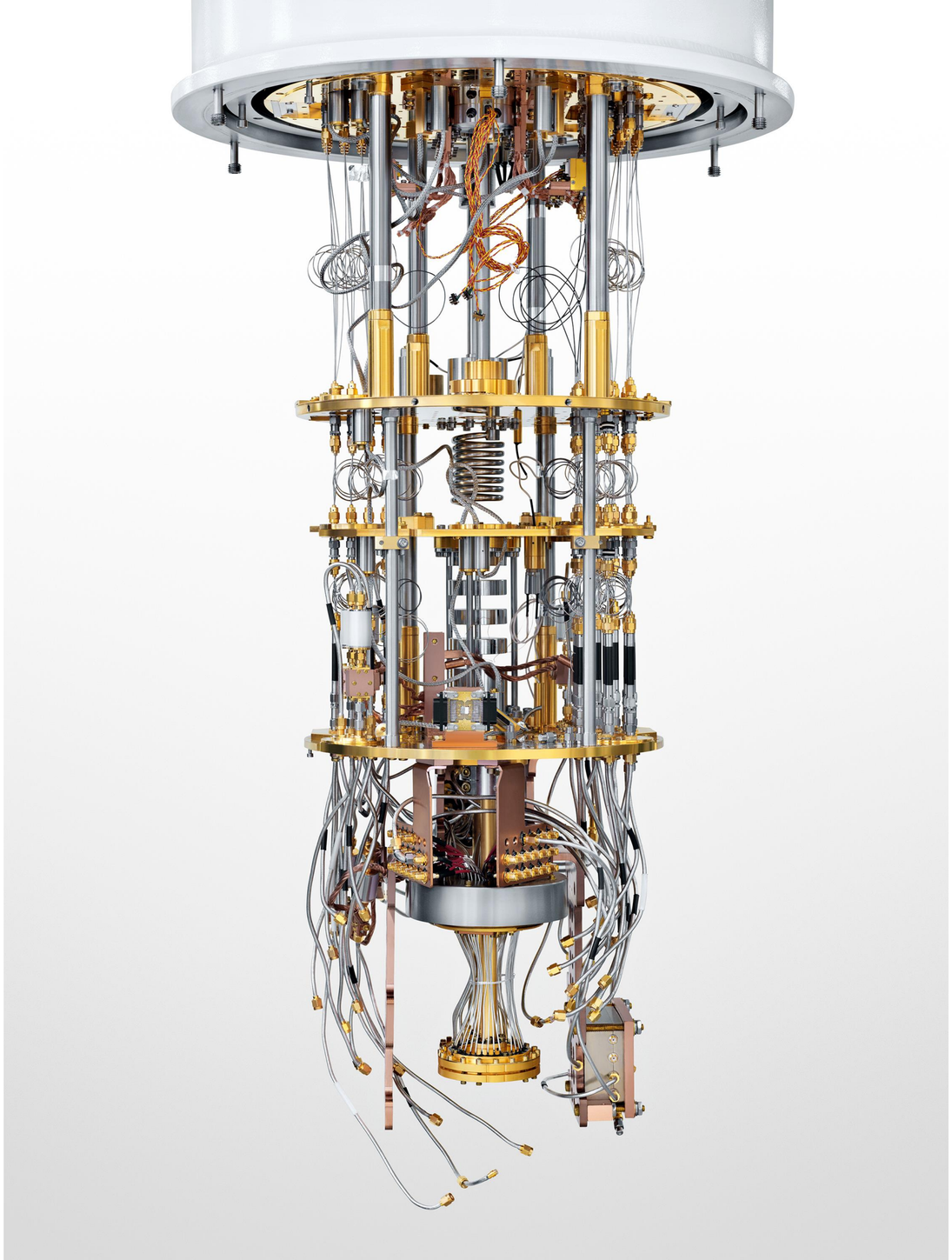
Alejandro Gómez Cadavid

Kipu Quantum GmbH

3rd Quantum Computing School ICTP-SAIFR

São Paulo, 2026

Impulse Regime



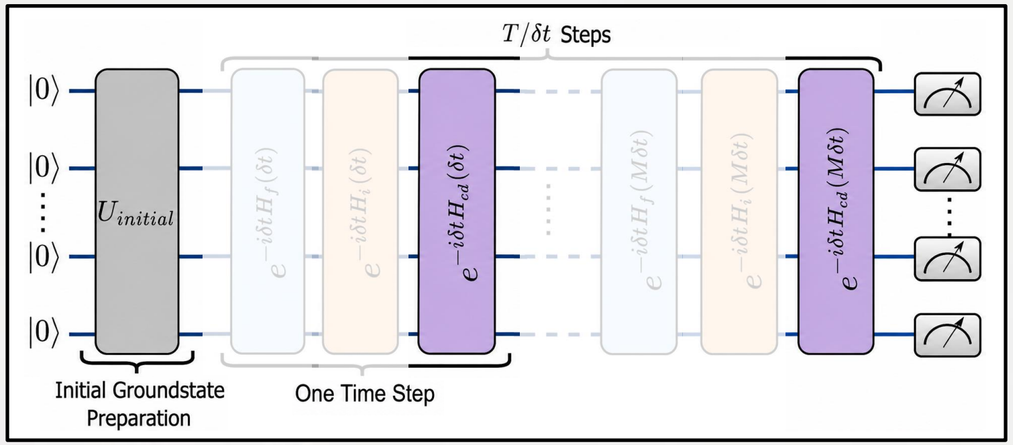
DCQO in the impulse regime

Enhancing quantum optimization at small evolution times

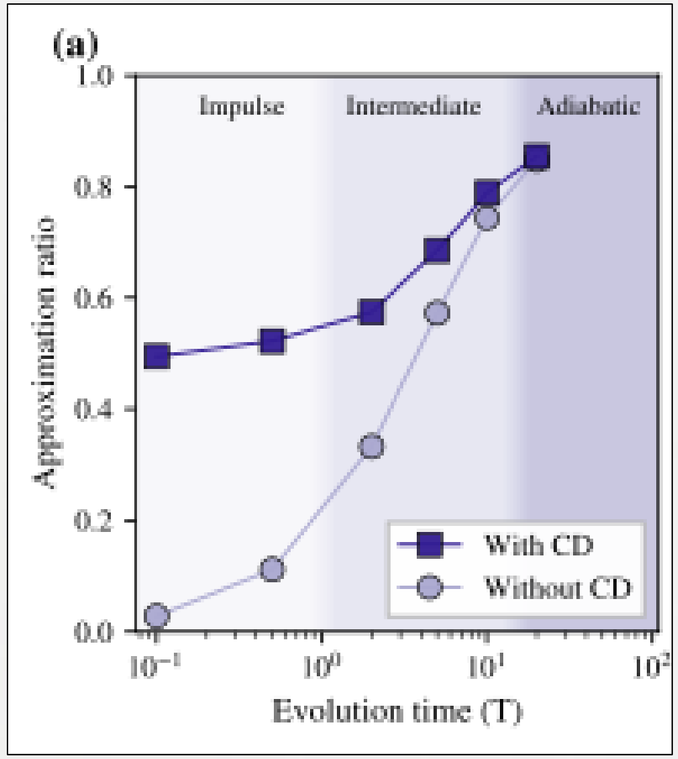
$$\langle n|\boldsymbol{A}_{\boldsymbol{\lambda}}|m\rangle = i\hbar \frac{\langle n|\boldsymbol{\nabla}_{\boldsymbol{\lambda}} H_0|m\rangle}{E_m(\boldsymbol{\lambda}) - E_n(\boldsymbol{\lambda})}$$

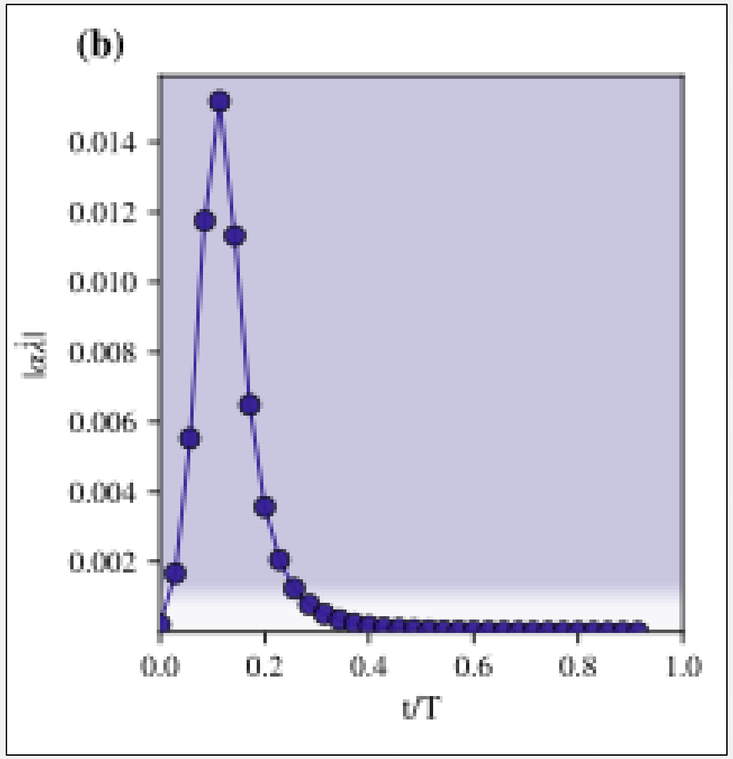
$$H(\boldsymbol{\lambda}) = H_0(\boldsymbol{\lambda}) + \underbrace{\dot{\boldsymbol{\lambda}}(t) \cdot \boldsymbol{A}_{\boldsymbol{\lambda}}^{(\ell)}}_{\dot{\boldsymbol{\lambda}} \propto \frac{1}{T}} \qquad H_0(t) = \boldsymbol{\lambda}\left(\frac{t}{T}\right) \cdot \boldsymbol{\mathcal{H}}$$

Simple impulse regime condition $\left|\alpha\dot{\boldsymbol{\lambda}}\right| \gg |\boldsymbol{\lambda}|$



$$H(\boldsymbol{\lambda}) \approx \dot{\boldsymbol{\lambda}}(t) \cdot \boldsymbol{A}_{\boldsymbol{\lambda}}^{(\ell)}$$

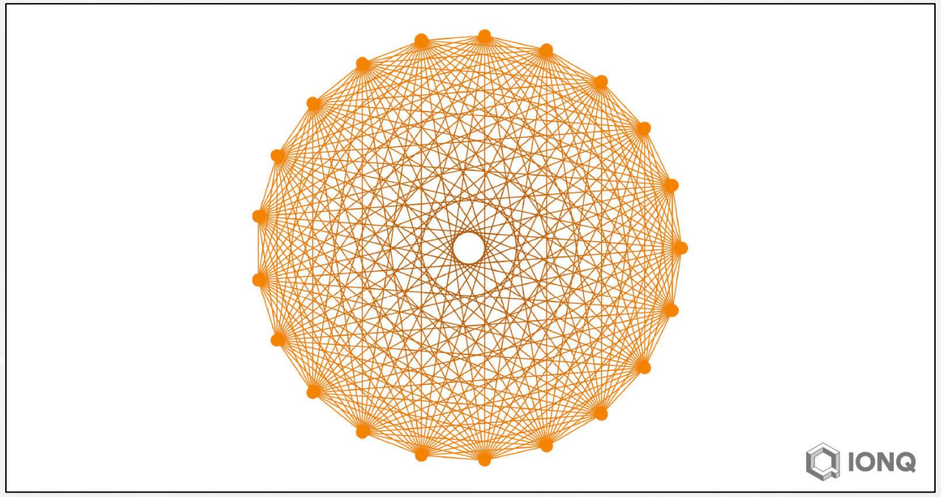




DCQO in the impulse regime

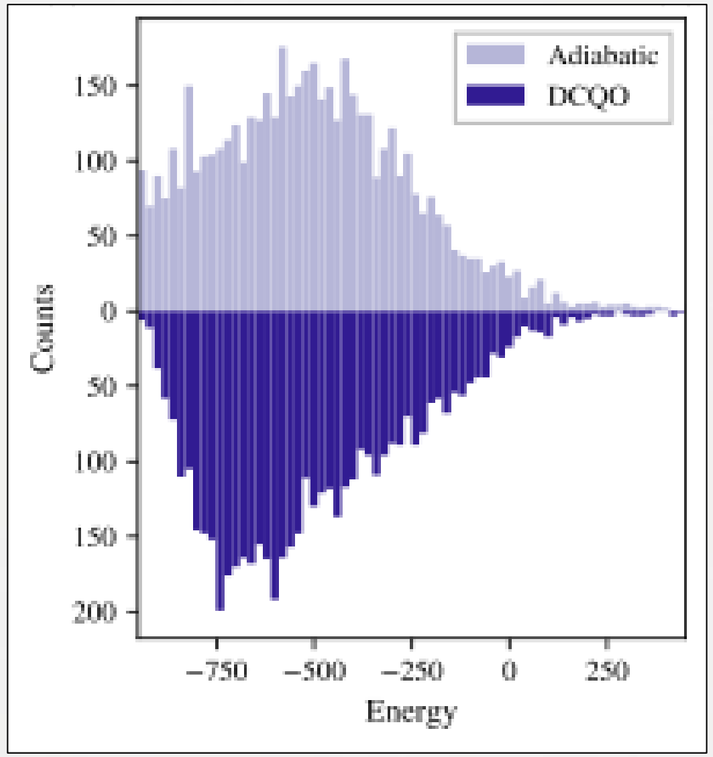
Experimental validation on trapped ion devices

Device : IonQ Aria

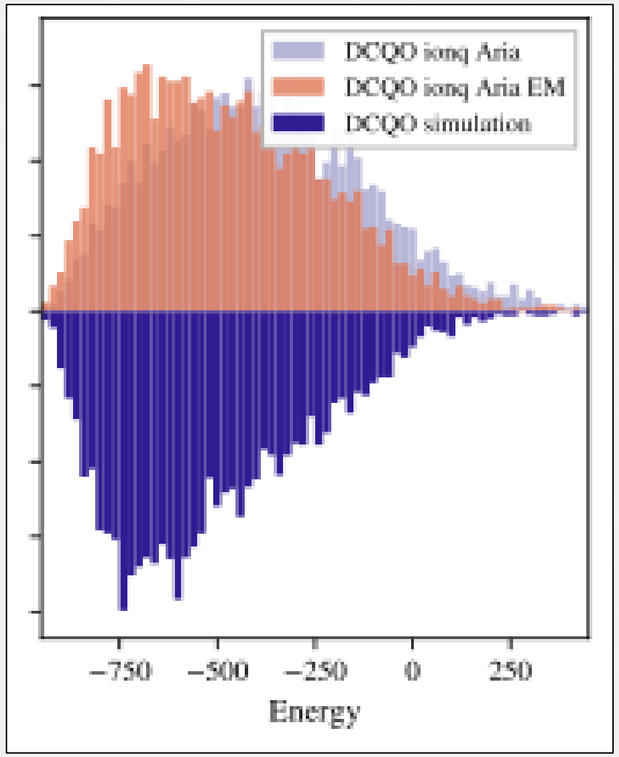


- DQA (80 steps) → $AR = 0.54$ (simulation)
- DCQO (1 effective step) → $AR = 0.53$ (simulation)
  $AR = 0.50$ (experiment)

Simulation

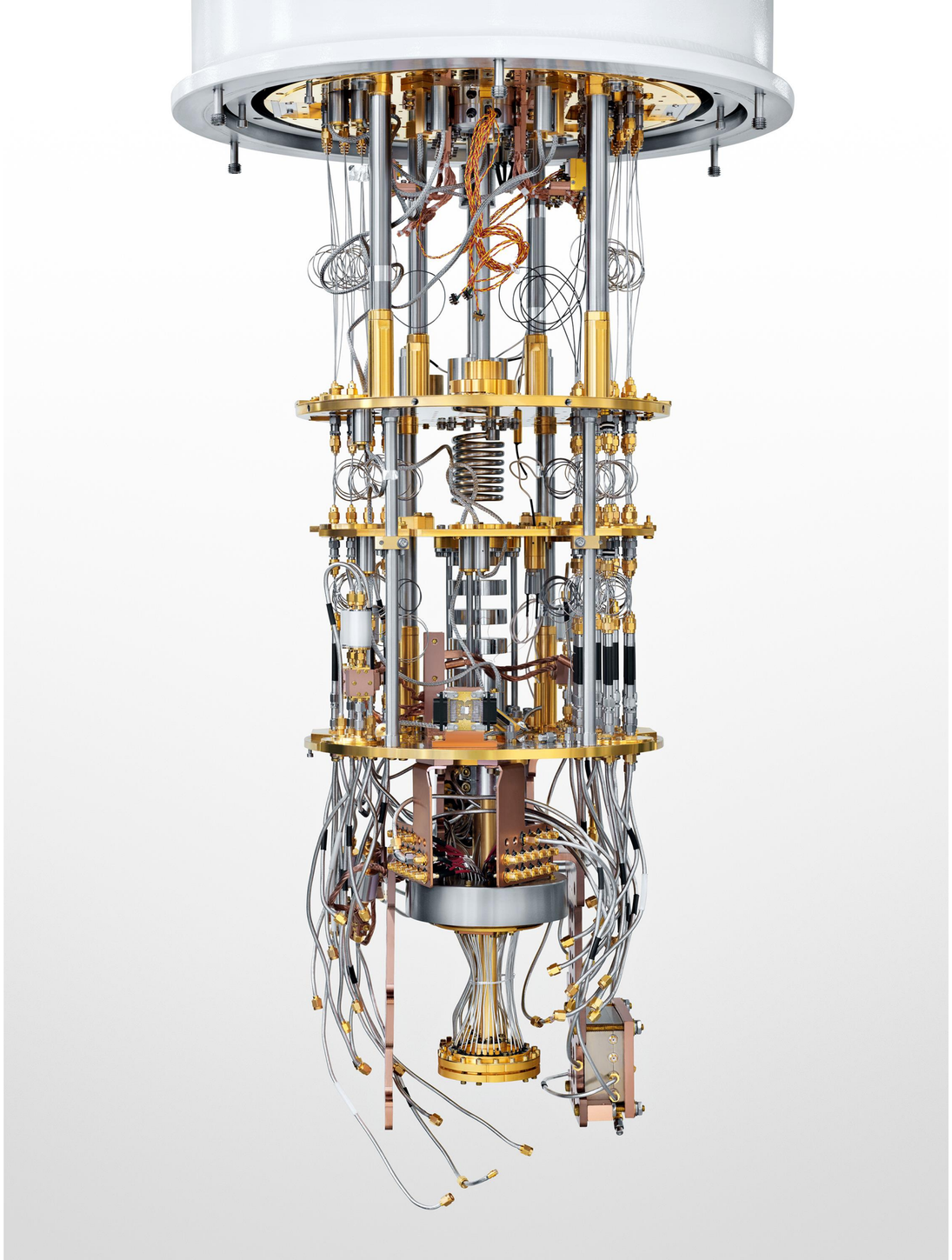


Experiment vs Simulation



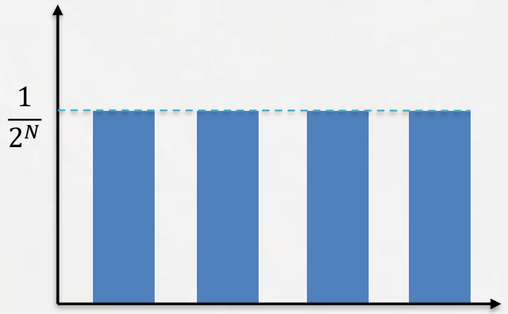
Cadavid, et al. Phys. Rev. Applied 22, 054037 (2024)

Bias-Field DCQO

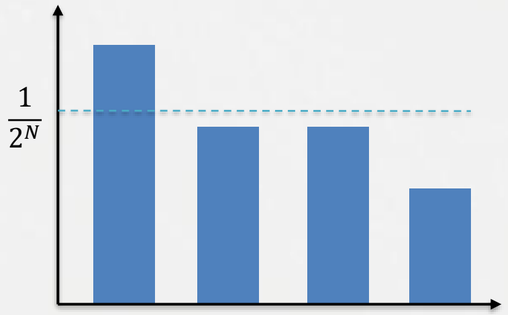


Biasing the initial state

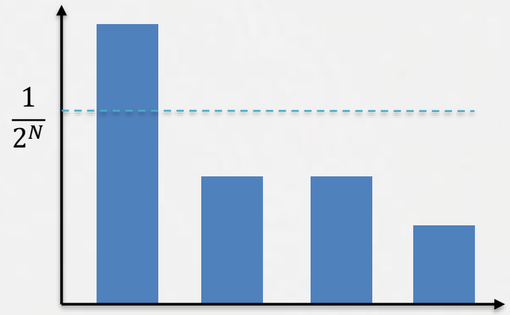
Warm starting via longitudinal bias fields



$$H_I = -\sum_k \sigma_k^x$$



$$H_I = \sum_k \left(-\sigma_k^x + h_k^b \sigma_k^z\right)$$

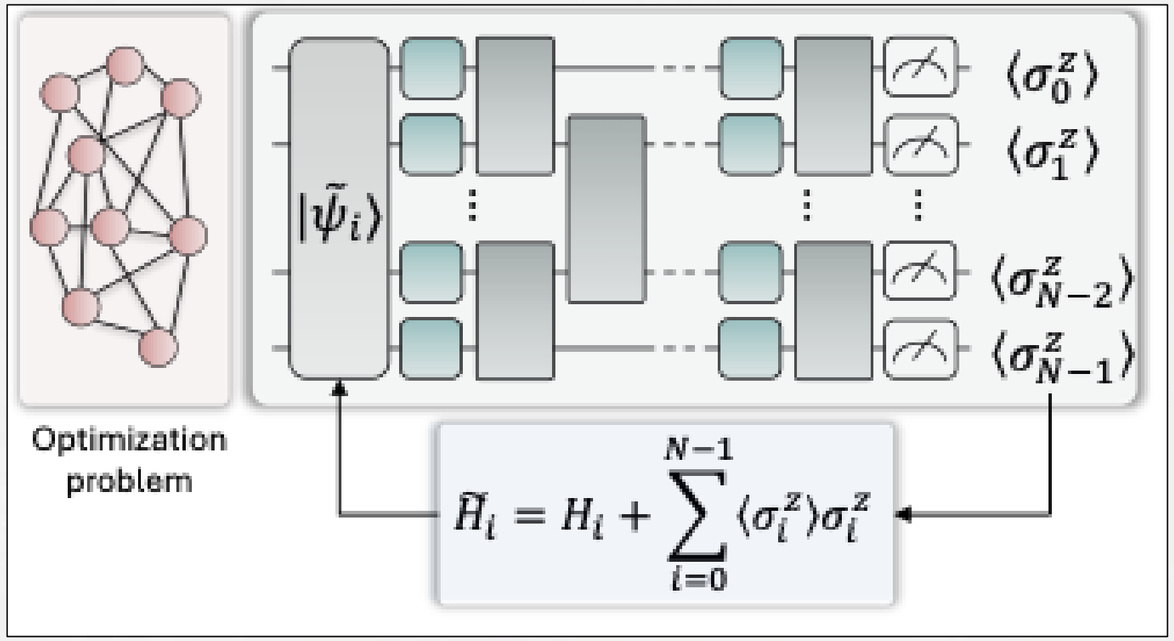


$$H_I = \sum_k \left(-\sigma_k^x + h_k^b \sigma_k^z\right)$$

Biasing the initial Hamiltonian modifies the initial state, from which the quantum search starts.

Bias-Field DCQO

Using DCQO to obtain better bias fields



Cadavid, et al. Phys. Rev. Research 7, L022010 (2025)

Bias-Field DCQO

Scaling enhancement with respect to DCQO and DQA

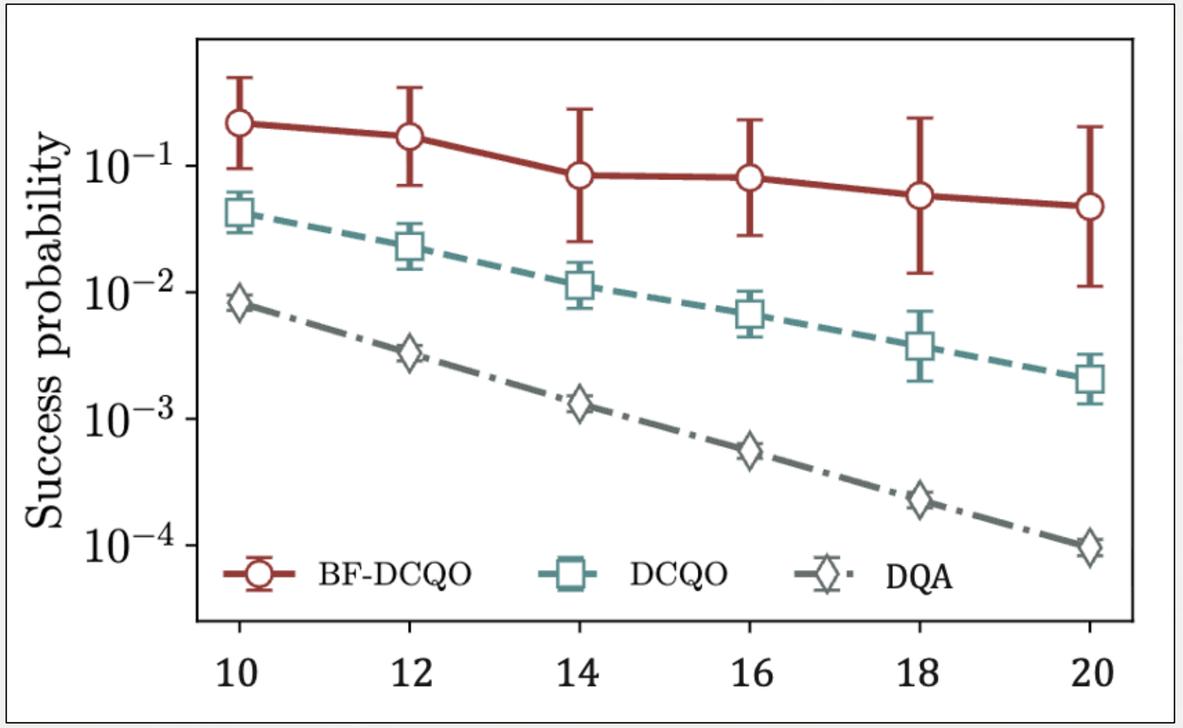
The benchmark:

- 400 randomly generated instances $\mathcal{N}(0,1)$
- 10 iterations of BF-DCQO

Assuming the exponential decay model for the success probability

$$p_{gs} = c_0 \cdot 10^{-rN}$$

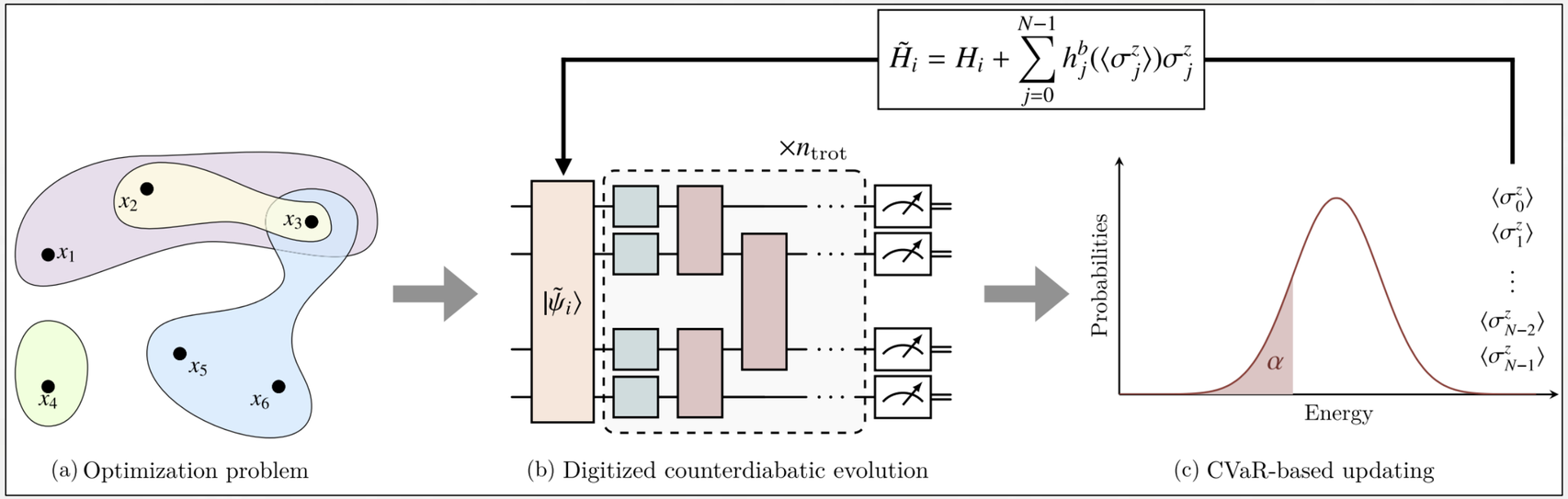
$r_{BF-DCQO} = 0.084$ $\quad r_{DCQO} = 0.128$ $\quad r_{DQA} = 0.194$



Cadavid, et al. Phys. Rev. Research 7, L022010 (2025)

Bias-Field DCQO

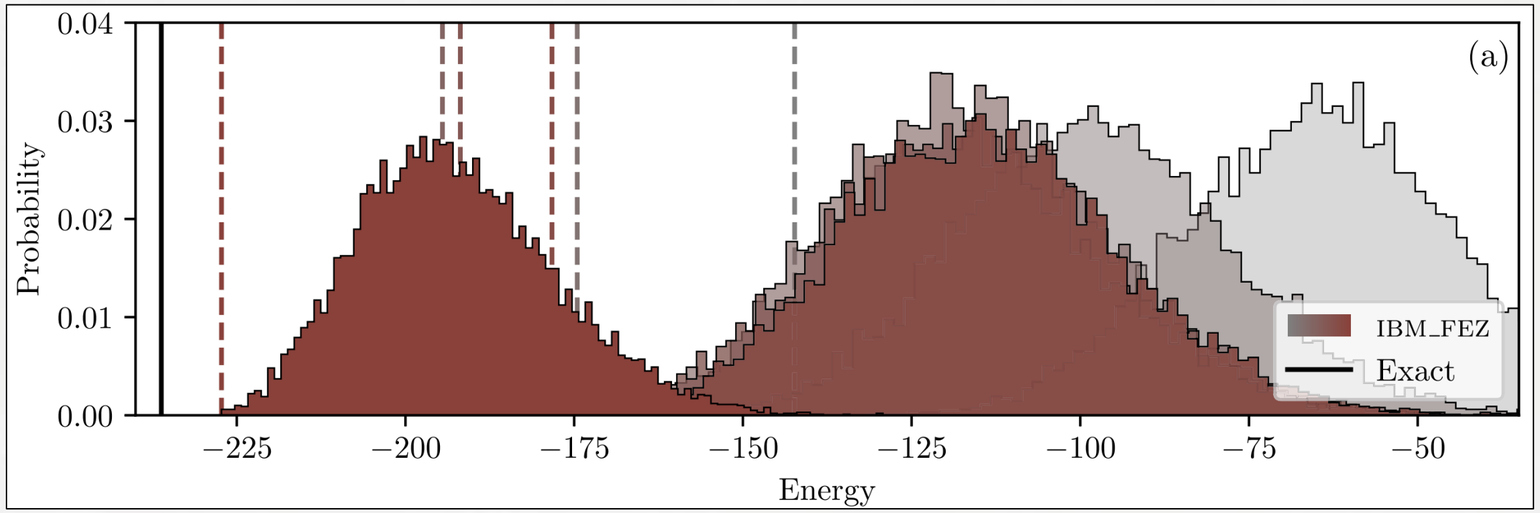
CVaR update



Romero, et. al Communications Physics, 8(1), 348 (2025)

Bias-Field DCQO

Experimental validation on superconducting devices

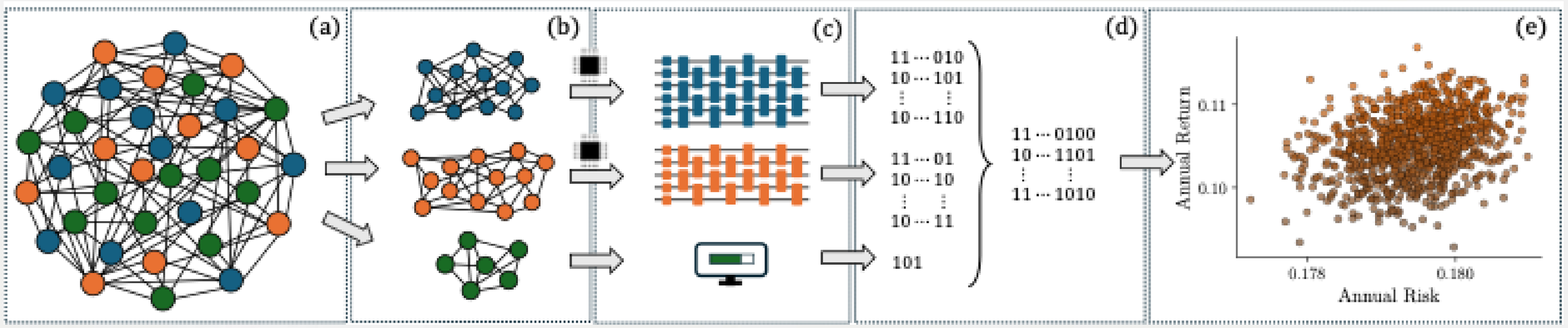


IBM **Quantum**

Romero, et. al Communications Physics, 8(1), 348 (2025)

Decomposition techniques

What can we do when the problem size exceeds the available qubit count?



We select 250 assets from S&P 500

- Original problem : 250 assets, budget = 125 assets

- Available hardware : 36, 64 qubits

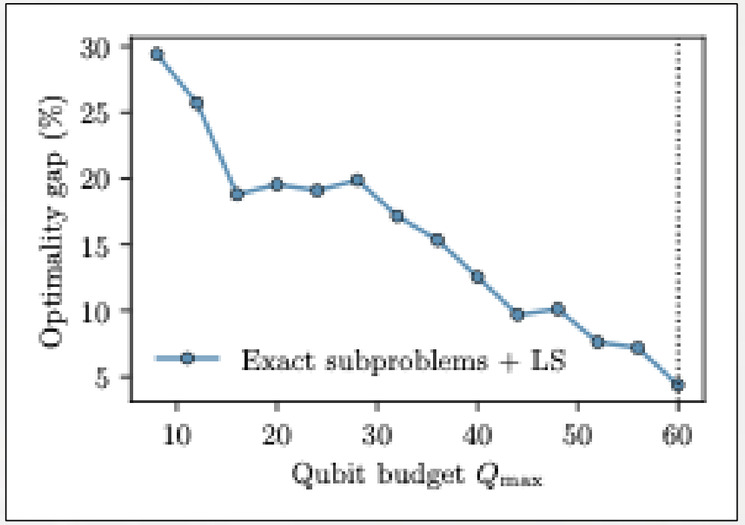
Hardware	# Clusters	Cluster sizes
36 qubits	14	[2,8,**36**,5,10,13,10,1,26,24,**36**,19,**36**,24]
64 qubits	11	[2,8,**41**,10,13,10,1,26,24,**55**,**60**]

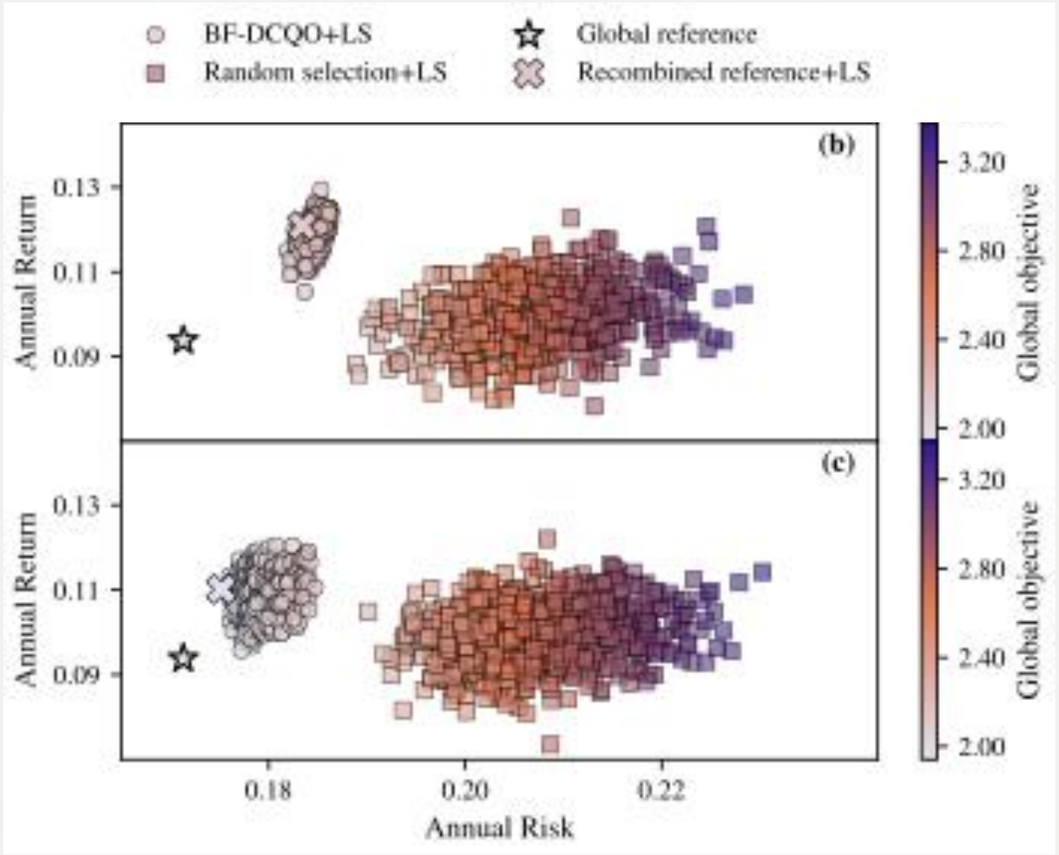
Cadavid, et al. arXiv:2602.23976 (2026)

Portfolio distributions

Experimental results on trapped-ion devices

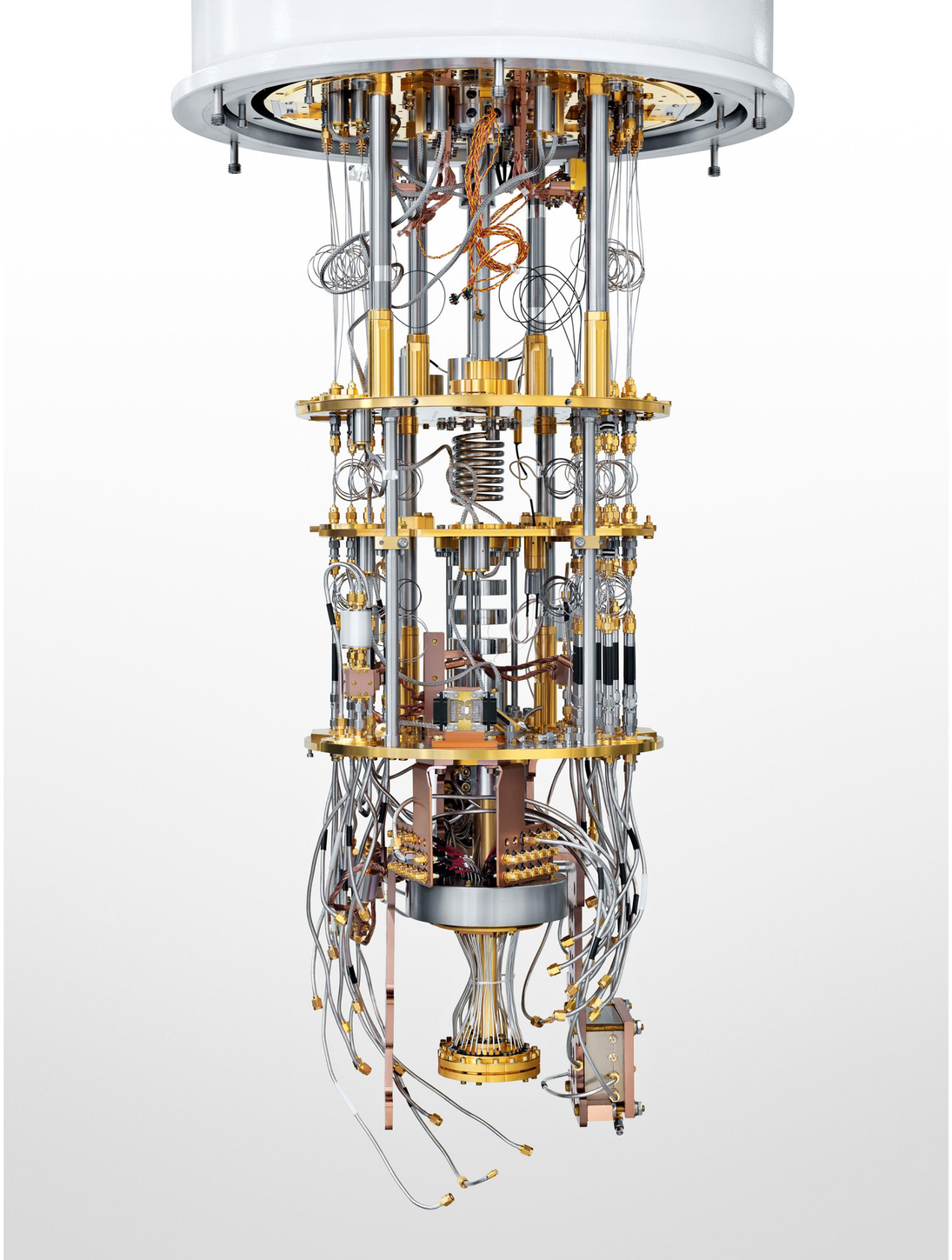
- The portfolios coming from the quantum pipeline are better located against random.
- Increasing the qubit budget improves the performance.



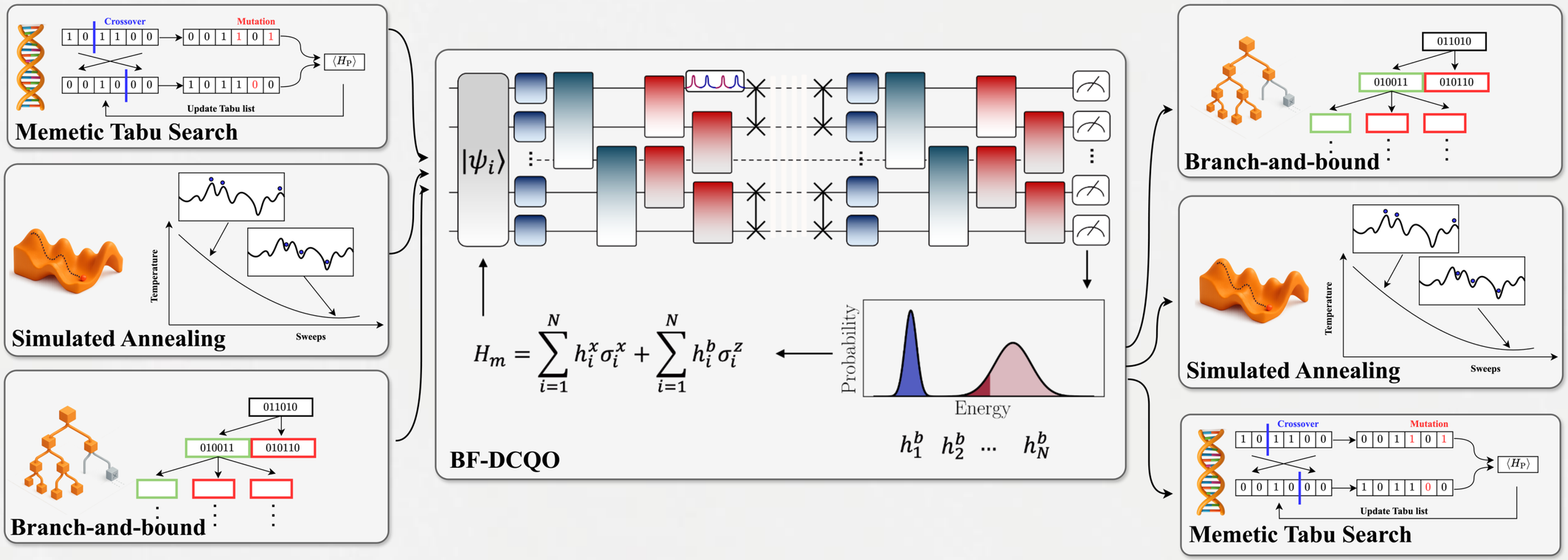


Cadavid, et al. arXiv:2602.23976 (2026)

Hybrid sequential Quantum optimization



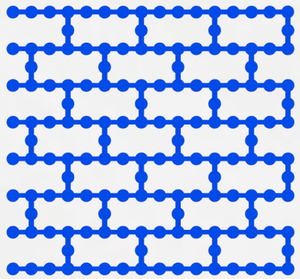
Classical – Quantum – Classical



Chandarana, et al. arXiv:2603.13607 (2026)

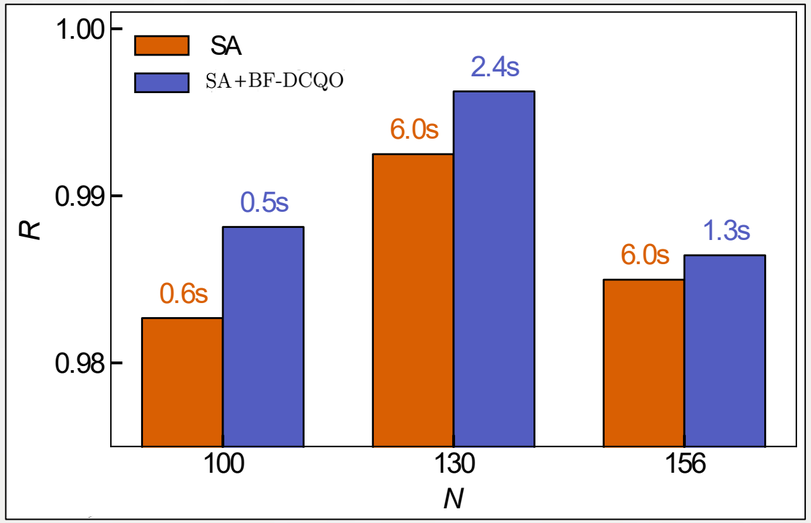
Classical – Quantum

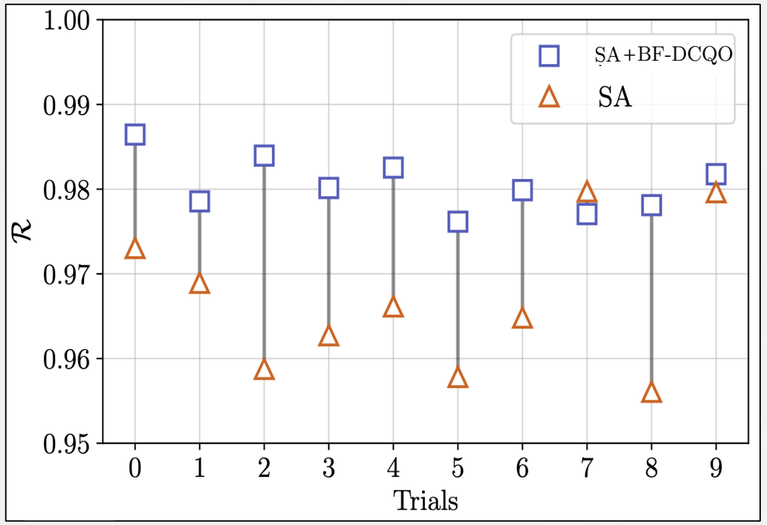
1 QPU



1 CPU







Utility scale quantum optimization

- Higher order binary optimization with up to 156 qubits.
- 3-body terms: 400+

Heavy-tailed distributions

- Coefficients chosen from heavy tailed distributions

Benchmarking metric

- Runtime to achieve an approximate solution.
- Classical sampling rate in GHz.
- Quantum sampling rate in kHz.

Classical baseline

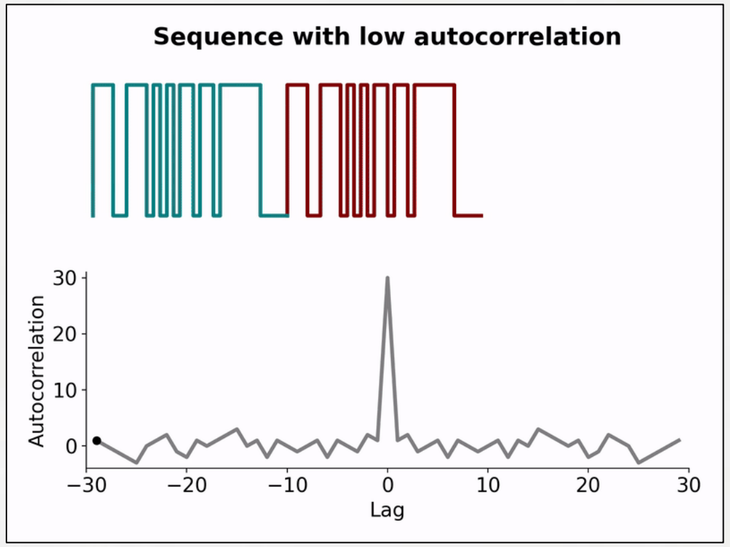
- Simulated Annealing.

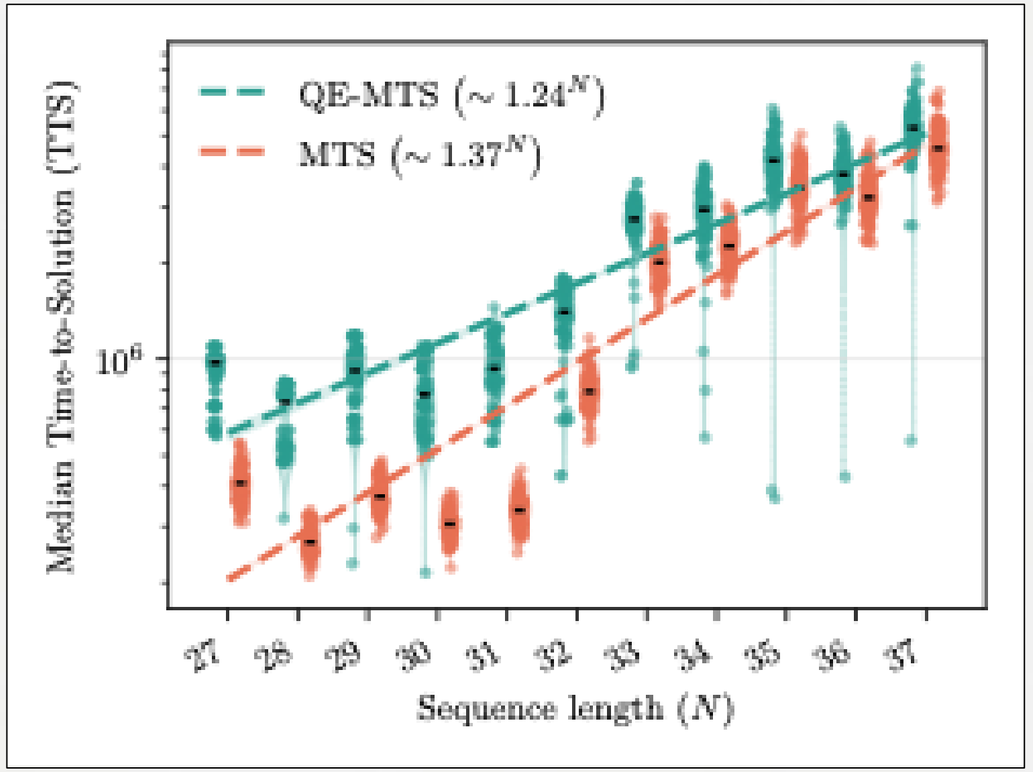
Chandarana, et al. arXiv:2603.13607 (2026)

Quantum–Classical

Scaling advantage for the low autocorrelation binary sequences (LABS) problem

$$E(s) = \sum_{k=1}^{N-1} C_k^2 \qquad C_k = \sum_{i=1}^{N-k} s_i s_{i+k}$$



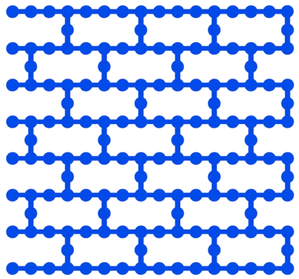


Cadavid, Alejandro Gomez, et al. Quantum Mach. Intell. 8, 92 (2026)

Classical–Quantum–Classical

Single QPU and multiple CPUs / GPUs

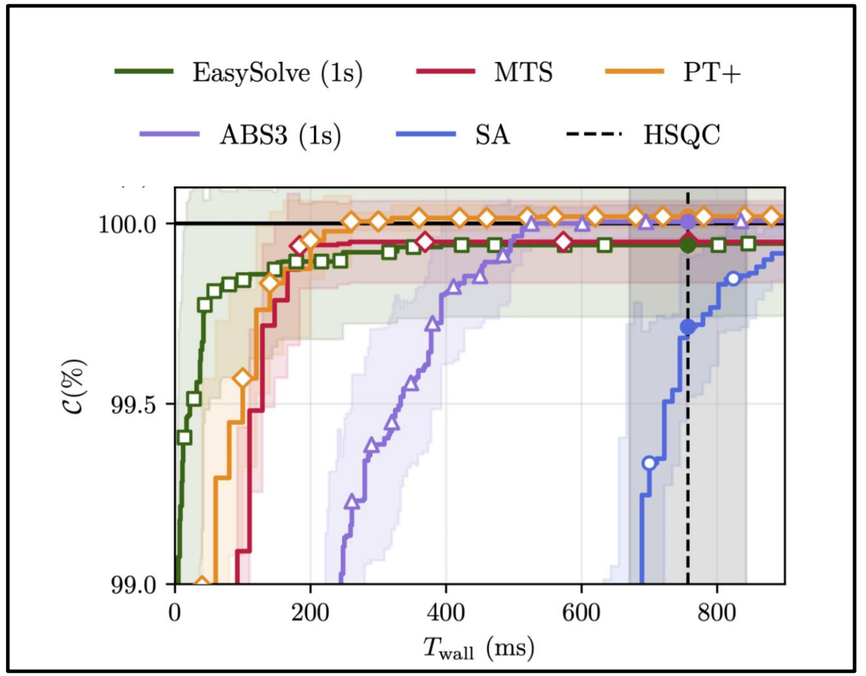
1 QPU



Multiple CPUs / GPUs







Utility scale quantum optimization

- Higher order binary optimization with up to 156 qubits.
- 3-body terms: 1000+

Heavy-tailed distributions

- Coefficients chosen from heavy tailed distributions

Chandarana, et al. arXiv:2603.13607 (2026)

The end

