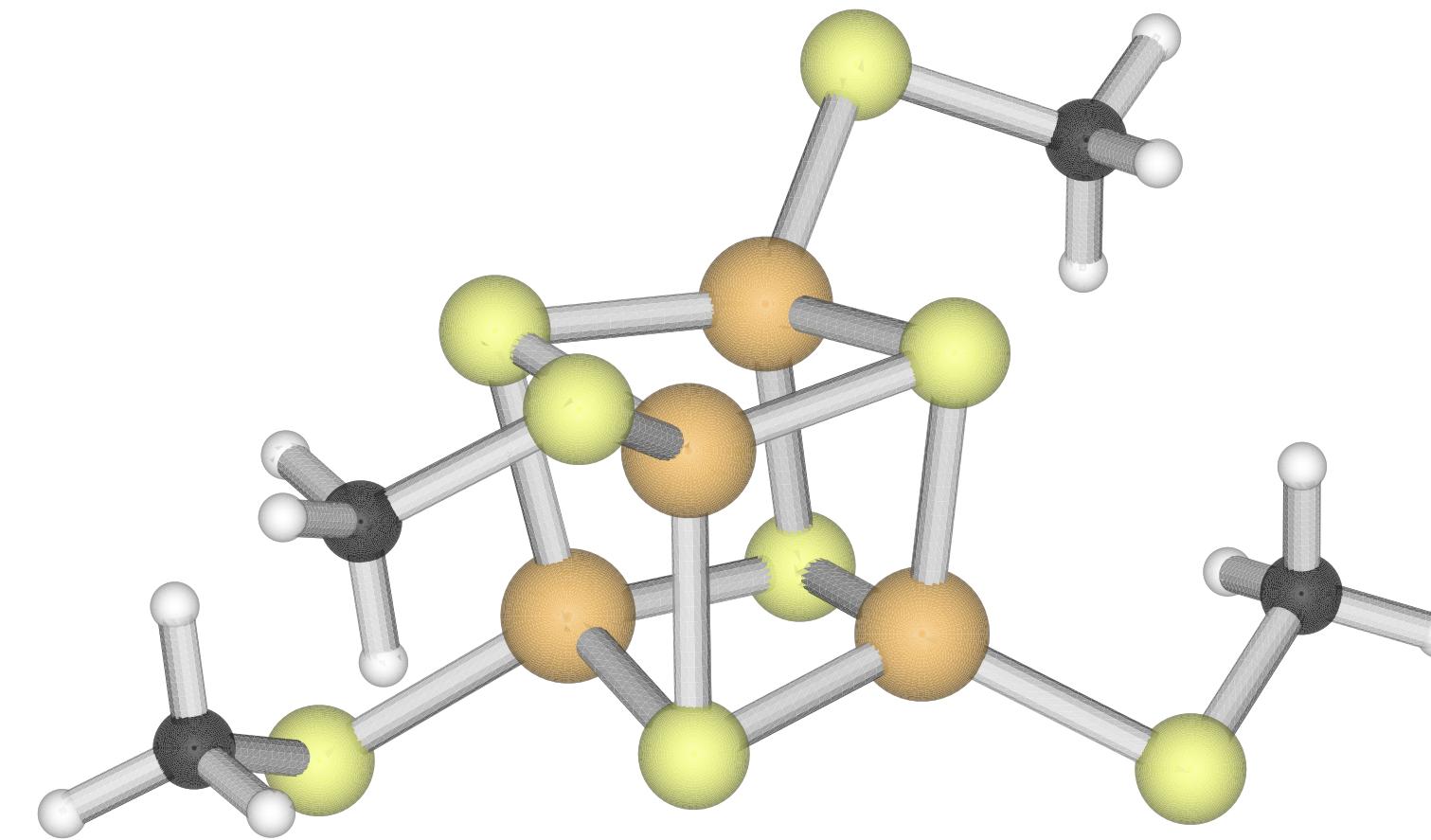


Chemistry Beyond Exact Solutions of the Schrödinger Equation on a Quantum-centric Supercomputer

Robledo-Moreno, Motta, Haas, Javadi, Jurcevic, Kirby, K.
Sharma, S. Sharma, Shirakawa, Sitdikov, Sun, Sung, Takita, Tran,
Yunoki, Mezzacapo

The current belief is that
pre-fault-tolerant quantum computers
in isolation cannot deal with realistic
formulations of nature

The cost for the accurate prediction of nature



Fe₄S₄ on **72** qubits (TZP-DKH basis set): **6.7M** Pauli operators

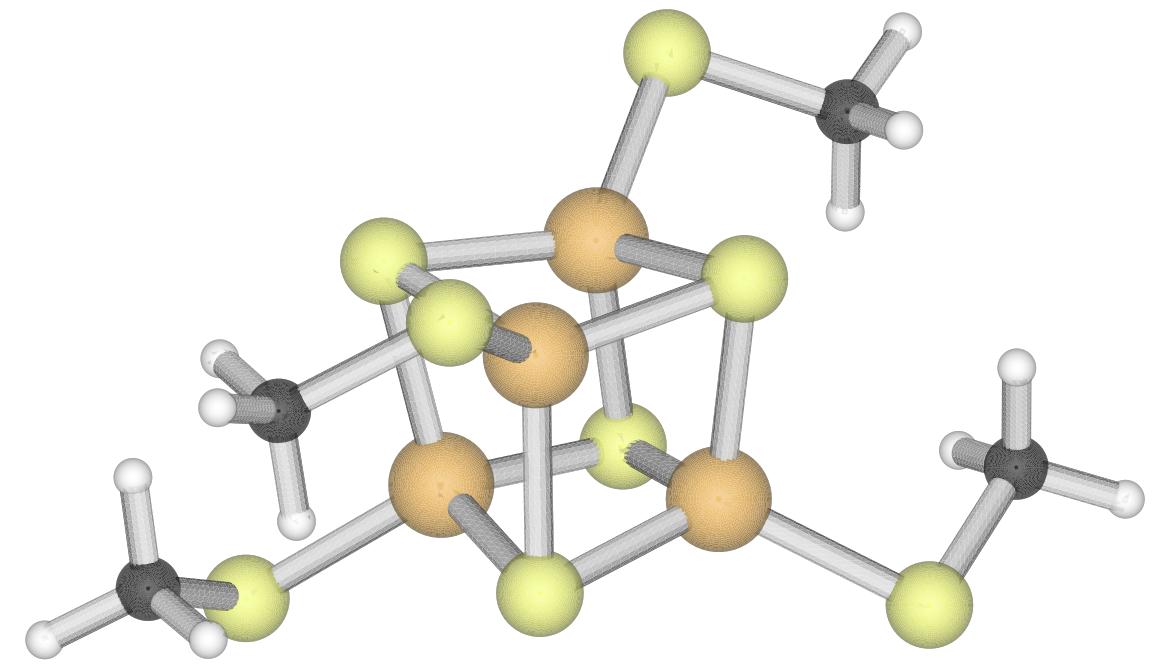
10⁻⁷ precision on each operator for milliHartree precision

Each circuit must be executed **10¹⁴** times

Runtime at 10μs/circuit ~**30 years**

Classical distributed computing enables quantum
to address complex use cases with a few large
quantum circuits

Quantum-centric supercomputing: a new computational framework



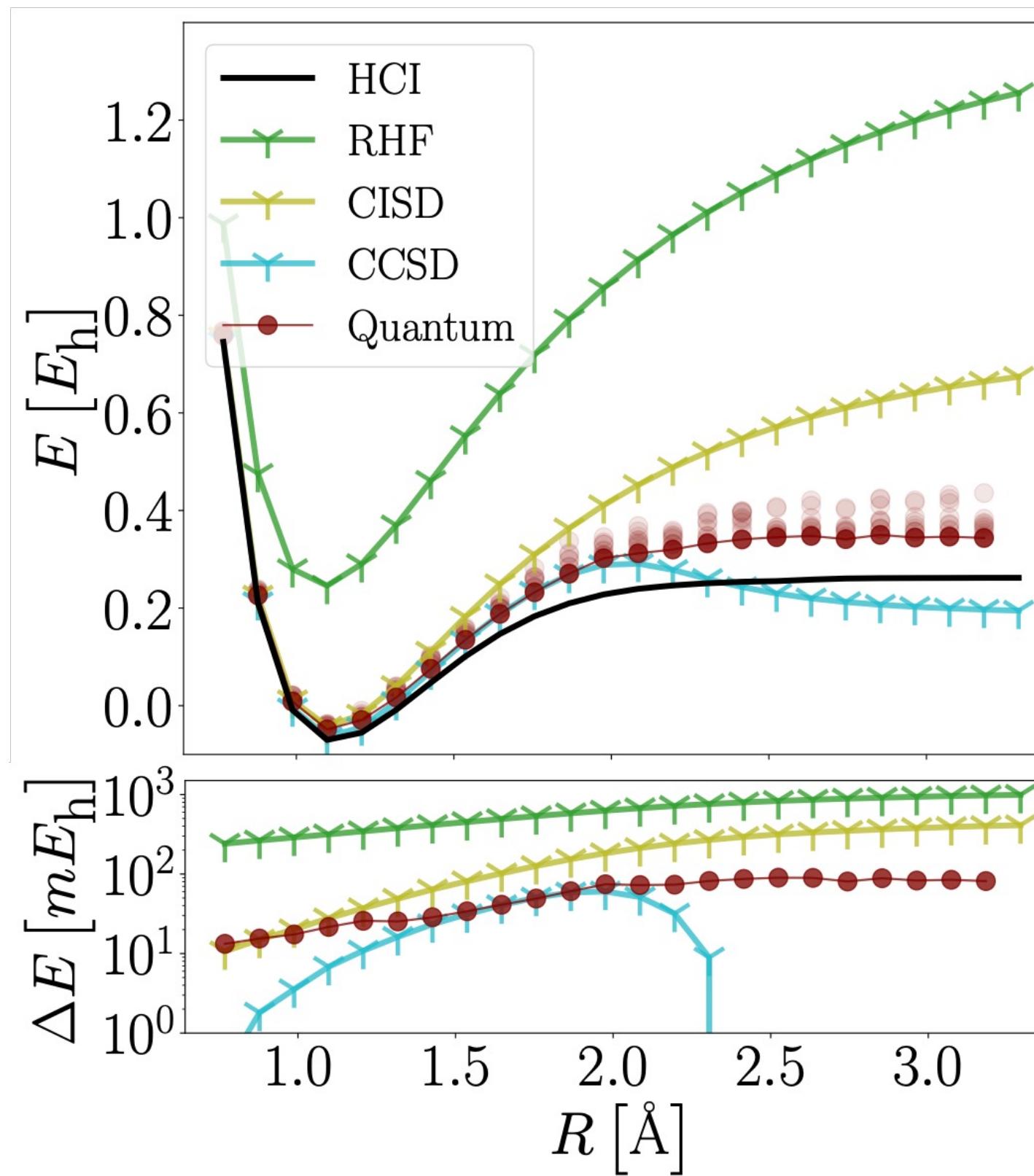
Fe₄S₄ on **72** qubits (TZP-DKH basis set): **6.7M** Pauli operators

Fault-tolerant	Phase estimation qubits: 4.53M 13 days runtime*
Pre-fault tolerant	VQE estimation at 10μs/circuit ~30 years
Quantum-centric supercomputing	Subspace estimation at 10μs/circuit ~2 hours

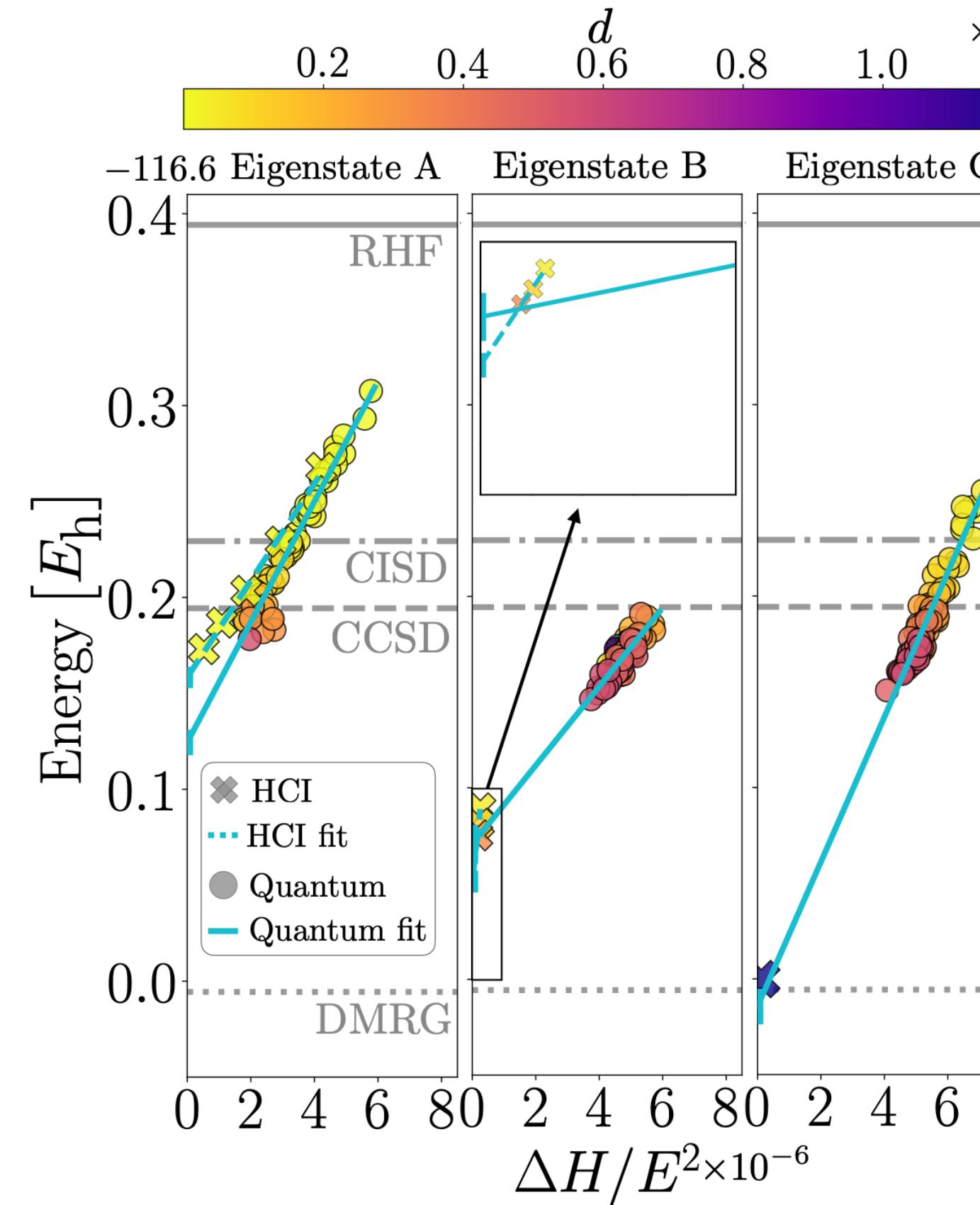
QCSC enables complex use cases such as quantum chemistry before fault-tolerance

Chemistry Beyond Exact Solutions of the Schrödinger Equation on a Quantum-centric Supercomputer

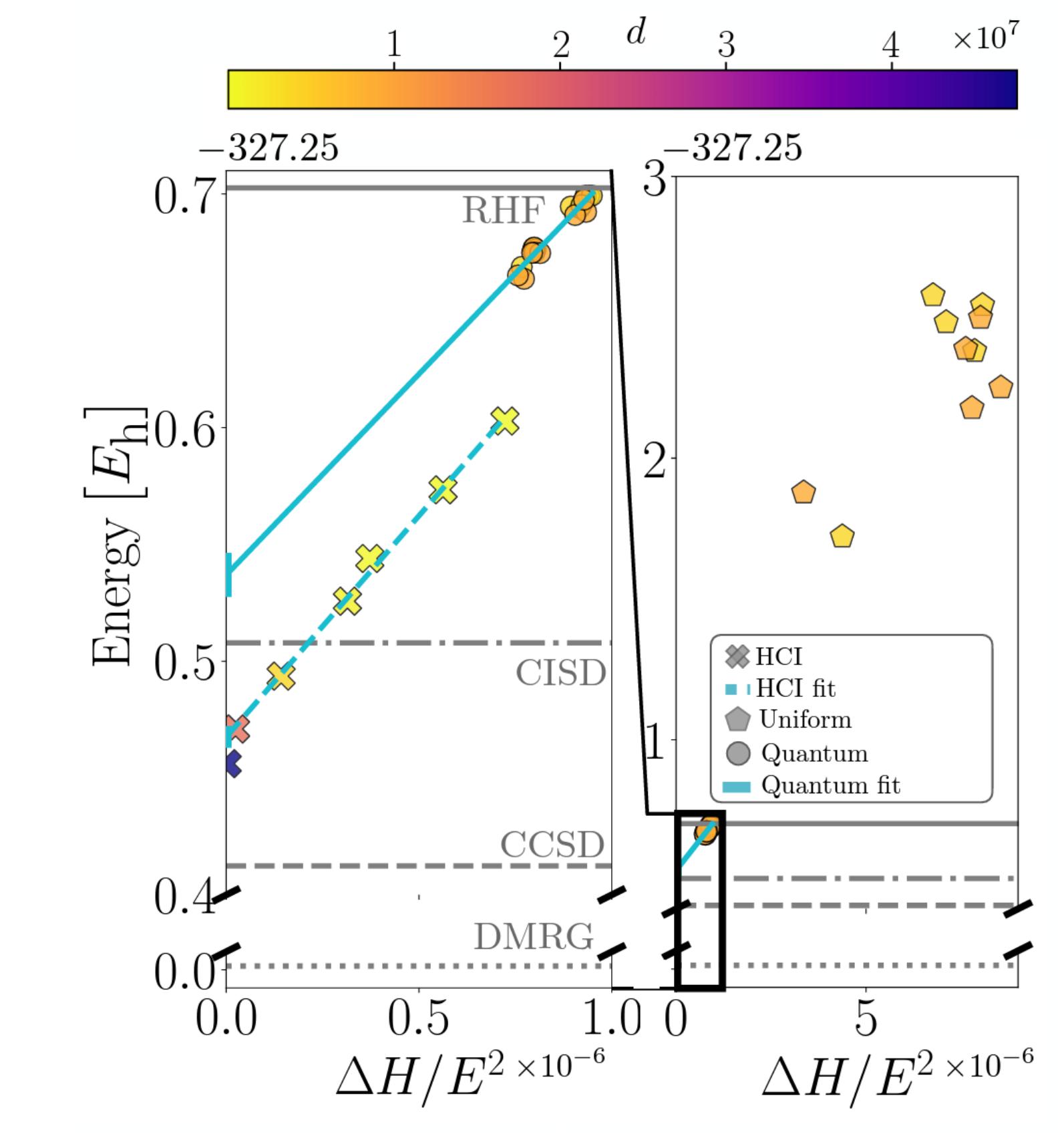
N_2 : Bond breaking on large basis set



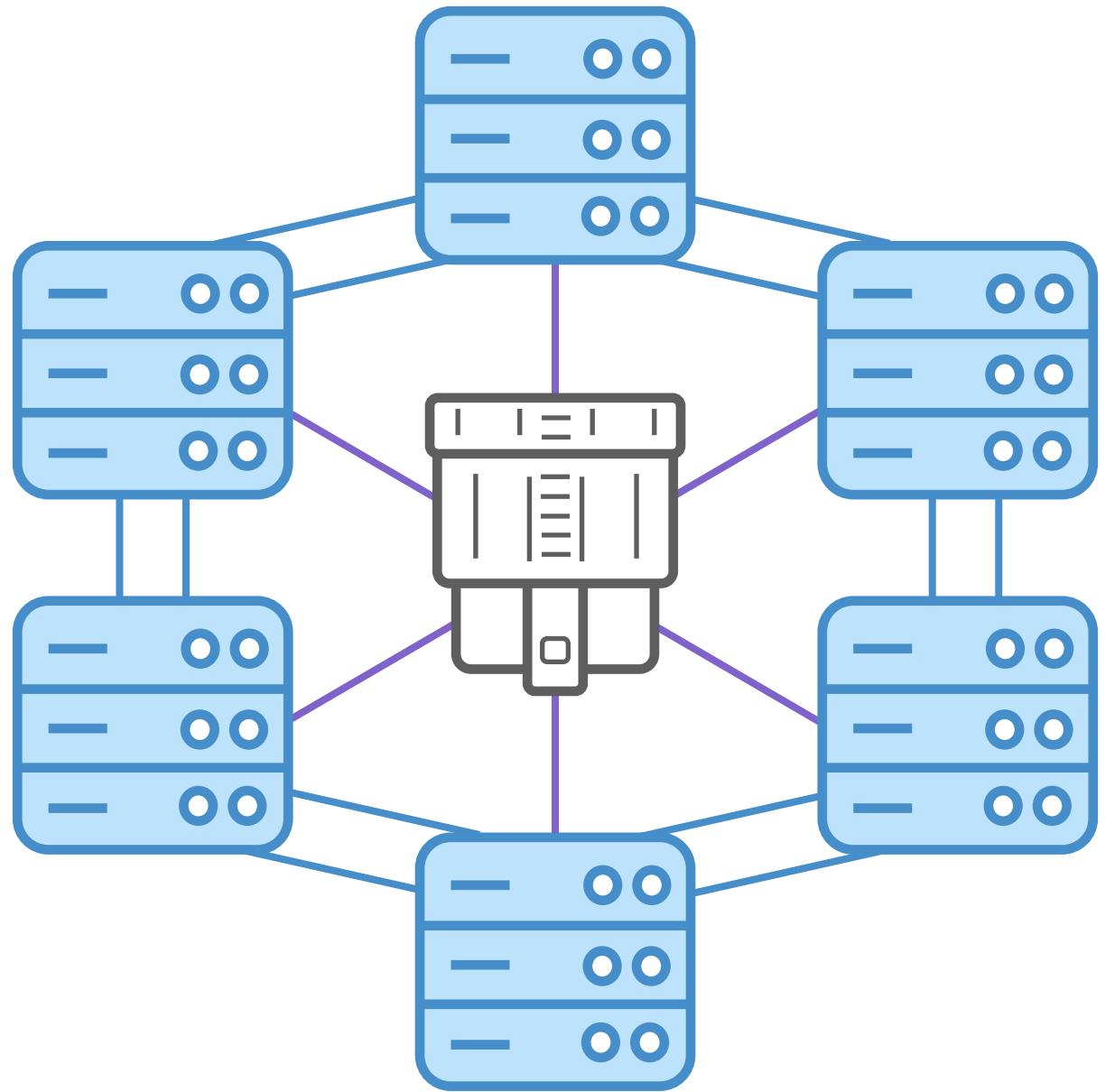
Fe_2S_2 : Precision many-body physics



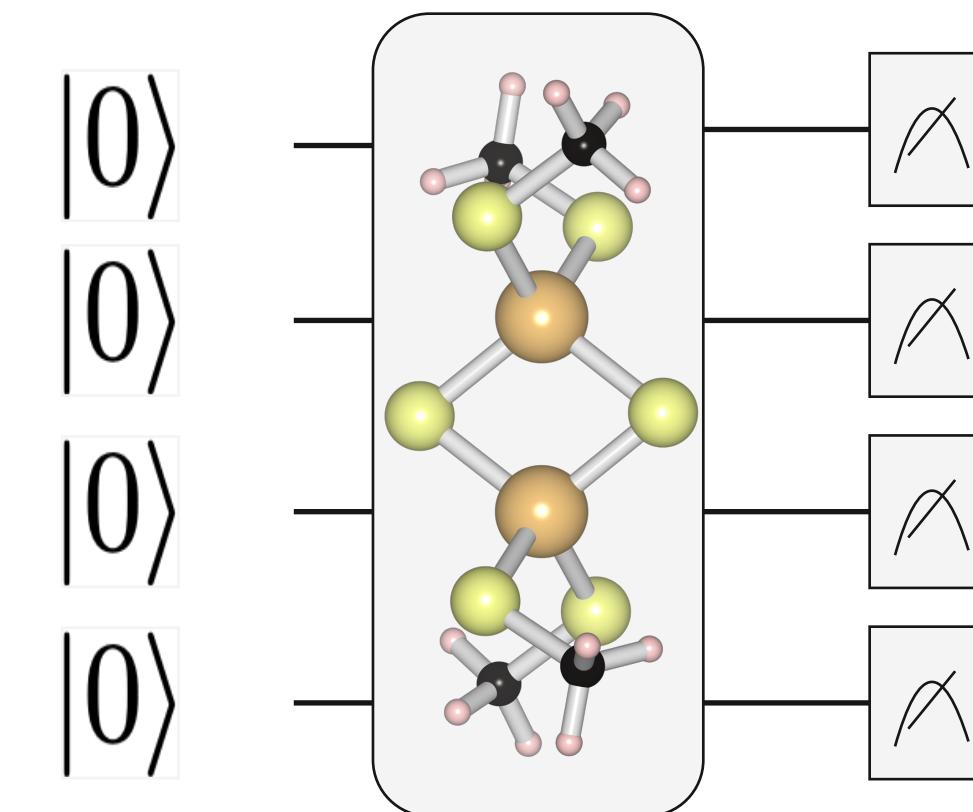
Fe_4S_4 : Pushing hardware capabilities



Chemistry on quantum-centric supercomputing



A quantum estimator that uses massive classical computing to process individual quantum samples



A class of quantum circuits of tunable depth for the accurate preparation of molecular ground states

HPC quantum estimator: accuracy, error tolerance

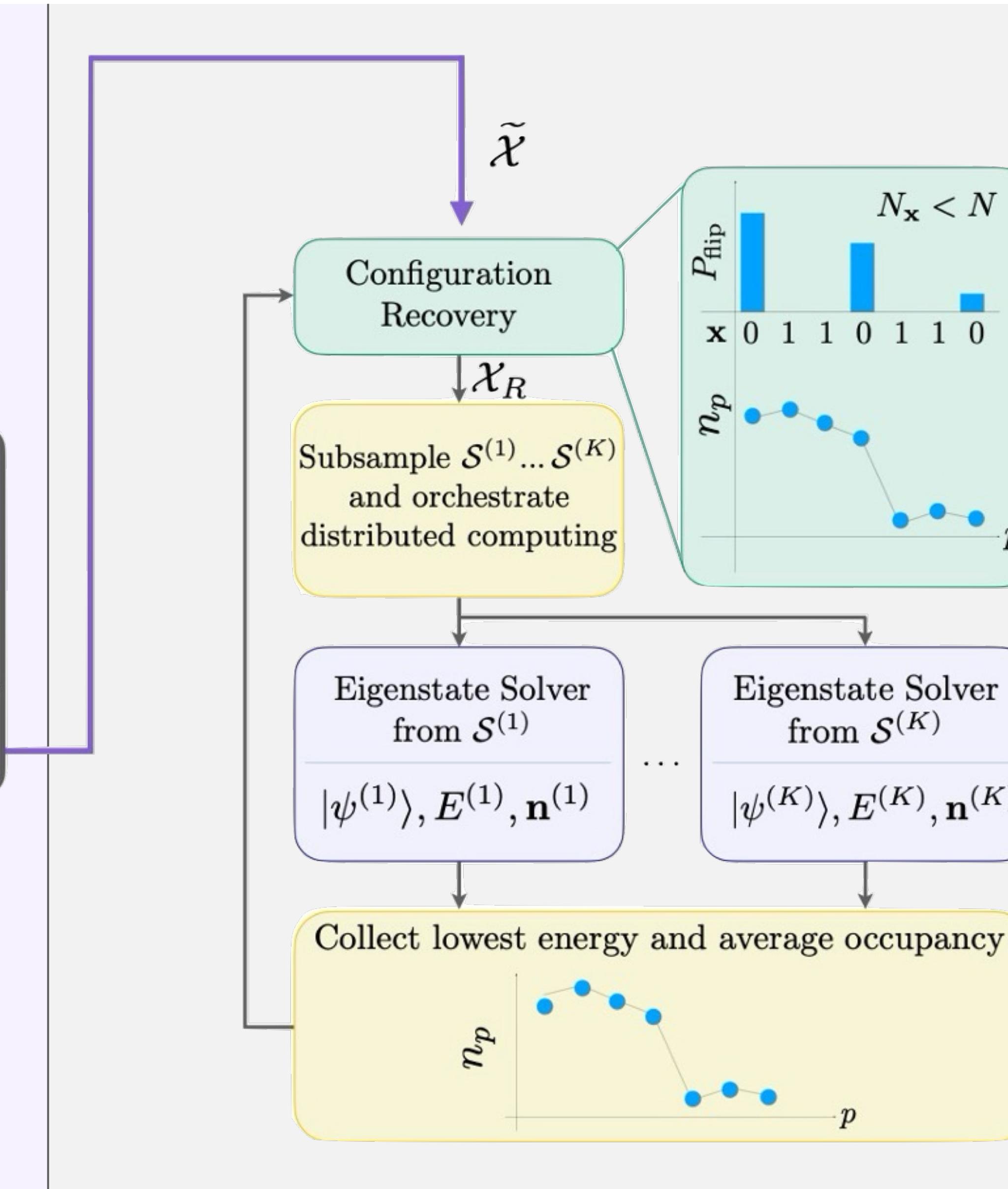
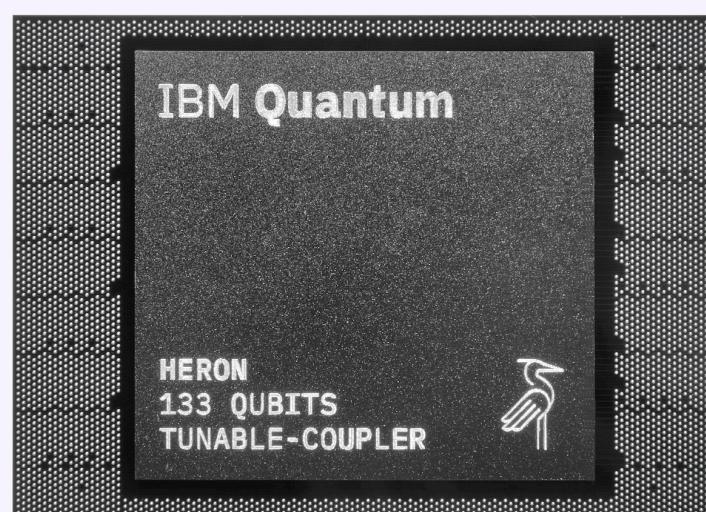
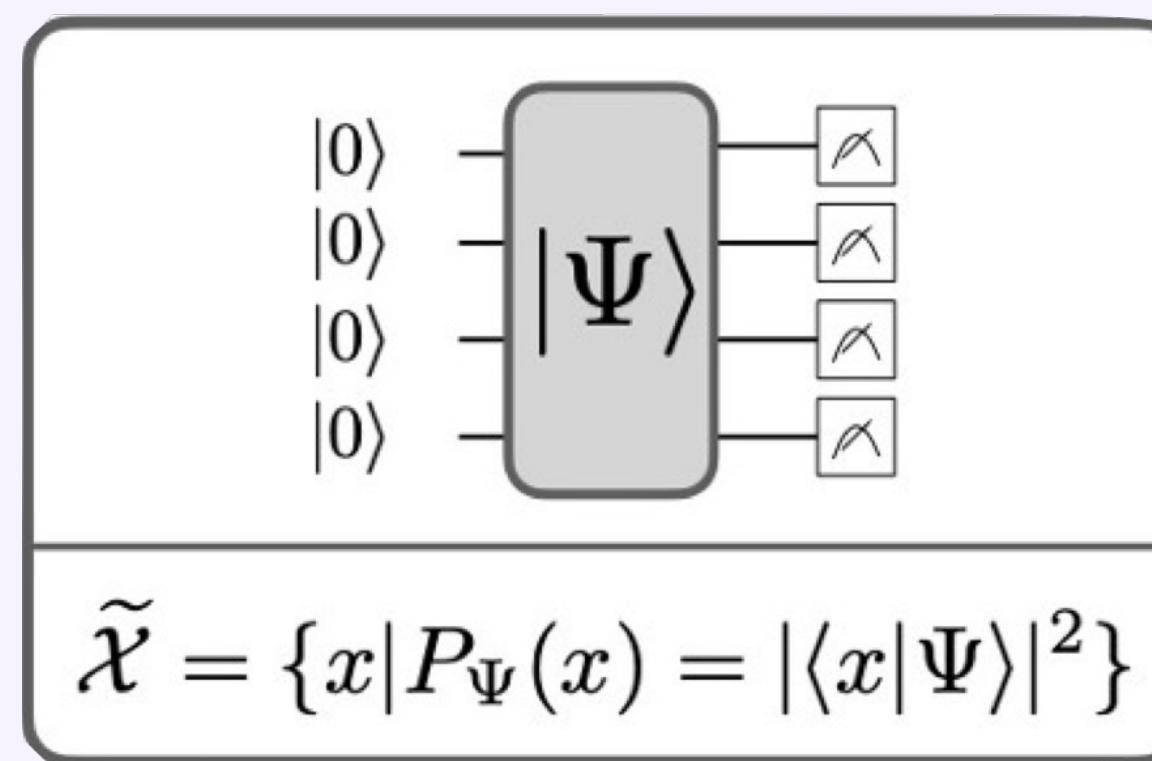
HPC quantum estimator: Properties & Highlights

- Observables with **millions** of Pauli operators
- No **false positives** for ground states: estimations are always upper bounds
- Massive **distributed** classical computations
- Complete removal of the effect of **quantum noise** with an exponential runtime in the noise rate
- Its performance is guaranteed for **concentrated wavefunction**

HPC quantum estimator: Quantum-Classical workflow

Execute using Qiskit
Runtime Primitives. Quantum
compute generates samples
from an exponentially large
space

Sampler()



Post-process using potential advanced classical compute to extract meaningful information

$$H_{\mathcal{S}^{(k)}} = P_{\mathcal{S}^{(k)}} H P_{\mathcal{S}^{(k)}}$$

with $P_{\mathcal{S}^{(k)}} = \sum_{\mathbf{x} \in \mathcal{S}^{(k)}} |\mathbf{x}\rangle\langle\mathbf{x}|$

$$H_{\mathcal{S}^{(k)}} |\psi^{(k)}\rangle = E^{(k)} |\psi^{(k)}\rangle$$

Harrison, J. Chem. Phys. 94, 5021-5031
Kanno et al., arXiv 2302.11320



Quantum-centric supercomputing: a new computational framework

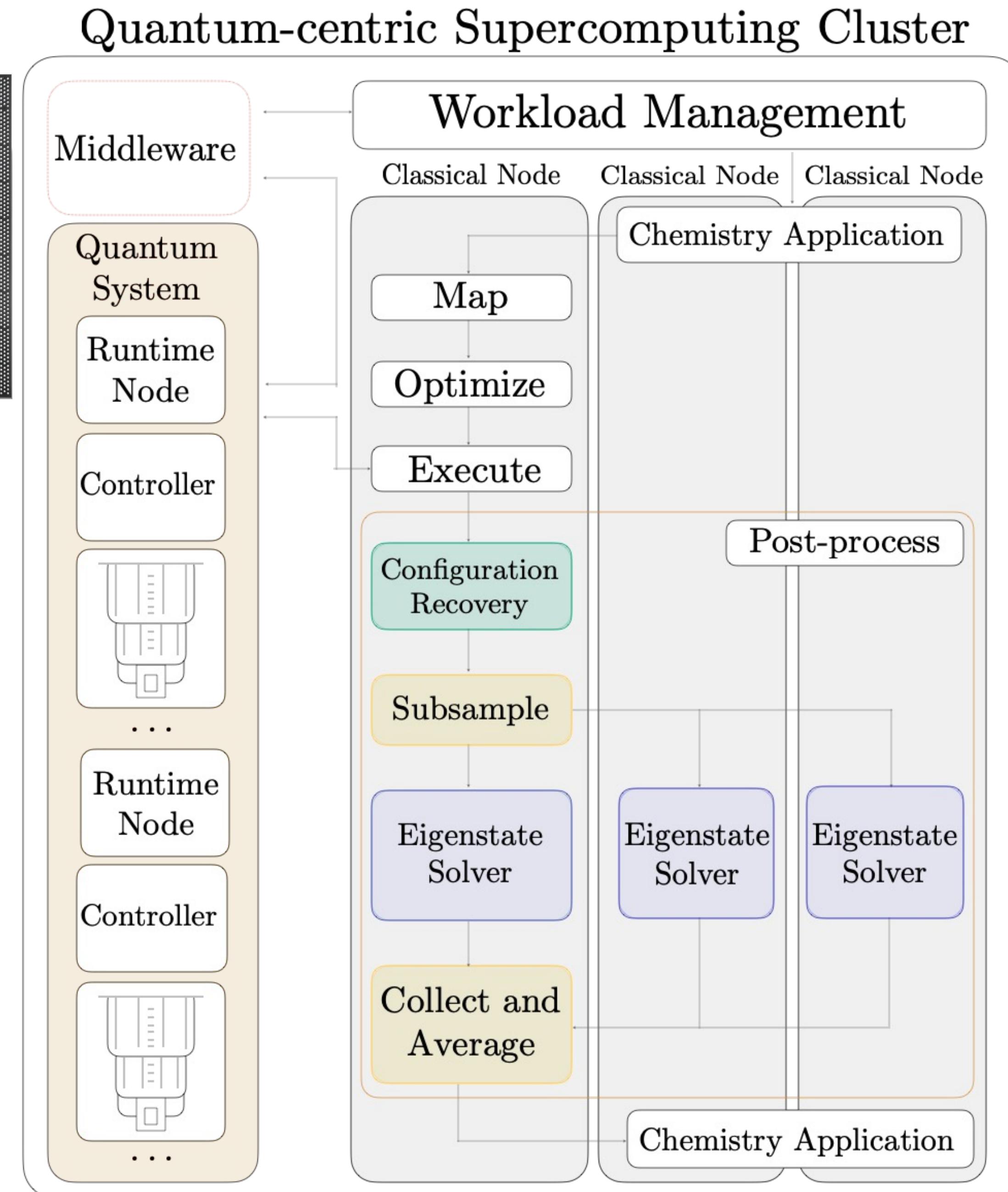


77 qubits
10570 quantum gates
3590 two-qubit gates



6400 nodes @
32 GB
1024 GB/s
48 cores

IBM Quantum

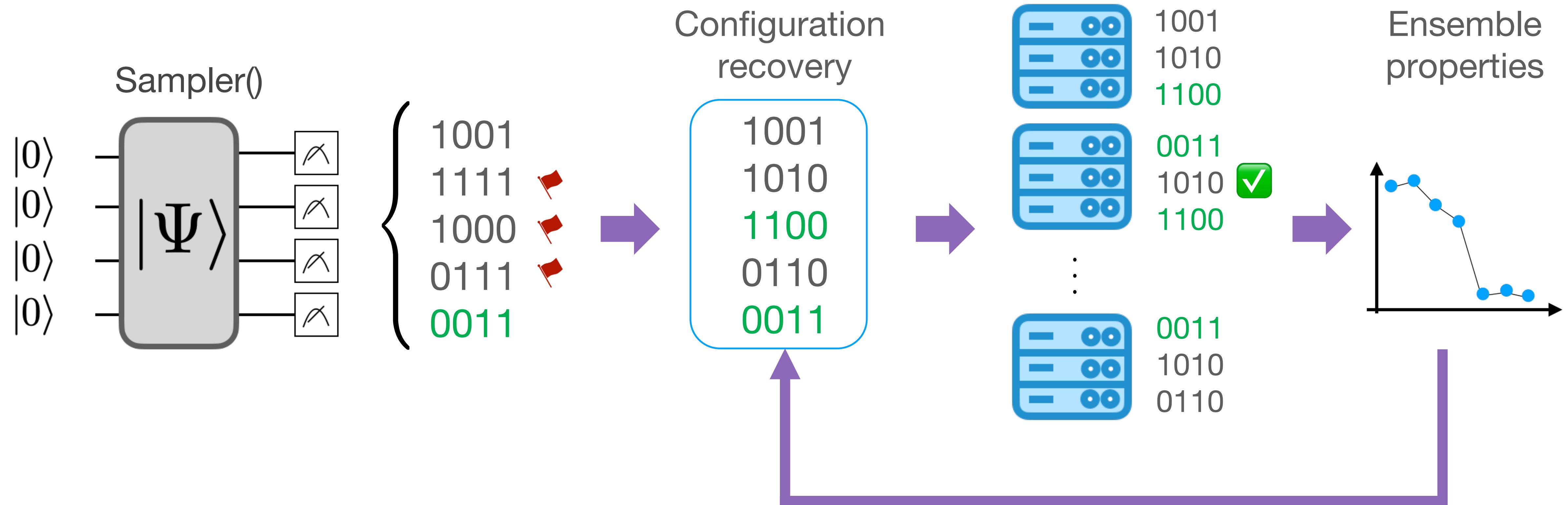


We use largest possible **data transfer** via the sampler primitive

Here 1Mbps for optimal transfer

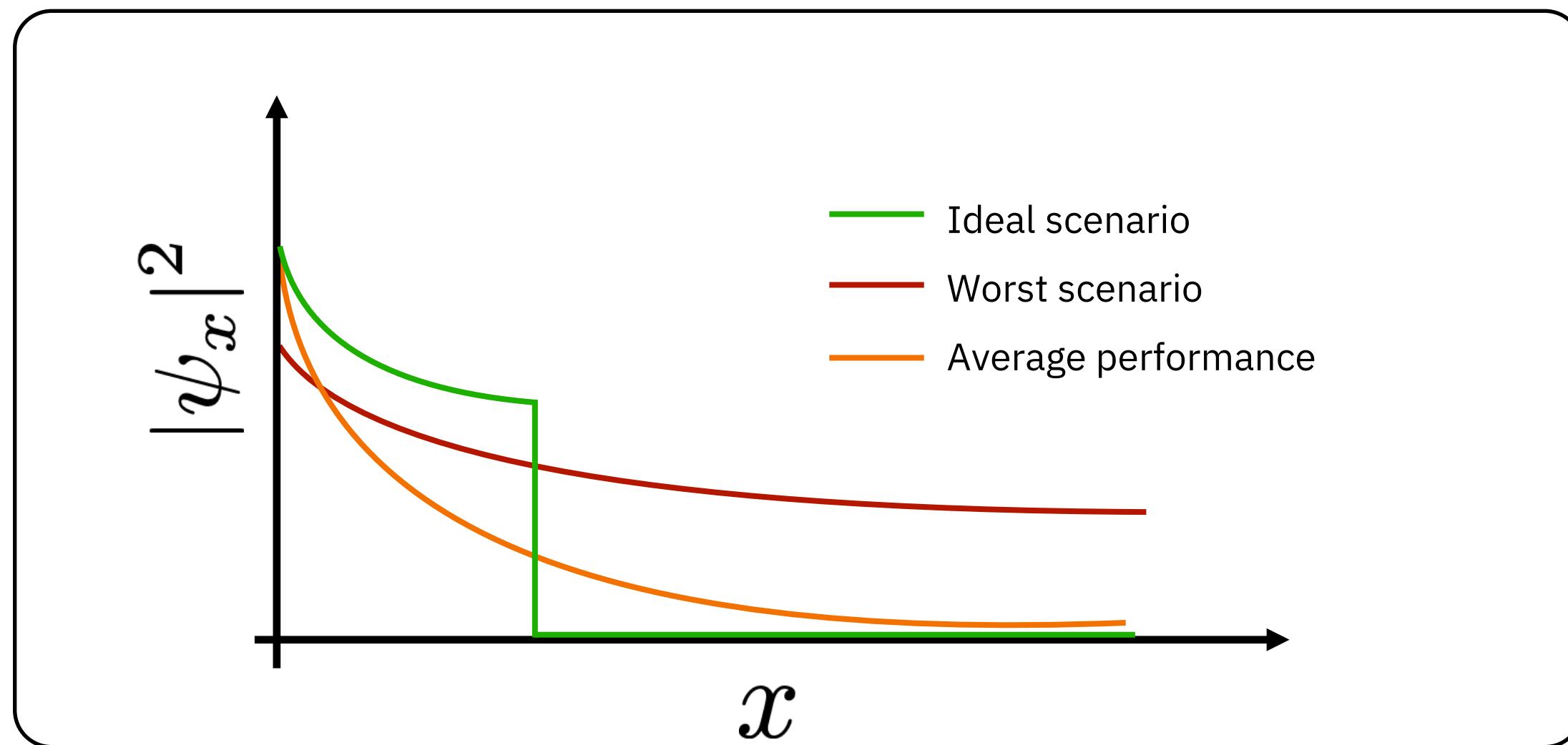
Information from all the samples is used to perform **accurate, noise-resistant estimations**

HPC quantum estimator: Quantum-Classical workflow

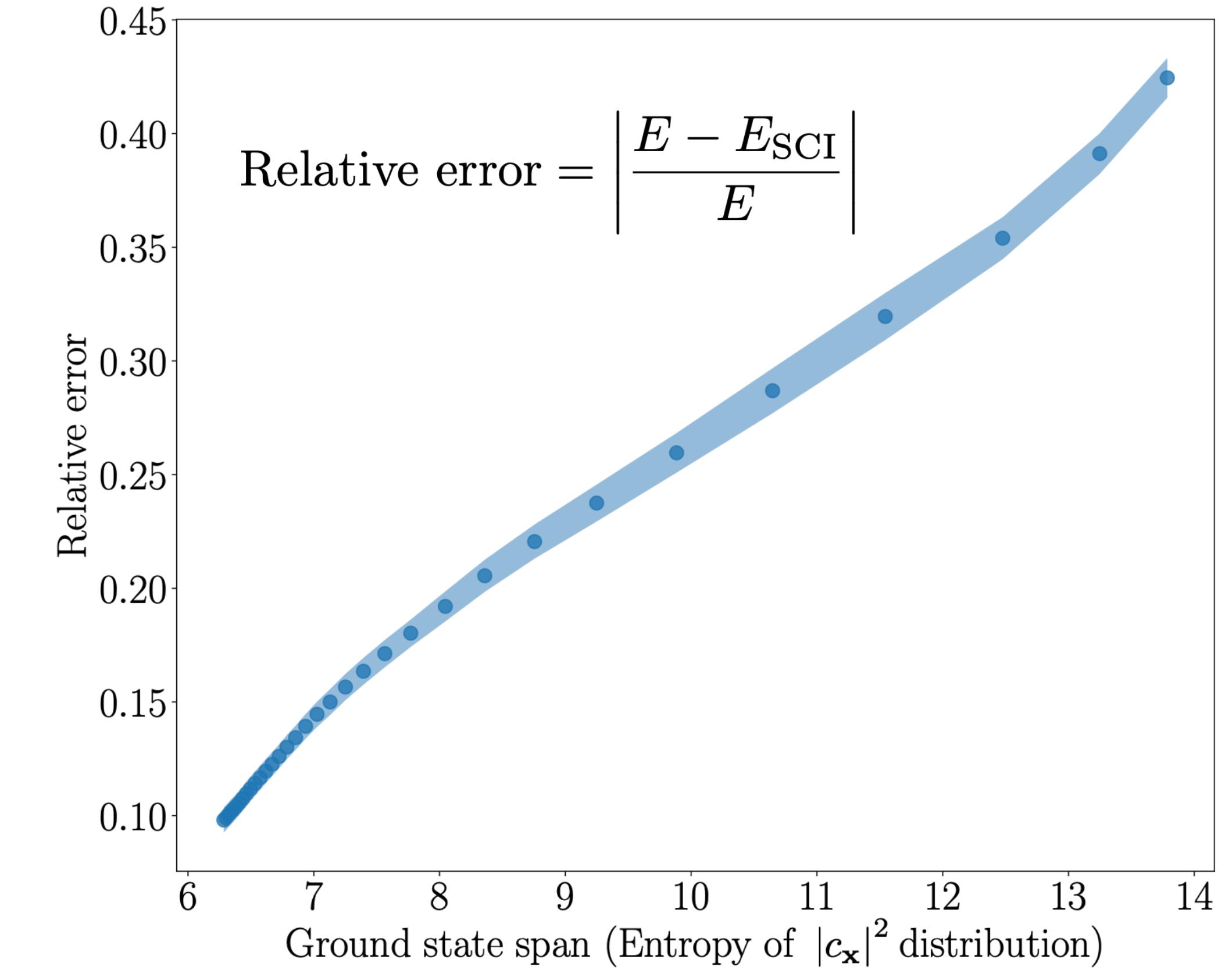


$$|\text{GroundTruth}\rangle = \frac{|1100\rangle + |0011\rangle}{\sqrt{2}}$$

HPC quantum estimator: Wavefunction Concentration



$$H = -\frac{1}{\sqrt{L}} \sum_{\substack{p,q=1 \\ \sigma=\{\uparrow,\downarrow\}}}^L t_{pq} c_{p\sigma}^\dagger c_{q\sigma} + U \sum_{p=1}^L n_{p\uparrow} n_{p\downarrow}$$



Shared assumption of classical Selected Configuration Interaction (SCI)

HPC quantum estimator: noise tolerance

$$\begin{aligned}\vec{a} &= 0011 \\ \vec{b} &= 1010 \\ \vec{c} &= 1100\end{aligned} \longrightarrow$$

$$H_{\mathcal{S}^{(k)}} = P_{\mathcal{S}^{(k)}} H P_{\mathcal{S}^{(k)}}$$

$$\text{with } P_{\mathcal{S}^{(k)}} = \sum_{\mathbf{x} \in \mathcal{S}^{(k)}} |\mathbf{x}\rangle\langle\mathbf{x}|$$

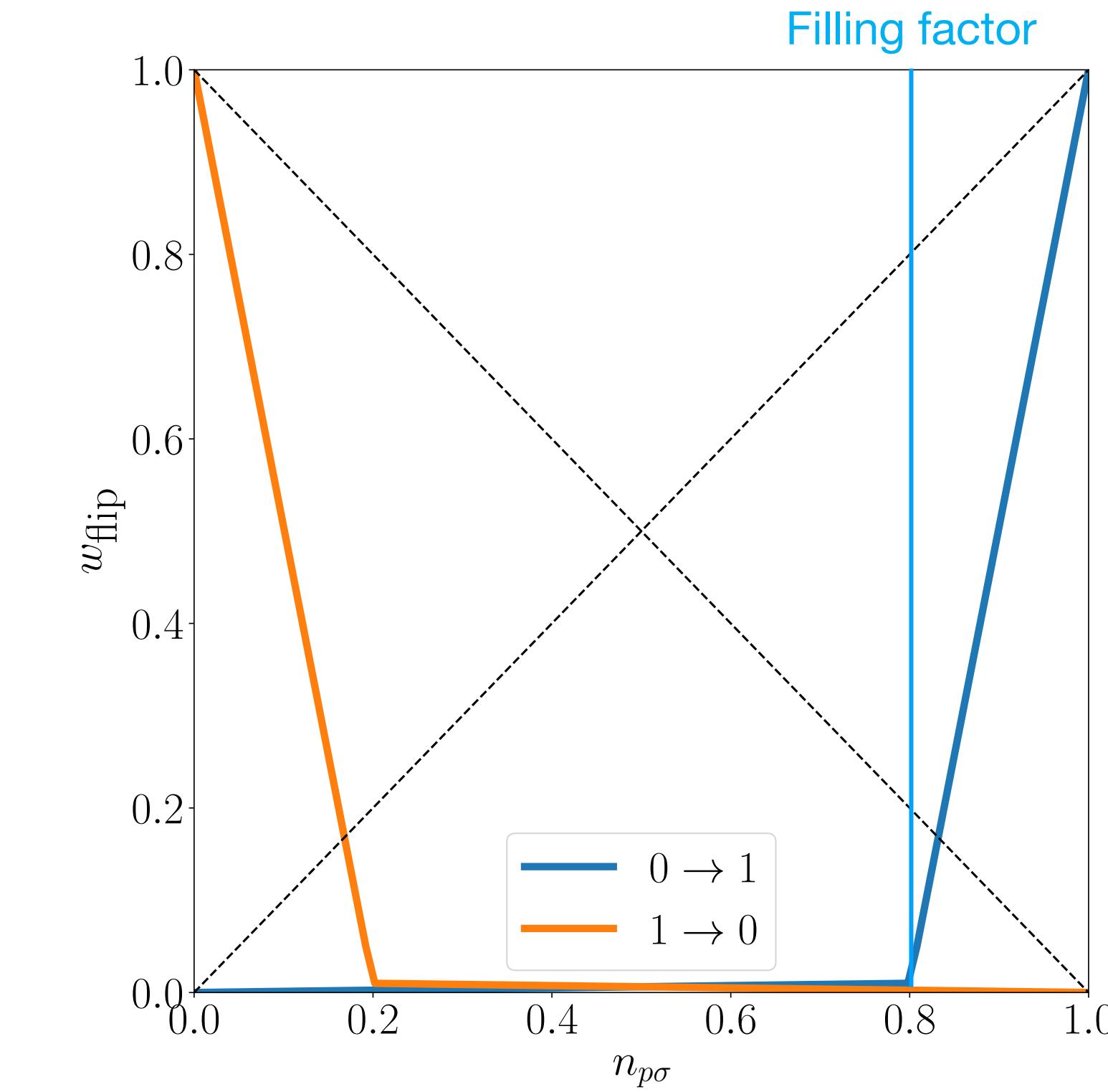
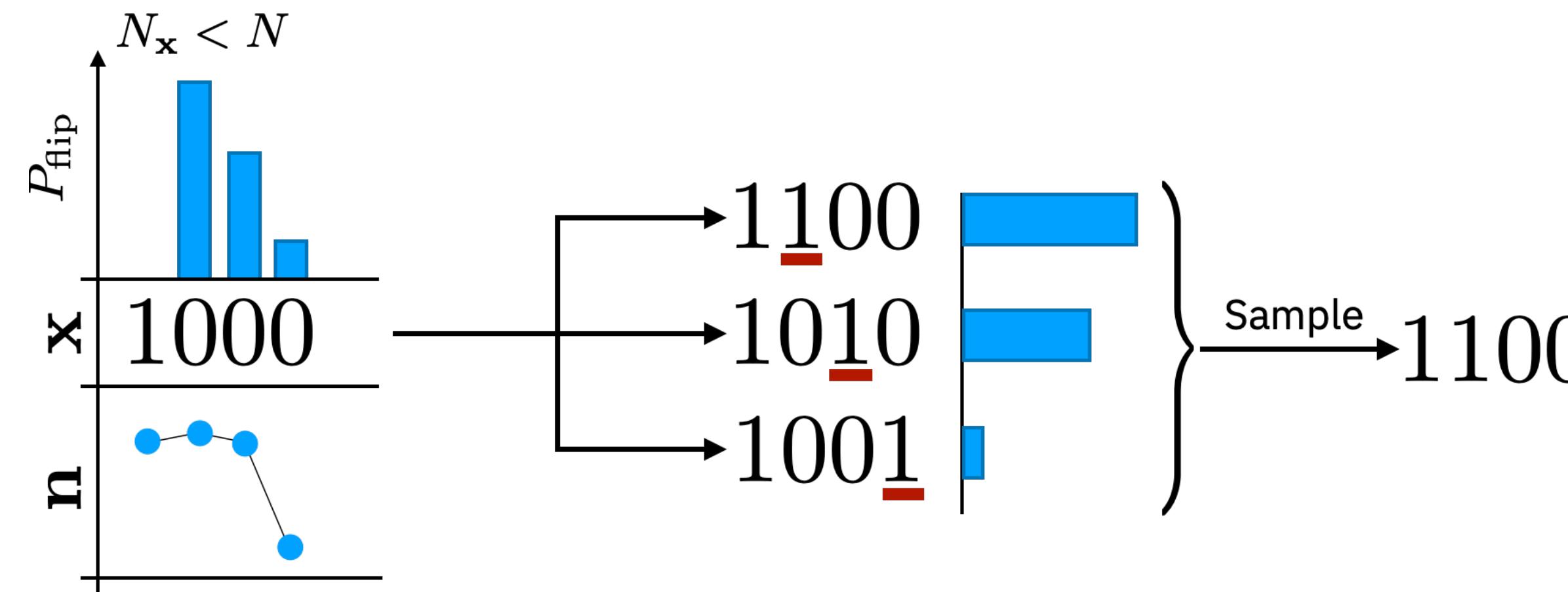
$\langle \vec{a} H \vec{a} \rangle$	$\langle \vec{a} H \vec{b} \rangle$	$\langle \vec{a} H \vec{c} \rangle$
$\langle \vec{b} H \vec{a} \rangle$	$\langle \vec{b} H \vec{b} \rangle$	$\langle \vec{b} H \vec{c} \rangle$
$\langle \vec{c} H \vec{a} \rangle$	$\langle \vec{c} H \vec{b} \rangle$	$\langle \vec{c} H \vec{c} \rangle$

$$H_{\mathcal{S}^{(k)}} |\psi^{(k)}\rangle = E^{(k)} |\psi^{(k)}\rangle$$

$$|\psi^{(k)}\rangle = \frac{1}{\sqrt{2}}\vec{a} + 0 \cdot \vec{b} + \frac{1}{\sqrt{2}}\vec{c}$$

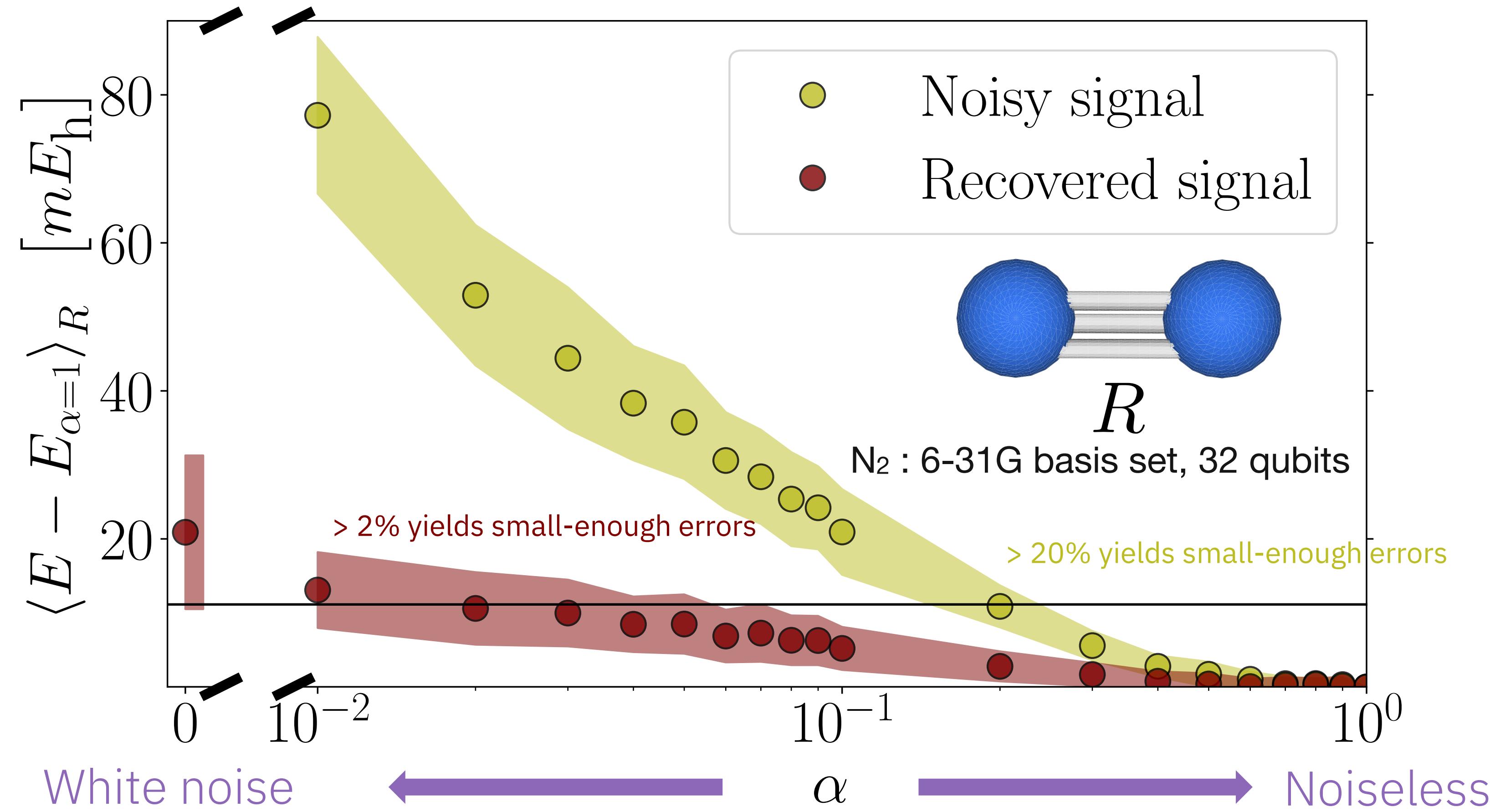
$$|\text{GroundTruth}\rangle = \frac{|1100\rangle + |0011\rangle}{\sqrt{2}}$$

HPC quantum estimator: self-consistent configuration recovery



Particle number restored by flipping bits according to a probability informed by the difference between the value of the bit and the average occupancy of the orbital and the filling factor

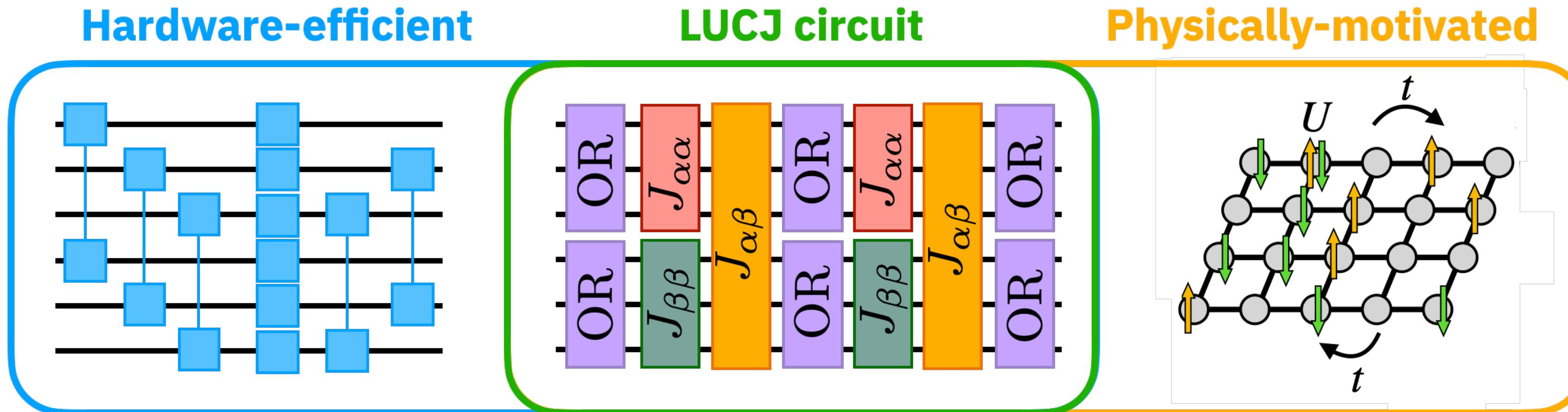
HPC quantum estimator: self-consistent configuration recovery



$$P_{\text{noisy}}(x) = \alpha \cdot P_{\text{noiseless}}(x) + (1 - \alpha) \cdot \frac{1}{2^M}$$

Quantum circuits for chemistry: Local Unitary Coupled Jastrow

Quantum circuits for chemistry: Local Unitary Coupled Jastrow



Compatible with hardware connectivity/gates/depth...
Hard to initialize and optimize

Parameters from classical calculations, easy to optimize
All-to-all connectivity, high circuit depth and gate count



Local Unitary Cluster Jastrow

Physical motivation (derived from coupled-cluster)
Hardware friendliness (device-specific connectivity)

Quantum circuits for chemistry: Local Unitary Coupled Jastrow

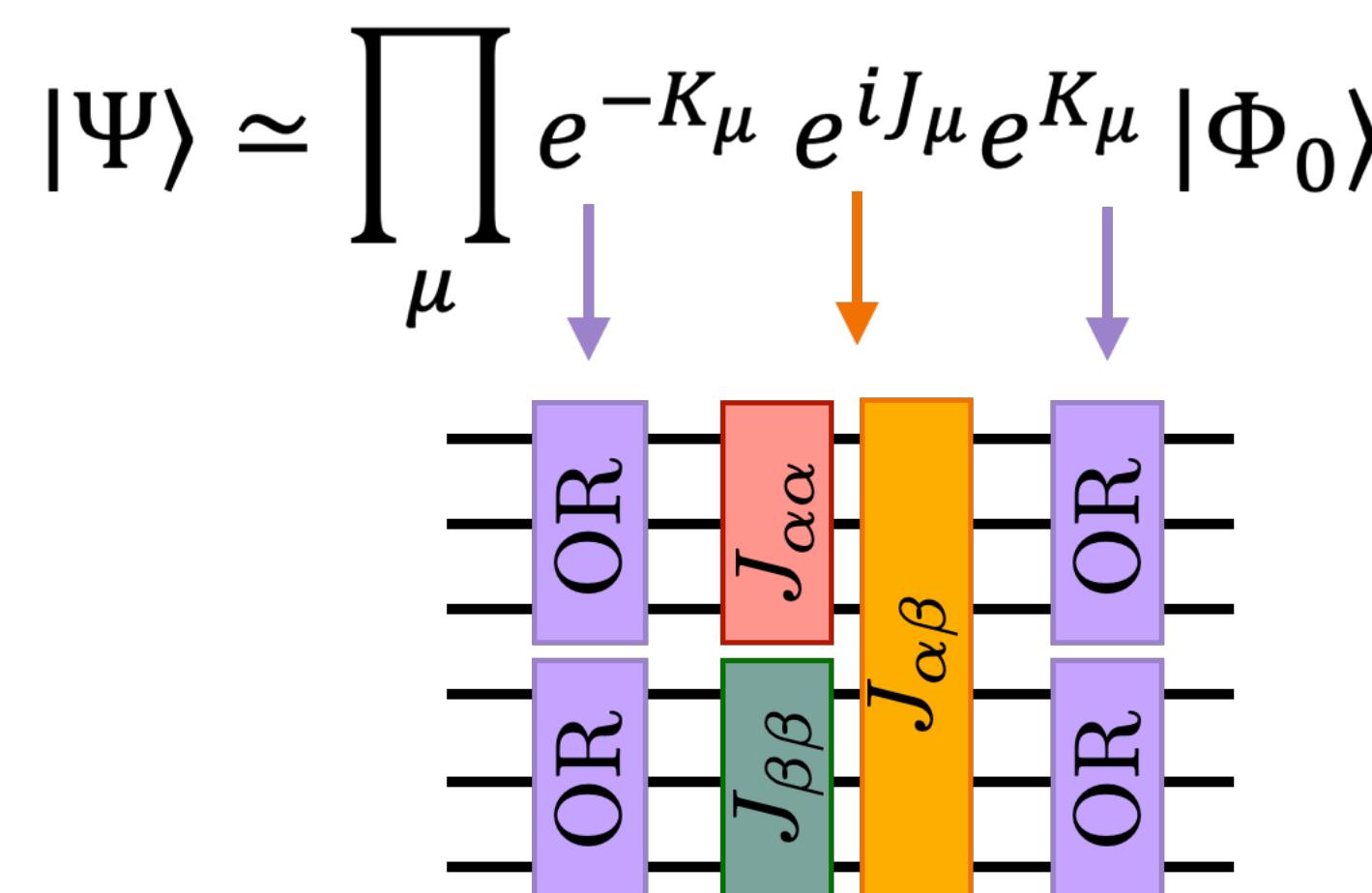
- Unitary coupled-cluster singles and doubles (UCCSD)

$$|\Psi\rangle = e^{T-T^+} |\Phi_0\rangle \quad T = \sum_{ai} t_i^a \hat{c}_a^\dagger \hat{c}_i + \sum_{abij} t_{ij}^{ab} \hat{c}_a^\dagger \hat{c}_b^\dagger \hat{c}_j \hat{c}_i$$

- Jastrow form of the UCCSD:

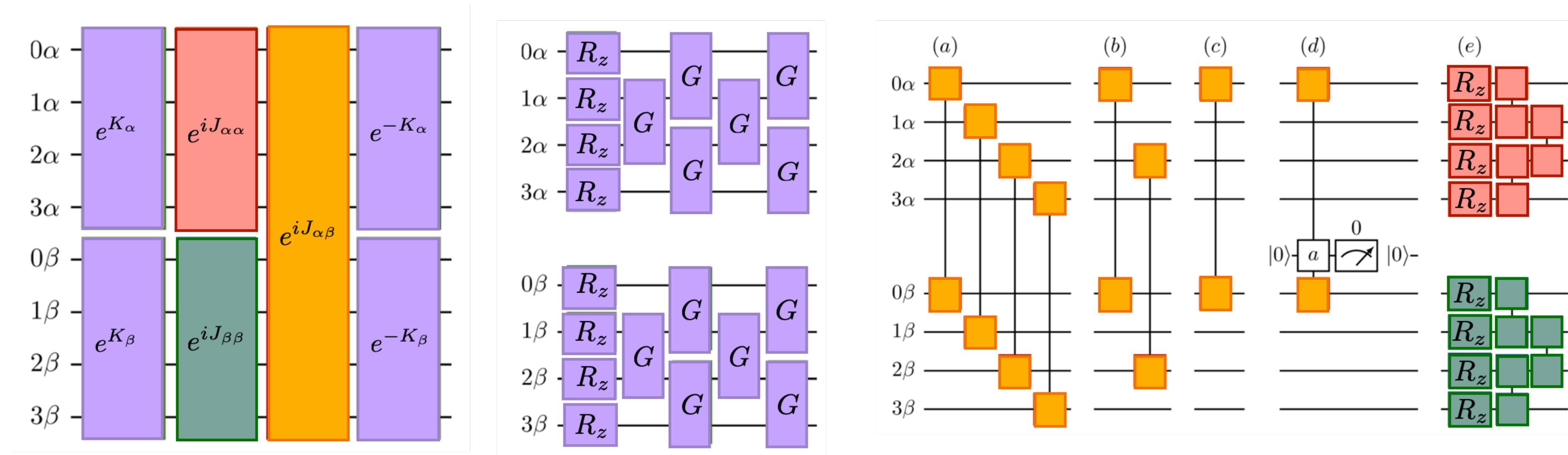
$$T - T^+ \simeq \sum_{\mu} e^{-K_{\mu}} iJ_{\mu} e^{K_{\mu}}, \quad K_{\mu} = \sum_{pr,\sigma} K_{pr}^{\mu} \hat{c}_{p\sigma}^\dagger \hat{c}_{r\sigma}, \quad J_{\mu} = \sum_{pr,\sigma\tau} J_{pr}^{\sigma\tau} \hat{n}_{p\sigma} \hat{n}_{r\tau}$$

- Local Unitary Cluster Jastrow ansatz^{*}

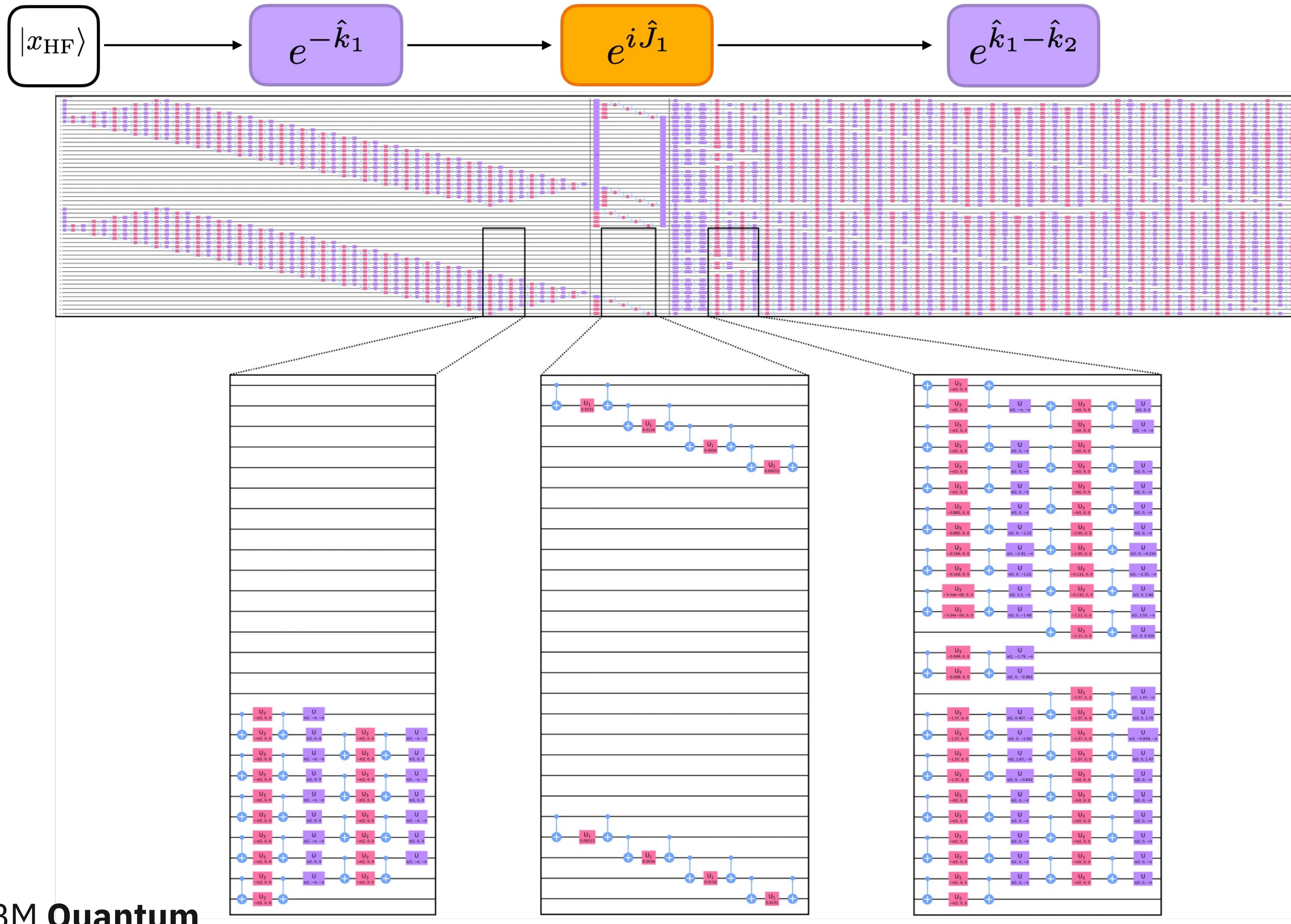


Quantum circuits for chemistry: Local Unitary Coupled Jastrow

- These circuits can take as input parameters from classical solutions
- There is evidence that they can perform better than their classical counterpart*
- The circuits are hard to simulate classically
- Compatible with quantum hardware

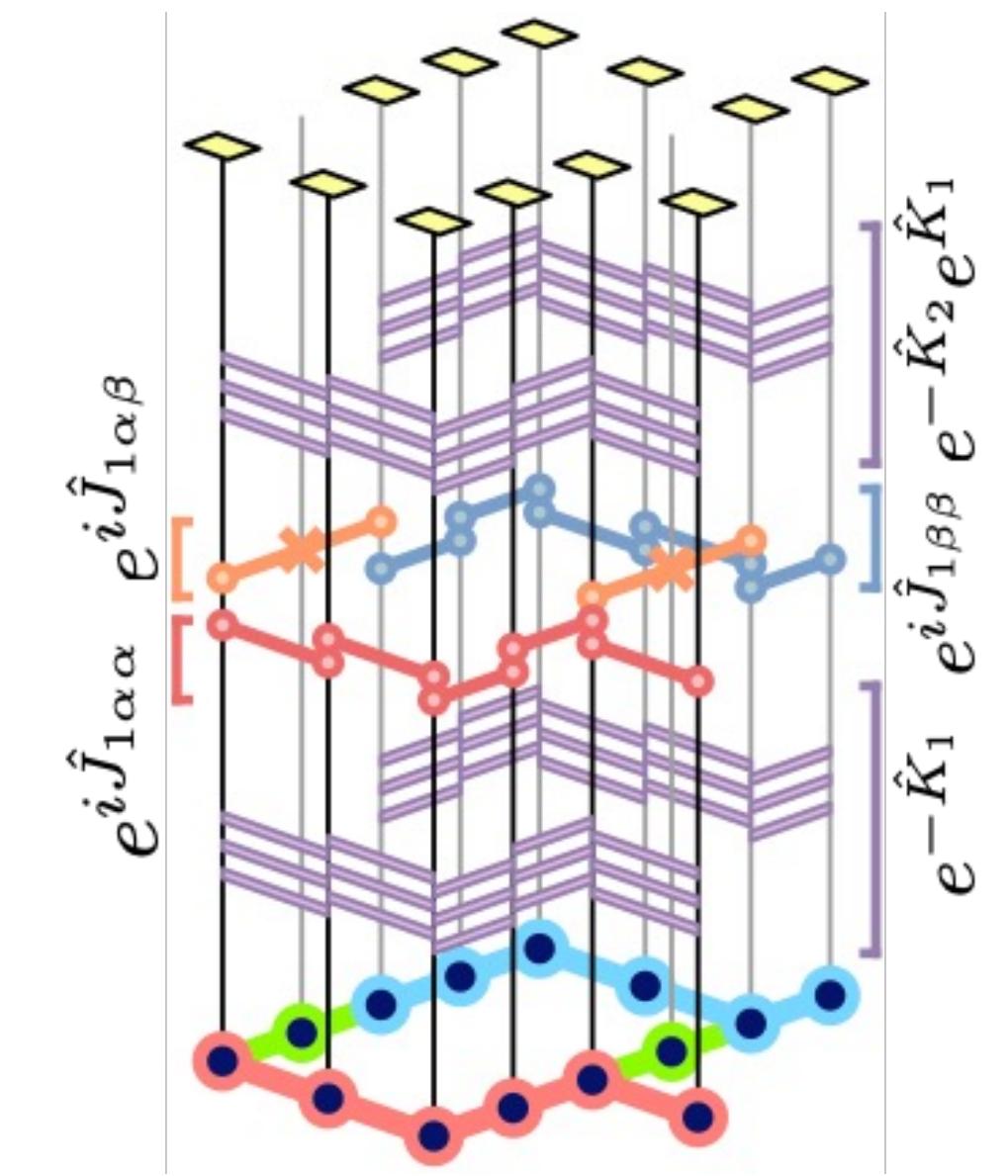


Quantum circuits for chemistry: Local Unitary Coupled Jastrow

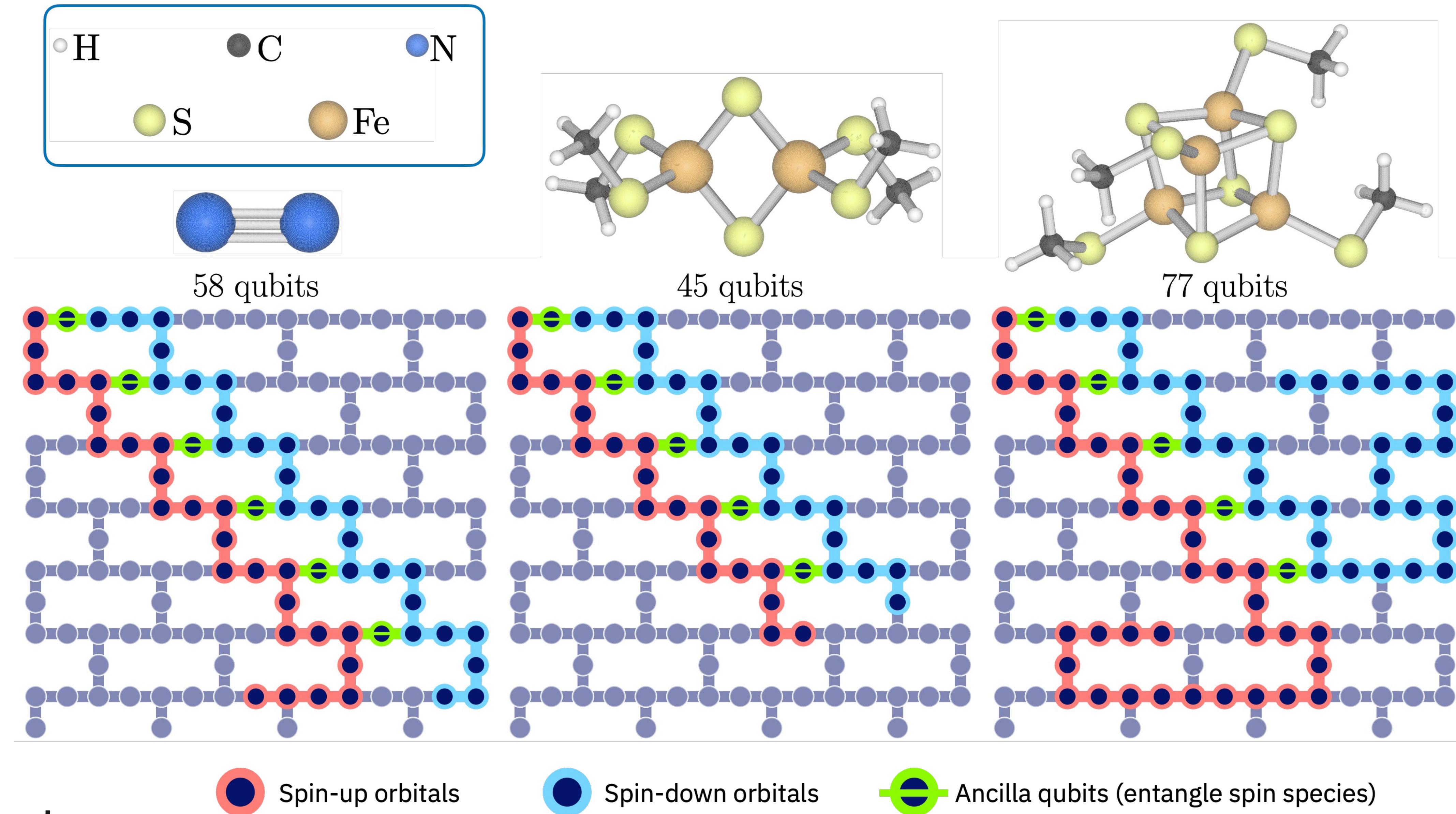


- Circuit diagram N_2 52-qubit experiment

Qubits	CNOT layers	Number of CNOTS
52	108	~ 1760



Quantum circuits for chemistry: Local Unitary Coupled Jastrow



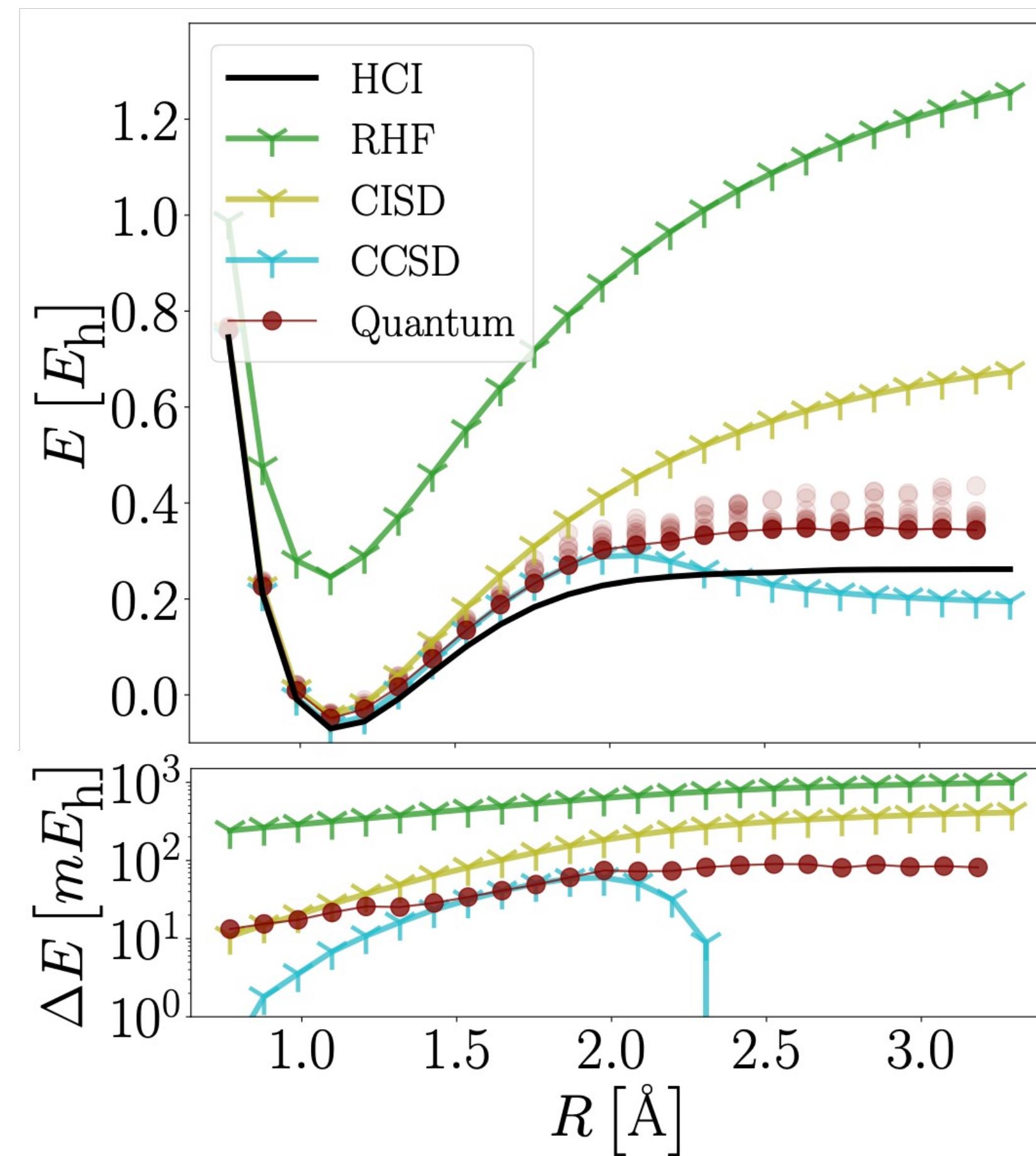
Chemistry experiments on quantum hardware

Chemistry on large basis sets: Brute-force requirements

Systems	Basis	Qubits	Memory vector storage	Reference
BeH	STO-3G	6	1KiB	<i>Nature</i> 549, 242 (2017)
Fe ₂ S ₂	Cc-pvtz (DKH)	40	2 GiB	This work
N ₂	Cc-pvdz	52	33 GiB	This work
C ₃ H ₈	STO-3G	Classical	7TiB	<i>J. Chem. Theory Comput.</i> 20, 1185 (2024)
Fugaku Storage	-	-	4.85PiB	-
Fe ₄ S ₄	Cc-pvtz (DKH)	72	63 PiB	This work

Classical SOTA

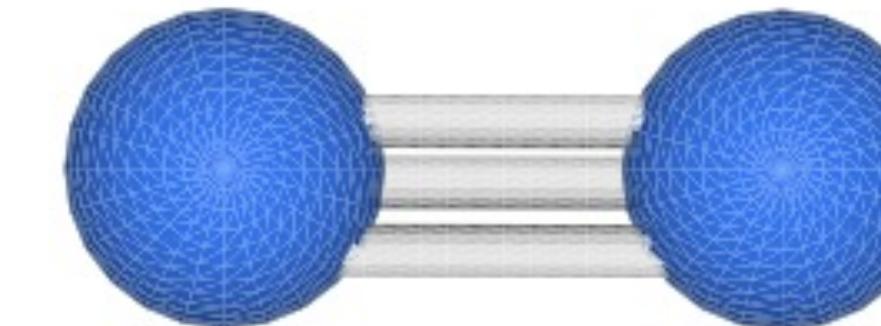
Bond breaking on large basis sets



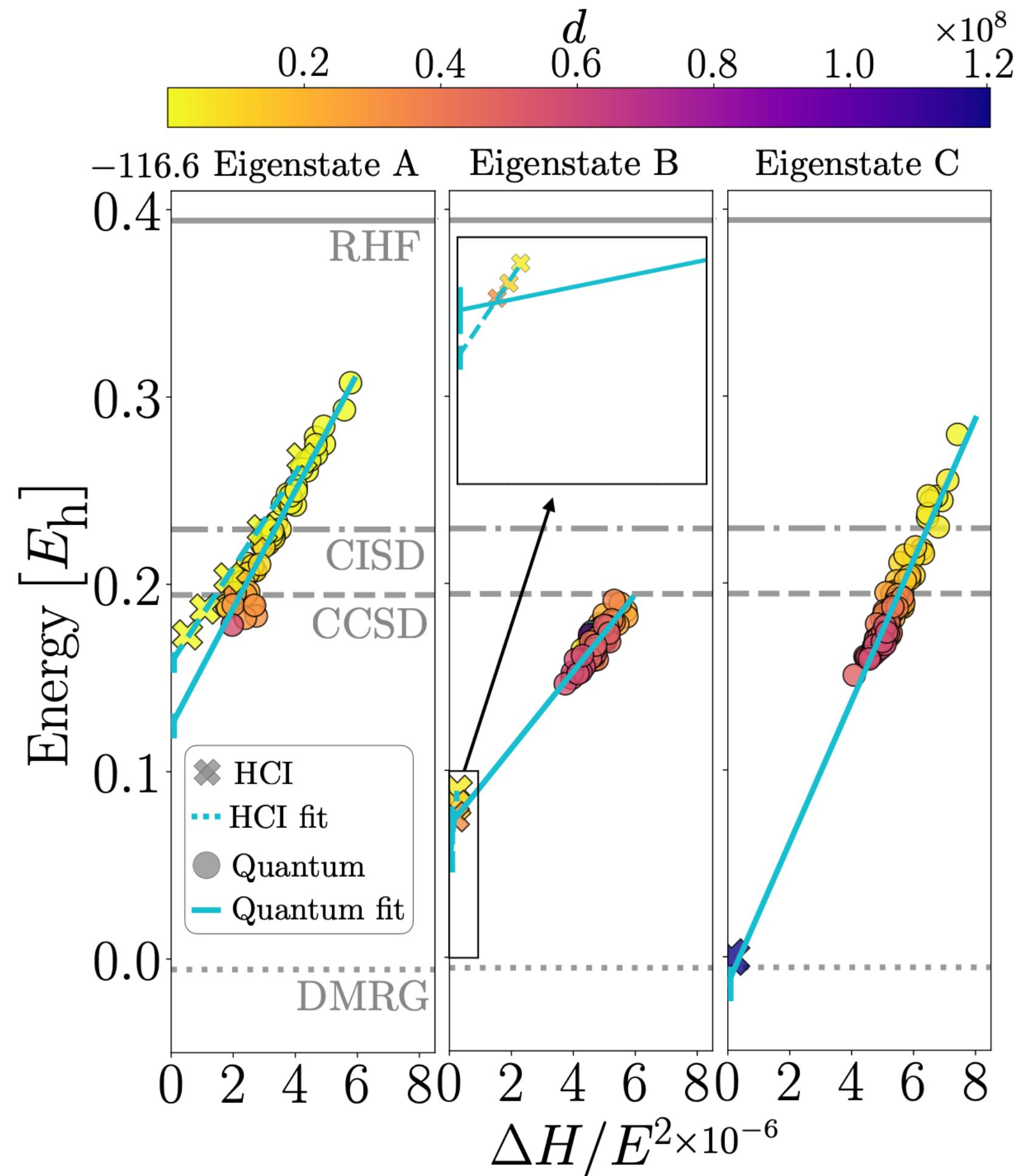
58 qubits
5204 quantum gates
1792 two-qubit gates

The breaking of the N_2 bond is a recognized test of the accuracy for electronic structure methods against static electronic correlation

Quantum data can qualitatively capture multi-reference ground states on molecular systems of 52 spin-orbitals



Hardware results: precision many-body physics

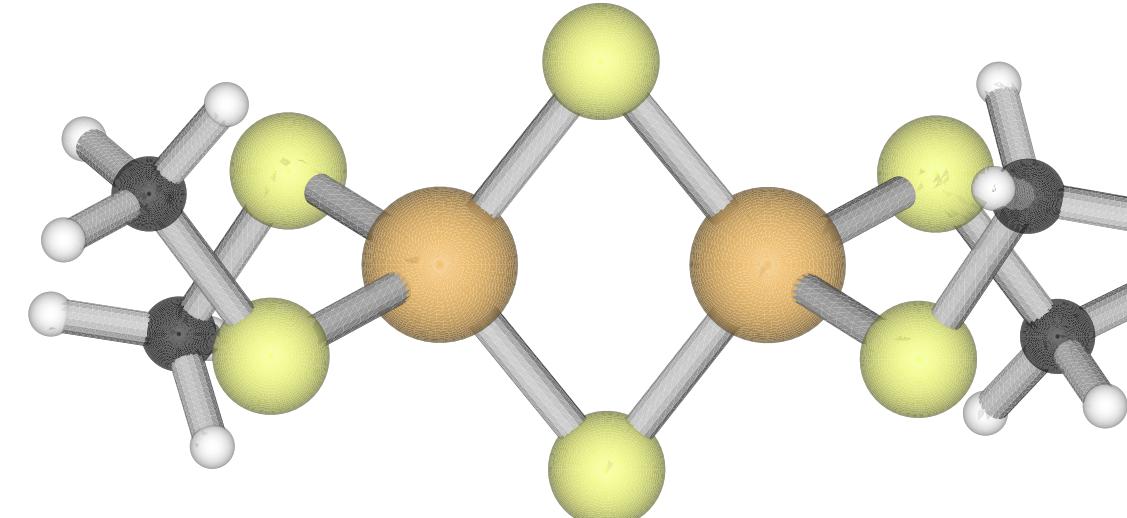


45 qubits
3170 quantum gates
1100 two-qubit gates

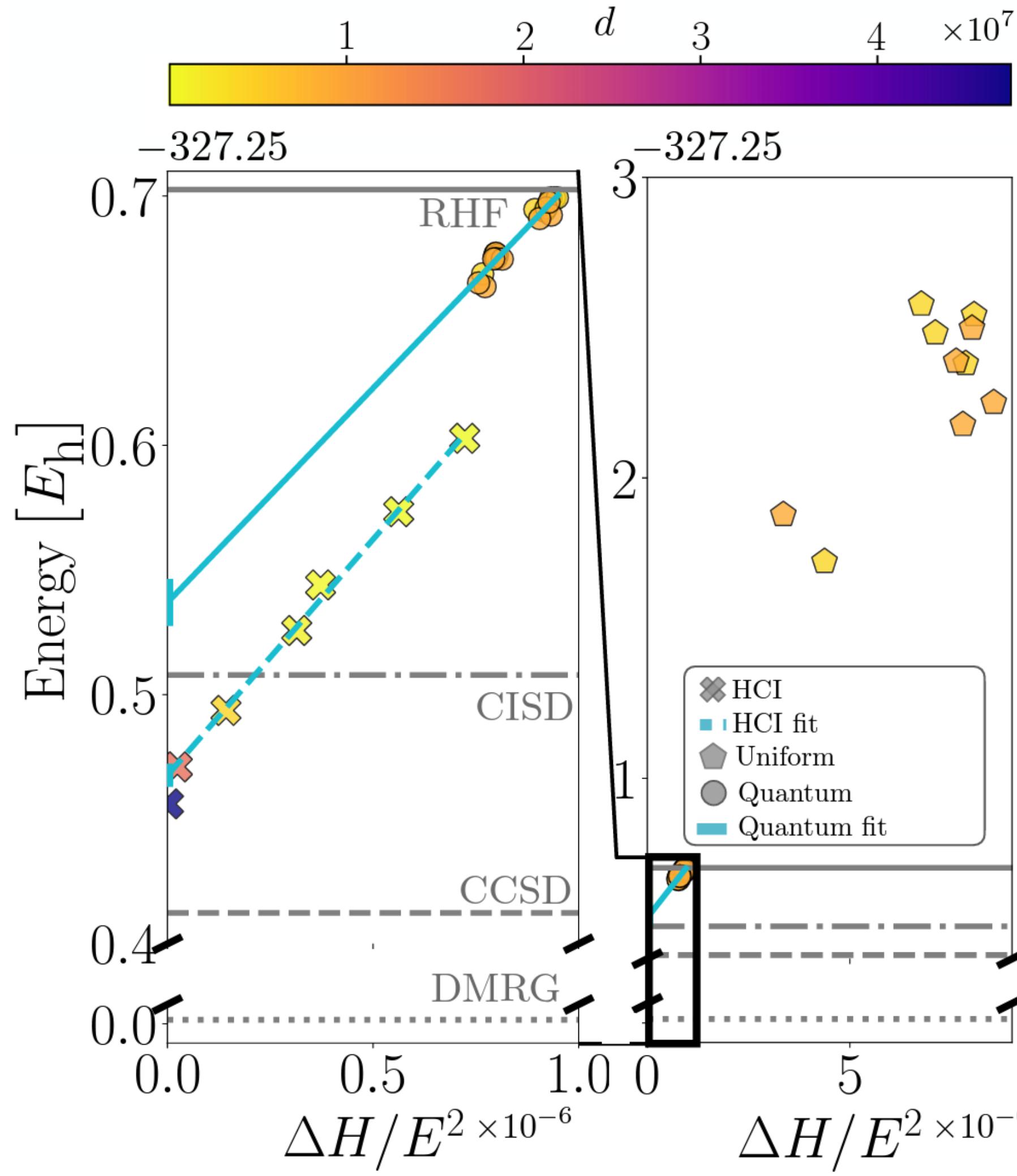
No false positives: the energies produced are upper bounds, and lower energies are certifiably better

A linear arrangement of points in the energy-variance plane signals that we are sampling an eigenstate

Linear extrapolations of the quantum data confirms that we have found three eigenstates of Fe_2S_2

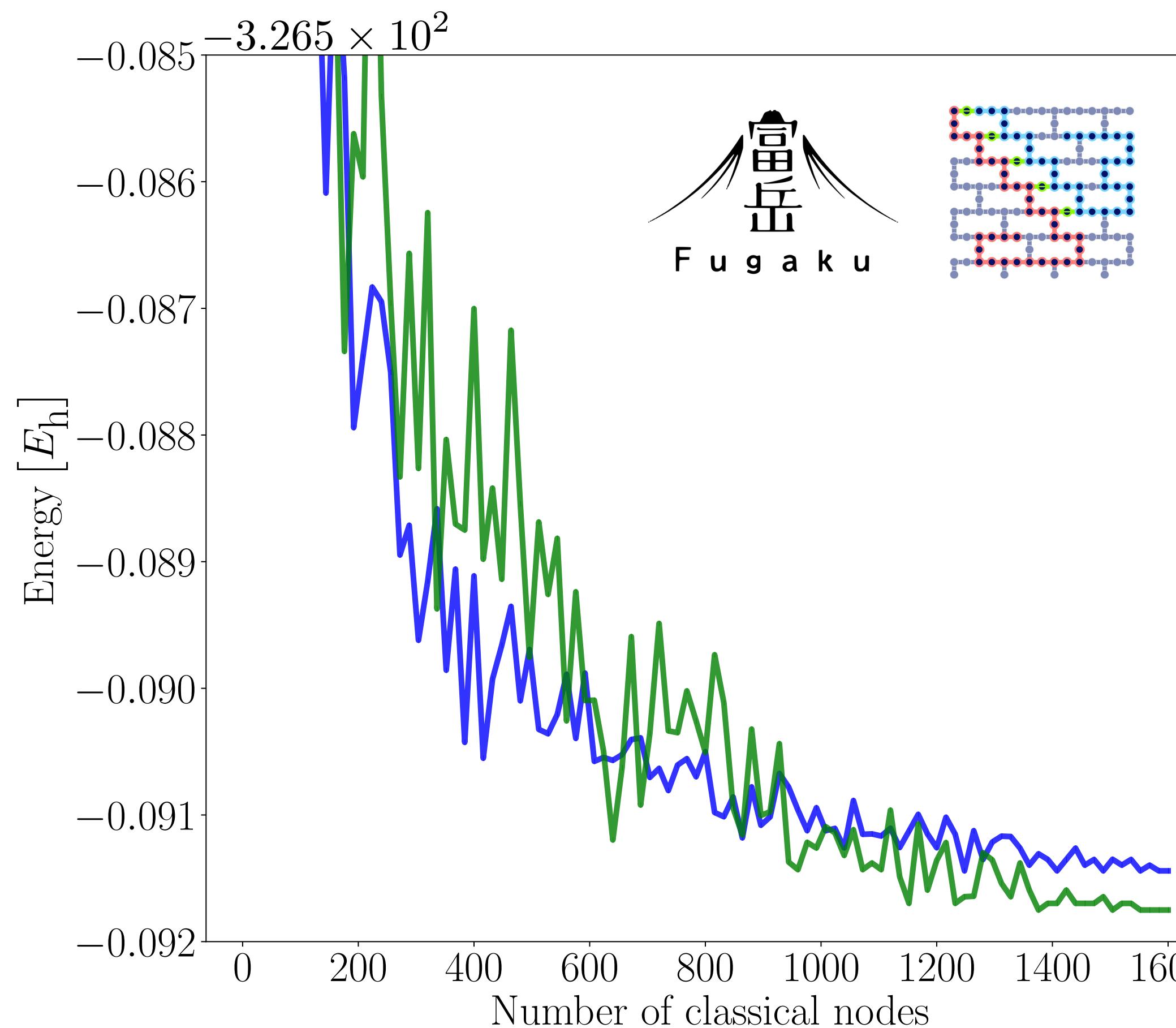


Pushing hardware capabilities



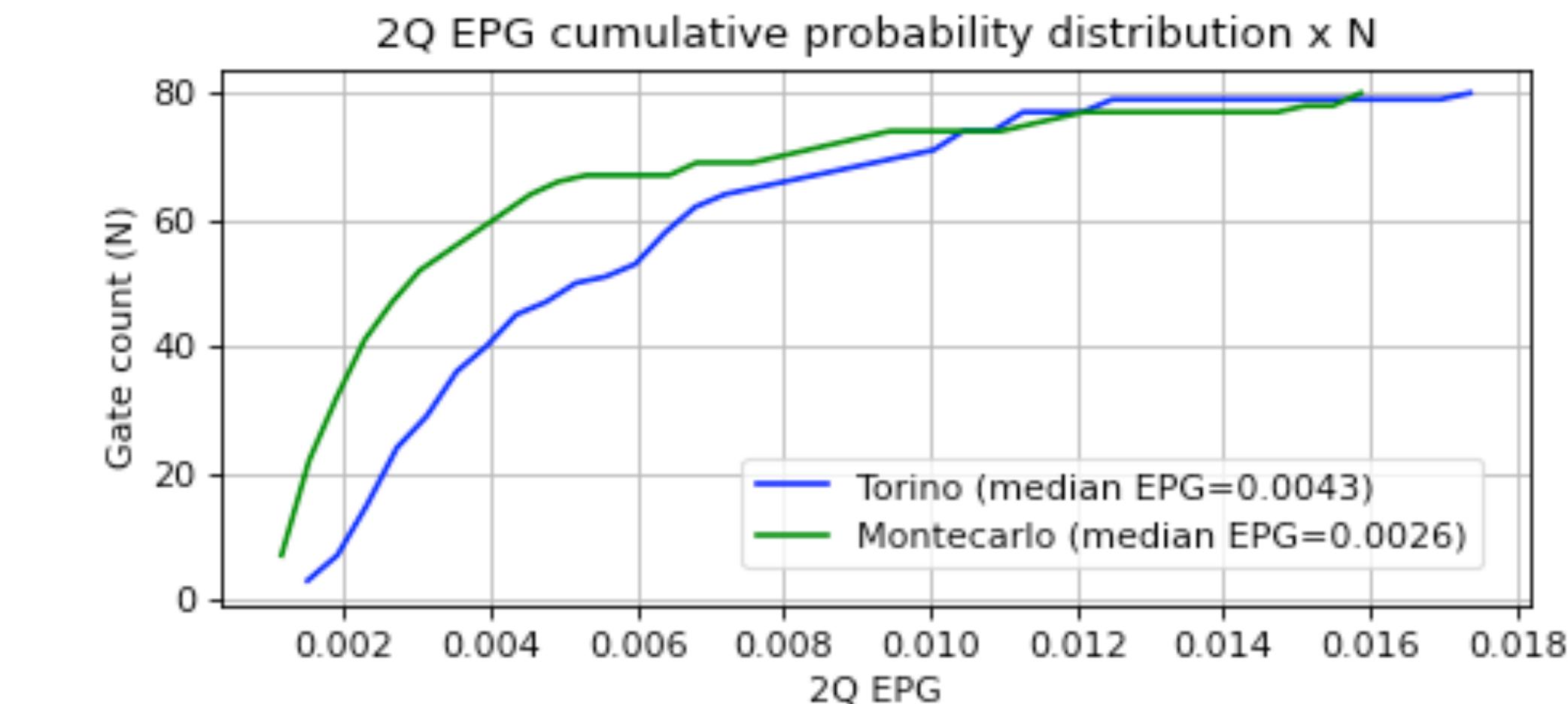
More classical compute extracts better signals out
of noisy quantum data

Pushing hardware capabilities



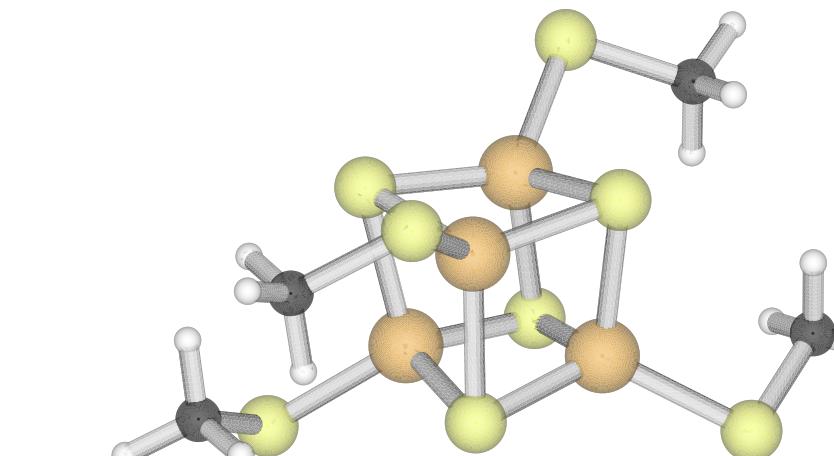
77 qubits
10570 quantum gates
3590 two-qubit gates

IBM Quantum

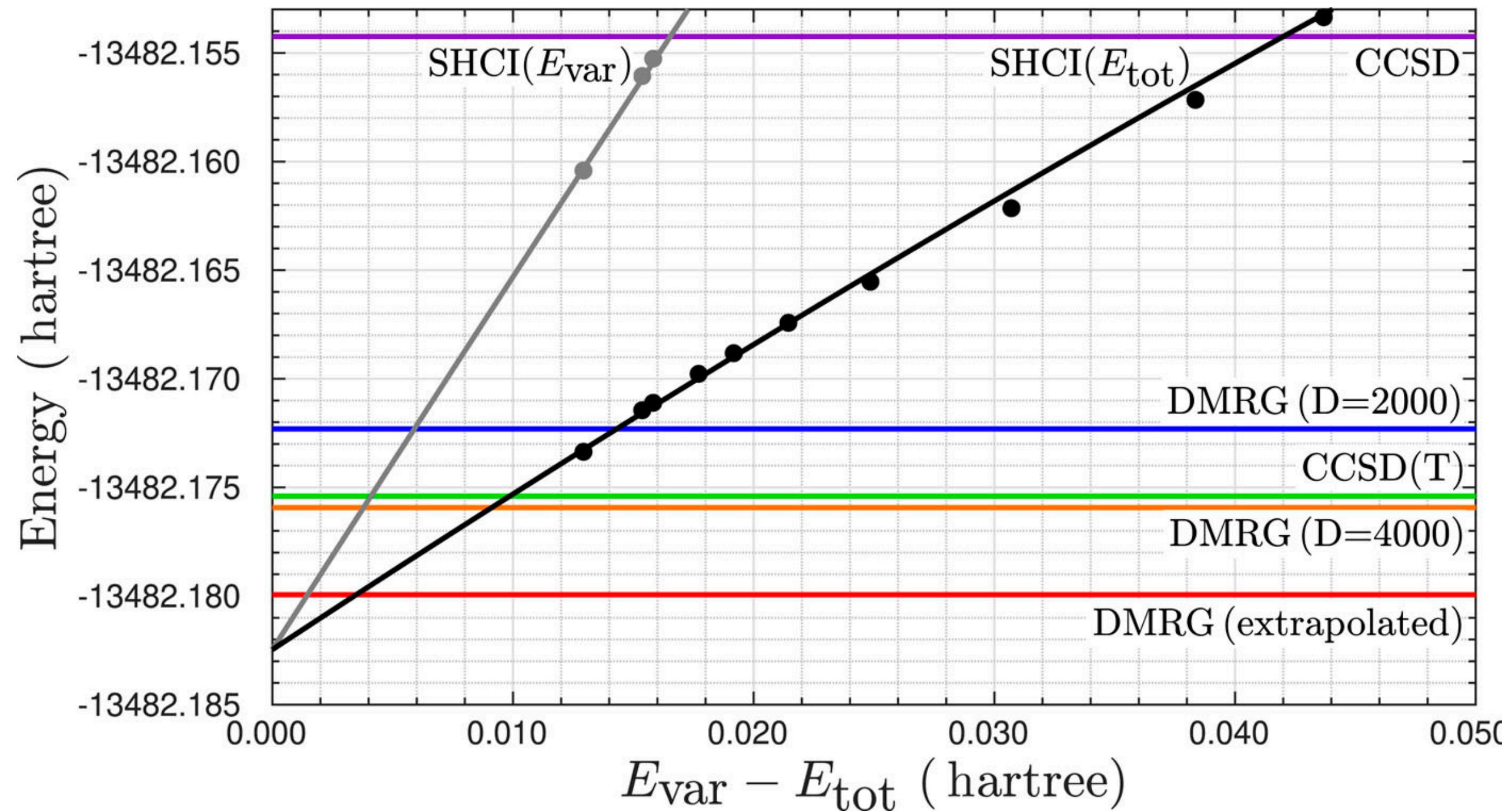


Increased classical processing helps extract better quantum signal from measurements

Better quantum error rates improve solution quality



Looking forward: application to FeMoCo

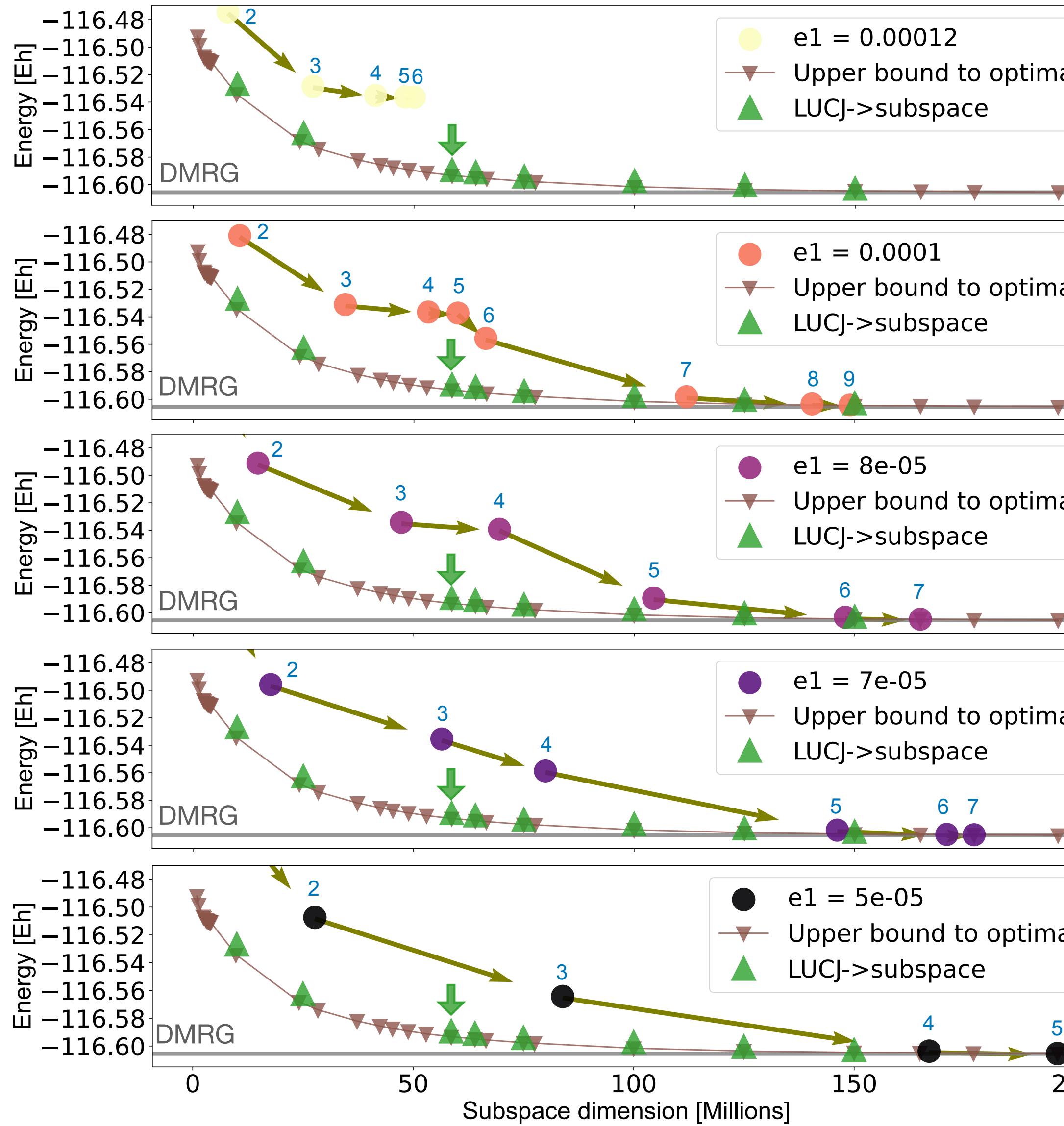


Extrapolations are routinely used in the computational chemistry community to assess performance

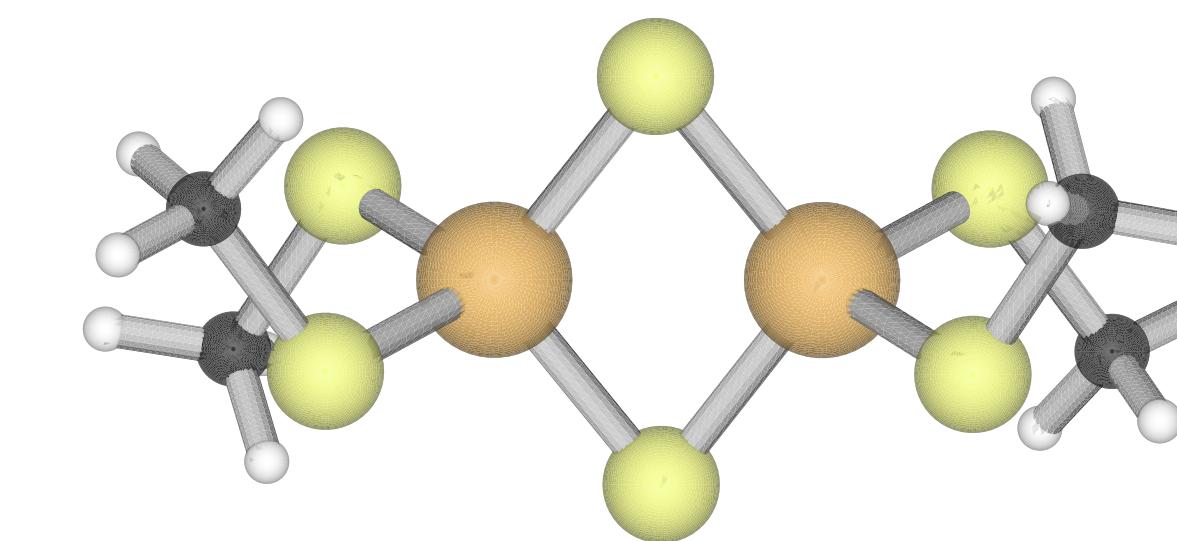
A quantum extrapolation can add value in these scenarios

108 qubits active space

Looking forward: sampling efficiency of LUCJ circuits



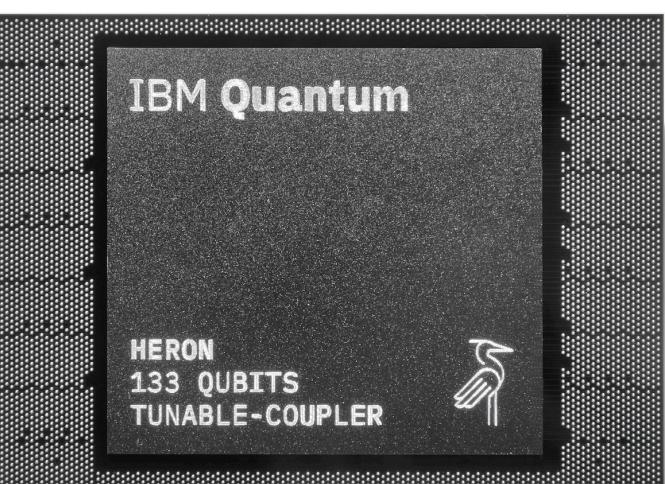
Efficient quantum circuits that
can sample **more efficiently**
than SCI methods exist



45 qubits
3170 quantum gates
1100 two-qubit gates

Main Takeaways

Quantum-centric supercomputing enables realistic use cases beyond problems tailored to the device connectivity. These use cases are believed to be fault-tolerant for quantum computers in isolation



Chemistry is a first: classical processes large classical data associated with chemistry Hamiltonians, quantum executes a few large quantum circuits

Quantum simulations of molecules beyond the reach of exact classical solutions: 77 qubits, 3.5k two-qubit gates with a useful quantum signal

Processing of quantum data at the sample level: no false positive solutions and certifiable advantage

The methods are applicable to any ground state problem that can be captured by sparse wavefunctions

There is value in trying the approach with different input circuits because any solution can be certified and ranked by quality

SQD Workshop

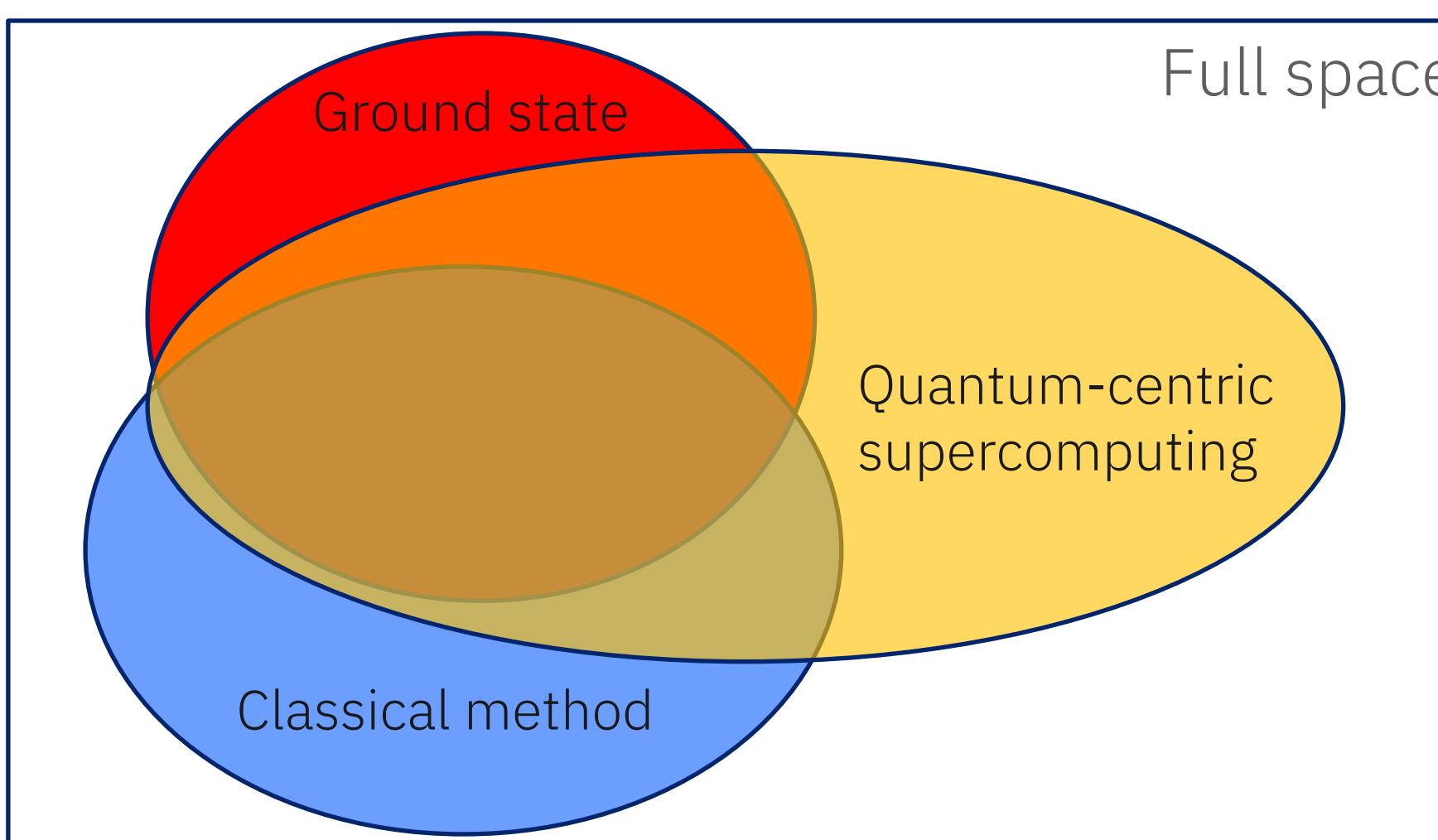
1. Create a new Python virtual environment.
2. `pip install "qiskit-addon-sqd[notebook-dependencies]"`
3. Download the tutorial notebook and run it on your computer (“Download notebook” link on right sidebar).
4. Windows is not supported. If you’re using Windows, you can try using Windows Subsystem for Linux (WSL).

<https://learning.quantum.ibm.com/tutorial/improving-energy-estimation-of-a-fermionic-hamiltonian-with-sqd>



Backup - FAQs

What do we need for quantum advantage in chemistry?



Orange circle: Advantage region

1. The class of ground state problems we consider allows to certify the quality of the individual solutions produced. This permits to certify quantum advantage

2. Obtaining a single quantum point of lower energy than SOTA classical methods – with shorter runtimes and costs - would be a certifiable quantum advantage

3. Extrapolated solutions on an energy-variance plane cannot be certified, however they are routinely used as a benchmark of computational methods

4. If we extrapolate our quantum points, we agree within error bars with state-of-the-art classical certifiable methods (DMRG, HCI) for Fe_2S_2 . This is the best result we have in terms of accuracy

5. Obtaining an extrapolated quantum solution better than extrapolated classical methods would be considered a strong evidence for advantage

How can I use these results?

An open-source implementation is available at

<https://qiskit.github.io/qiskit-addon-sqd/>

pip install qiskit-addon-sqd



Do I need a supercomputer to run these simulations?

We have used a supercomputer to push the limits of quantum and classical hardware together

On a laptop, it is possible to run the largest experiments performed here – Fe_4S_4 for up to 10 M determinants for just one sample batch. This will give one datapoint with a decent accuracy, not at the accuracy level obtained with Fugaku

Are the methods applicable to other use cases?

For the methods to work, it is sufficient that a wavefunction with a polynomial support gives a good approximation of the ground state energy

Any materials science or high-energy physics model that satisfies that assumption will do

Machine learning applications that rely on ground state problems can be explored as well

What primitives/capabilities have we used?

We have only used the sampler primitive throughout

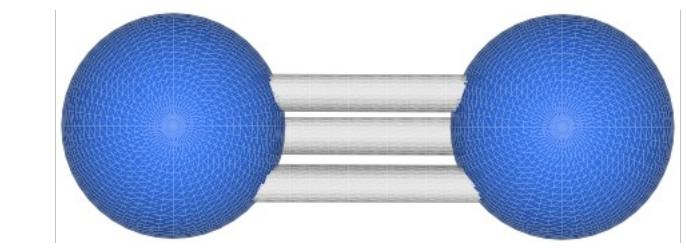
Any improvement in error suppression methods will result in major performance boost

We have used mid-circuit measurements for optimal initialization

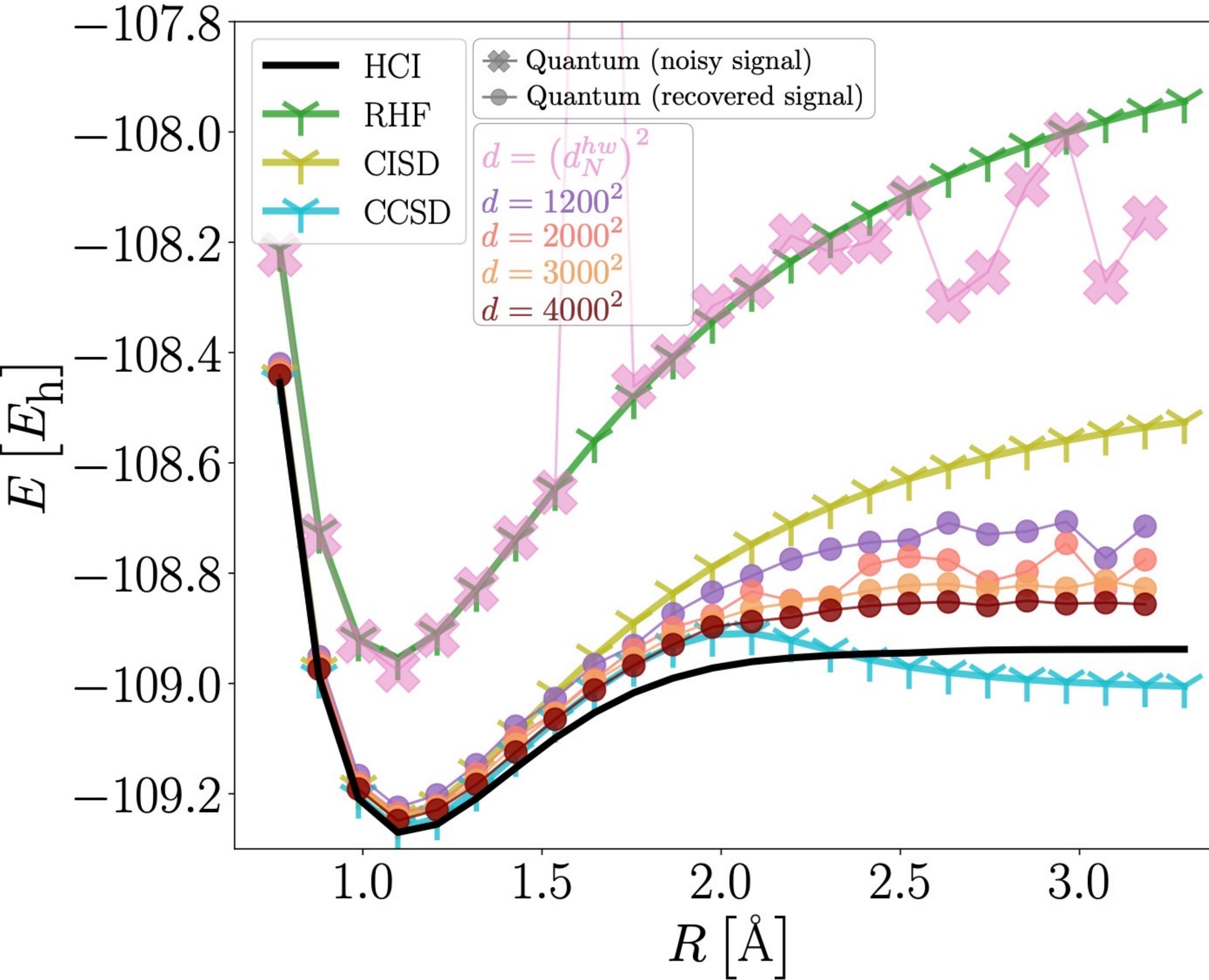
Interactions of the methods presented with other building blocks/techniques will be explored with selected partners and clients in second half of 2024

Backup Slides

Hardware results: noisy signal vs recovered signal



(cc-pVDZ)

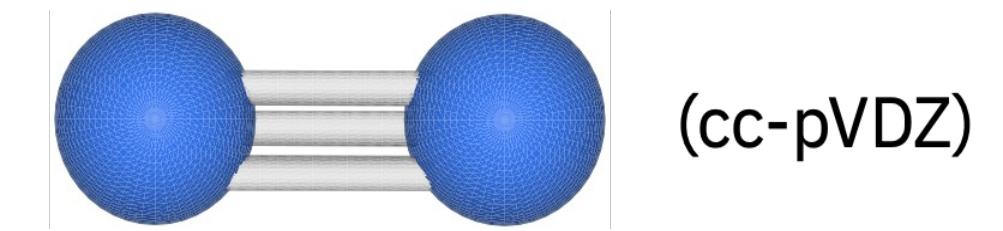


Qubits	CNOT layers	Number of CNOTS	Total number of gates (not counting DD)
52	108	~1760	~5176

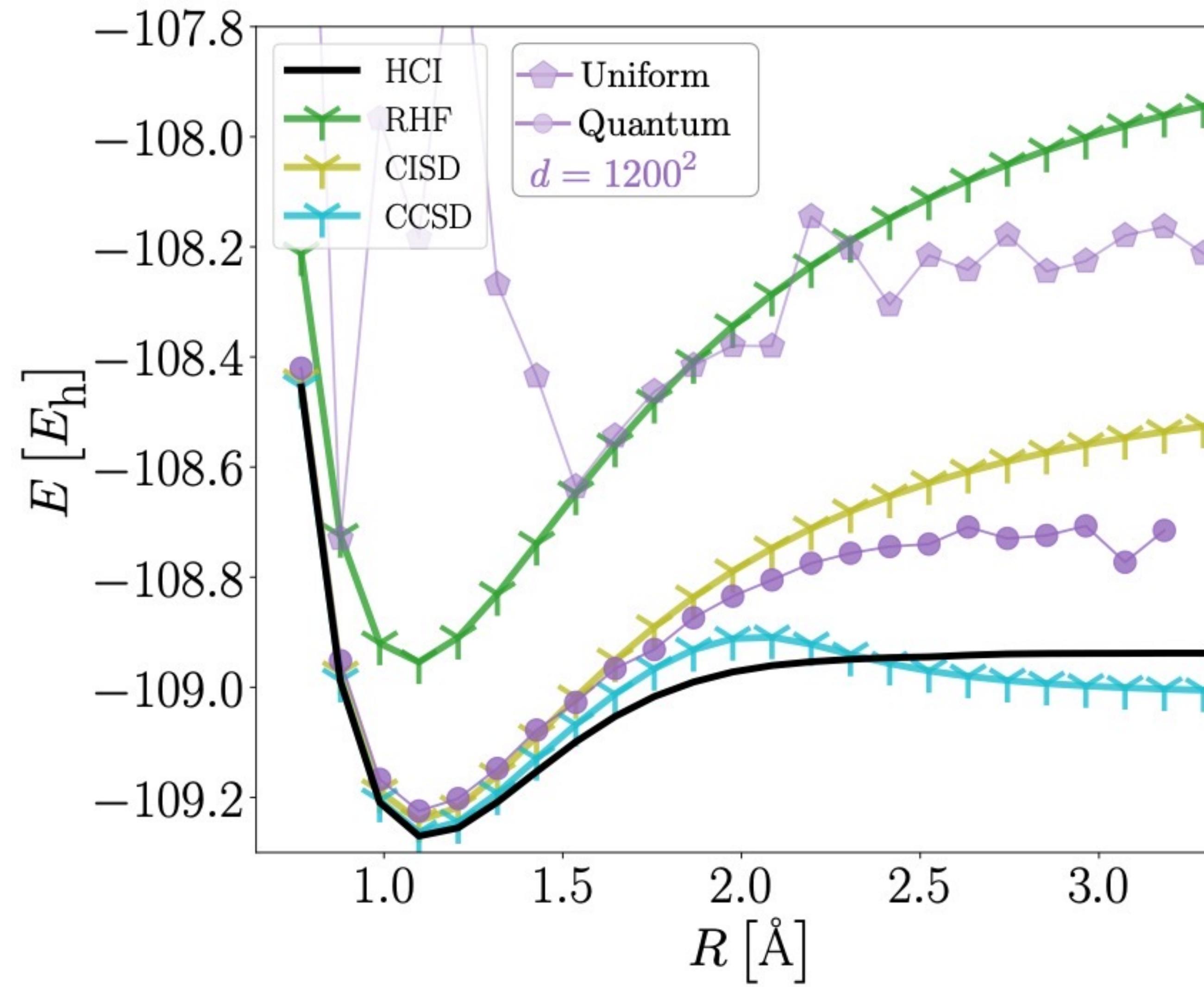
50k from Quantum \rightarrow 2.5 B sampled configs. per point in the dissociation curve

More Quantum and classical processing \rightarrow higher accuracy

Hardware results: Estimator Quantum vs Estimator uniform



of samples from uniform = # of samples on quantum hardware



Uniform in right particle sector

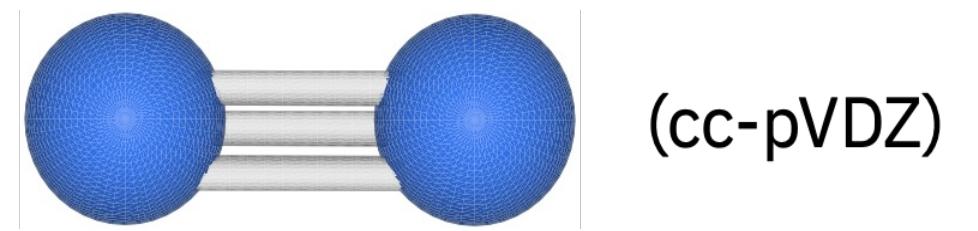
50k from Uniform \rightarrow 2.5 B sampled configs. per point in the dissociation curve

Qubits	CNOT layers	Number of CNOTS	Total number of gates (not counting DD)
52	108	~1760	~5176

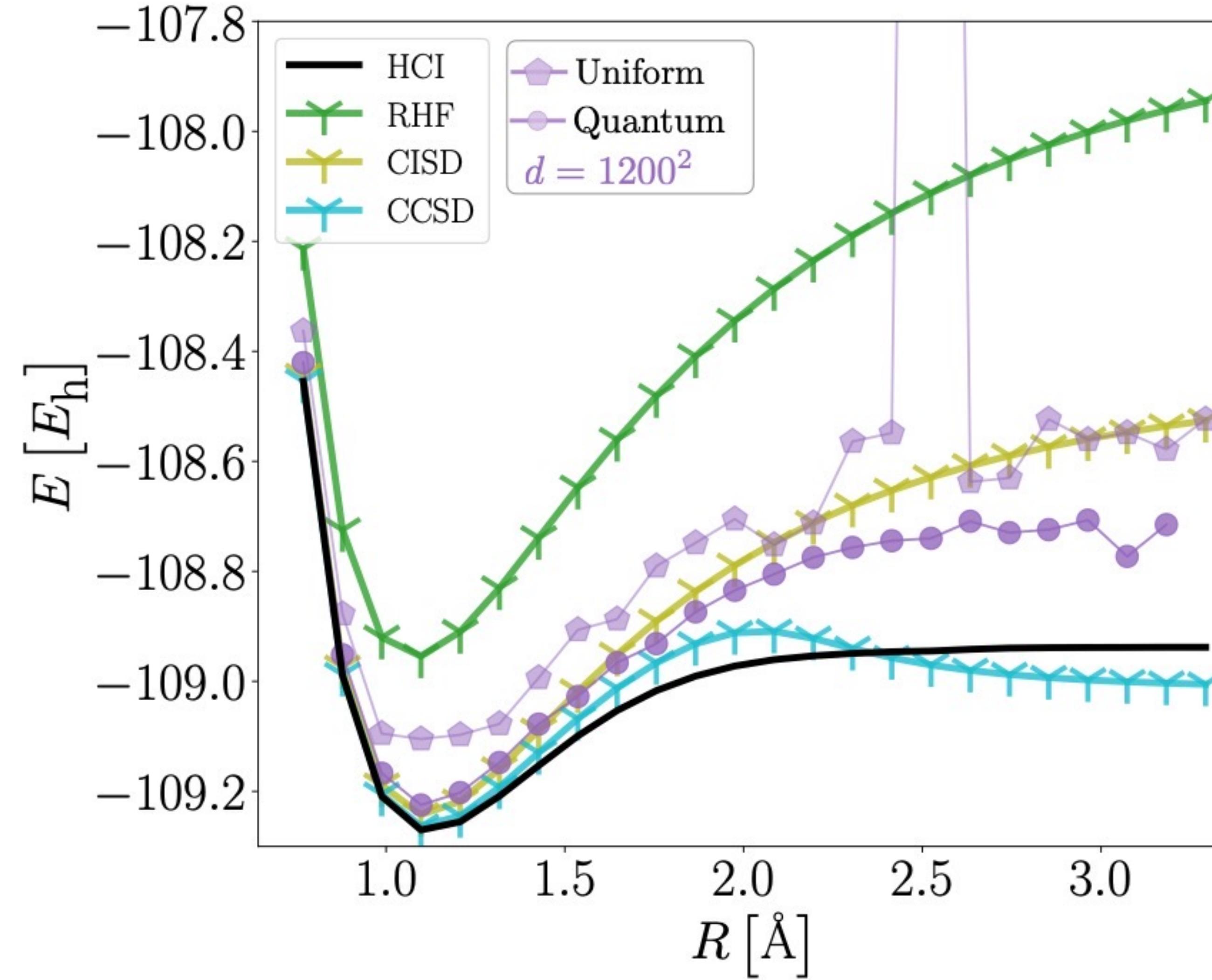
50k from Quantum \rightarrow 2.5 B sampled configs. per point in the dissociation curve

Useful signal from quantum

Hardware results: Estimator Quantum vs Estimator uniform



of samples from uniform = $50 \times$ (# of samples on quantum hardware)



Uniform over Fock space

2.5M from Uniform \rightarrow 2500 B sampled configs. per point in the dissociation curve

Qubits	CNOT layers	Number of CNOTS	Total number of gates (not counting DD)
52	108	~1760	~5176

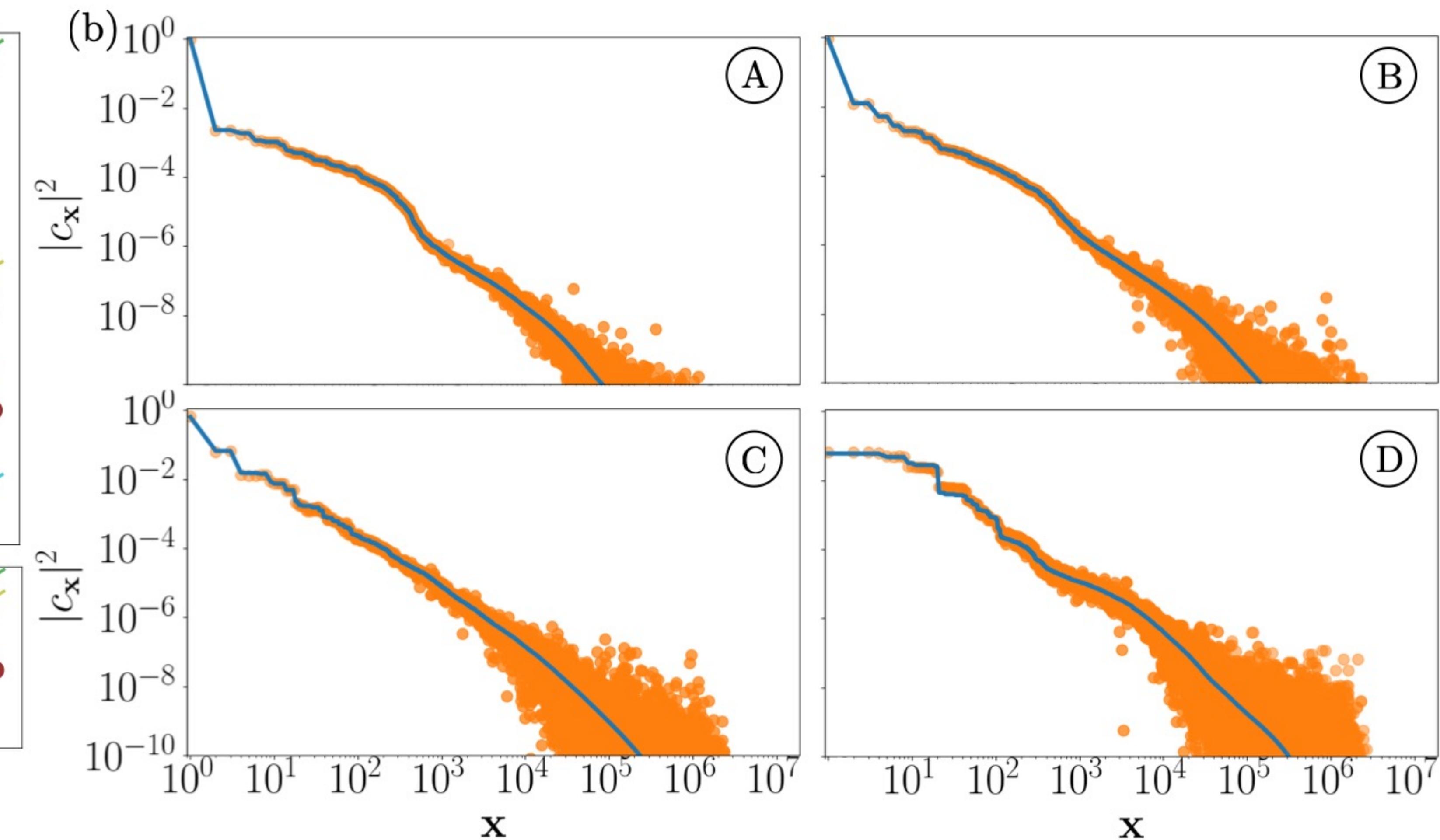
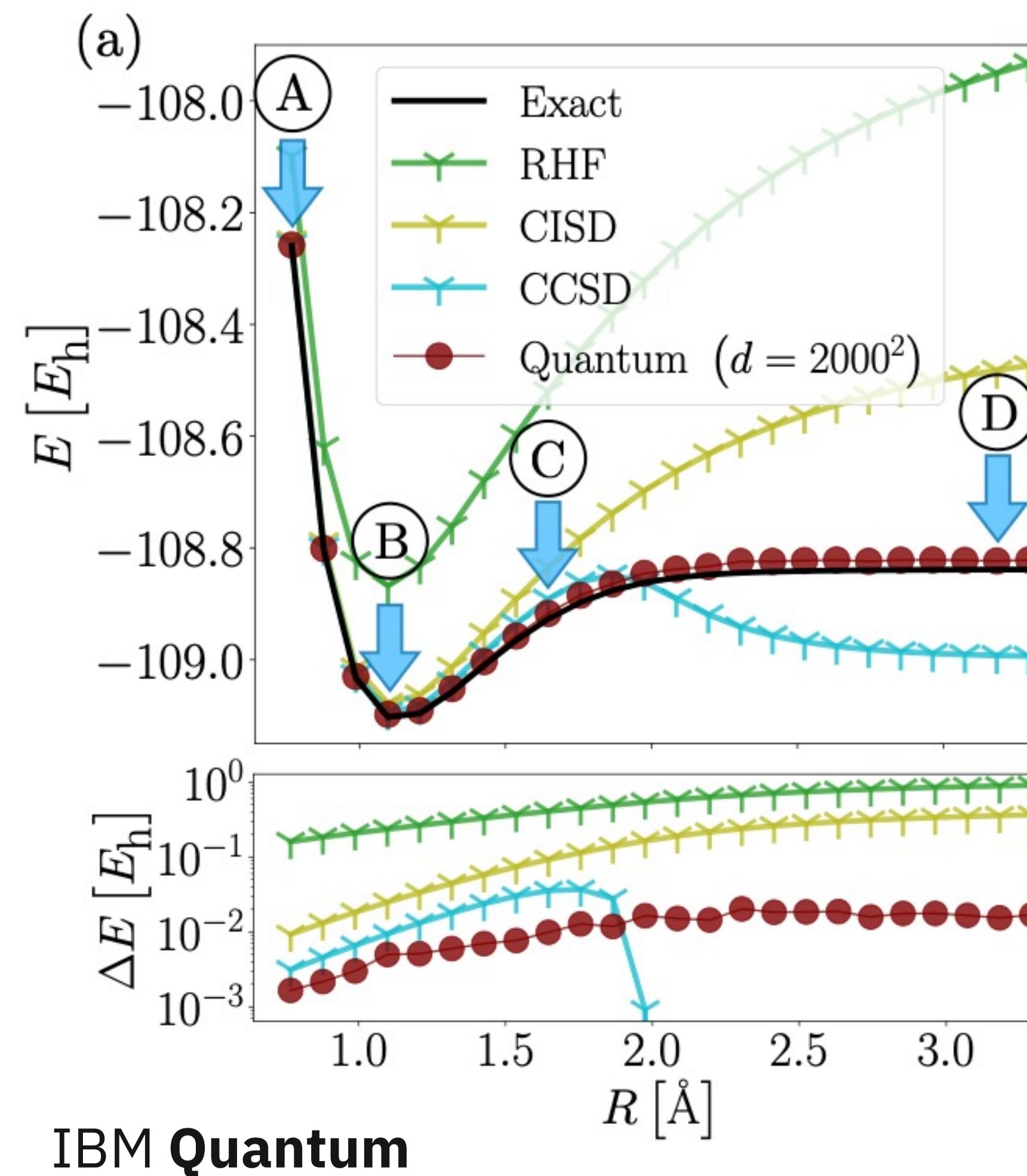
50k from Quantum \rightarrow 2.5 B sampled configs. per point in the dissociation curve

Useful signal from quantum

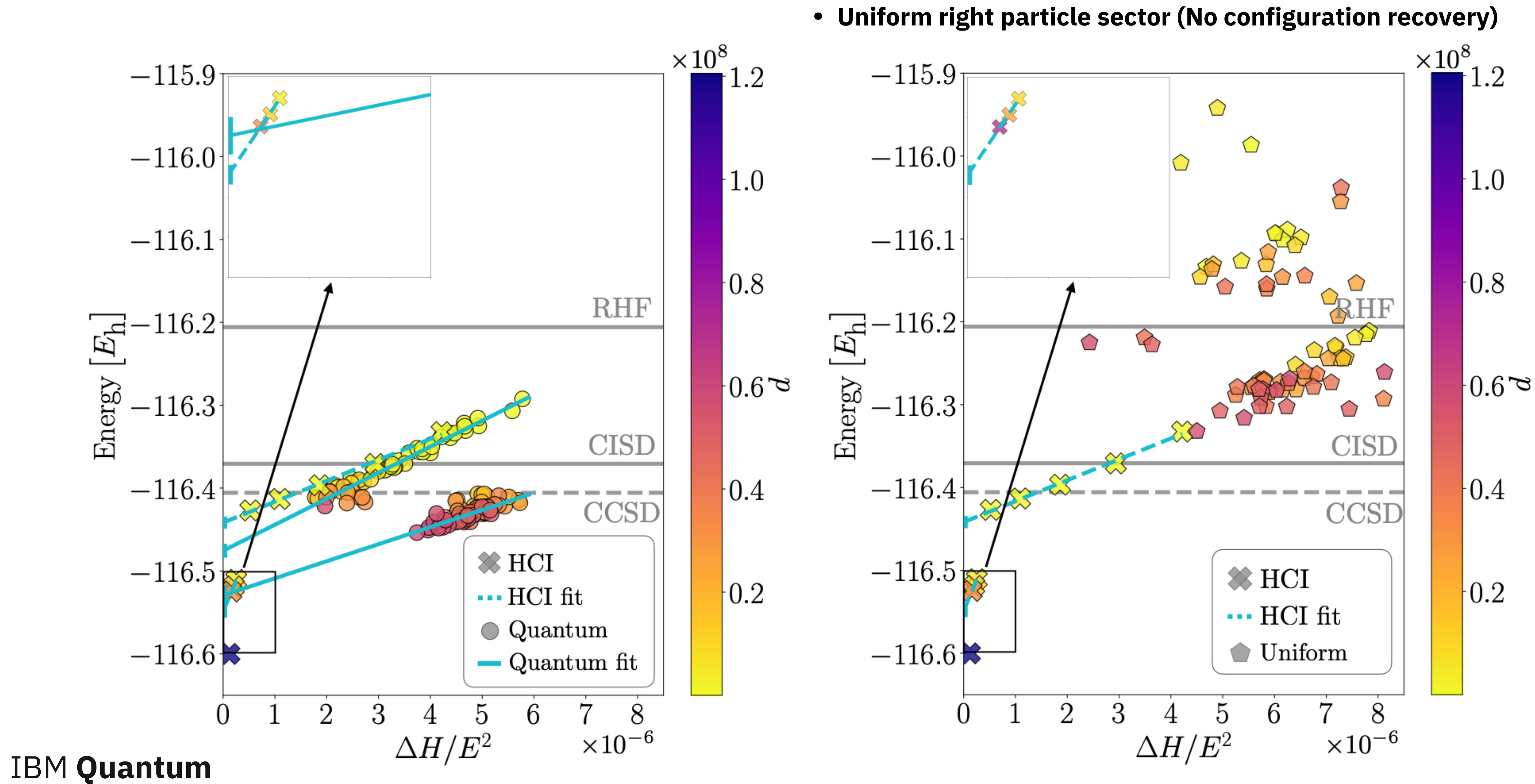
Hardware results: visualization of the wave function amplitudes

- Dissociation of N₂ in the 6-31g basis (32 qubits on Eagle)
- Compare the exact ground state wave function amplitudes with the HPC quantum estimator amplitudes

— Exact ground state
● HPC quantum estimator



Hardware results: estimator on quantum VS estimator on uniform



Hardware results: estimator on quantum VS estimator on uniform

• Uniform in Fock space

