Scaling to utility sized experiments

Meet the Speaker

Ibrahim Shehzad

Professional background

- Quantum Algorithm Developer, IBM (July 2023 – Present)
  Focusing on execution of quantum algorithms at scale

- Postdoctoral associate, Cornell University (June 2022 - June 2023), with a focus on observational cosmology (Project: https://microdevices.jpl.nasa.gov/news/looking-back-to-the-first-stars-formation/)

Educational background

- PhD in Theoretical Physics, Cornell University (August 2016 - June 2022)
  With a focus on the interface of theoretical gravitational and high energy physics

Contact: Ibrahim.Shehzad@ibm.com

Scaling to utility sized experiments

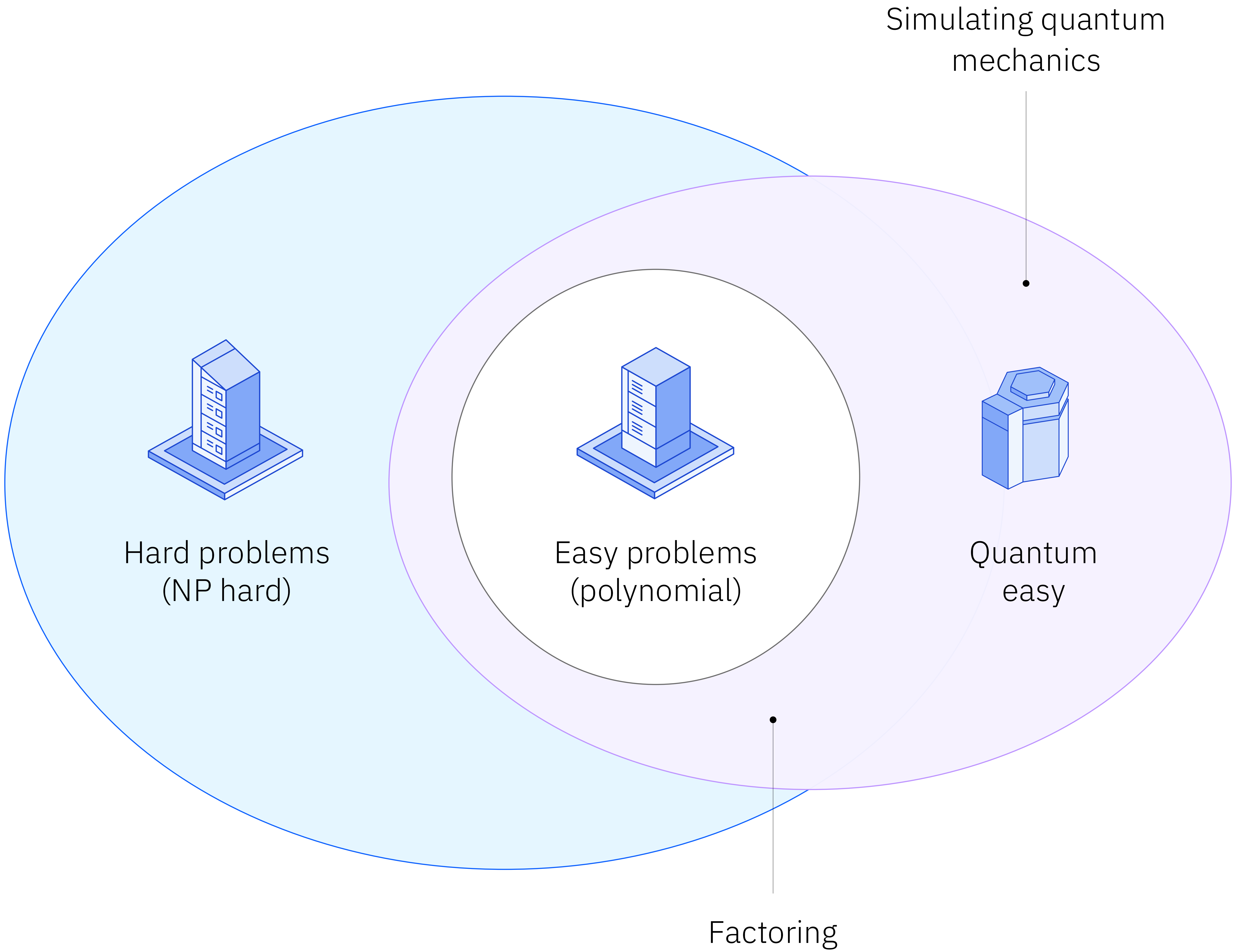
Ibrahim Shehzad
Quantum Algorithm Engineering team
IBM Quantum

Quantum computers: game changers in the field of computing

Quantum computing is not just a faster or better version of classical systems—it is an entirely new branch of computing.

Quantum computing follows the laws of nature to represent data in ways that mimic the randomness and unpredictability of the natural world.

Ultimately, GPUs and classical hardware are not built for this.



Application use cases for quantum computing

Simulating nature

- Batteries
- Solar cells
- Catalysts
- Drug discovery
- High-energy physics

Quantum-centric Supercomputing for Materials Science: A Perspective on Challenges and Future Directions
https://arxiv.org/abs/2312.09733

Quantum Computing for High-Energy Physics: State of the Art and Challenges
https://arxiv.org/abs/2307.03236

Mathematics and processing data with complex structure

- Factorization
- Unstructured search
- Classification problems

Towards quantum-enabled cell-centric therapeutics
https://arxiv.org/abs/2307.05734

Search and optimization

- Logistics
- Electronic design
- Finance

Quantum optimization: Potential, challenges, and the path forward
https://arxiv.org/abs/2312.02279

Quantum Algorithms: within reach

Grover's Search

ML · Optimization

General search algorithm with polynomial speed-up.

Caveats: requires quantum oracle construction

Quantum Phase Estimation

Chemistry · Physics

QPE can be used for general eigenvalue problems such as finding energy states.

HHL

ML · Physics

Useful for solving problems described by a linear system of equations, such as in machine learning and engineering.

Caveats: only sparse systems

Quantum Fourier Transform

General subroutine used in other quantum algorithms (e.g. Schor) for Fourier analysis and period-finding.

Quantum Algorithms: implementable today

Sample-based Quantum Diagonalization (SQD)

Chemistry

Solve eigenvalue problems by using quantum computer to sample from a quantum state, then using classical HPC to recover valid configurations and diagonalize in reduced subspace.

Krylov Subspace Methods

Physics

Solve many-body eigenvalue problems using combination of quantum computer and classical computer with subspace methods.

Variational Quantum Eigensolver (VQE)

Chemistry Physics Optimization

Useful for eigenvalue problems described by or mapped to a Hamiltonian operator

Caveat: poor scaling for large Hamiltonians. Consider SQD.

Time evolution (Trotterization)

Chemistry Physics

Dynamics simulation over time of a quantum mechanical system described by a Hamiltonian operator. Includes real-time and imaginary time (QITE) algorithm variants.

Quantum Kernel Estimation

ML

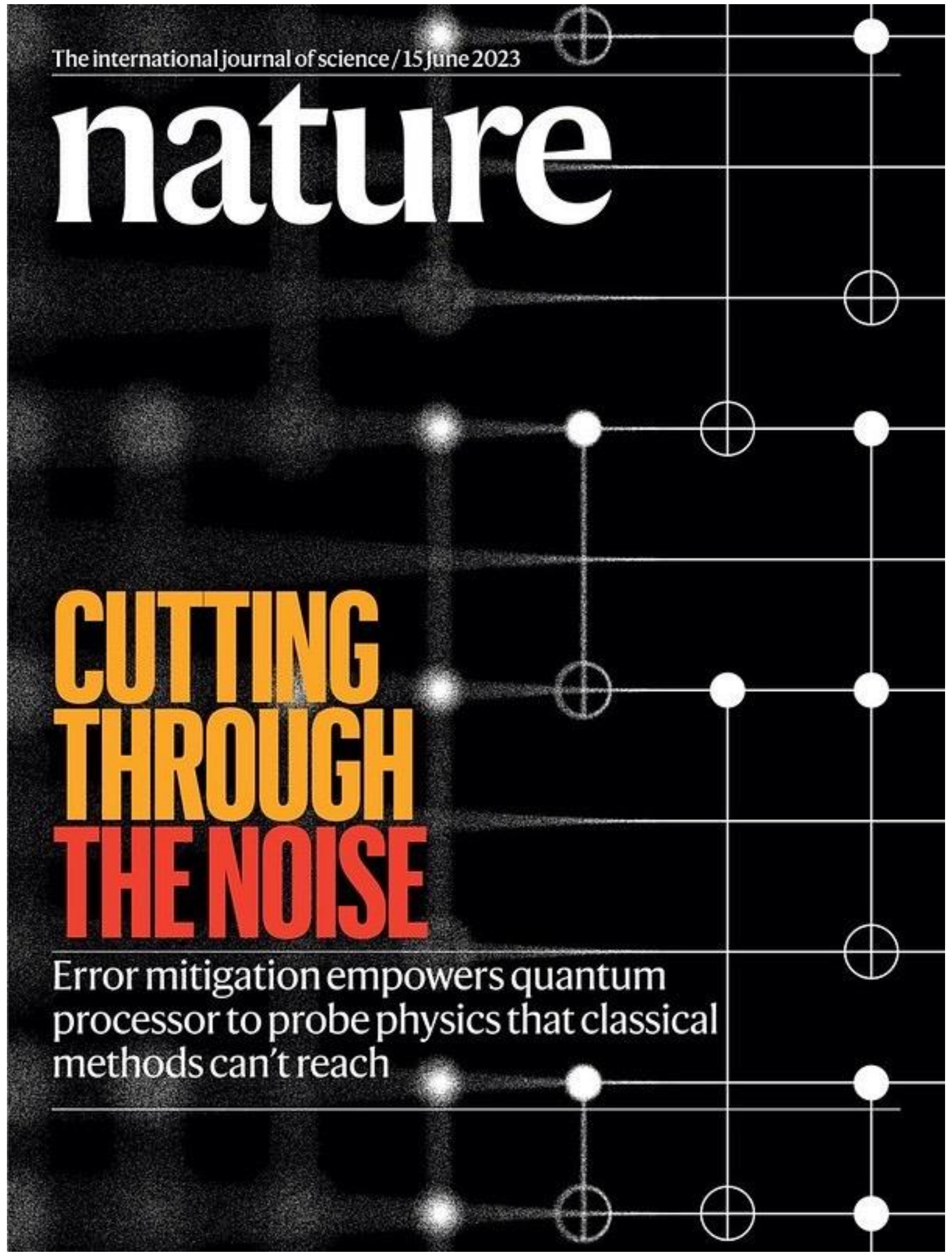
Leverage quantum-enhanced feature space for machine learning methods such as support vector machines and clustering

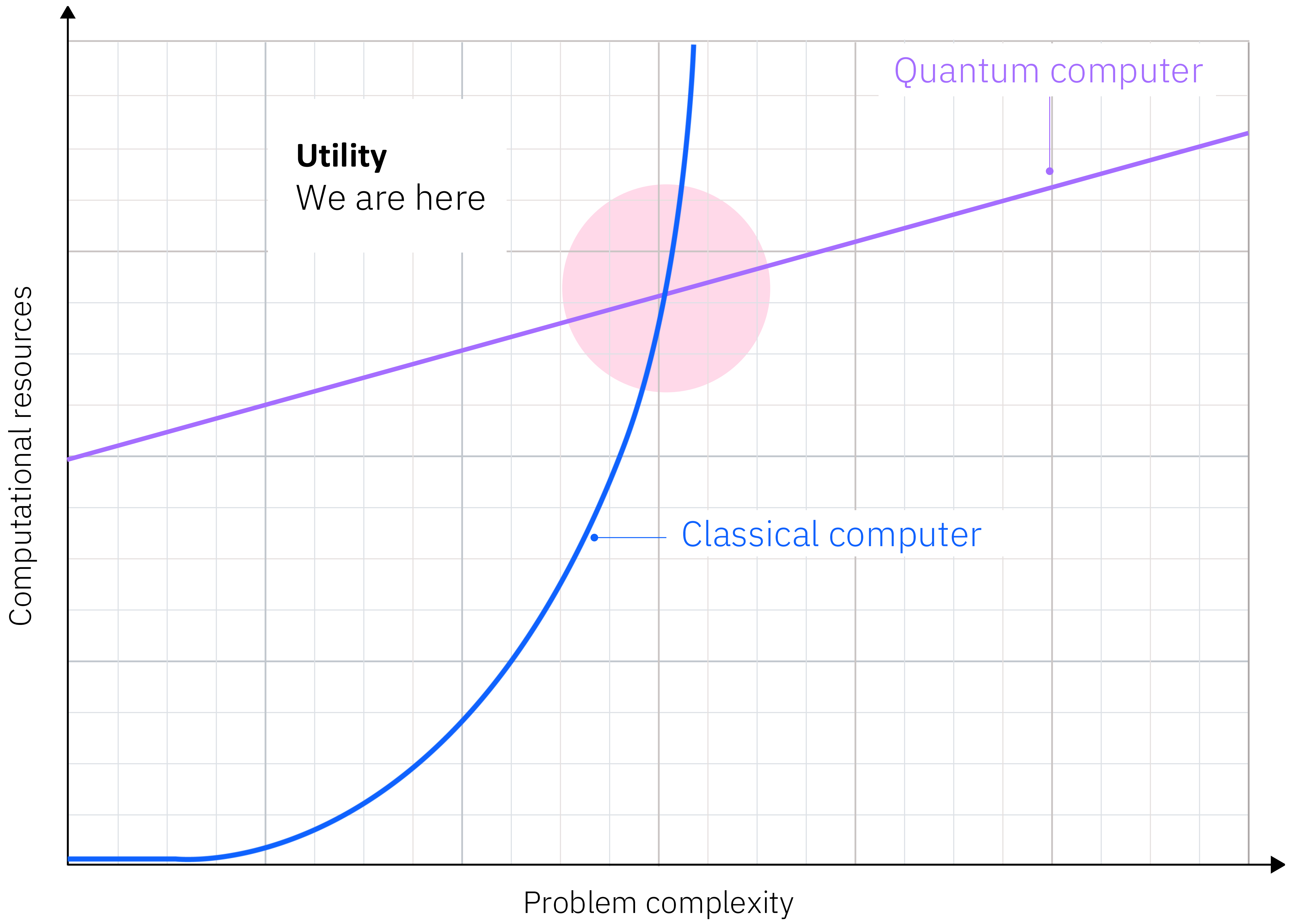
Quantum Approximate Optimization Algorithm (QAOA)

Optimization

Useful for solving combinatorial quadratic optimization problems that can be mapped to the Ising Hamiltonian

The era of quantum utility





A noisy quantum computer produces accurate expectation values on 127 qubits and 2880 gates, outside of brute force classical computation.

Y. Kim, A. Eddins, et al, Nature. 618, 500–505 (2023)

Quantum Utility

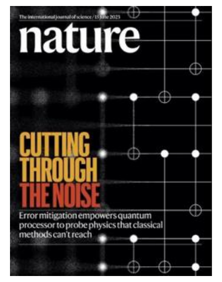
Gates ⇨		3K	5K	100M
	PRE-UTILITY		ERROR MITIGATION	ERROR CORRECTION
Year ⇨		2023	2024	2029

All circuits can be brute force run on classical hardware

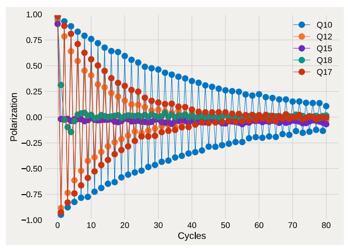
Only approximate classical solutions exists and is the regime for exploring quantum advantage

Era of fault tolerance

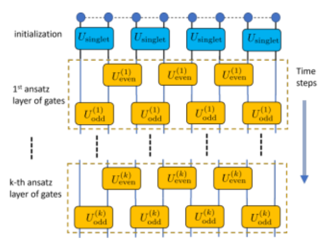
Examples of other utility-scale results (1/2)



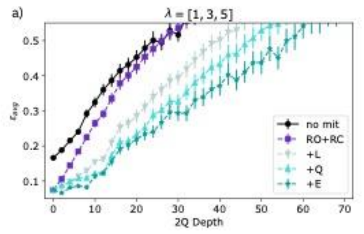
Evidence for the utility of quantum computing before fault tolerance
Nature, **618**, 500 (2023)
127 qubits
simulation



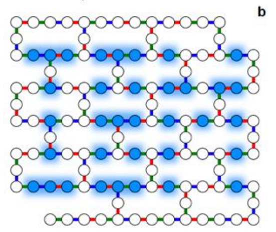
Characterizing quantum processors using discrete time crystals
arXiv:2301.07625
80 qubits
simulation



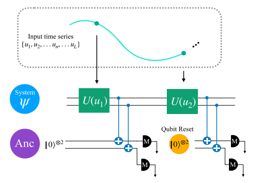
Simulating large-size quantum spin chains on cloud-based superconducting quantum computers
Phys. Rev. Research **5**, 013183 (2023)
102 qubits
simulation



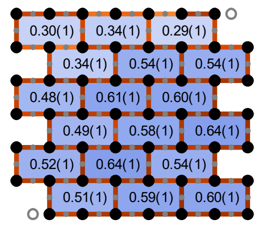
Best practices for quantum error mitigation with digital zero-noise extrapolation
arXiv:2307.05203
104 qubits
tools



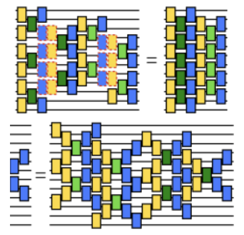
Uncovering Local Integrability in Quantum Many-Body Dynamics
arXiv:2307.07552
124 qubits
simulation



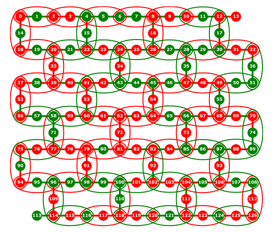
Quantum reservoir computing with repeated Measurements on superconducting devices
arXiv:2310.06706
120 qubits
QML



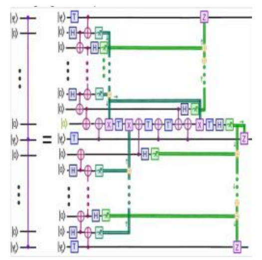
Realizing the Nishimori transition across the error threshold for constant-depth quantum circuits
Nature Physics, 21, 161-167 (2025)
125 qubits
simulation



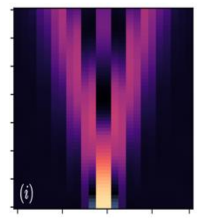
Scalable Circuits for Preparing Ground States on Digital Quantum Computers: The Schwinger Model Vacuum on 100 Qubits
PRX Quantum **5**, 020315 (2024)
100 qubits
simulation



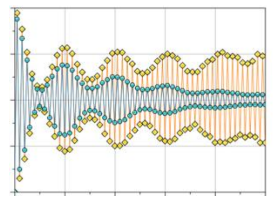
Scaling Whole-Chip QAOA for Higher-Order Ising Spin Glass Models on Heavy-Hex Graphs
arXiv:2312.00997
127 qubits
optimization



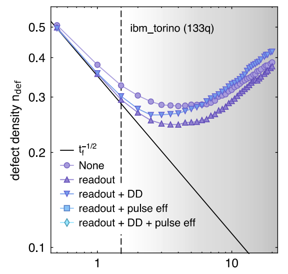
Efficient Long-Range Entanglement using Dynamic Circuits
PRX Quantum **5**, 030339 (2024)
101 qubits
tools



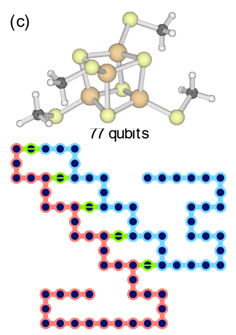
Quantum Simulations of Hadron Dynamics in the Schwinger Model using 112 Qubits
Phys. Rev. D **109**, 114510 (2024)
112 qubits
simulation



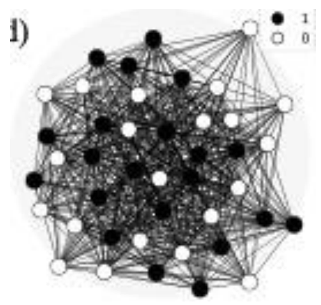
Unveiling clean two-dimensional discrete time quasicrystals on a digital quantum computer
arXiv:2403.16718
133 qubits
simulation



Benchmarking digital quantum simulations and optimization above hundreds of qubits using quantum critical dynamics
arXiv:2404.08053
133 qubits
simulation

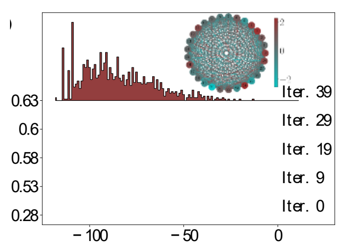


Chemistry Beyond Exact Solutions on a Quantum-Centric Supercomputer
arXiv:2405.05068
77 qubits
simulation

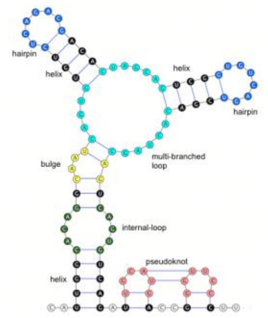


Towards a universal QAOA protocol: Evidence of quantum advantage in solving combinatorial optimization problems
arXiv:2405.09169
109 qubits
optimization

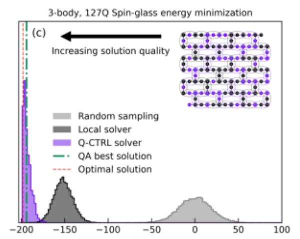
Examples of other utility-scale results (2/2)



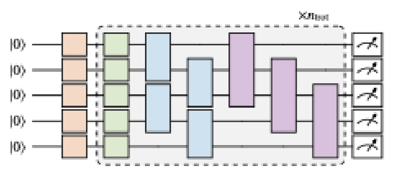
Bias-field digitized counteradiabatic quantum optimization
arXiv:2405.13898
100 qubits
optimization



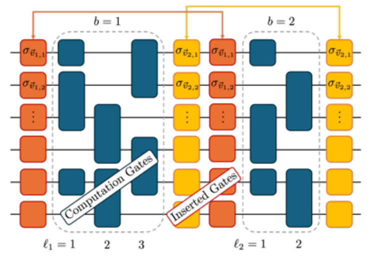
mRNA secondary structure prediction using utility-scale quantum computers
arXiv:2405.20328
80 qubits
simulation



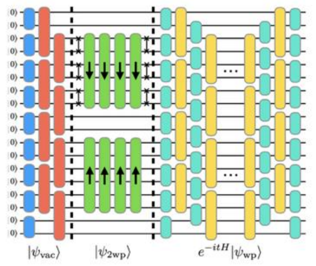
Quantum optimization using a 127-qubit gate-model IBM quantum computer can outperform quantum annealers for nontrivial binary optimization problems
arXiv:2406.01743
127 qubits
optimization



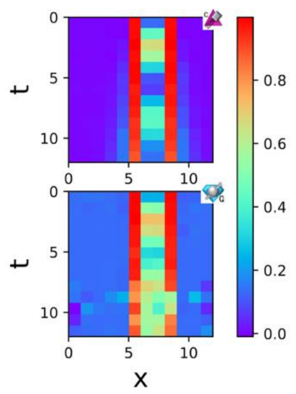
Bias-Field Digitized Counterdiabatic Quantum Algorithm for Higher-Order Binary Optimization
arXiv:2409.04477
156 qubits
optimization



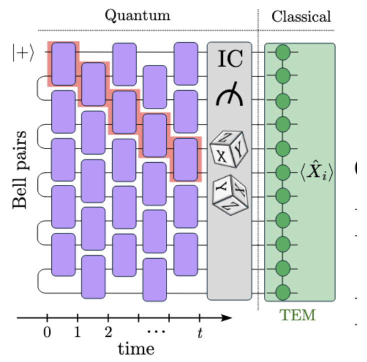
Robust Error Accumulation Suppression for Quantum Circuits
arXiv:2401.16884
100 qubits
tools



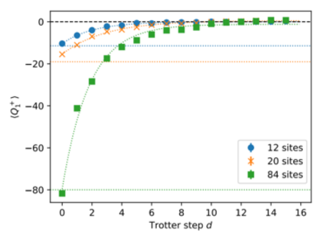
Scalable Quantum Simulations of Scattering in Scalar Field Theory on 120 Qubits
arXiv:2411.02486
120 qubits
simulation



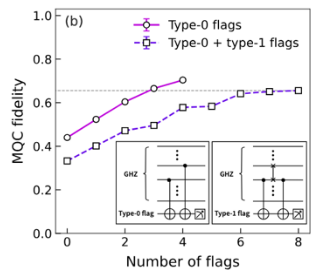
String Breaking in the Heavy Quark Limit with Scalable Circuits
arXiv:2411.05915
104 qubits
simulation



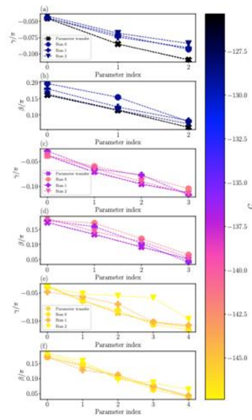
Dynamical simulations of many-body quantum chaos on a quantum computer
arXiv:2411.00765
91 qubits
simulation



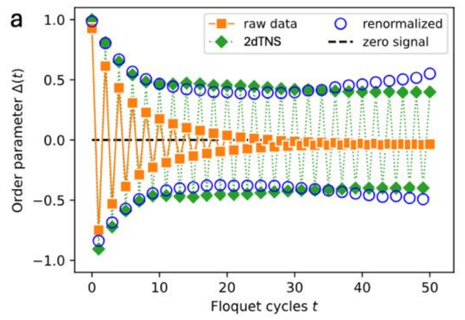
Conserved charges in the quantum simulation of integrable spin chains
arXiv:2208.00576
84 qubits
simulation



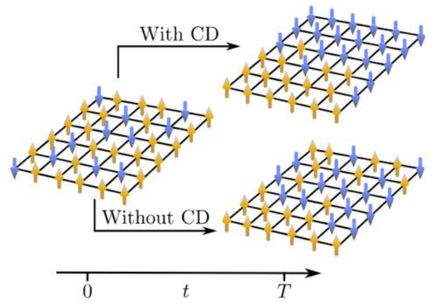
Achieving computational gains with quantum error correction primitives: Generation of long-range entanglement enhanced by error detection
arXiv:2411.14638
75 qubits
tools



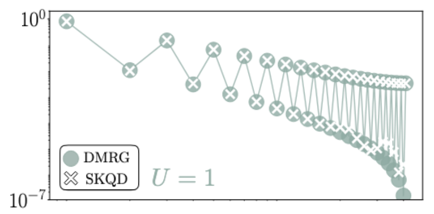
Efficient Online Quantum Circuit Learning with No Upfront Training
arXiv:2501.04636
127 qubits
tools



Realization of Two-dimensional Discrete Time Crystals with Anisotropic Heisenberg Coupling
arXiv:2501.18036
144 qubits
simulation

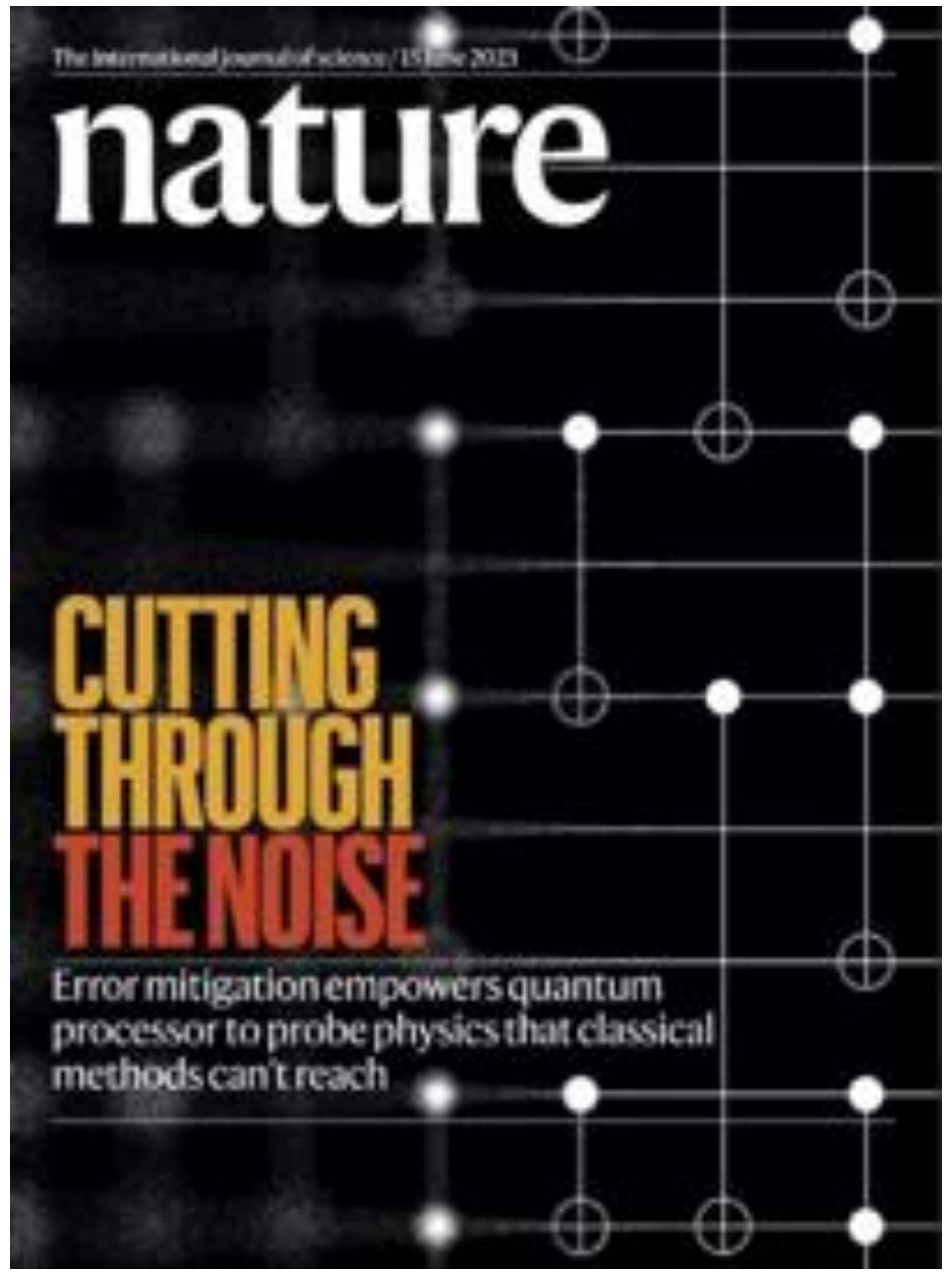


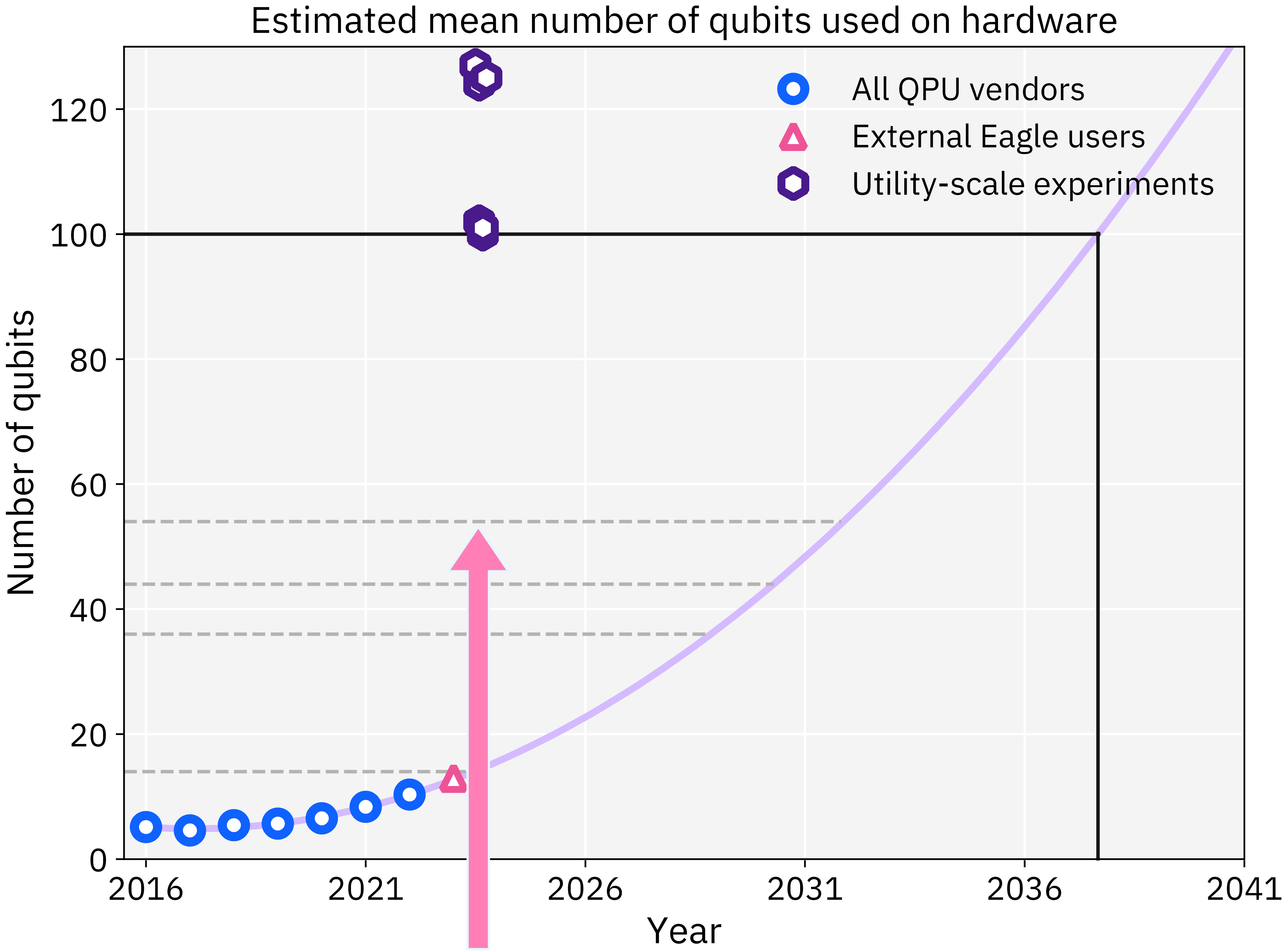
Digitized counterdiabatic quantum critical dynamics
arXiv:2502.15100
156 qubits
optimization



Quantum-Centric Algorithm for Sample-Based Krylov Diagonalization
arXiv:2501.09702
85 qubits
simulation

IBM Quantum Systems + Qiskit = path to utility and on to advantage





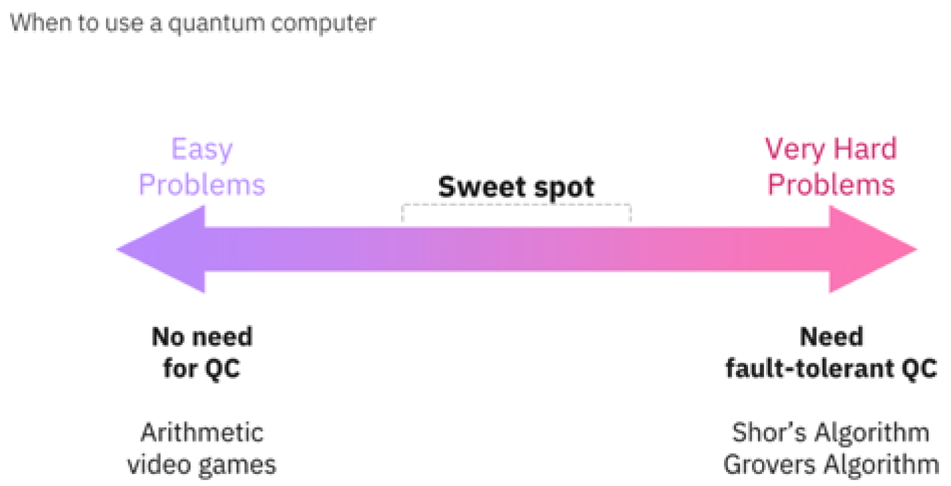
Roadmap for scientific breakthroughs with today's quantum computers

0. Keep scaling in mind.

Scale: necessary to go to a regime where classical computers cannot compete

1. Problem choice

Select the right problems



2. Requirements for useful QC

To get the best out of performant software and hardware, pick efficient mappings to quantum hardware.

```
pass_manager = generate_preset_pass_manager(
optimization_level=3, backend=backend, initial_layout= initial_layout)

isa_circuit = pass_manager.run(circuit)
```

3. Practice makes perfect

To learn execution on quantum hardware: start early and use it extensively.

4. Scaling towards breakthroughs

Going to ~100+ qubits takes us to a regime that is ripe for scientific breakthroughs.

Error mitigation with Qiskit

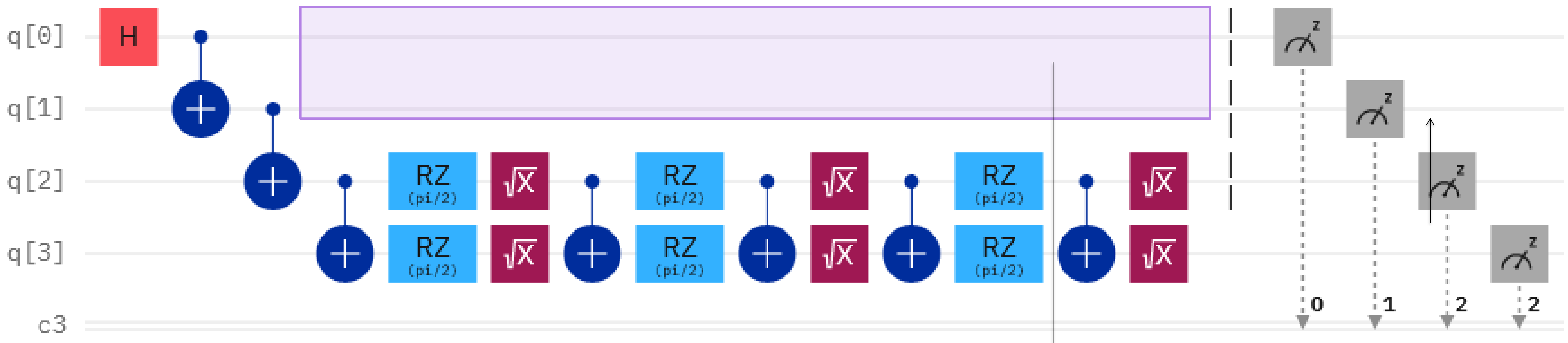
1 Noise in quantum systems

2 Error suppression techniques

3 Error mitigation techniques

4 Combining techniques

Noise in quantum systems



Bath/system coupling

Environmental noise: relevant for deep sparse circuits

ibm_torino

OpenQASM 3

Details

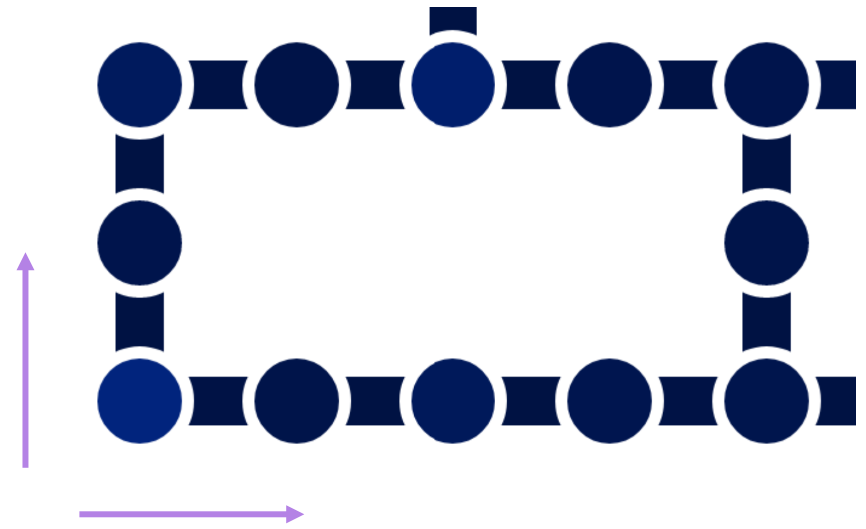
133 Qubits

0.8% EPLG

3.8K CLOPS

Status:	Online
Total pending jobs:	43 jobs
Processor type ⓘ:	Heron r1
Version:	1.0.16
Basis gates:	CZ, ID, RZ, SX, X
Your instance usage:	0 jobs

Median CZ error:	4.384e-3
Median SX error:	3.161e-4
Median readout error:	1.770e-2
Median T1:	175.89 us
Median T2:	134.83 us



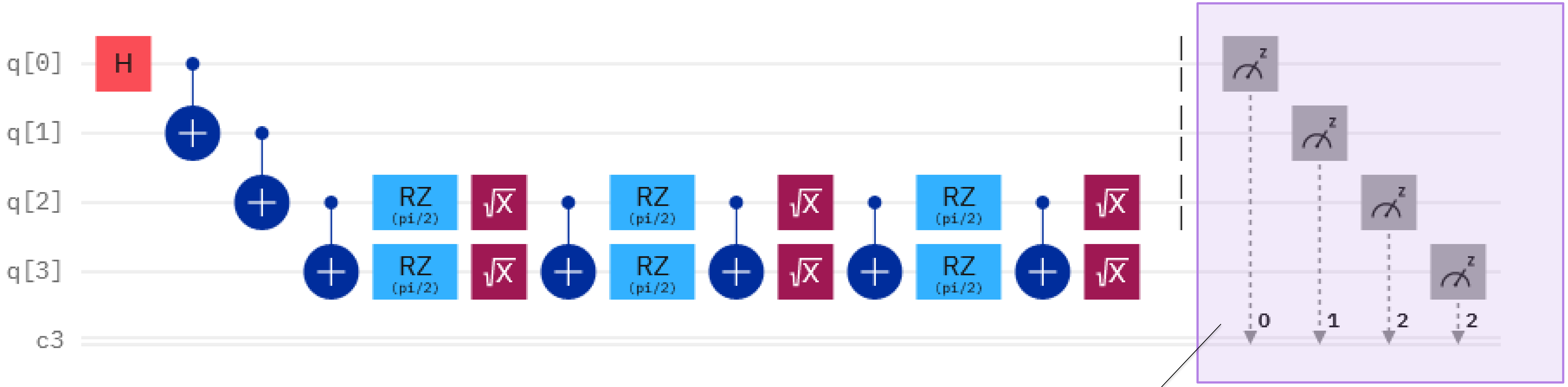
Decoherence

Information loss over time.

Crosstalk

Idle qubits interact with their neighbors. (Mitigated by default in our latest devices!)

Noise in quantum systems



ibm_torino

OpenQASM 3

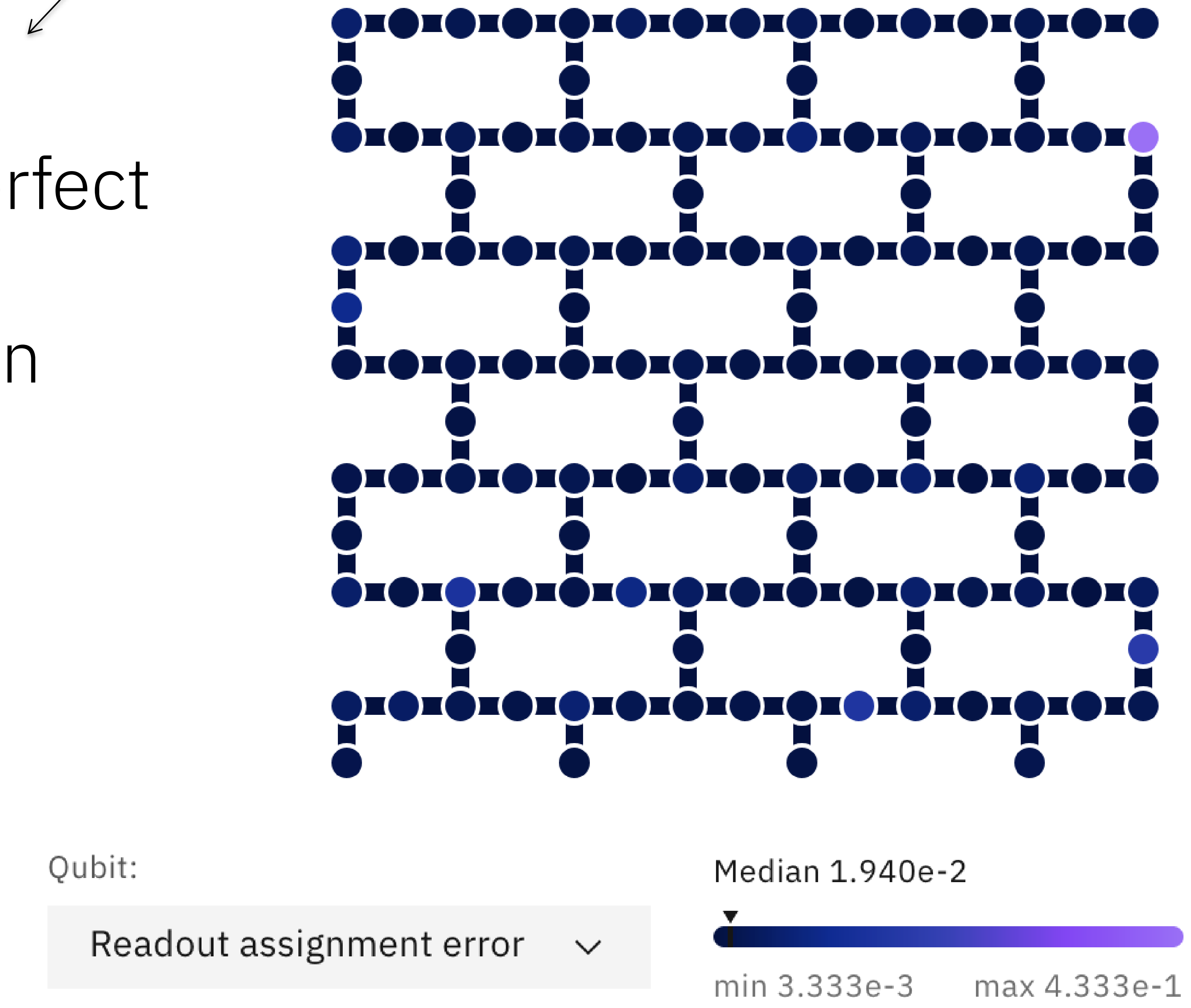
Details

133	Qubits
0.8%	EPLG
3.8K	CLOPS

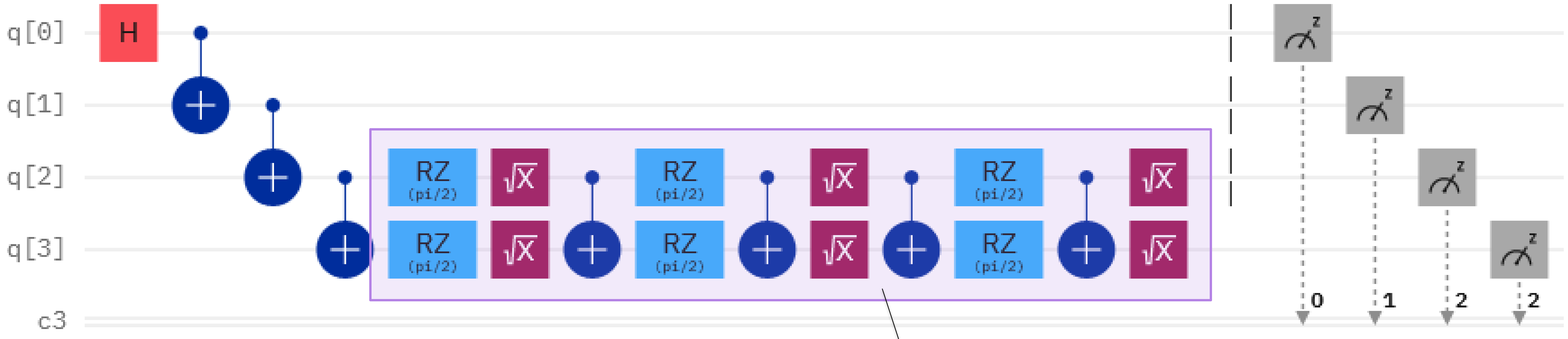
Status:	Online
Total pending jobs:	43 jobs
Processor type ⓘ:	Heron r1
Version:	1.0.16
Basis gates:	CZ, ID, RZ, SX, X
Your instance usage:	0 jobs

Median CZ error:	4.384e-3
Median SX error:	3.161e-4
Median readout error:	1.770e-2
Median T1:	175.89 us
Median T2:	134.83 us

Readout errors because of imperfect measurements. Most important in shallow circuits.



Noise in quantum systems



Gate errors
Because of imperfect operations on qubits.

ibm_torino

OpenQASM 3

Details

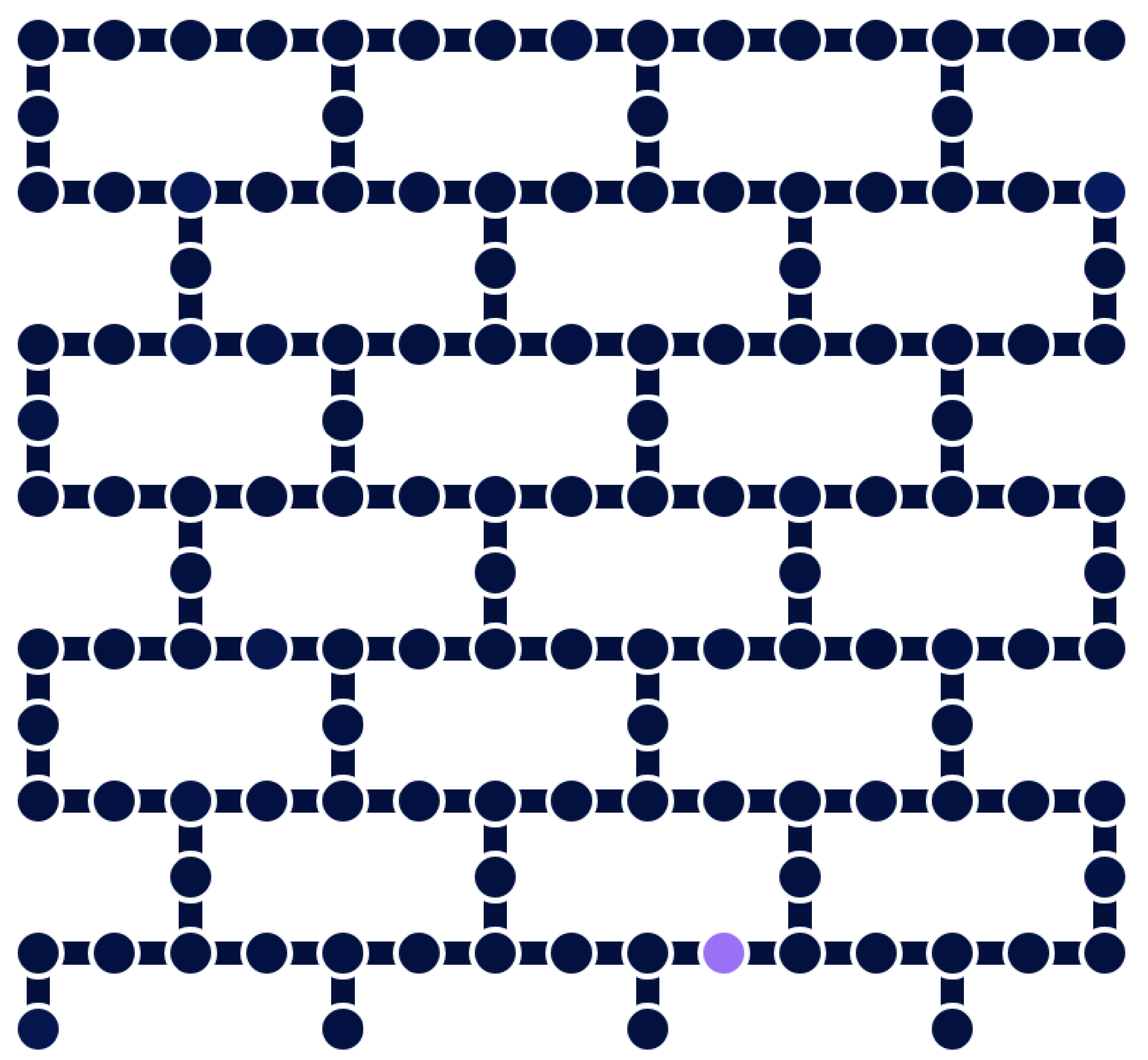
133
Qubits

0.8%
EPLG

3.8K
CLOPS

Status:	Online
Total pending jobs:	43 jobs
Processor type ⓘ:	Heron r1
Version:	1.0.16
Basis gates:	CZ, ID, RZ, SX, X
Your instance usage:	0 jobs

Median CZ error:	4.384e-3
Median SX error:	3.161e-4
Median readout error:	1.770e-2
Median T1:	175.89 us
Median T2:	134.83 us



Qubit: √x (sx) error
Median 3.299e-4
min 1.335e-4 max 8.469e-2

Connection: CZ error
Median 1.000e+0
min 1.000e+0 max 1.000e+0

Fighting noise in quantum systems

Before fault tolerance...

Error suppression techniques minimize the impact of noise by either preventing errors from happening or modifying the noise structure.

- Dynamical decoupling (DD)
- Pauli Twirling (PT)

Before or during execution (typically)

Additional classical resources dominate

Error mitigation techniques allow errors to occur and reduce their effect by modeling the device noise that was present at the time of execution.

- Twirled Readout Error eXtinction (TREX)
- Zero Noise Extrapolation (ZNE)

After or during execution (typically)

Additional quantum resources dominate

Fault tolerance

Error correction builds up redundancies that allow us to detect and correct errors when they occur, leading to essentially error-free logical qubits!

"Well, your quantum computer is broken in every way possible simultaneously."

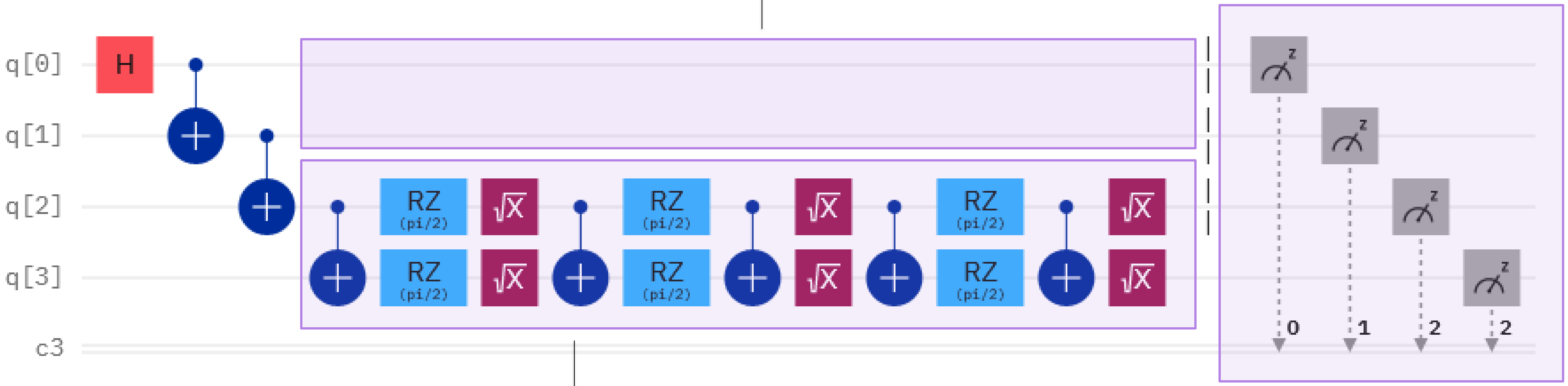
https://www.ibm.com/quantum/blog/quantum-error-suppression-mitigation-correction

Fighting noise in quantum systems

Dynamical decoupling (DD)

Readout errors

Environmental noise



Gate errors

Twirled Readout Error eXtinction (TREX)

Pauli Twirling (PT)

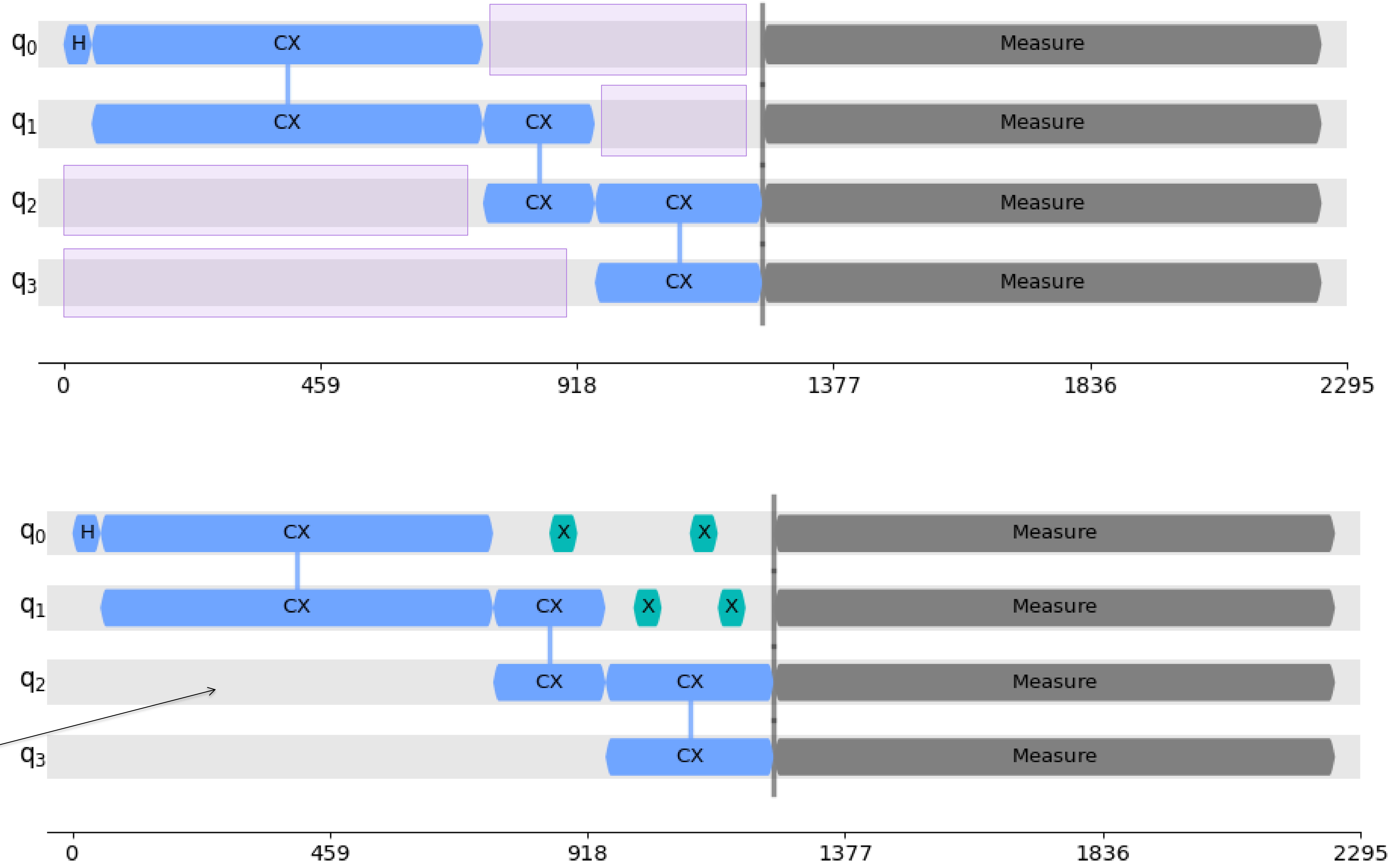
Zero Noise Extrapolation (ZNE)

Dynamical decoupling (DD) (see also https://youtu.be/67jRWQuW3Fk?si=ejmSCg-01NnCxTFk)

Crosstalk can be suppressed by applying gates which add up to the identity.

Such gates will also introduce errors, so there is a balance to be found.

No need to apply gates to qubits which are not initialized.



Dynamical decoupling (DD)

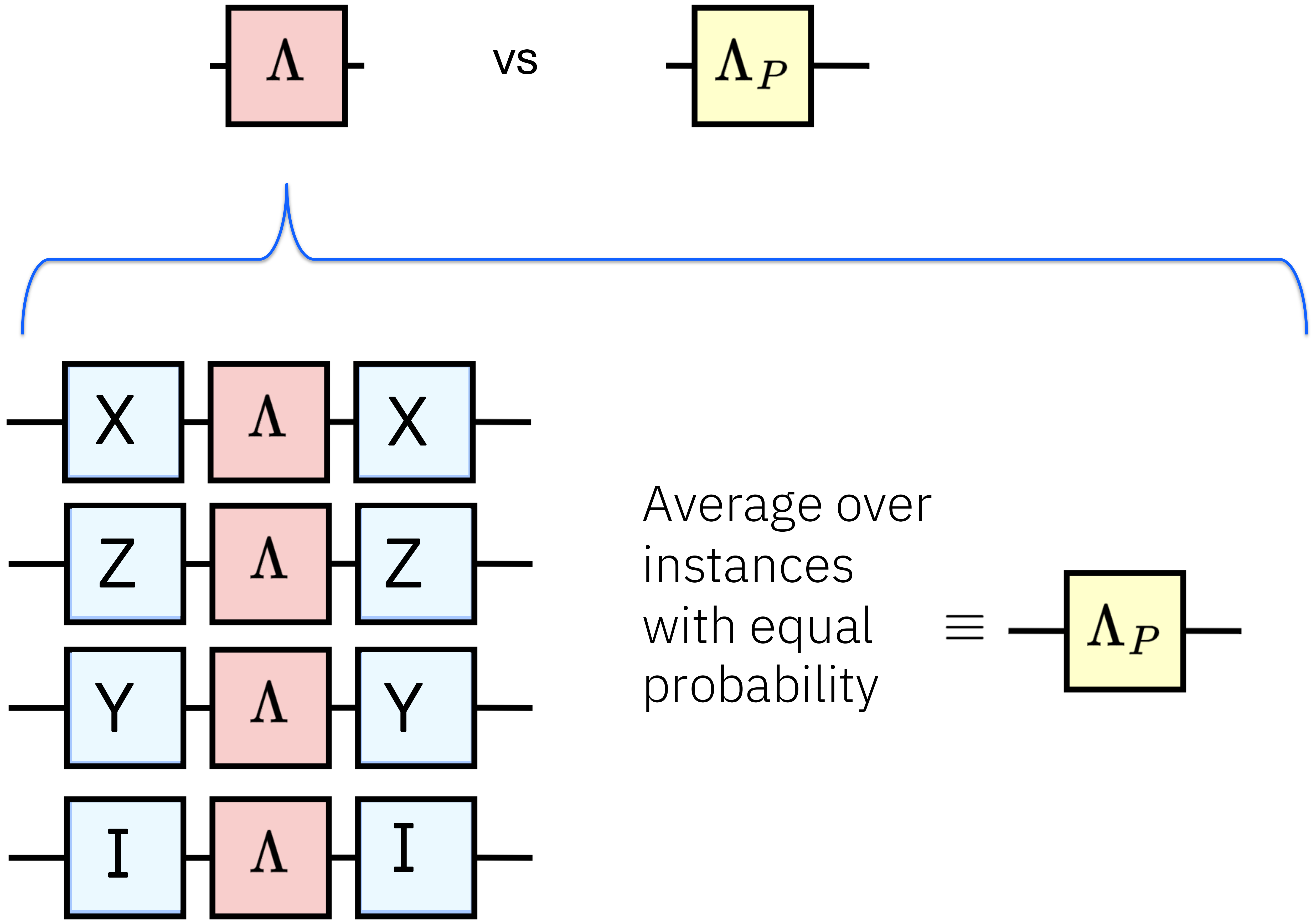
Options	Sub-options	Sub-sub-options	Choices	Default
dynamical_decoupling	enable		True / False	False
	sequence_type		'XX' / 'XpXm' / 'XY4'	'XX'
	extra_slack_distribution		'middle' / 'edges'	'middle'
	scheduling_method		'asap' / 'alap'	'alap'

```
from qiskit_ibm_runtime import SamplerOptions, EstimatorOptions

options = SamplerOptions(default_shots=1024)  # or...
options = EstimatorOptions(default_shots=1024, optimization_level=0, resilience_level=0)

## Configure Dynamical Decoupling
options.dynamical_decoupling.enable = True
options.dynamical_decoupling.sequence_type = 'XX'
options.dynamical_decoupling.extra_slack_distribution = 'middle'
options.dynamical_decoupling.scheduling_method = 'alap'
```


https://docs.quantum.ibm.com/api/qiskit-ibm-runtime/qiskit_ibm_runtime.options.DynamicalDecouplingOptions

Pauli Twirling (PT)

Arbitrary noise channels can be converted into Pauli noise.



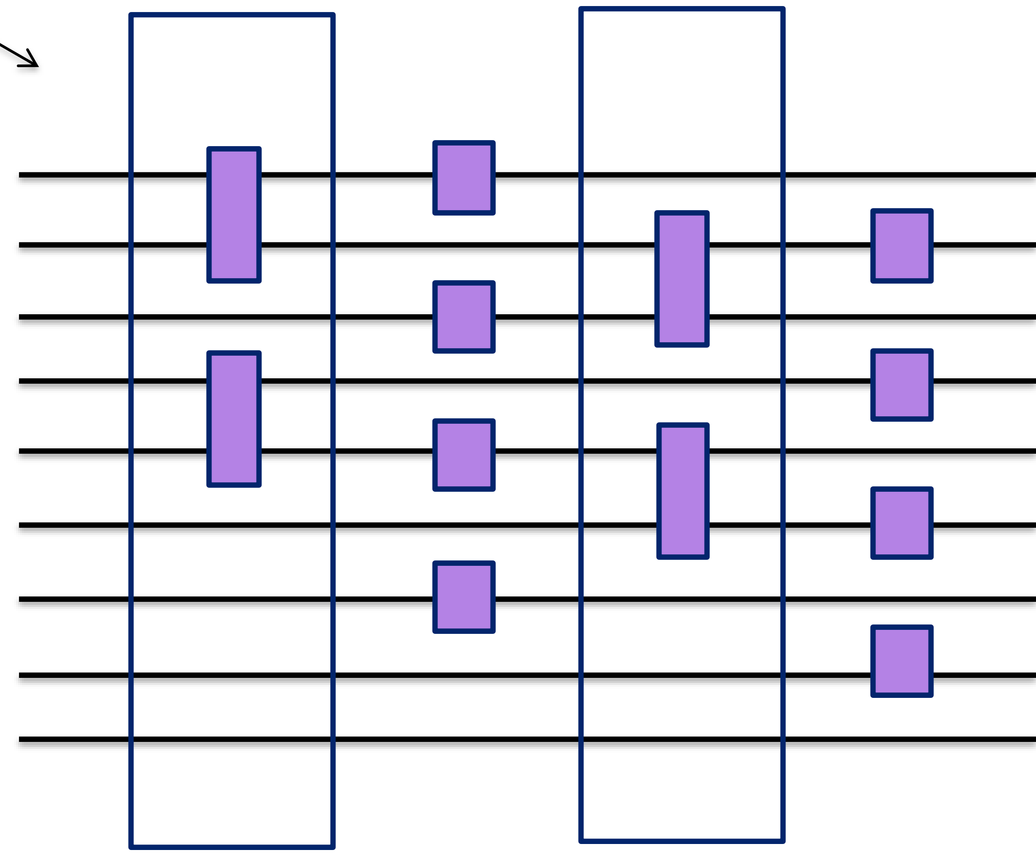
Pauli noise accumulates linearly, in contrast to coherent noise, which accumulates quadratically!

Pauli Twirling (PT)

Options	Sub-options	Sub-sub-options	Choices	Default
twirling	enable_gates		True/False	False
	enable_measure		True/False	True
	num_randomizations			'auto'
	shots_per_randomization			'auto'
	strategy		'active'/ 'active-circuit'/ 'active-accum'/ 'all'	'active-accum'

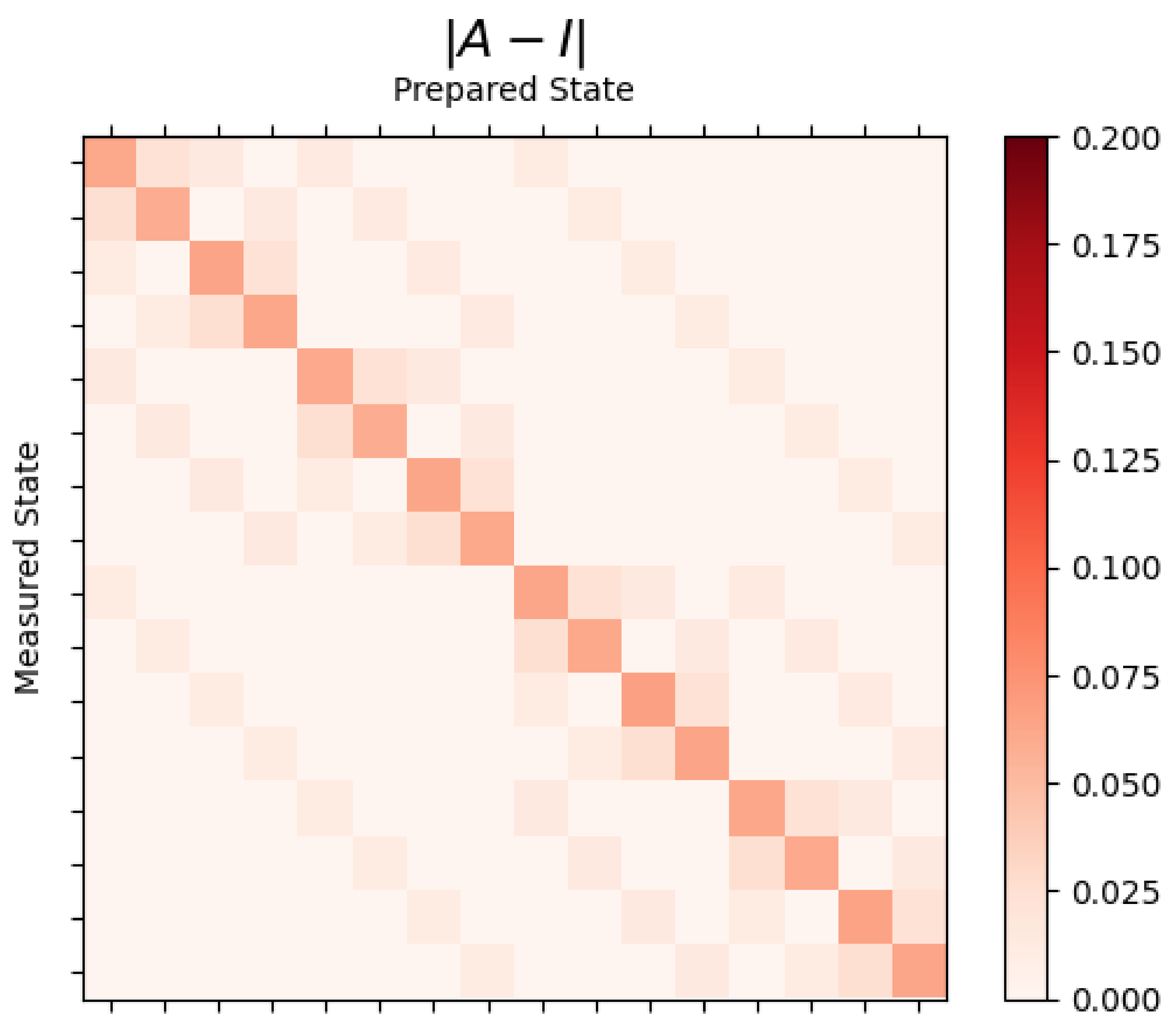
(later)

```
from qiskit_ibm_runtime import SamplerOptions, EstimatorOptions

options = SamplerOptions(default_shots=1024)  # or...
options = EstimatorOptions(default_shots=1024, optimization_level=0, resilience_level=0)

## Configure Twirling
options.twirling.enable_gates = True
options.twirling.enable_measure = False
options.twirling.num_randomizations = 'auto'
options.twirling.shots_per_randomization = 'auto'
options.twirling.strategy = 'active-accum'
```



https://docs.quantum.ibm.com/api/qiskit-ibm-runtime/qiskit_ibm_runtime.options.TwirlingOptions

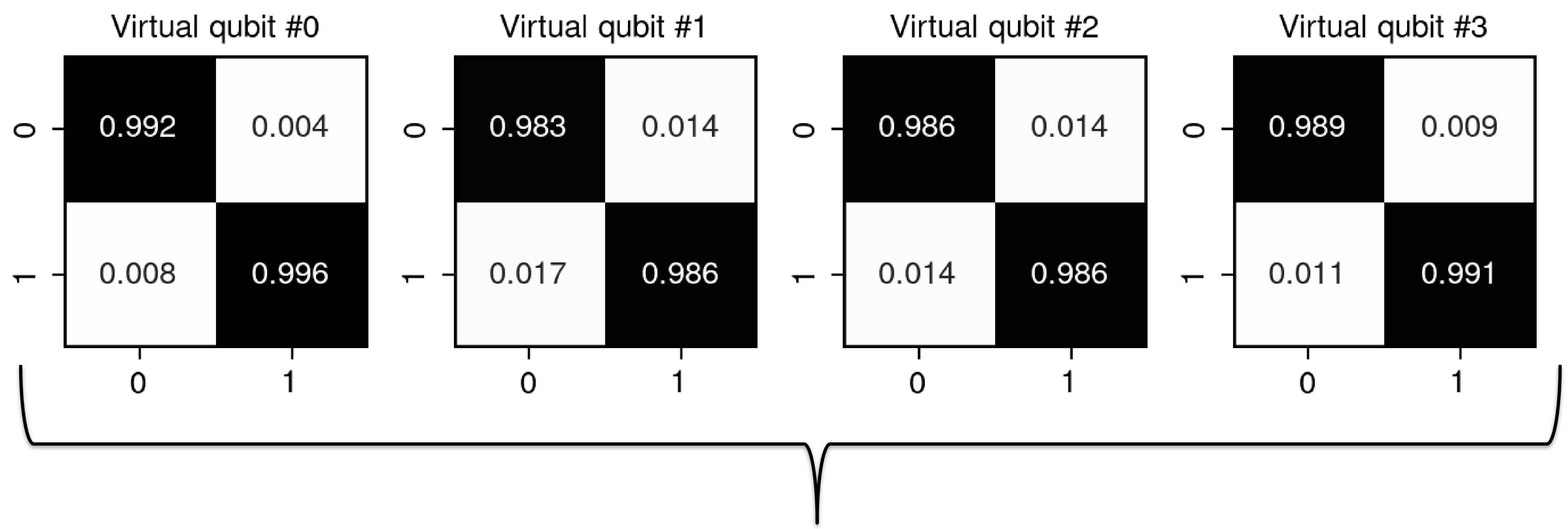
Twirled Readout Error eXtinction (TREX)



Readouts errors cause the wrong states to be measured.

This is depicted in the readout-error transfer matrix.

In scenarios when the noise is not correlated between qubits, measurement errors can be measured per qubit and the full transfer matrix is then reconstructed as a tensor product.



$$2^N \times 2^N$$

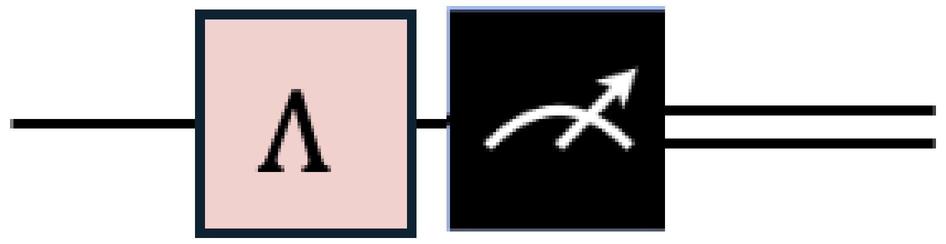
↓

The inverse of the transfer matrix can be used for error mitigation, but it cannot be obtained efficiently in general!

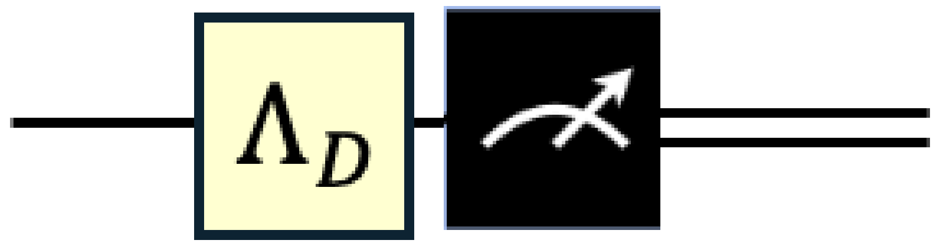
Twirled Readout Error eXtinction (TREX)

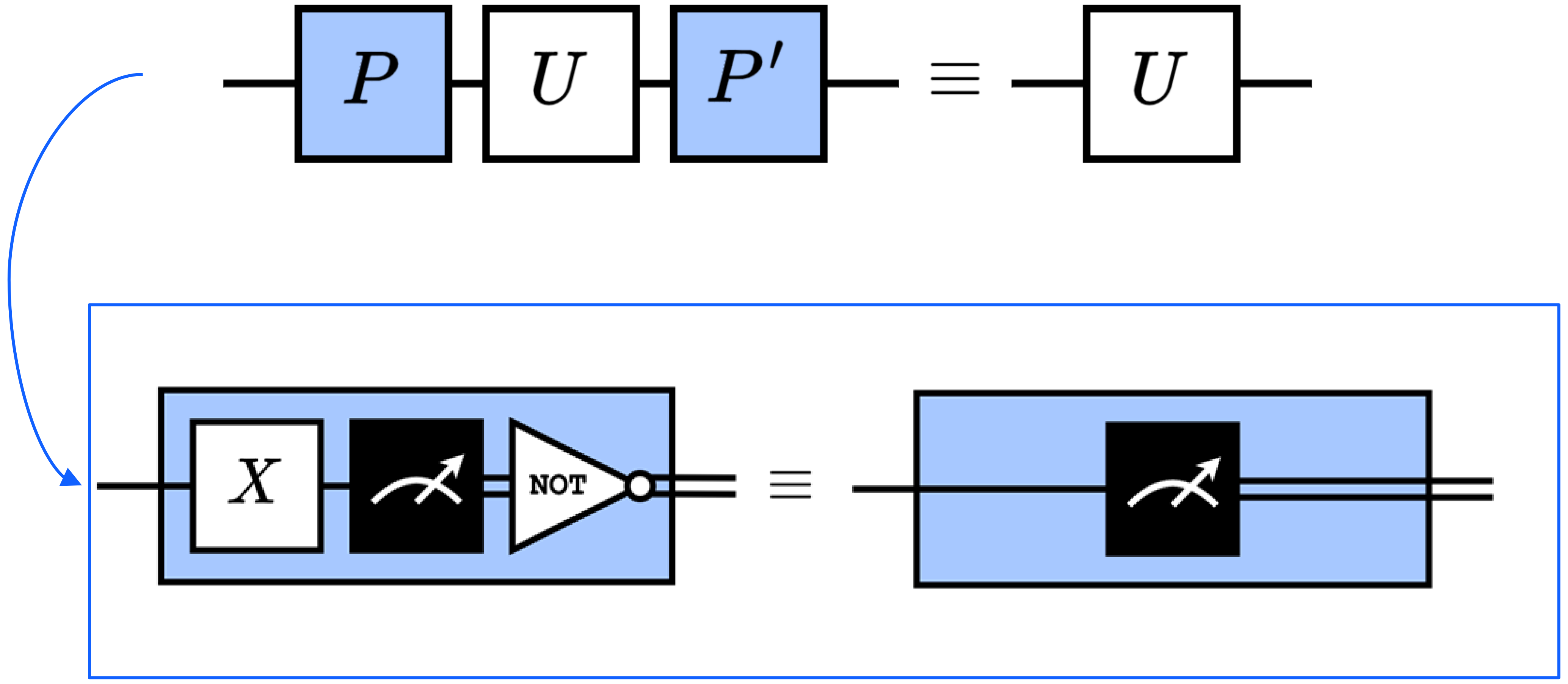
Via **measurement twirling we can diagonalize the** readout-error transfer matrix.

Arbitrary transfer matrix

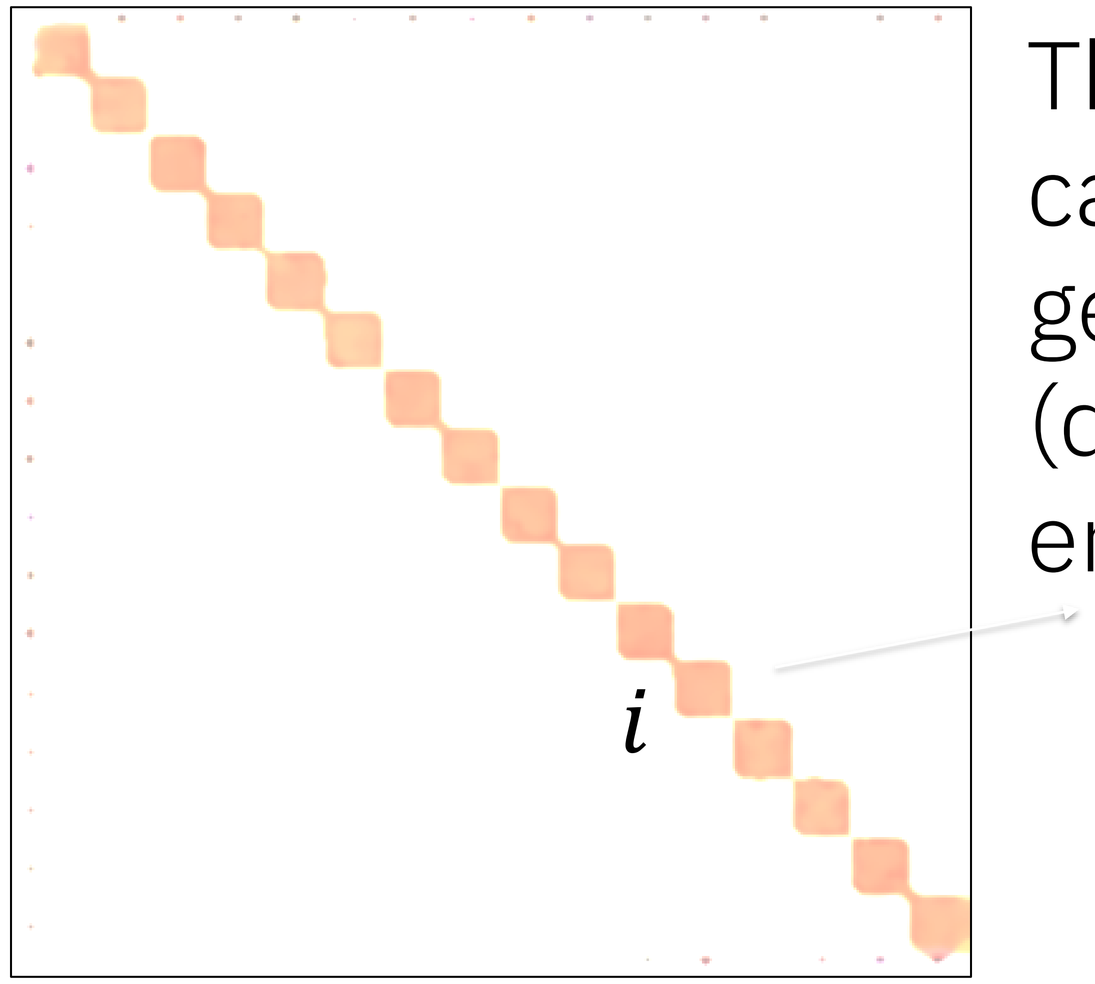


Diagonal transfer matrix





Average over random bitflips that are undone in classical postprocessing.



Then, we need to run calibration circuits to get the inverse of the (diagonal) readout error transfer matrix.

$$< O_i > = \frac{< \widetilde{O_i} >}{E_i}$$

Only valid for expectation values!

Twirled Readout Error eXtinction (TREX)

Options	Sub-options	Sub-sub-options	Choices	Default
resilience	measure_mitigation		True/False	True
	measure_noise_learning	num_randomizations		32
		shots_per_randomization		'auto'

```
from qiskit_ibm_runtime import EstimatorOptions

options = EstimatorOptions(default_shots=1024, optimization_level=0, resilience_level=0)

## Configure TREX
options.resilience.measure_mitigation = True
options.resilience.measure_noise_learning.num_randomizations = 32
options.resilience.measure_noise_learning.shots_per_randomization = 'auto'

options.twirling.enable_measure = True # Automatically set by TREX
```

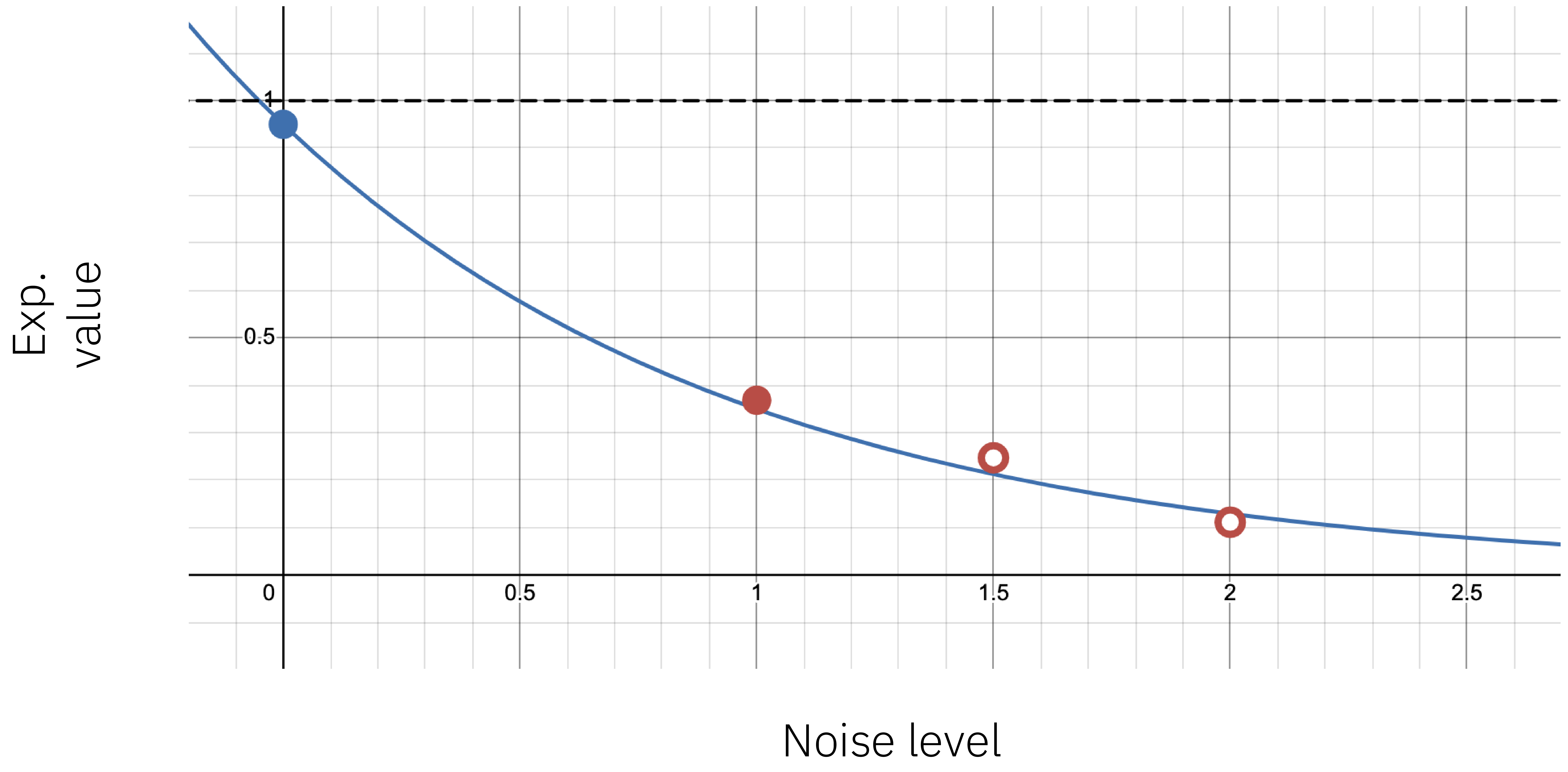

https://docs.quantum.ibm.com/api/qiskit-ibm-runtime/qiskit_ibm_runtime.options.ResilienceOptionsV2

Zero Noise Extrapolation (ZNE)

Measures the effects of amplified noise to infer what the results would look like in the absence of noise.

1. Noise amplification:
   the original circuit unitary is executed at different levels of noise.

2. Extrapolation:
   the zero-noise limit is inferred from the noisy expectation-value results.



Zero Noise Extrapolation (ZNE)

Options	Sub-options	Sub-sub-options	Choices	Default
resilience	zne_mitigation		`True`/`False`	`False`
	zne	noise_factors		`(1, 1.5, 2) for PEA, and (1, 3, 5) otherwise`
		amplifier	`'gate_folding'`/ `'gate_folding_front'`/ `'gate_folding_back'`/ `'pea'`	`gate_folding`
		extrapolator	`'exponential'`/ `'linear'`/ `'double_exponential'`/ `'polynomial_degree_(1 <= k <= 7)'`	(`'exponential'`, `'linear'`)

```
from qiskit_ibm_runtime import EstimatorOptions

options = EstimatorOptions(default_shots=1024, optimization_level=0, resilience_level=0)

## Configure ZNE
options.resilience.zne_mitigation = True
options.resilience.zne.noise_factors = (1, 3, 5)
options.resilience.zne.amplifier = 'gate_folding'
options.resilience.zne.extrapolator = ('exponential', 'linear')
```

[Can have fractional noise factors, e.g. (1, 1.5, 2)]

Choosing the right noise factors and extrapolator is tricky!

https://docs.quantum.ibm.com/api/qiskit-ibm-runtime/qiskit_ibm_runtime.options.ResilienceOptionsV2

Combining techniques

```
from qiskit_ibm_runtime import QiskitRuntimeService, EstimatorV2, EstimatorOptions

options = EstimatorOptions(default_shots=1024, optimization_level=0, resilience_level=0)

## Configure Dynamical Decoupling
options.dynamical_decoupling.enable = True
options.dynamical_decoupling.sequence_type = 'XX'
options.dynamical_decoupling.extra_slack_distribution = 'middle'
options.dynamical_decoupling.scheduling_method = 'alap'

## Configure Twirling
options.twirling.enable_gates = True
options.twirling.enable_measure = True  # Needed for TREX
options.twirling.num_randomizations = 'auto'
options.twirling.shots_per_randomization = 'auto'
options.twirling.strategy = 'active-accum'

## Configure TREX
options.resilience.measure_mitigation = True
options.resilience.measure_noise_learning.num_randomizations = 32
options.resilience.measure_noise_learning.shots_per_randomization = 'auto'

## Configure ZNE
options.resilience.zne_mitigation = True
options.resilience.zne.noise_factors = (1, 3, 5)
options.resilience.zne.extrapolator = 'exponential'

service = QiskitRuntimeService()
backend = service.least_busy()
estimator = EstimatorV2(backend, options=options)
```

Combining techniques

Useful summary of all techniques:
https://docs.quantum.ibm.com/guides/error-mitigation-and-suppression-techniques

```python
from qiskit_ibm_runtime import EstimatorV2

estimator = EstimatorV2(backend=backend)
estimator.options.resilience_level = 2
job = estimator.run([(transpiled_circuit, transpiled_observable)], precision=0.01)
print(f">>> Job ID: {job.job_id()}")
print(f">>> Job Status: {job.status()}")
```

✓ 2.1s Python

Resilience Level	Definition	Technique
0	No mitigation	None
1 [Default]	Minimal mitigation costs: Mitigate error associated with readout errors	Twirled Readout Error eXtinction (TREX) measurement twirling
2	Medium mitigation costs. Typically reduces bias in estimators, but is not guaranteed to be zero-bias.	Level 1 + Zero Noise Extrapolation (ZNE) and gate twirling

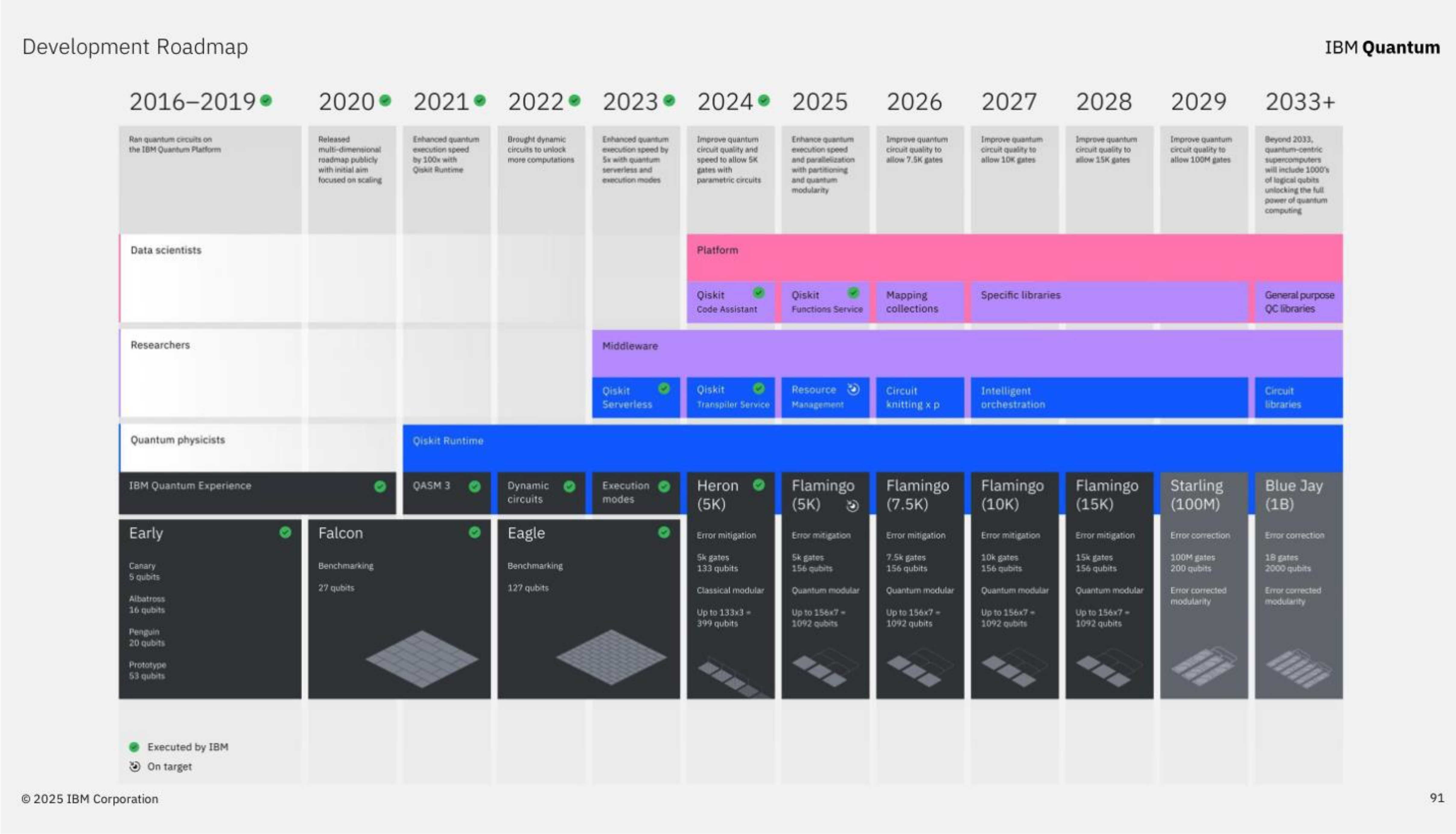
resilience level is only available for Estimator!

Heron-R2: the latest processor on our QEC directed roadmap

Tunable coupler architecture with improved calibrations

Integrated two-level system mitigation controls

156 high quality qubits allow true utility scale experiments and explorations of quantum advantage



ibm_kingston		
QPU status	Online	
Processor type	Heron r2	
Qubits	2Q error (best/layered)	CLOPS
156	1.16e-3/3.44e-3	250K

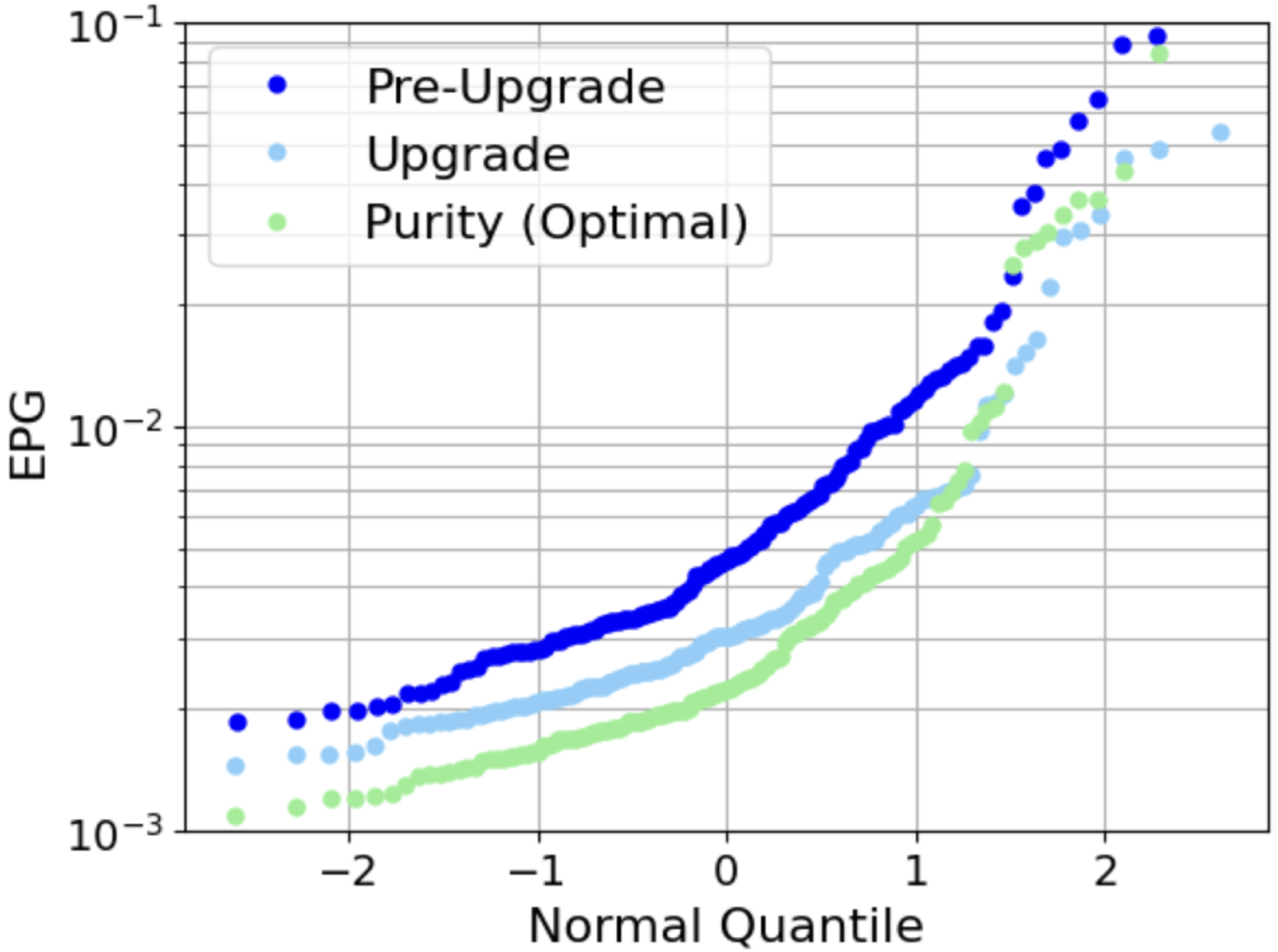
ibm_aachen		
QPU status	Online	
Processor type	Heron r2	
Qubits	2Q error (best/layered)	CLOPS
156	1.25e-3/3.46e-3	250K

Better tunable coupler calibrations

Improved calibrations bring performance closer to optimal attainable from the coherence of device

We upgraded Torino in August 2024 from median 5e-3 to median 3e-3 for 2 qubit gate error

This technique is now part of all our newest Heron deployments



Improvements in instruction set architecture leading to:

1. Depth reduction,

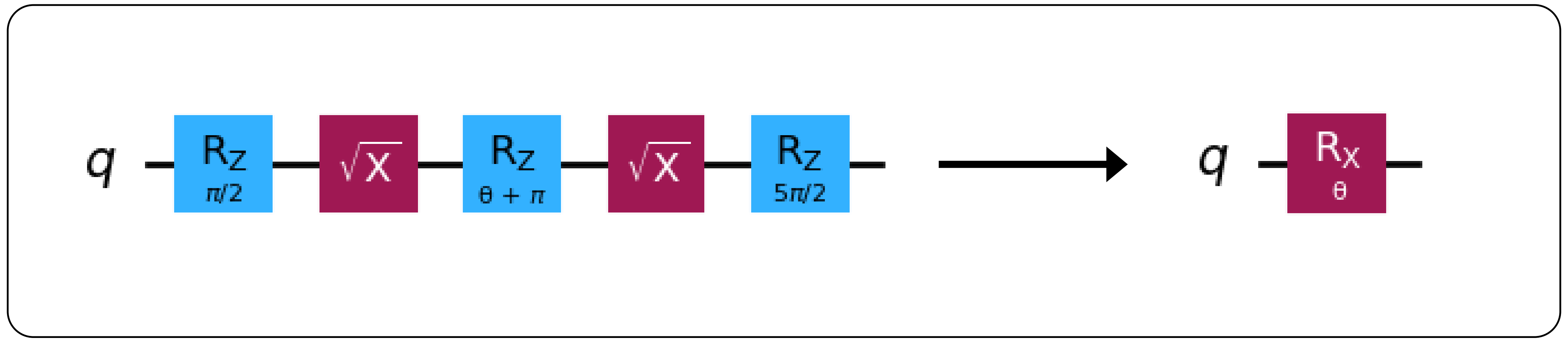
2. More efficient execution of dynamic circuits

Rolling out now: fractional one and two-qubit gates that reduce circuit depth in a variety of settings.

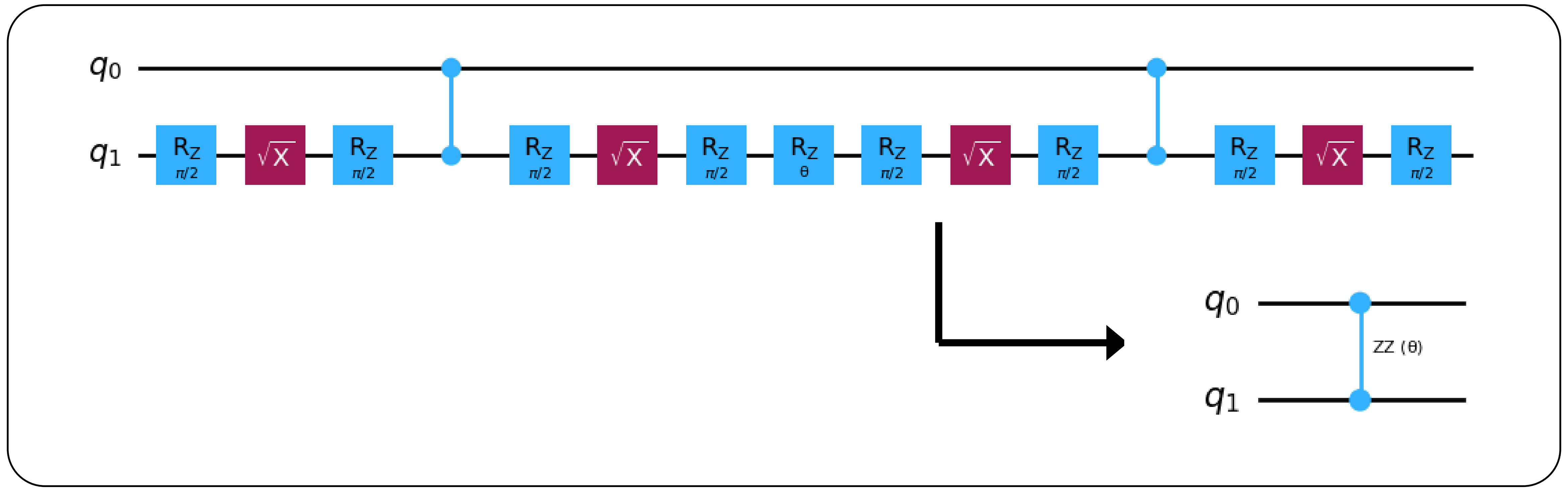
$R_{ZZ}(\theta)$ for $0 < \theta \leq \pi/2$

$R_X(\theta)$ for any θ

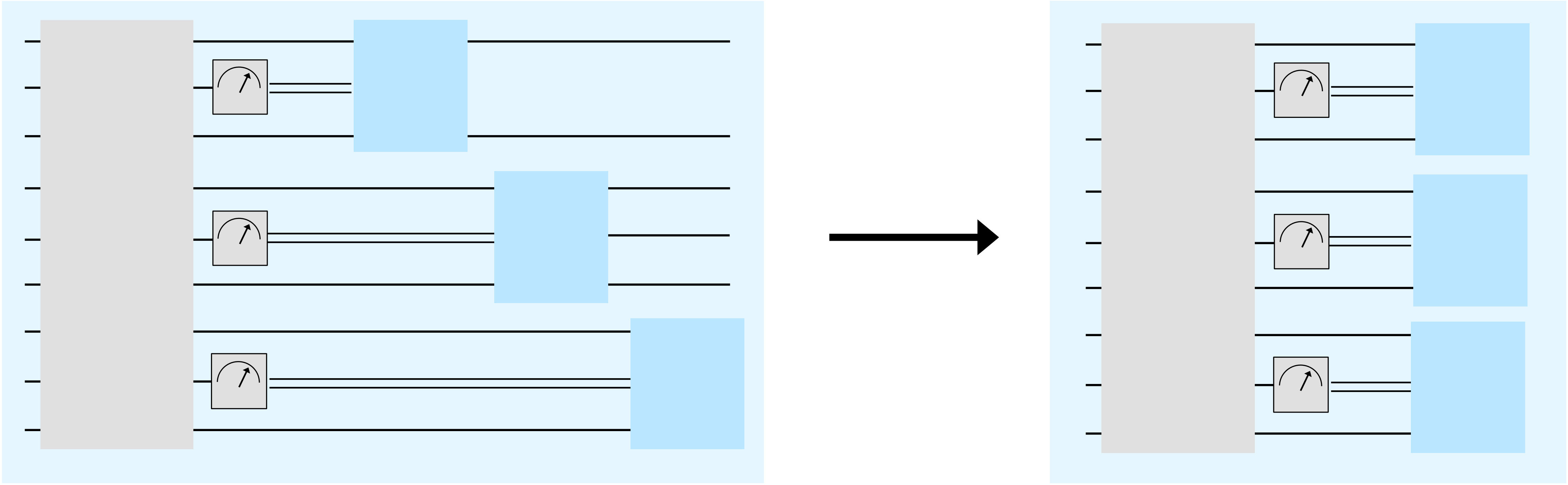
1Q fractional gates compresses:



2Q fractional gates offer even better compression:



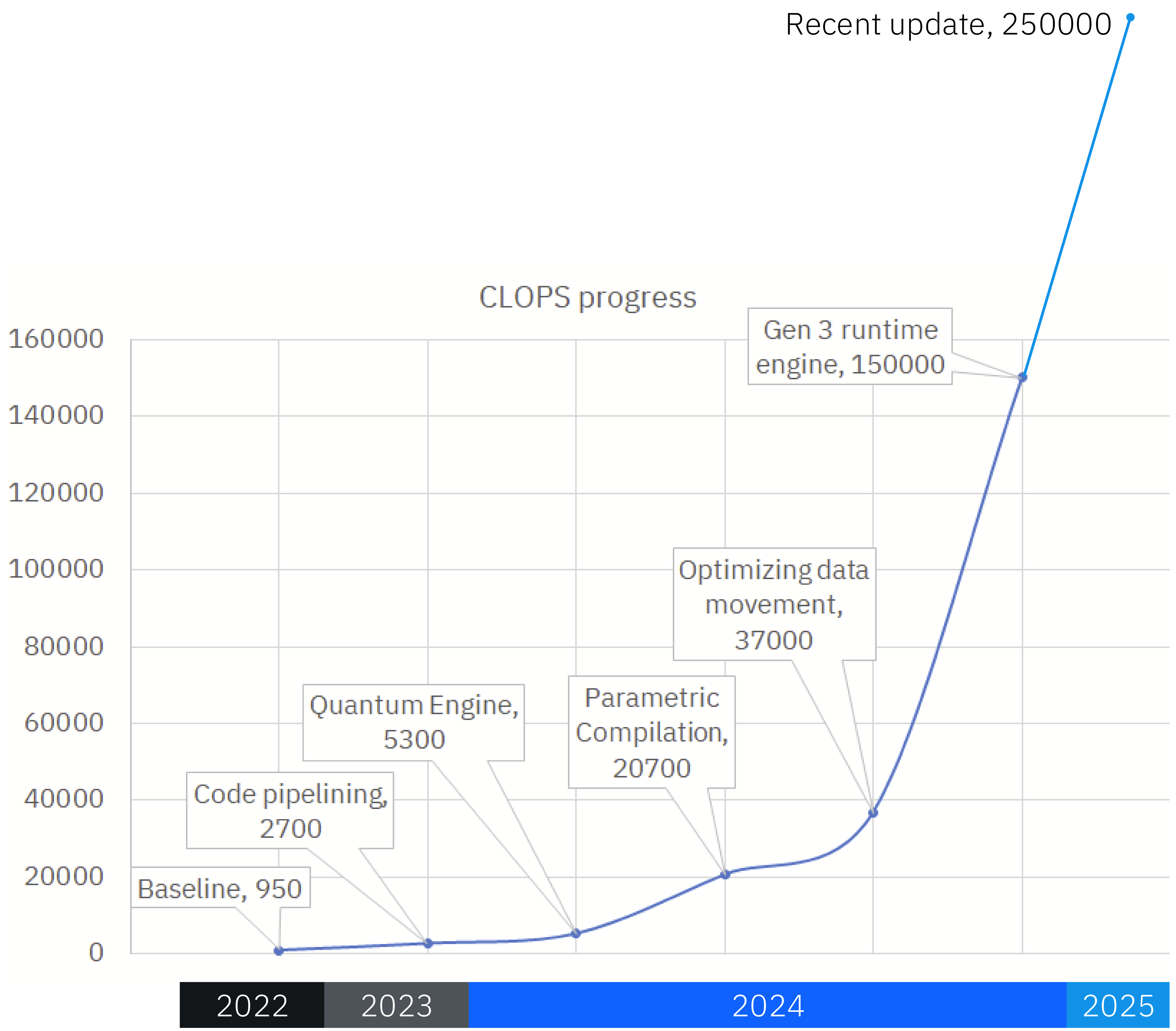
In June: modifications to improve the performance of dynamic circuits, through compression of circuit duration with parallel execution of conditional blocks



Robust software for utility-scale explorations

Speed up in execution of
Utility experiment

↑75x



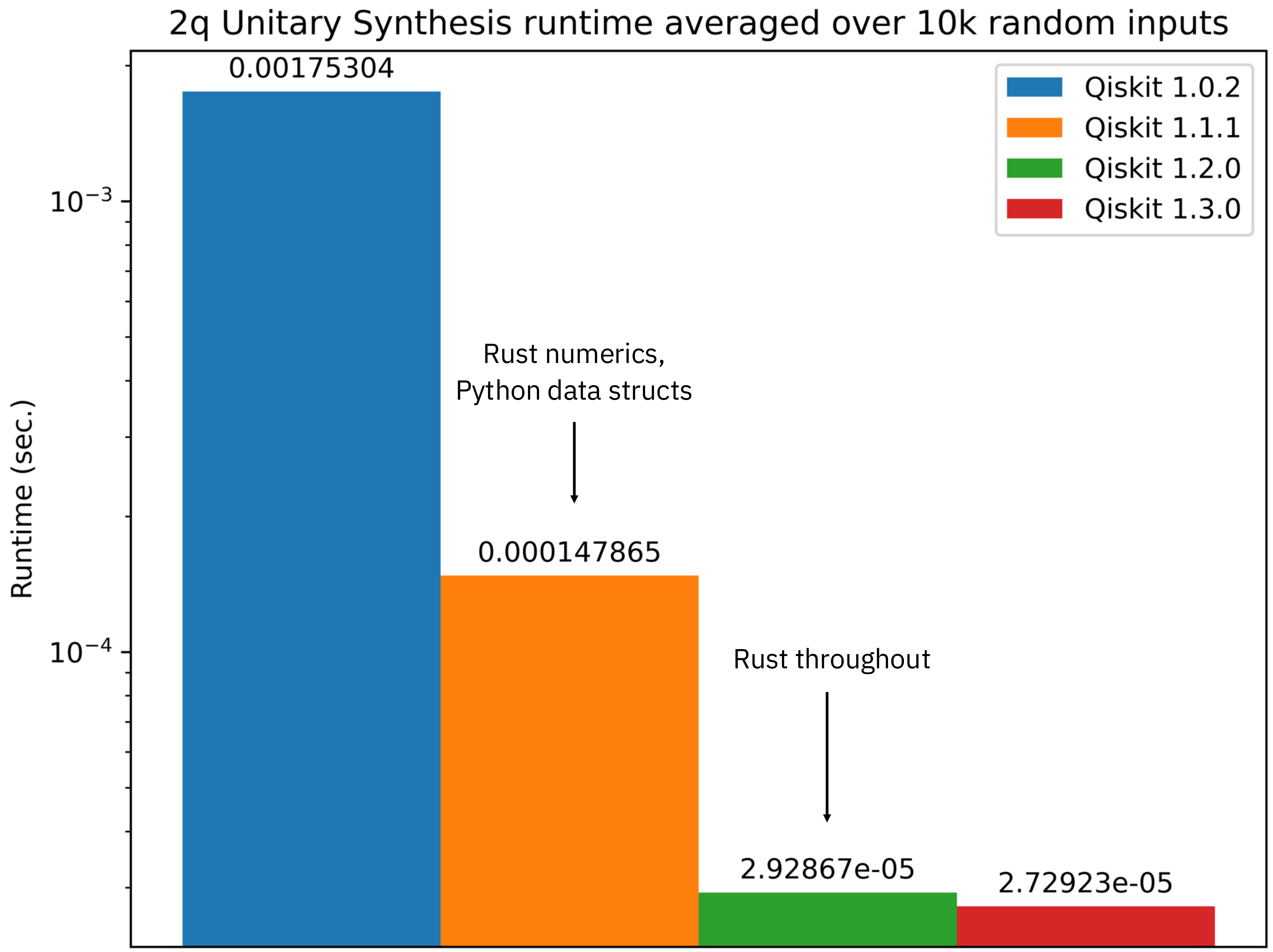
Qiskit SDK: Built with a focus on performance

Since 2024 we have focused on moving the bulk of Qiskit's transpilation pipelines into Rust.

Currently on-going, but already two-orders of magnitude faster in many areas.

Improvement

↑60x



arXiv:2405.08810

Qiskit Transpiler Service and AI Passes

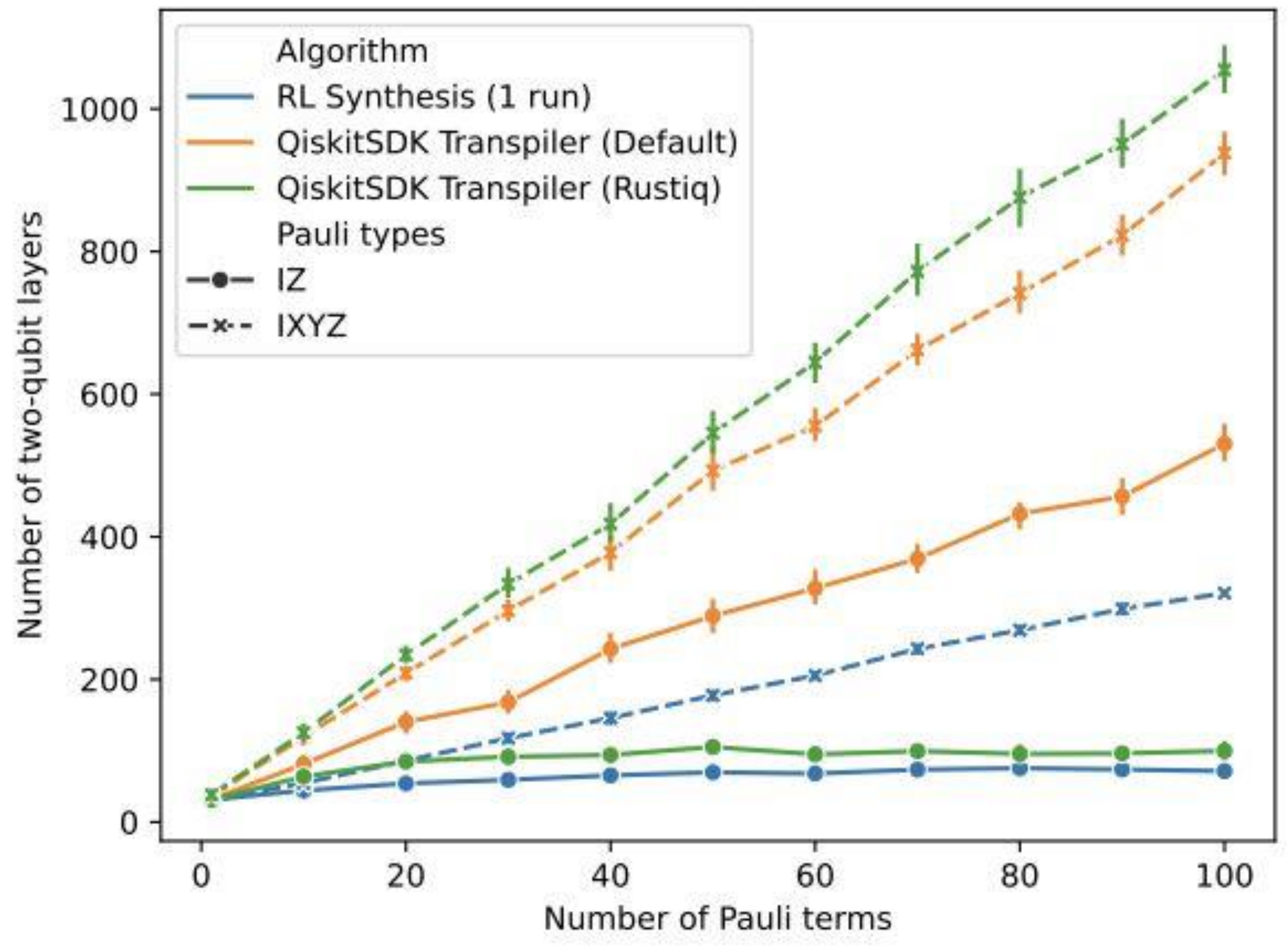
With the Qiskit Transpiler Service, you can transpile on the cloud and leverage the power of the AI transpiler passes.

Three major upgrades:
↓

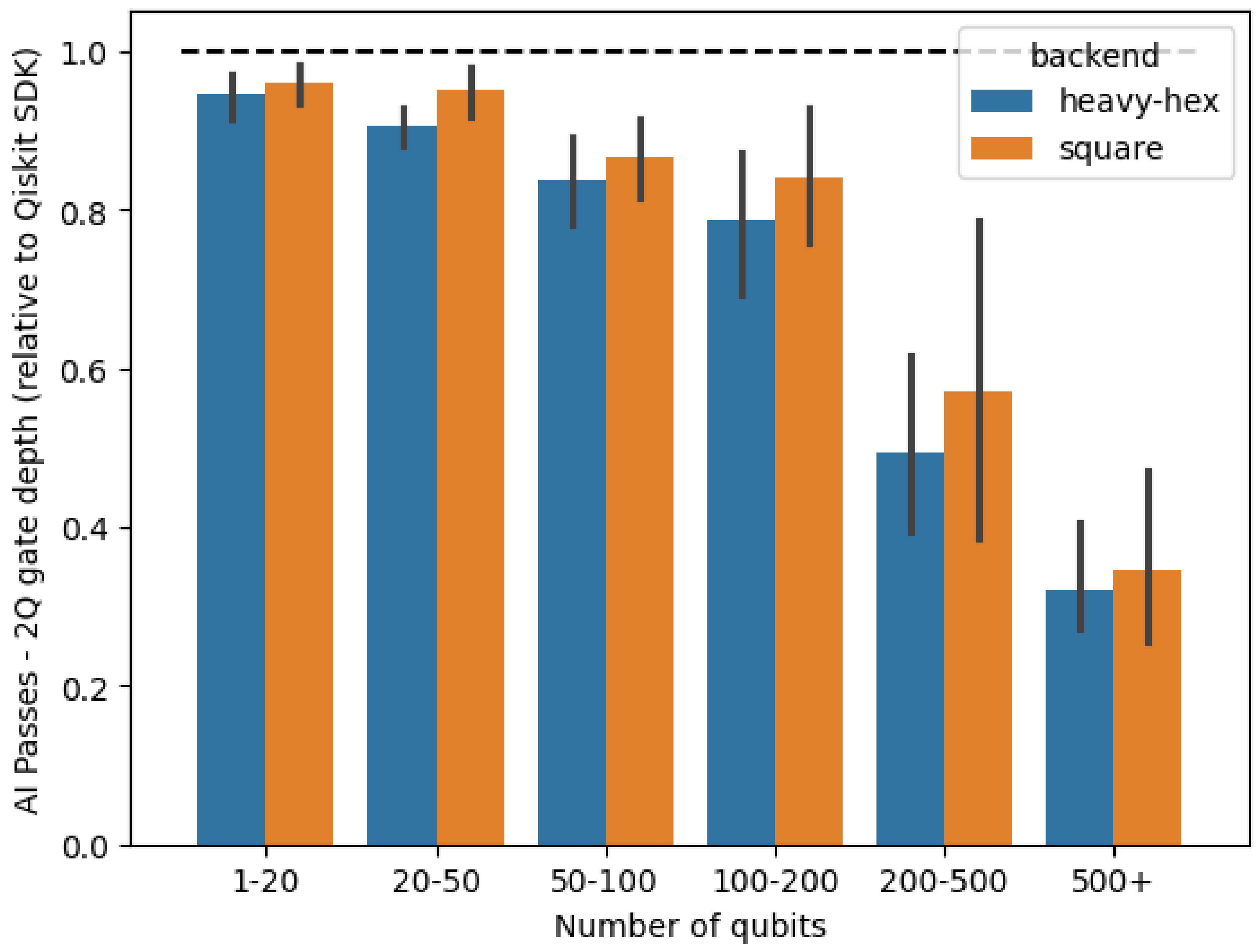
01
Improved stability & performance
Tested on 1000 qubits and 1M gates, 30%+ depth improvement on benchpress circuits.

02
New AI Synthesis Pass
Pauli Network synthesis, perfect for chemistry circuits.

03
Local AI mode
You can now use some of the AI passes locally!



We extend our earlier work on RL-based synthesis of Clifford circuits to general Pauli Networks. We obtain reduction in circuit depth of around 20% on average, and in many cases over 40%.



Circuit depth for circuits transpiled with Qiskit+AI, relative to standard Qiskit, for the Benchpress circuits (lower is better)

New Qiskit 2.0

Our second major version release setting the stage for efficient quantum-centric supercomputing

https://www.ibm.com/quantum/blog/qiskit-2-0-release-summary

The biggest updates are:

A brand-new C API to interact with the `SparseObservable` class. It's the first step toward building a robust C interface for the Qiskit SDK.

https://www.ibm.com/quantum/blog/supercomputing-24

Continued performance improvements by oxidizing (Rust) Qiskit, further speedups in circuit synthesis.

Introducing boxes and stretches features for grouping your instructions for later processing and for more control of the timing of gates.

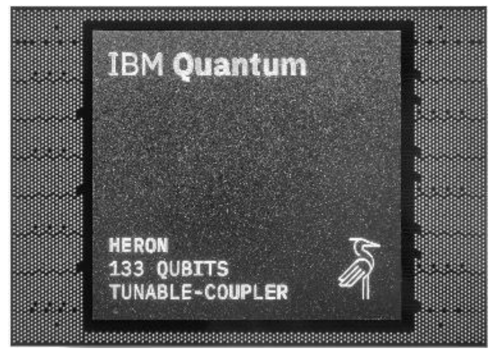
Qiskit addons:

A collection of research capabilities developed as modular tools that can plug into a workflow to design new algorithms at the utility scale

Starting with multi-product formulas (MPF), approximate quantum compilation (AQC-Tensor), operator backpropagation (OBP), and sample-based quantum diagonalization (SQD)*.

Map	Optimize	Execute	Post-Process
AQC-Tensor	OBP		SQD
MPF	Circuit cutting		M3
Qiskit Circuit Library	Transpiler	Primitives	Quantum Info
Input: Domain inputs	Input: Circuits, observable	Input: ISA circuit, observable	Input: Expectation value/samples
Output: Circuits, observable	Output: ISA circuit, observable	Output: Expectation value/samples	Output: Data objects/visualizations

Chemistry beyond the capabilities of brute-force classical simulation

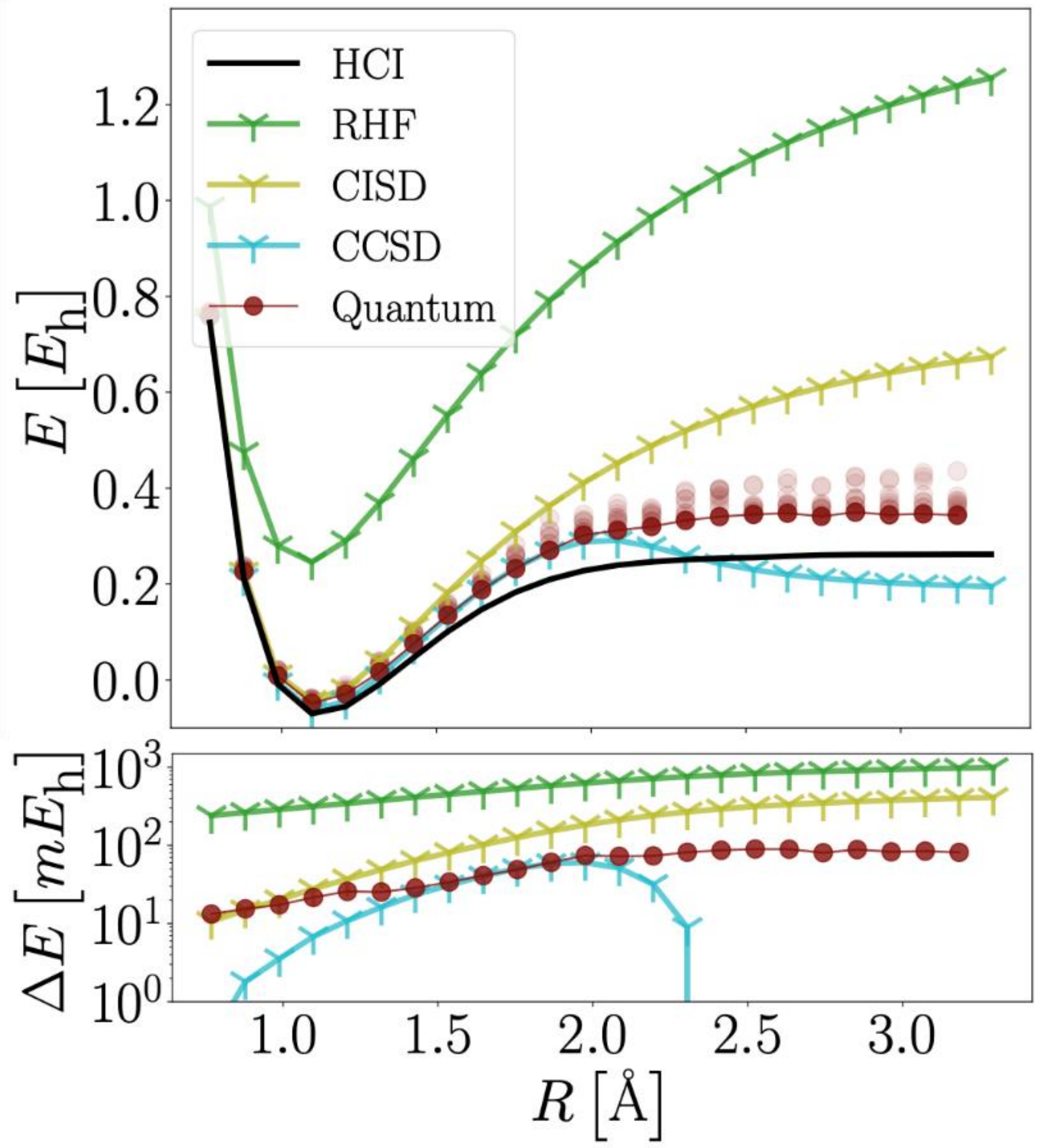


77 qubits
10,570 quantum gates
3,590 two-qubit gates

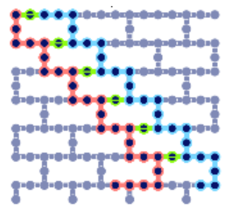
Fugaku

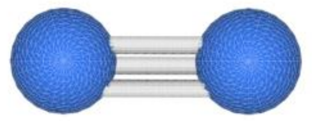
6,400 nodes @
32 GB
1,024 GB/s
48 cores

N_2 – Bond breaking on large basis set



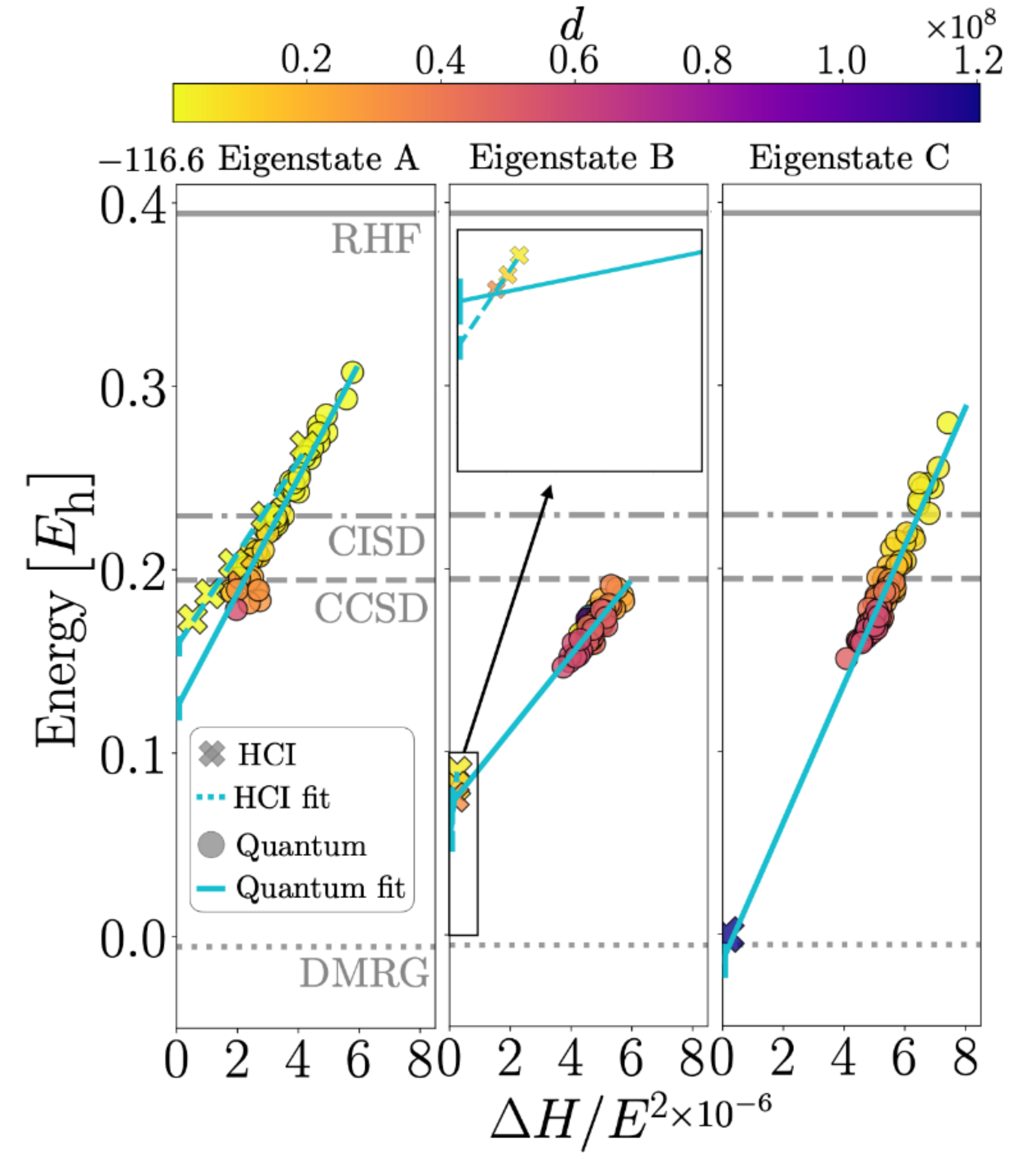
Fugaku



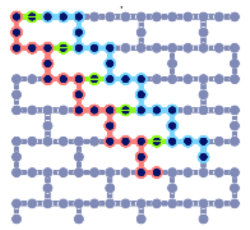


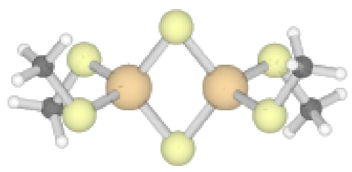
58 qubits are used to model the N_2 dissociation (cc-pVDZ basis set)

Fe_2S_2 – Precision many-body physics



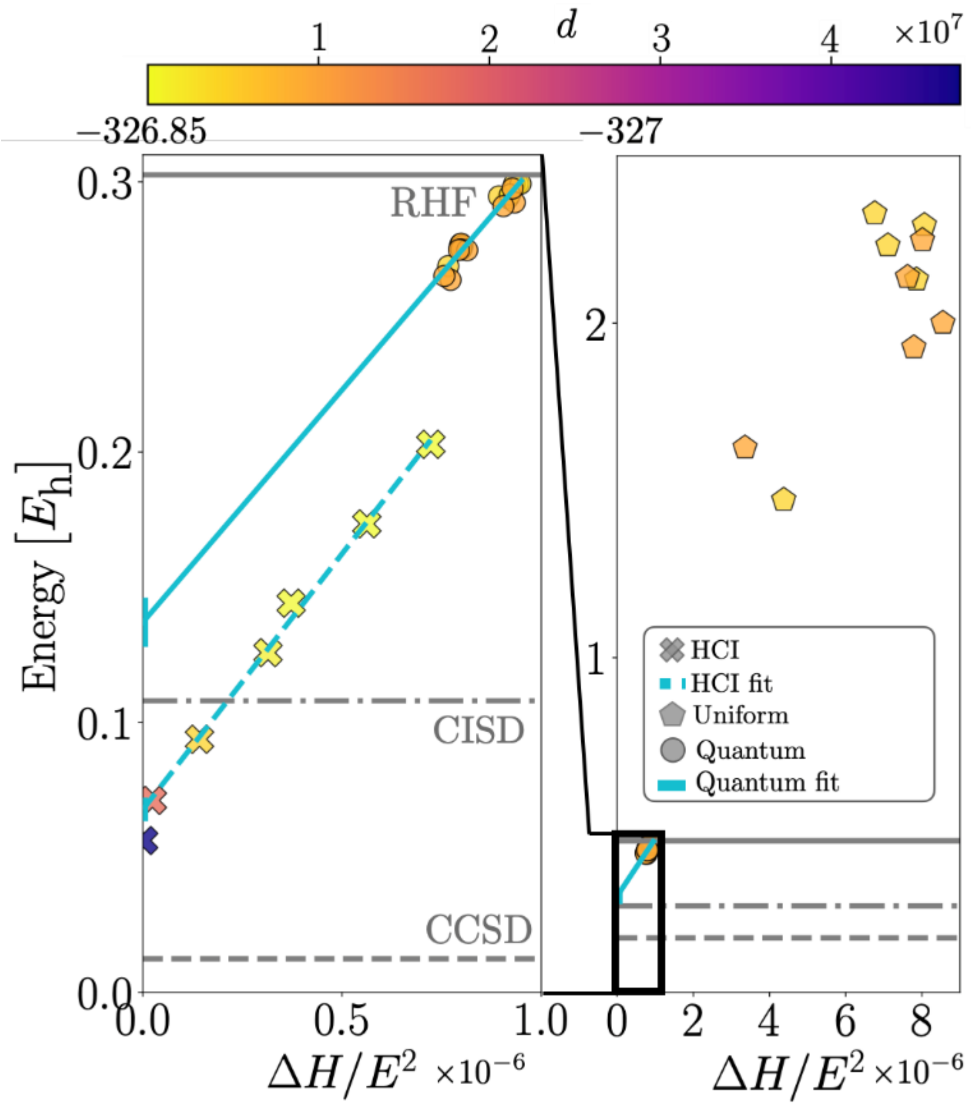
Fugaku



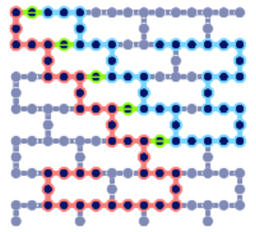


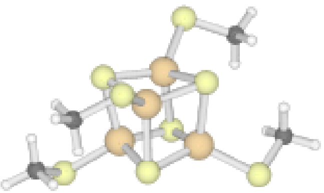
45 qubits are used to model the Fe_2S_2 cluster (TZP-DKH basis set)

Fe_4S_4 – Pushing hardware capabilities



Fugaku



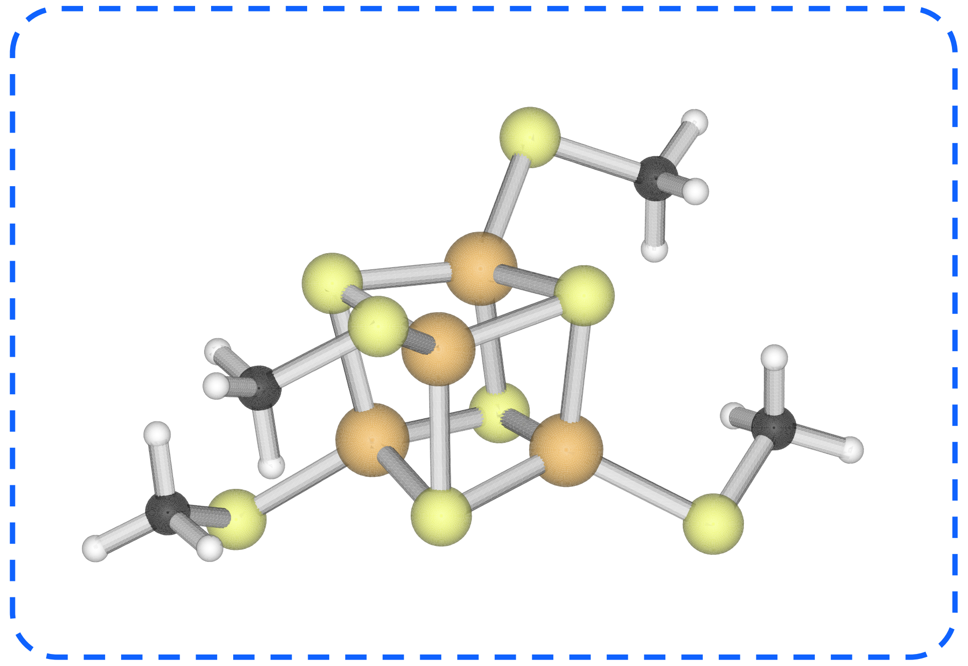


77 qubits are used to model the Fe_4S_4 cluster (TZP-DKH basis set)

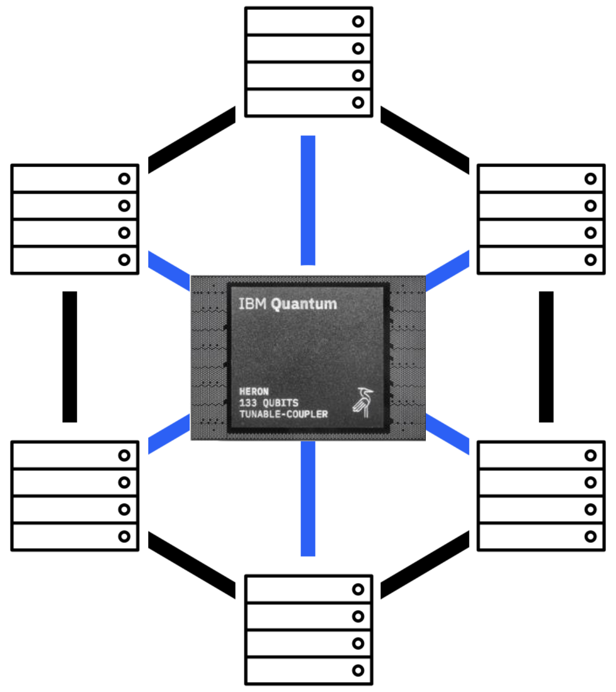
arXiv:2405.05068

What are these problems?

Modeling molecules, atoms, electrons, and quarks with unprecedented accuracy



[4Fe-4S] using an active space of 54 electrons in 36 orbitals from the TZP-DKH basis set



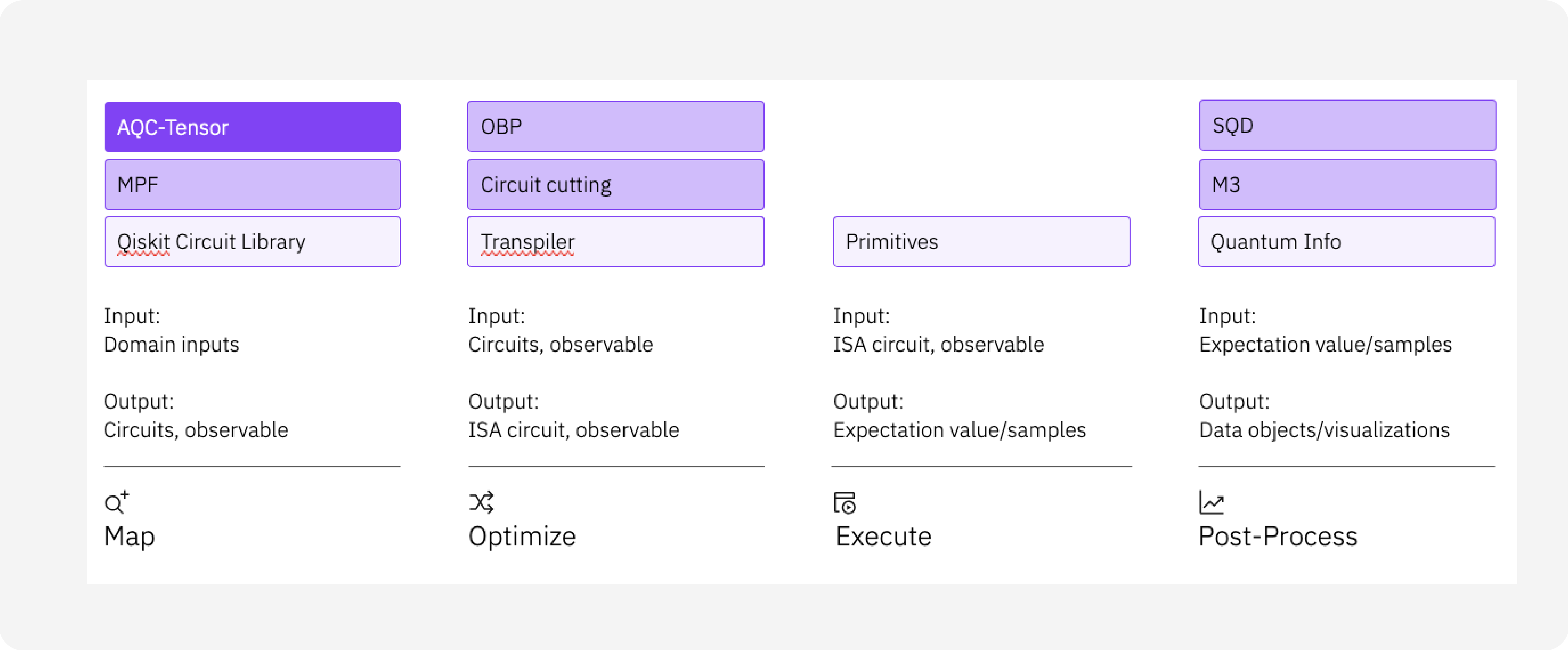
$$H\Psi = E\Psi$$

Classical exact method	63 PiB of memory[1]
Classical approximate method (DMRG)	8 hours[2,3]
Quantum method (QCSC)	12 min using 72 qubits + 1.5 hours supercomputer time[1]

We are already using quantum (Heron, 72 qubits) and HPC (Supercomputer Fugaku) to achieve results comparable with the best classical approximate methods (DMRG) in accuracy and timing

1. arXiv:2405.05068
2. J. Chem. Theory Comp. 20 (2024): 775–786
3. J. Chem. Phys. 159, 234801 (2023)

A 5x speedup in research using the Qiskit addons



Faster research
Up to:

↑5x

1 year of research

"Chemistry Beyond Exact Solutions on a Quantum-Centric Supercomputer"

—

arXiv:2405:05068

RIKEN

2.5 months of research

"Accurate quantum-centric simulations of supramolecular interactions"

—

arXiv:2410:09209

Cleveland Clinic

"Quantum-Centric Study of Methylene Singlet and Triplet States"

—

arxiv.org/abs/2411.04827

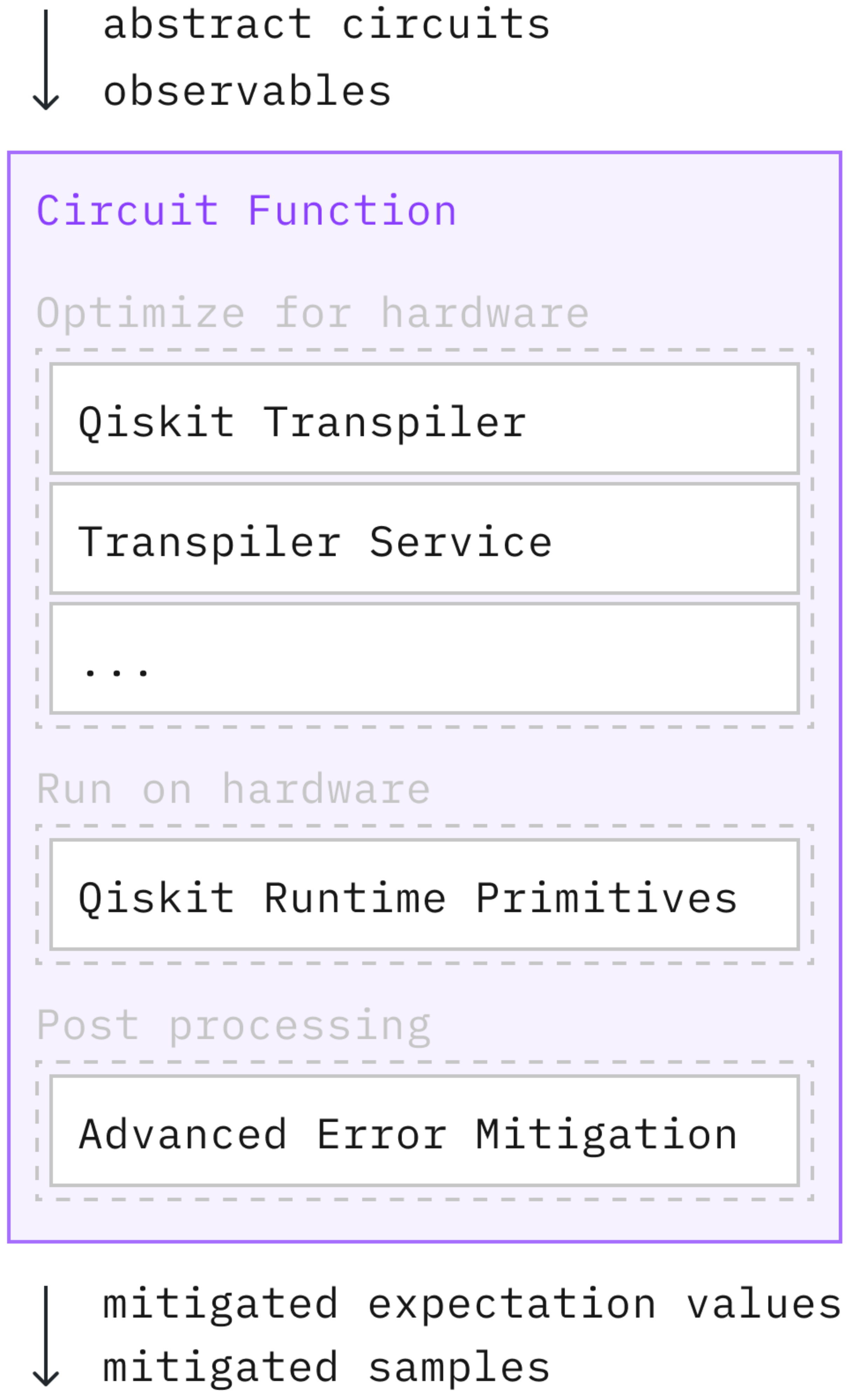
LOCKHEED MARTIN

Qiskit Functions: abstracted services, designed to accelerate development

In September 2024, we previewed a catalog of managed, utility-scale services to unlock new users:

- **Circuit Functions**: Enabling quantum computational scientists to discover new algorithms and applications, without needing to manage transpilation, error suppression, or error mitigation.

- **Application Functions**: Enabling data scientists and enterprise developers to integrate quantum into industry workflows, while leveraging familiar domain abstractions.



Circuit Functions –

Enable quantum computational scientists to discover new algorithms and applications

Circuit function

IBM

Generate accurate hardware results with AI-powered circuit optimization, and advanced error mitigation methods.

Enabled

Performance management

Q-CTRL

Reduce errors without sampling overhead, with automated, AI-driven error suppression.

Not licensed

QESEM

Qedma

Accurate results for large active volumes using characterization, transpilation, error suppression, and mitigation.

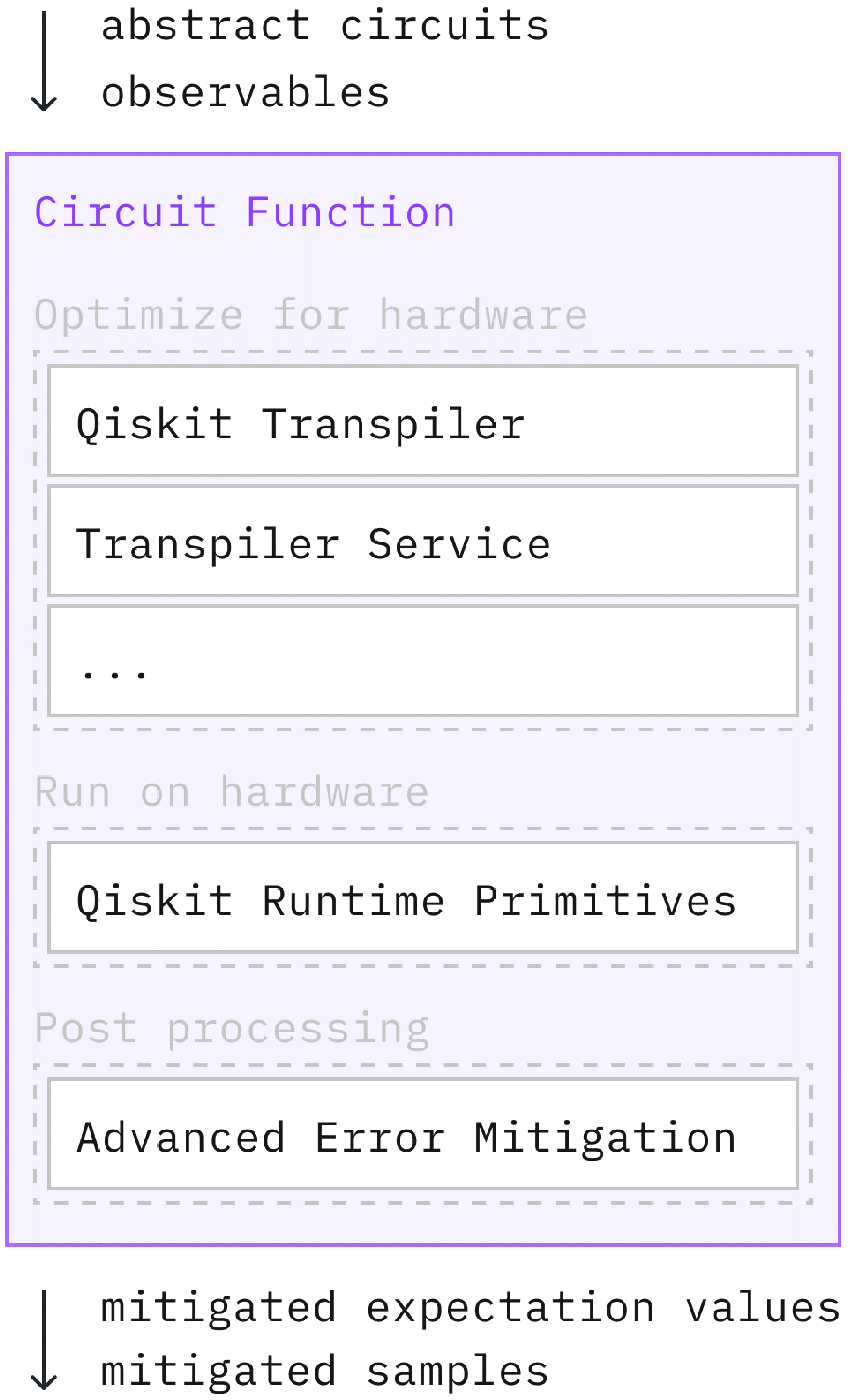
Not licensed

Tensor-Network Error Mitigation

Algorithmiq

Unbiased tensor-network error mitigation with low shot overhead and reduced runtime usage.

Not licensed



Applications Functions –

Enable enterprise developers and data scientists to integrate quantum into industry workflows

QURI chemistry

Qunasys

Compute ground state energy and electron configuration distribution.

Not licensed

Optimization solver

Q-CTRL

Solve optimization problems by inputting the high-level problem definition, and the solver takes care of the rest.

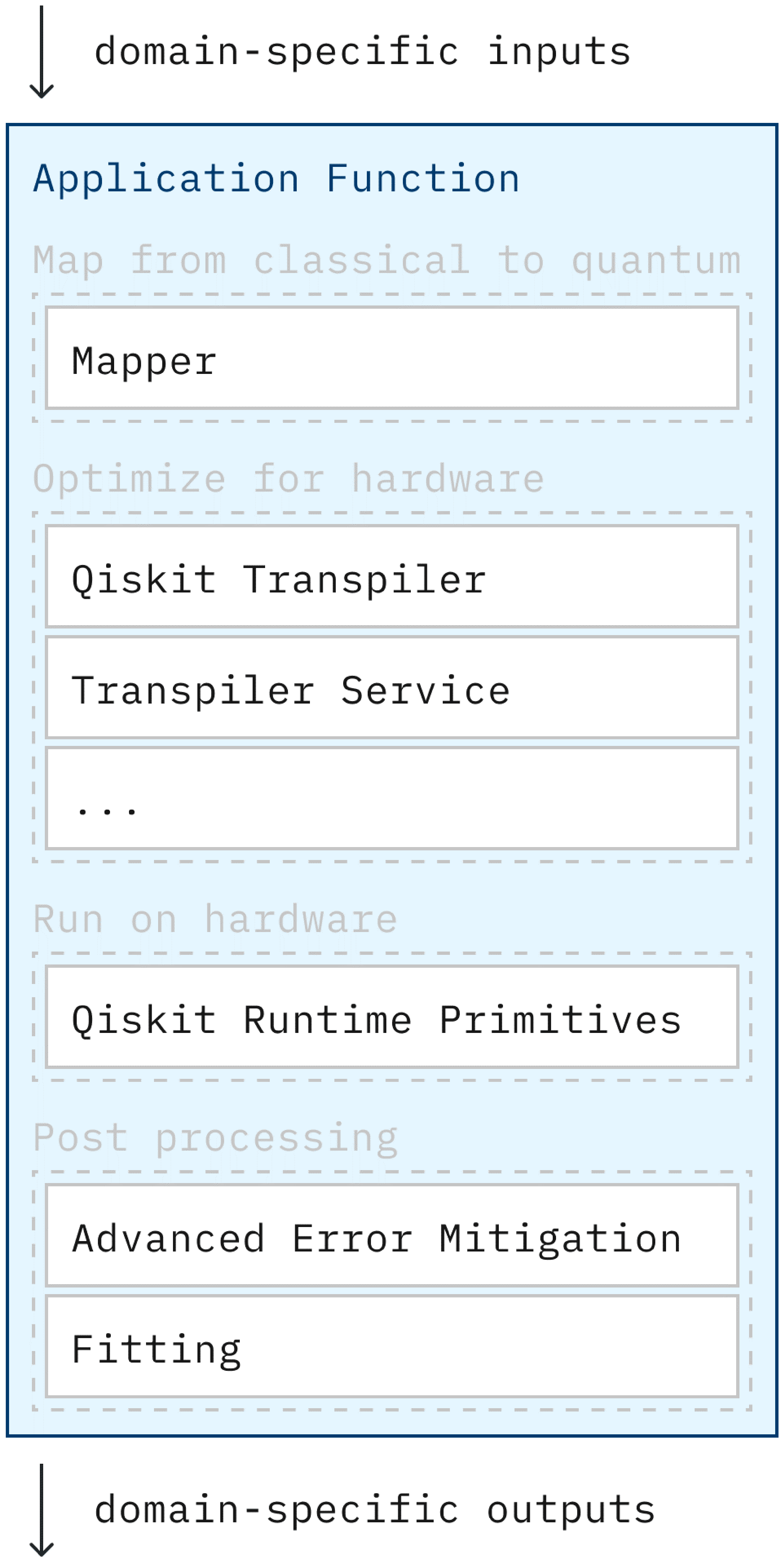
Not licensed

Singularity Machine Learning

Multiverse Computing

Solve real-world classification problems on quantum hardware without requiring quantum expertise.

Not licensed



IBM has the most performant quantum computers

Key factors

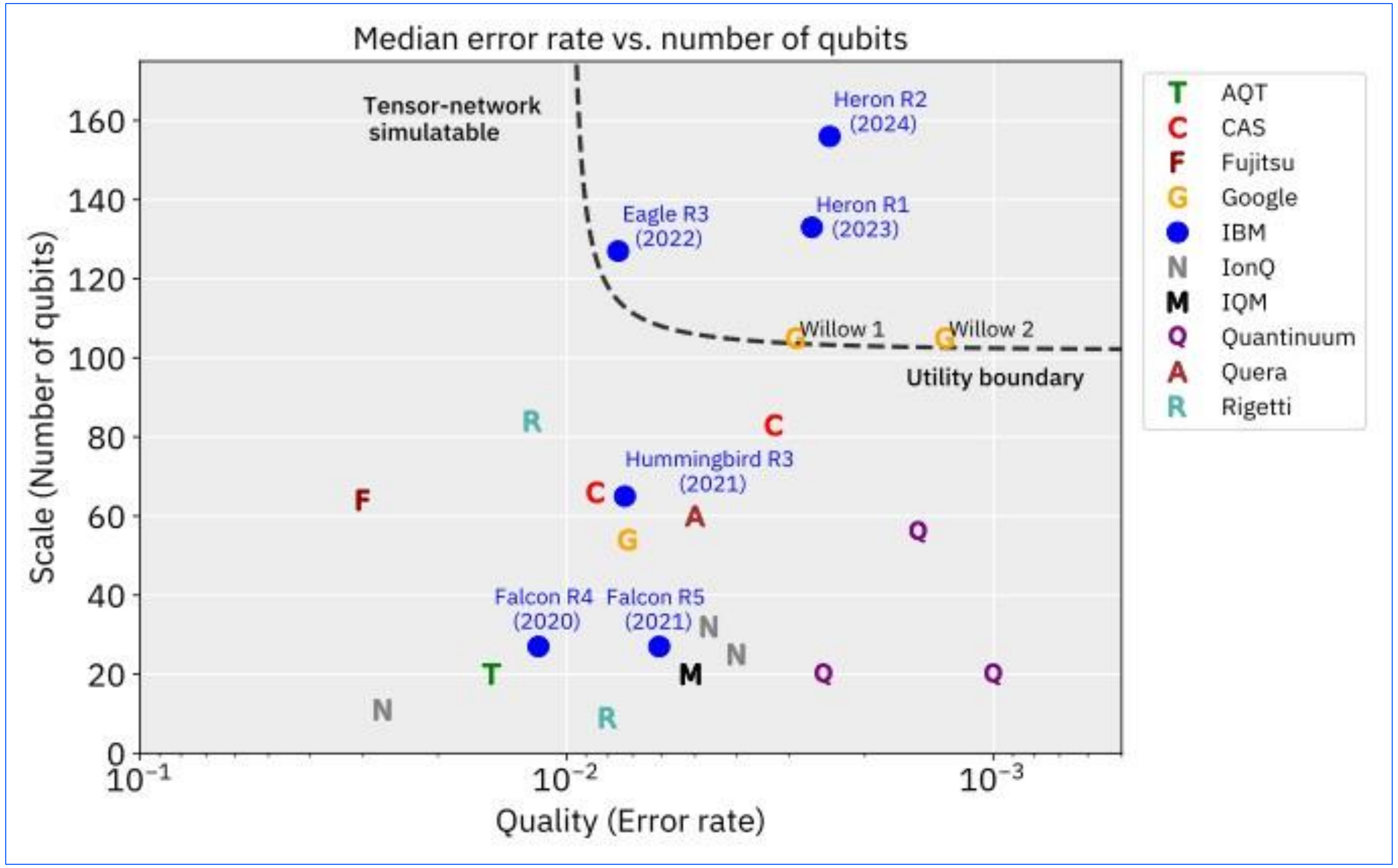
What really matters when comparing execution on various hardware?

Scale: Can I run the job?

Quality: Was the result correct?

Speed: How long does it take to get a result?

Cost: How much money was I charged?



.System size and error rates sourced from publicly available system data sheets and published research papers. (Utility scale defined as proven to be able to run a circuit larger than can be simulated classically by brute-force)

Superconducting vs Ions

Speed

400x-2,000x

faster

@ 500 shots & 56 Qubits

Cost

1,200x-70,000x

cheaper

@ 500 shots & 56 Qubits

Qiskit is the most popular open-source quantum SDK

Key factors

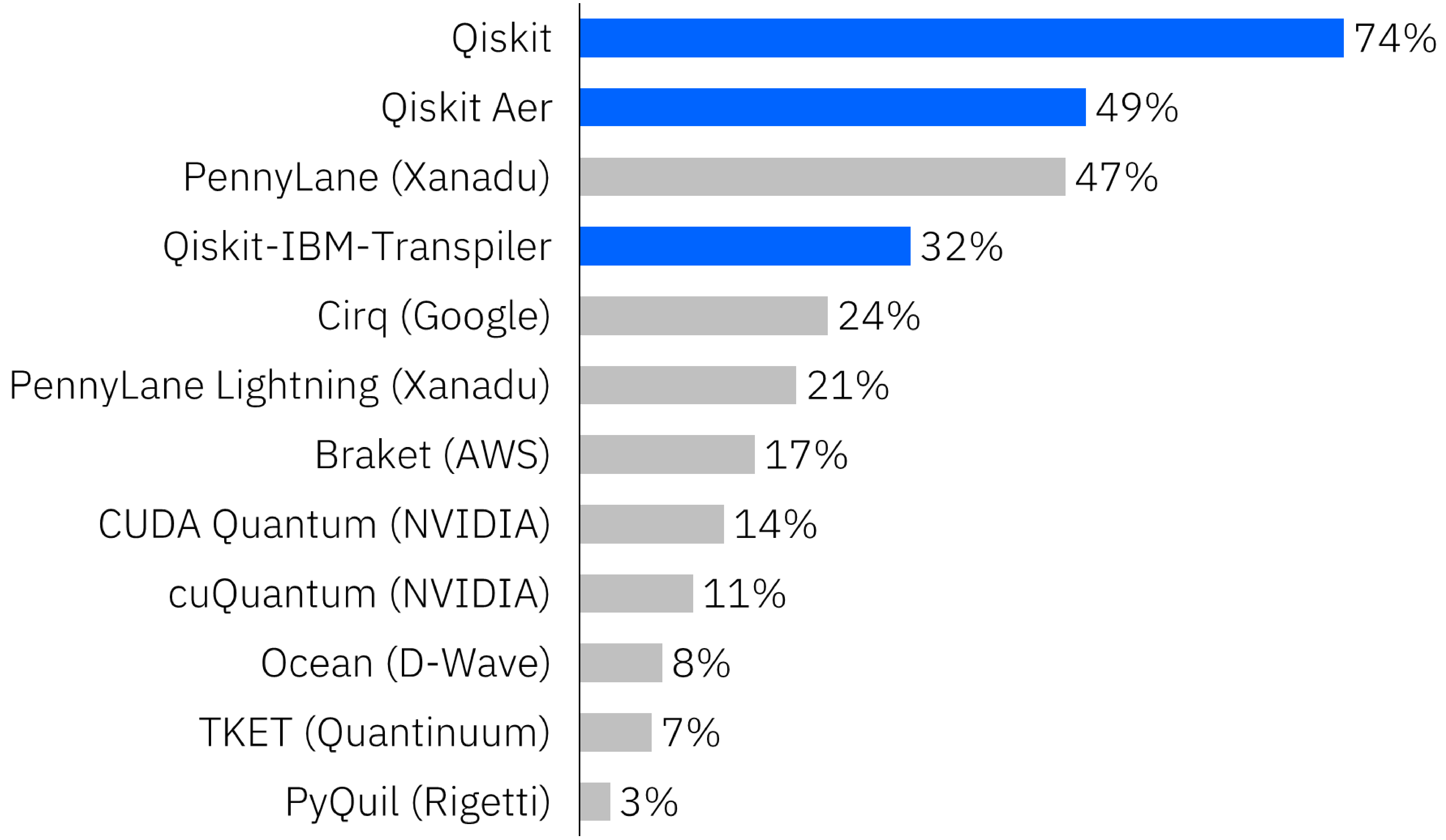
What really matters when choosing a development platform?

Ease of use: Is there comprehensive documentation?

Stability: Is the code base stable and updated on a regular cadence?

Performance: Can I run my workloads fast?

Full-stack Development Platforms Adoption



Source: Unitary Foundation 2024 Quantum Open Source Software Survey

IBM

Qiskit is the most built-upon quantum platform

Key factors

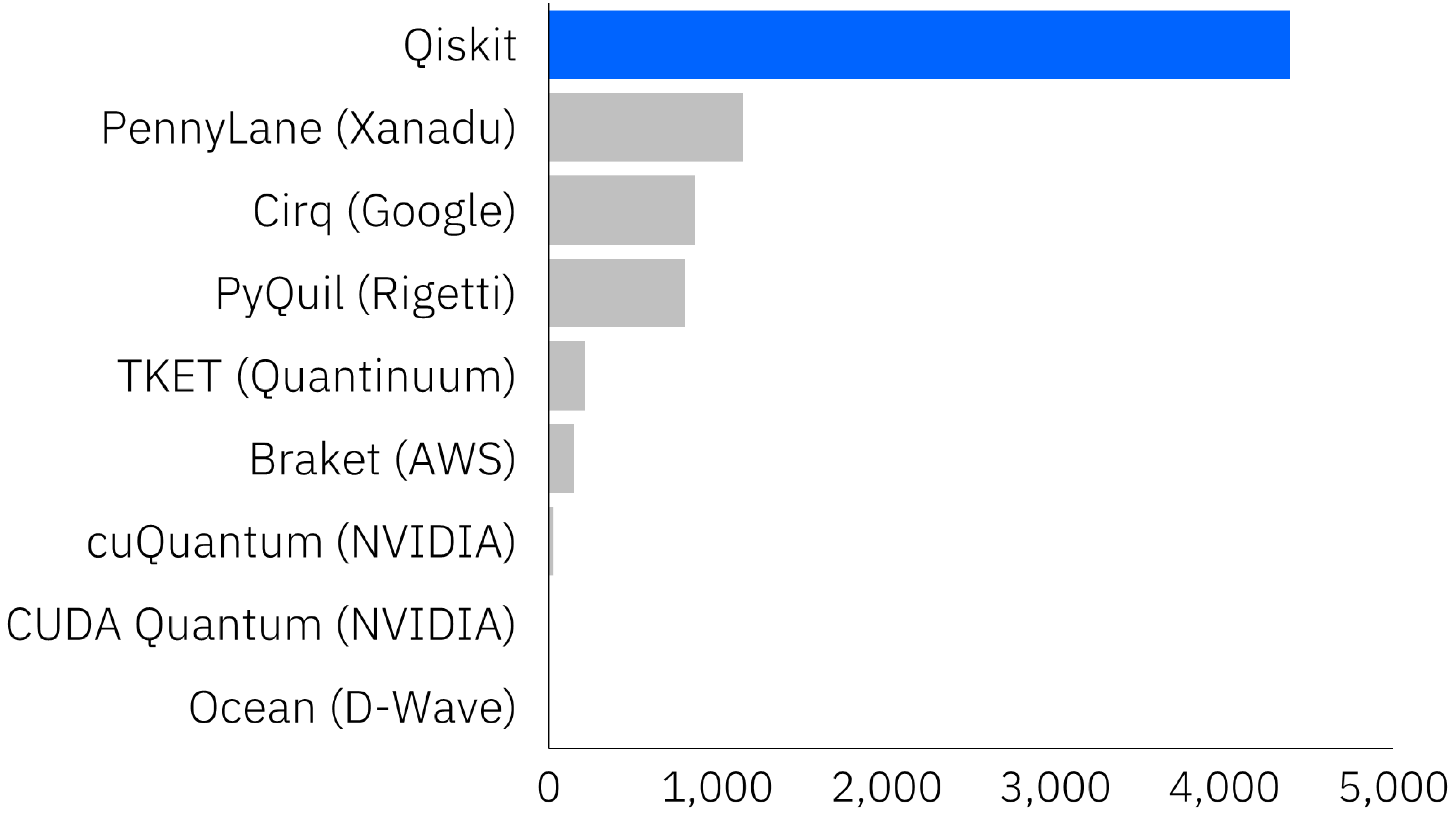
What really matters when building upon a development platform?

Ease of Use: Can I easily integrate the code base into my workflow?

Extensibility: Can others use my workflow?

Performance: Can my users run their workloads fast using my solution?

Quantum SDK Number of Dependent Projects



Source: GitHub insights dependency graphs for PennyLaneAI/pennylane, quantumlib/Cirq, rigetti/pyquil, CQCL/tket, amazon-braket/amazon-braket-schemas-python, NVIDIA/cuQuantum, NVIDIA/cuda-quantum, dwavesystems/dwave-ocean-sdk.
Qiskit data taken from GitHub Insights Dependency API, with duplicate entries within the Qiskit-terra package removed
Data as of 01/22/25

Scaling with Quantum algorithm engineering

Our goal: provide expertise across key areas of quantum computing, including

a. problem co-design, b. mapping, c. scalability, d. hardware execution, and e. performance optimization.

.

What we look for: Subject matter experts looking to run classically hard problems on quantum hardware at scale.

Domains where our team has expertise: Healthcare and life sciences, High Energy Physics, Condensed Matter Physics, Chemistry, Optimization, and Materials science.

Please get in touch: ibrahim.shehzad@ibm.com

IBM