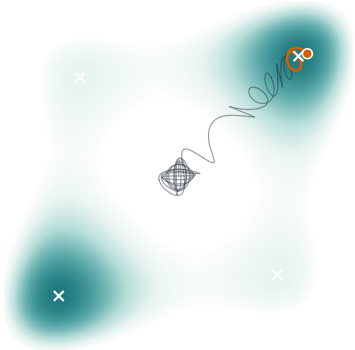
2nd Slovenian Quantum Computing Forum

Closing the quantum–classical scaling gap in approximate optimization



Jakub Pawłowski, Paweł Tarasiuk, Jan Tuziemski, Łukasz Pawela, Bartłomiej Gardas

Wrocław University of Science and Technology • Quantumz.io
University of Gdańsk • IITiS PAN, Gliwice

Novo mesto, Slovenia • 5 November 2026

RUDOLFOVO QMZ Politechnika Wrocławska IITiS

The team behind this work



Jakub Pawłowski
Wrocław Tech
Quantumz.io



Paweł Tarasiuk
Quantumz.io



Jan Tuziemski
Univ. of Gdańsk
Quantumz.io



Łukasz Pawela
IITiS PAN
Quantumz.io



Bartłomiej Gardas
IITiS PAN
Quantumz.io

Quantumz.io: a Polish deep tech startup aiming to accelerate the shift to quantum with physics-inspired and hybrid QUBO solvers

Toward quantum scaling advantage in approximate optimization
J. Pawłowski *et al.*, PR Applied **26**, 014024 (2026)

VeloxQ: A fast and efficient QUBO solver
J. Pawłowski *et al.*, arXiv:2501.19221

What is quantum advantage?

Quantum advantage: a quantum computer performs a (useful) task **better than the best classical computer** — under appropriate definition of “better”.

What is quantum advantage?

Quantum advantage: a quantum computer performs a (useful) task **better than the best classical computer** — under appropriate definition of “better”.

A fair comparison needs three ingredients:

1 Well defined task

Ideally something useful.

What is quantum advantage?

Quantum advantage: a quantum computer performs a (useful) task **better than the best classical computer** — under appropriate definition of “better”.

A fair comparison needs three ingredients:

1 Well defined task

Ideally something useful.

2 State-of-the-art classical computation

Classical methods keep improving.

What is quantum advantage?

Quantum advantage: a quantum computer performs a (useful) task **better than the best classical computer** — under appropriate definition of “better”.

A fair comparison needs three ingredients:

1 Well defined task

Ideally something useful.

2 State-of-the-art classical computation

Classical methods keep improving.

3 Honest measurement

Well defined, reproducible protocols.

Quantum advantage comes in several flavours

Theoretical

Complexity-theoretic separations.

e.g. exponential in Simon's problem

Quantum advantage comes in several flavours

Theoretical

Complexity-theoretic separations.

e.g. exponential in Simon's problem

Empirical scaling

A heuristic whose cost grows more slowly with problem size.

e.g. annealers on a family of spin glasses.

Quantum advantage comes in several flavours

Theoretical

Complexity-theoretic separations.

e.g. exponential in Simon's problem

Empirical scaling

A heuristic whose cost grows more slowly with problem size.

e.g. annealers on a family of spin glasses.

Runtime

Less end-to-end time to an equally good answer.

Important for actual user experience.

Quantum advantage comes in several flavours

Theoretical

Complexity-theoretic separations.

e.g. exponential in Simon's problem

Empirical scaling

A heuristic whose cost grows more slowly with problem size.

e.g. annealers on a family of spin glasses.

Runtime

Less end-to-end time to an equally good answer.

Important for actual user experience.

Other resources

Energy, sensing and metrology, sample complexity, ...

This talk is about empirical scaling.

Random circuit sampling

The task

Run a random quantum circuit, measure all qubits, repeat millions of times; the output strings must follow the circuit's statistics.

A benchmark chosen to be hard for classical computers, not a practical application.

2019 Google Sycamore, 53 qubits: a million samples in about 200 s, estimated 10,000 years on a supercomputer

F. Arute *et al.*, Nature 574, 505 (2019); F. Pan, K. Chen, P. Zhang, PRL 129, 090502 (2022); X. Zhao *et al.*, arXiv:2406.18889 (Natl. Sci. Rev. 2025); A. Morvan *et al.*, Nature 634, 328 (2024).

Random circuit sampling

The task

Run a random quantum circuit, measure all qubits, repeat millions of times; the output strings must follow the circuit's statistics.

A benchmark chosen to be hard for classical computers, not a practical application.

2019 Google Sycamore, 53 qubits: a million samples in about 200 s, estimated 10,000 years on a supercomputer

2022 Tensor networks: the same task in about 15 hours on 512 GPUs

F. Arute *et al.*, Nature 574, 505 (2019); F. Pan, K. Chen, P. Zhang, PRL 129, 090502 (2022); X. Zhao *et al.*, arXiv:2406.18889 (Natl. Sci. Rev. 2025); A. Morvan *et al.*, Nature 634, 328 (2024).

Random circuit sampling

The task

Run a random quantum circuit, measure all qubits, repeat millions of times; the output strings must follow the circuit's statistics.

A benchmark chosen to be hard for classical computers, not a practical application.

2019 Google Sycamore, 53 qubits: a million samples in about 200 s, estimated 10,000 years on a supercomputer

2022 Tensor networks: the same task in about 15 hours on 512 GPUs

2024 1,432 GPUs: seven times faster than the Sycamore experiment itself

F. Arute *et al.*, Nature 574, 505 (2019); F. Pan, K. Chen, P. Zhang, PRL 129, 090502 (2022); X. Zhao *et al.*, arXiv:2406.18889 (Natl. Sci. Rev. 2025); A. Morvan *et al.*, Nature 634, 328 (2024).

Random circuit sampling

The task

Run a random quantum circuit, measure all qubits, repeat millions of times; the output strings must follow the circuit's statistics.

A benchmark chosen to be hard for classical computers, not a practical application.

2019 Google Sycamore, 53 qubits: a million samples in about 200 s, estimated 10,000 years on a supercomputer

2022 Tensor networks: the same task in about 15 hours on 512 GPUs

2024 1,432 GPUs: seven times faster than the Sycamore experiment itself

2024 Google, 67 qubits: lower noise, deeper circuits, beyond known classical methods — unrefuted so far

F. Arute *et al.*, Nature 574, 505 (2019); F. Pan, K. Chen, P. Zhang, PRL 129, 090502 (2022); X. Zhao *et al.*, arXiv:2406.18889 (Natl. Sci. Rev. 2025); A. Morvan *et al.*, Nature 634, 328 (2024).

Quantum simulation on an annealer

The task

Change a quantum magnet suddenly (a “quench”) and measure how correlations between its spins build up.

A real physics problem, hard for classical computers because of entanglement.

2025 D-Wave annealers: quenches of 2D, 3D and infinite-dimensional spin glasses, leading classical methods could not match the accuracy

A. D. King *et al.*, Science 388, 199 (2025); J. Tindall *et al.*, arXiv:2503.05693; L. Mauron, G. Carleo, arXiv:2503.08247; A. D. King *et al.*, arXiv:2504.06283.

Quantum simulation on an annealer

The task

Change a quantum magnet suddenly (a “quench”) and measure how correlations between its spins build up.

A real physics problem, hard for classical computers because of entanglement.

2025 D-Wave annealers: quenches of 2D, 3D and infinite-dimensional spin glasses, leading classical methods could not match the accuracy

2025 Tensor networks: 2D and 3D with “modest computational resources”

A. D. King *et al.*, Science 388, 199 (2025); J. Tindall *et al.*, arXiv:2503.05693; L. Mauron, G. Carleo, arXiv:2503.08247; A. D. King *et al.*, arXiv:2504.06283.

Quantum simulation on an annealer

The task

Change a quantum magnet suddenly (a “quench”) and measure how correlations between its spins build up.

A real physics problem, hard for classical computers because of entanglement.

2025 D-Wave annealers: quenches of 2D, 3D and infinite-dimensional spin glasses, leading classical methods could not match the accuracy

2025 Tensor networks: 2D and 3D with “modest computational resources”

2025 Neural-network states: small errors up to 128 spins in 3D

A. D. King *et al.*, Science 388, 199 (2025); J. Tindall *et al.*, arXiv:2503.05693; L. Mauron, G. Carleo, arXiv:2503.08247; A. D. King *et al.*, arXiv:2504.06283.

Quantum simulation on an annealer

The task

Change a quantum magnet suddenly (a “quench”) and measure how correlations between its spins build up.

A real physics problem, hard for classical computers because of entanglement.

2025 D-Wave annealers: quenches of 2D, 3D and infinite-dimensional spin glasses, leading classical methods could not match the accuracy

2025 Tensor networks: 2D and 3D with “modest computational resources”

2025 Neural-network states: small errors up to 128 spins in 3D

2025 Reply: “a significant improvement”, but the hardest cases remain unsolved

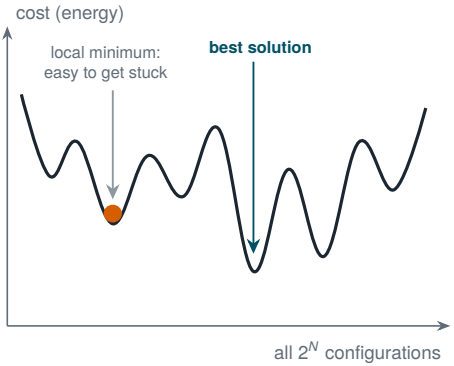
A. D. King *et al.*, Science 388, 199 (2025); J. Tindall *et al.*, arXiv:2503.05693; L. Mauron, G. Carleo, arXiv:2503.08247; A. D. King *et al.*, arXiv:2504.06283.

Approximate optimisation

- Many decisions are **binary choices**: which truck takes which route, which asset to hold, which machine runs which job

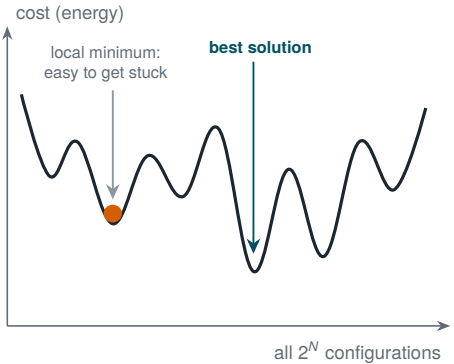
Approximate optimisation

- Many decisions are **binary choices**: which truck takes which route, which asset to hold, which machine runs which job
- Can be formulated as QUBO problems, equivalent to the description of interacting magnets (Ising model)



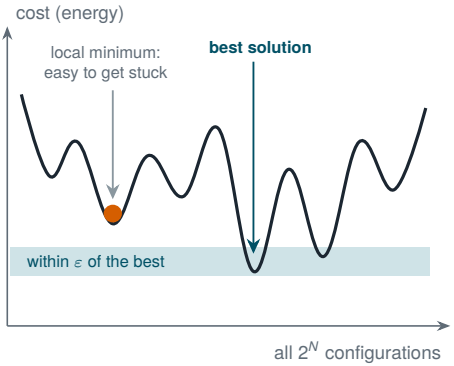
Approximate optimisation

- Many decisions are **binary choices**: which truck takes which route, which asset to hold, which machine runs which job
- Can be formulated as QUBO problems, equivalent to the description of interacting magnets (Ising model)
- 2^N possibilities: a few hundred variables give more combinations than atoms in the observable universe



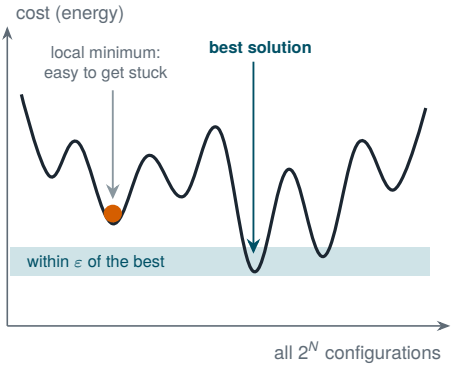
Approximate optimisation

- Many decisions are **binary choices**: which truck takes which route, which asset to hold, which machine runs which job
- Can be formulated as QUBO problems, equivalent to the description of interacting magnets (Ising model)
- 2^N possibilities: a few hundred variables give more combinations than atoms in the observable universe
- The exact optimum is hard to find, so we ask for **approximate** answers: within ε (e.g. 1%) of the best



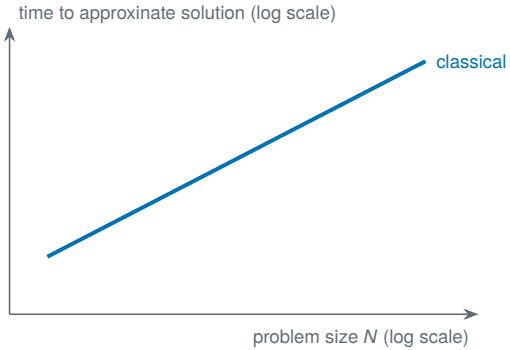
Approximate optimisation

- Many decisions are **binary choices**: which truck takes which route, which asset to hold, which machine runs which job
- Can be formulated as QUBO problems, equivalent to the description of interacting magnets (Ising model)
- 2^N possibilities: a few hundred variables give more combinations than atoms in the observable universe
- The exact optimum is hard to find, so we ask for **approximate** answers: within ε (e.g. 1%) of the best



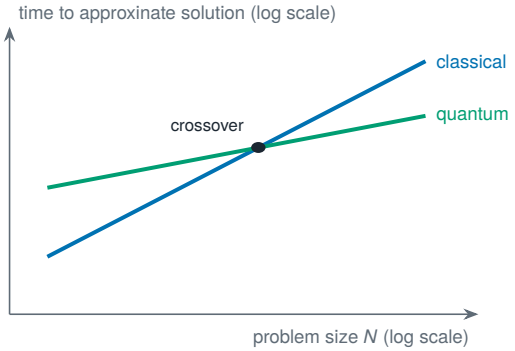
Quantum annealers are machines built for exactly this kind of problem.

Runtime vs. scaling



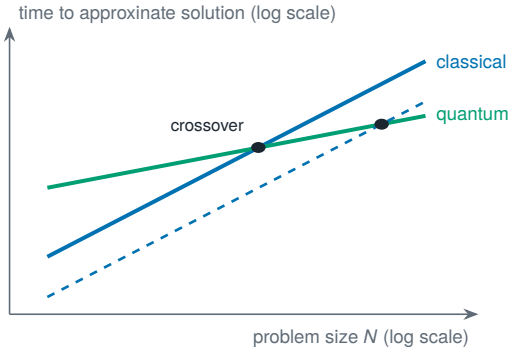
- **Runtime:** depends on many fine details.
- **Scaling**: how the time grows with problem size, e.g. $t \propto N^{\alpha}$.

Runtime vs. scaling



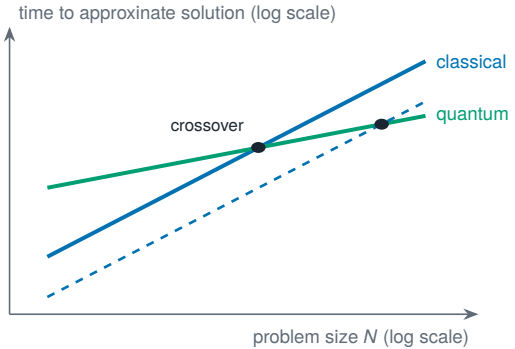
- **Runtime:** depends on many fine details.
- **Scaling**: how the time grows with problem size, e.g. $t \propto N^{\alpha}$.
- A smaller slope wins for large enough problems.

Runtime vs. scaling



- **Runtime:** depends on many fine details.
- **Scaling**: how the time grows with problem size, e.g. $t \propto N^{\alpha}$.
- A smaller slope wins for large enough problems.
- Faster hardware shifts the curve down, only a better method changes the slope.

Runtime vs. scaling



- **Runtime:** depends on many fine details.
- **Scaling**: how the time grows with problem size, e.g. $t \propto N^{\alpha}$.
- A smaller slope wins for large enough problems.
- Faster hardware shifts the curve down, only a better method changes the slope.

Scaling advantage: smaller α than the best classical method, for the same quality target.

Scaling advantage is a promise about future runtime advantage.

A claim of quantum scaling advantage

$$H = -\sum_{i<j} J_{ij}\, s_i s_j, \quad s_i = \pm 1$$

- D-Wave Advantage 4.1 with error suppression, **quantum annealing correction** (QAC): >1,300 logical qubits

A claim of quantum scaling advantage

$$H = -\sum_{i<j} J_{ij}\, s_i s_j, \quad s_i = \pm 1$$

- D-Wave Advantage 4.1 with error suppression, **quantum annealing correction** (QAC): >1,300 logical qubits
- Test problems: QAC-native spin glasses with 142 to 1,322 variables

A claim of quantum scaling advantage

$$H = -\sum_{i<j} J_{ij}\, s_i s_j, \quad s_i = \pm 1$$

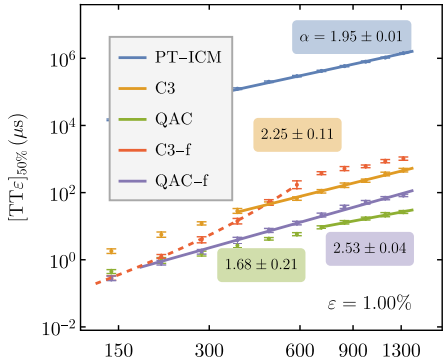
- D-Wave Advantage 4.1 with error suppression, **quantum annealing correction** (QAC): >1,300 logical qubits
- Test problems: QAC-native spin glasses with 142 to 1,322 variables
- Time to reach within ε of the optimal energy

A claim of quantum scaling advantage

$$H = -\sum_{i<j} J_{ij}\, s_i s_j, \quad s_i = \pm 1$$

- D-Wave Advantage 4.1 with error suppression, **quantum annealing correction** (QAC): >1,300 logical qubits
- Test problems: QAC-native spin glasses with 142 to 1,322 variables
- Time to reach within ε of the optimal energy
- Classical baseline: parallel tempering with isoenergetic cluster moves (**PT-ICM**)

A claim of quantum scaling advantage

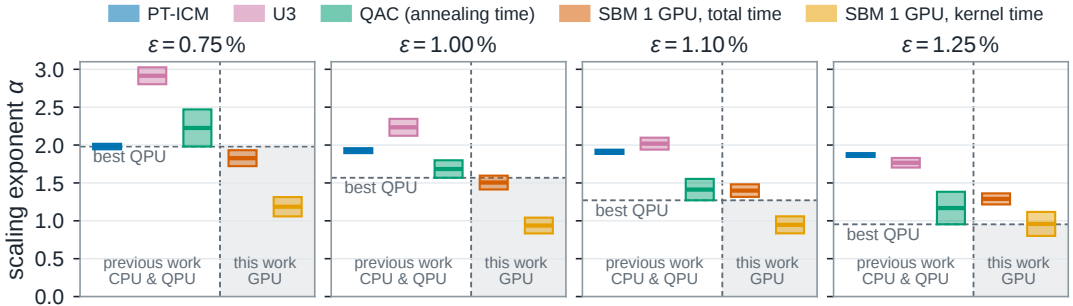


$$H = -\sum_{i<j} J_{ij}\, s_i s_j, \quad s_i = \pm 1$$

- D-Wave Advantage 4.1 with error suppression, **quantum annealing correction** (QAC): >1,300 logical qubits
- Test problems: QAC-native spin glasses with 142 to 1,322 variables
- Time to reach within ε of the optimal energy
- Classical baseline: parallel tempering with isoenergetic cluster moves (**PT-ICM**)
- At $\varepsilon = 1\%$: $\alpha_{\mathrm{QAC}} = 1.68 \pm 0.21$ vs. $\alpha_{\mathrm{PT\text{-}ICM}} = 1.95 \pm 0.01$

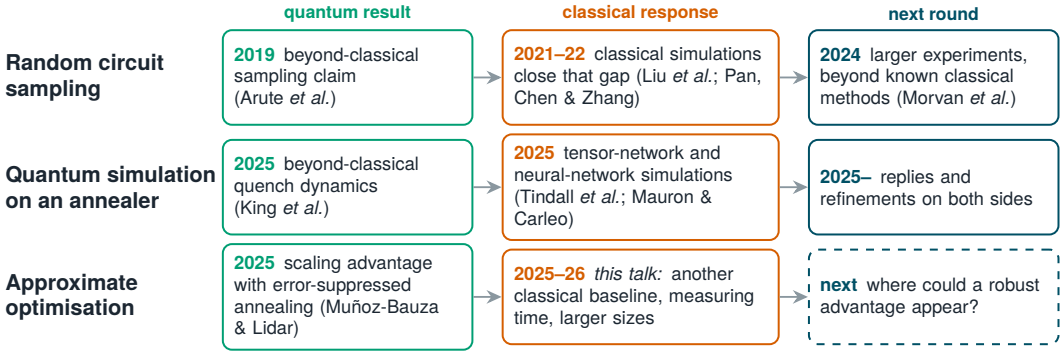
Figure source: part of Fig. 1 ($\varepsilon = 1\%$), H. Muñoz-Bauza & D. A. Lidar, PRL **134**, 160601 (2025), CC BY 4.0.

Main result



SBM on GPU closes the quantum–classical scaling gap.

Quantum claims and classical responses



Back-and-forth between quantum and classical drives progress.

Questions we asked

1 **Baseline: does the gap survive a different *kind* of classical algorithm?** → simulated bifurcation

Questions we asked

1. **Baseline: does the gap survive a different *kind* of classical algorithm?** → simulated bifurcation

2. **Time measurement: what does “time” mean for an annealer?** → annealing time vs. runtime

Questions we asked

1. **Baseline: does the gap survive a different *kind* of classical algorithm?** → simulated bifurcation
2. **Time measurement: what does "time" mean for an annealer?** → annealing time vs. runtime
3. **Size: are $N \leq 1{,}322$ spins enough to infer the asymptotic scaling?** → instances up to $N \approx 4 \times 10^4$

Questions we asked

1	**Baseline: does the gap survive a different *kind* of classical algorithm?**	→ simulated bifurcation
2	**Time measurement: what does "time" mean for an annealer?**	→ annealing time vs. runtime
3	**Size: are $N \leq 1{,}322$ spins enough to infer the asymptotic scaling?**	→ instances up to $N \approx 4 \times 10^4$
4	**Future: where could annealers still win?**	→ 3D spin glasses

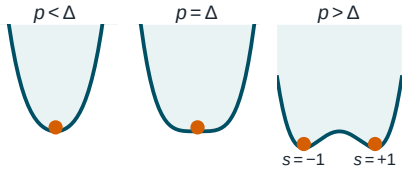
Quantum-inspired classical dynamics

Adiabatic simulated bifurcation (Goto *et al.* 2019)

Kerr parametric oscillators → classical amplitudes $x_i + \mathrm{i}\, y_i$

$$\dot{x}_i = \Delta\, y_i, \qquad \dot{y}_i = -\frac{\partial V}{\partial x_i}, \qquad s_i = \mathrm{sign}(x_i)$$

$$V = \sum_i \left[\tfrac{\Delta - p(t)}{2} x_i^2 + \tfrac{K}{4} x_i^4 \right] - \tfrac{\xi_0}{2} \sum_{ij} J_{ij} x_i x_j$$



one oscillator: one well → two wells

Deterministic dynamics, no thermal noise, naturally parallelizable

Quantum evolution and classical trajectory

Improved variants of simulated bifurcation

Time-to-ε: a measure of both quality and time

$$\mathrm{TT}\varepsilon = t_f \, \frac{\ln(1 - 0.99)}{\ln\left(1 - p_\varepsilon\right)}$$

- p_ε: probability of reaching $E \leq E_0 + \varepsilon |E_0|$
- t_f: time of one run — **how to measure?**

Time-to-ε: a measure of both quality and time

$$\mathrm{TT}\varepsilon = t_f \, \frac{\ln(1 - 0.99)}{\ln\left(1 - p_\varepsilon\right)}$$

- p_ε: probability of reaching $E \leq E_0 + \varepsilon |E_0|$
- t_f: time of one run — **how to measure?**
- median over ensemble of instances, then optimise the solver parameters θ at each size

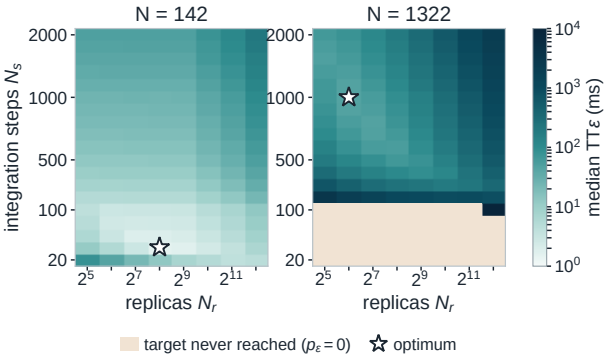
$$\mathrm{TT}\varepsilon^\star(N) = \min_\theta \operatorname{median}_I \mathrm{TT}\varepsilon(I;\theta) \propto N^\alpha$$

Time-to-ε: a measure of both quality and time

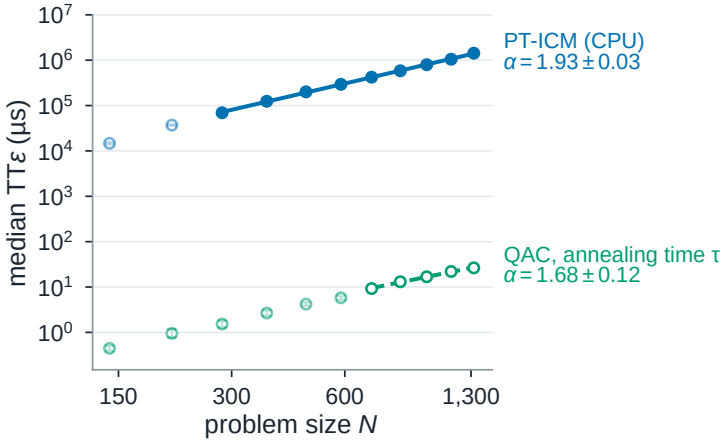
$$\mathrm{TT}\varepsilon = t_f \frac{\ln(1-0.99)}{\ln\left(1-p_\varepsilon\right)}$$

- p_ε: probability of reaching $E \leq E_0 + \varepsilon|E_0|$
- t_f: time of one run — **how to measure?**
- median over ensemble of instances, then optimise the solver parameters θ at each size

$$\mathrm{TT}\varepsilon^\star(N) = \min_\theta \operatorname*{median}_I \mathrm{TT}\varepsilon(I;\theta) \propto N^\alpha$$



Closing the gap with SBM

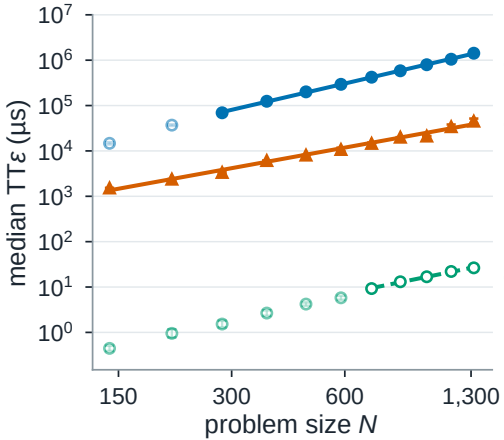


Sidon-28, $\varepsilon = 1\%$
median TT$\varepsilon \propto N^{\alpha}$

PT-ICM, CPU	1.93 ± 0.03
QAC, τ (set)	1.68 ± 0.12

PT-ICM and QAC points: H. Muñoz-Bauza and D. A. Lidar, Phys. Rev. Lett. **134**, 160601 (2025)

Closing the gap with SBM



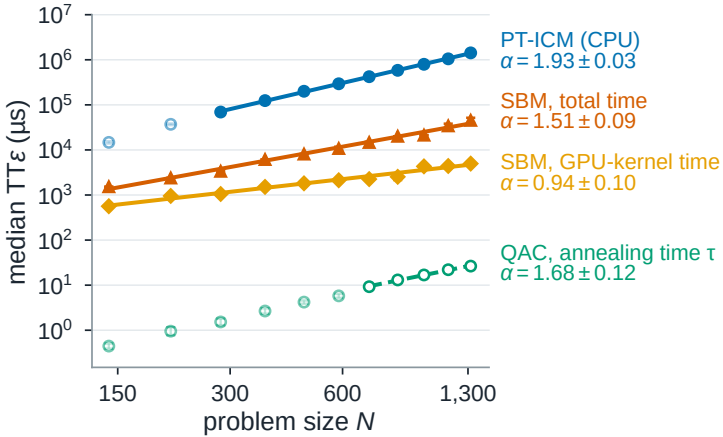
Sidon-28, $\varepsilon = 1\%$
median $\mathrm{TT}\varepsilon \propto N^{\alpha}$

PT-ICM, CPU	1.93 ± 0.03
QAC, τ (set)	1.68 ± 0.12
SBM, total	1.51 ± 0.09

All overheads included,
one GPU matches QAC.

PT-ICM and QAC points: H. Muñoz-Bauza and D. A. Lidar, Phys. Rev. Lett. **134**, 160601 (2025)

Closing the gap with SBM



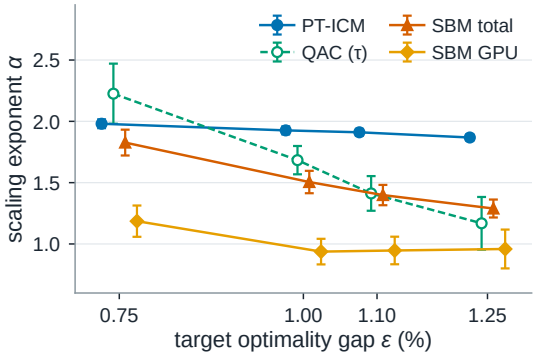
Sidon-28, $\varepsilon = 1\%$
median $\mathrm{TT}\varepsilon \propto N^{\alpha}$

PT-ICM, CPU	1.93 ± 0.03
QAC, τ (set)	1.68 ± 0.12
SBM, total	1.51 ± 0.09
SBM, GPU-only	0.94 ± 0.10

All overheads included,
one GPU matches QAC.

PT-ICM and QAC points: H. Muñoz-Bauza and D. A. Lidar, Phys. Rev. Lett. **134**, 160601 (2025)

The same picture at every target quality



- **QAC** beats **PT-ICM** for $\varepsilon \geq 1\%$: the reported advantage
- **SBM total** is below **QAC** at 0.75–1.00 % and comparable within errors at 1.10–1.25 %
- **SBM GPU** is lower throughout
- As $\varepsilon \to 0$ exponents grow: exact optimisation stays hard

With all overheads, one GPU matches or beats QAC's annealing-time scaling.

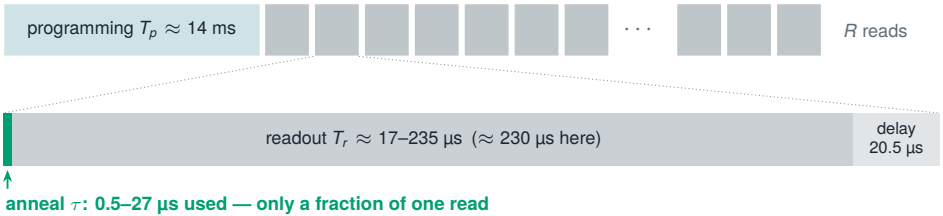
PT-ICM and QAC data from PRL **134**, 160601 (2025).

Anatomy of a D-Wave sampling job



D-Wave documentation, "Operation and timing". Advantage 4.1 values

Anatomy of a D-Wave sampling job



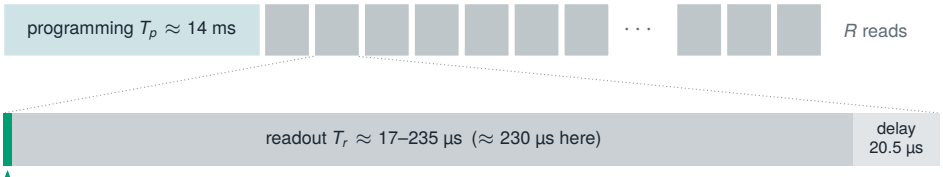
D-Wave documentation, "Operation and timing". Advantage 4.1 values

Anatomy of a D-Wave sampling job



D-Wave documentation, "Operation and timing". Advantage 4.1 values

Anatomy of a D-Wave sampling job

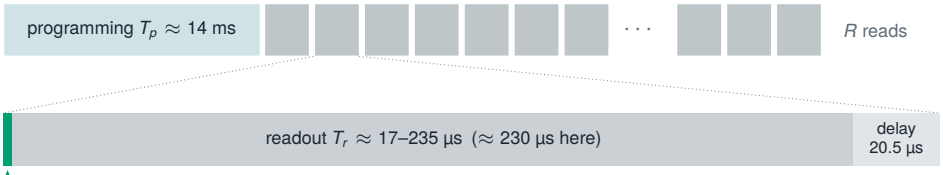


— **annealing time** τ set by the user

— **QPU access time** $T = T_p + T_s,\ T_s \approx R\,(\tau + T_r + T_d)$: reported by D-Wave

D-Wave documentation, "Operation and timing". Advantage 4.1 values

Anatomy of a D-Wave sampling job



- **annealing time τ** set by the user
- **QPU access time** $T = T_p + T_s$, $T_s \approx R\,(\tau + T_r + T_d)$: reported by D-Wave
- **external runtime** measured by the client: adds network, queue, …

D-Wave documentation, "Operation and timing". Advantage 4.1 values

Annealing time is not runtime

Annealing time τ

- directly impacts solution quality
- non-measurable by the user
- physics of the annealing

Annealing time is not runtime

Annealing time τ

- directly impacts solution quality
- non-measurable by the user
- physics of the annealing

Runtime T

- includes programming, annealing, read-out and delays
- measurable by the user
- time to get the approximate answer

Annealing time is not runtime

Annealing time τ

- directly impacts solution quality
- non-measurable by the user
- physics of the annealing

Runtime T

- includes programming, annealing, read-out and delays
- measurable by the user
- time to get the approximate answer

Analogous split for SBM:

- **GPU-kernel time** $\approx$ the anneal
- **total time** with CPU$\leftrightarrow$GPU transfers $\approx$ QPU access.

Annealing time is not runtime

Annealing time τ

- directly impacts solution quality
- non-measurable by the user
- physics of the annealing

Runtime T

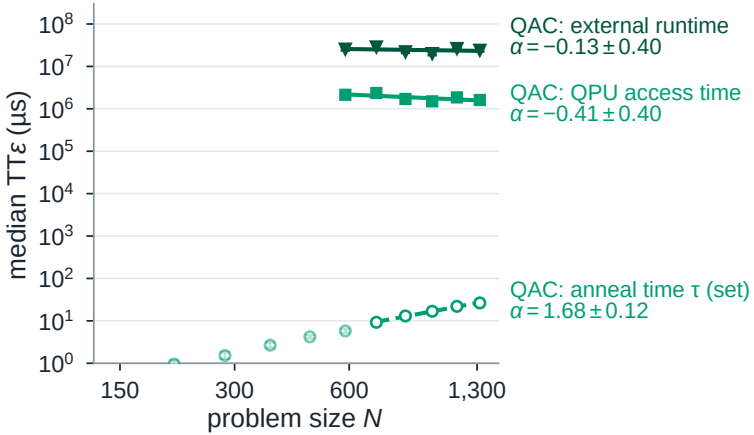
- includes programming, annealing, read-out and delays
- measurable by the user
- time to get the approximate answer

Analogous split for SBM:

- **GPU-kernel time** $\approx$ the anneal
- **total time** with CPU$\leftrightarrow$GPU transfers $\approx$ QPU access.

Advantage based on actual runtime is more operationally relevant.

Impact of the time measurement on QAC scaling



- Only the **set** τ grows with N
- **Measured time** results are flat: α consistent with zero

$\Rightarrow$ no exponent can be extracted for $N \leq 1{,}322$

QAC τ points from H. Muñoz-Bauza and D. A. Lidar, Phys. Rev. Lett. **134**, 160601 (2025)

Summary

1. A true scaling advantage claim needs the **best classical baseline** and **proper time measurement**.

Summary

1. A true scaling advantage claim needs the **best classical baseline** and **proper time measurement**.
2. A quantum-inspired classical solver — **SBM on one GPU** — matches or beats the reported QAC scaling, even when the annealer's runtime is its annealing time.

Summary

1. A true scaling advantage claim needs the **best classical baseline** and **proper time measurement**.
2. A quantum-inspired classical solver — **SBM on one GPU** — matches or beats the reported QAC scaling, even when the annealer's runtime is its annealing time.
3. Today's annealer runtime is dominated by **overheads** (programming, readout), the anneal itself is a small fraction.

Summary

1. A true scaling advantage claim needs the **best classical baseline** and **proper time measurement**.
2. A quantum-inspired classical solver — **SBM on one GPU** — matches or beats the reported QAC scaling, even when the annealer's runtime is its annealing time.
3. Today's annealer runtime is dominated by **overheads** (programming, readout), the anneal itself is a small fraction.
4. Beyond present QPU sizes the classical scaling stays robust: $\alpha \approx 1.6$ at $\varepsilon = 1\%$ up to $N \approx 4 \times 10^4$.

Summary

1. A true scaling advantage claim needs the **best classical baseline** and **proper time measurement**.
2. A quantum-inspired classical solver — **SBM on one GPU** — matches or beats the reported QAC scaling, even when the annealer's runtime is its annealing time.
3. Today's annealer runtime is dominated by **overheads** (programming, readout), the anneal itself is a small fraction.
4. Beyond present QPU sizes the classical scaling stays robust: $\alpha \approx 1.6$ at $\varepsilon = 1\%$ up to $N \approx 4 \times 10^4$.
5. **Promising routes for annealers**: problem classes such as 3D spin glasses, where SBM does not match the annealer.

What a convincing scaling/runtime advantage needs?

1 Measure runtime

Time the actual execution, not settings or runtime component.

What a convincing scaling/runtime advantage needs?

1 Measure runtime

Time the actual execution, not settings or runtime component.

2 Count all overheads

Readout, compilation, data transfer, tuning.

What a convincing scaling/runtime advantage needs?

1 Measure runtime

Time the actual execution, not settings or runtime component.

2 Count all overheads

Readout, compilation, data transfer, tuning.

3 Standard metrics

TTε at small ε; diverse instance families.

What a convincing scaling/runtime advantage needs?

1 Measure runtime

Time the actual execution, not settings or runtime component.

2 Count all overheads

Readout, compilation, data transfer, tuning.

3 Standard metrics

TTε at small ε; diverse instance families.

4 Sufficient size range

Extrapolate with care.

What a convincing scaling/runtime advantage needs?

1 Measure runtime

Time the actual execution, not settings or runtime component.

2 Count all overheads

Readout, compilation, data transfer, tuning.

3 Standard metrics

$\mathrm{TT}\varepsilon$ at small ε; diverse instance families.

4 Sufficient size range

Extrapolate with care.

5 Strong baselines

State-of-the-art classical algorithms on suitable hardware.

What a convincing scaling/runtime advantage needs?

1 Measure runtime

Time the actual execution, not settings or runtime component.

2 Count all overheads

Readout, compilation, data transfer, tuning.

3 Standard metrics

TTε at small ε; diverse instance families.

4 Sufficient size range

Extrapolate with care.

5 Strong baselines

State-of-the-art classical algorithms on suitable hardware.

6 Hybrid null test

Replace the quantum step classically: does performance drop?

VeloxQ: a physics-inspired solver for large problems

- Developed by **Quantumz.io**, runs on conventional hardware of today
- Physics-inspired: simulates a dynamical system

VeloxQ: a physics-inspired solver for large problems

- Developed by **Quantumz.io**, runs on conventional hardware of today
- Physics-inspired: simulates a dynamical system
- Any problem structure: **no embedding** onto a hardware graph

VeloxQ: a physics-inspired solver for large problems

- Developed by **Quantumz.io**, runs on conventional hardware of today
- Physics-inspired: simulates a dynamical system
- Any problem structure: **no embedding** onto a hardware graph
- Scales to $\sim 10^8$ variables on Pegasus/Zephyr-like sparse problems
- Preliminary results: up to $\sim 10^{11}$ variables on selected problems

VeloxQ: a physics-inspired solver for large problems

- Developed by **Quantumz.io**, runs on conventional hardware of today
- Physics-inspired: simulates a dynamical system
- Any problem structure: **no embedding** onto a hardware graph
- Scales to $\sim 10^8$ variables on Pegasus/Zephyr-like sparse problems
- Preliminary results: up to $\sim 10^{11}$ variables on selected problems
- Cloud access through an open **Python SDK**

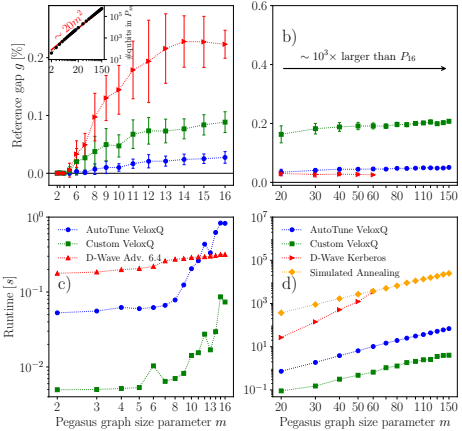
VeloxQ: a physics-inspired solver for large problems

- Developed by **Quantumz.io**, runs on conventional hardware of today
- Physics-inspired: simulates a dynamical system
- Any problem structure: **no embedding** onto a hardware graph
- Scales to $\sim 10^8$ variables on Pegasus/Zephyr-like sparse problems
- Preliminary results: up to $\sim 10^{11}$ variables on selected problems
- Cloud access through an open **Python SDK**
- Heuristic: very good solutions, but no proof of optimality

Benchmarked against

- D-Wave Advantage and Advantage2 (QPU and hybrid)
- Kipu Quantum BF-DCQO (gate-based)
- CPLEX and branch-and-bound
- Brute force and certified solvers
- Simulated bifurcation, parallel annealing

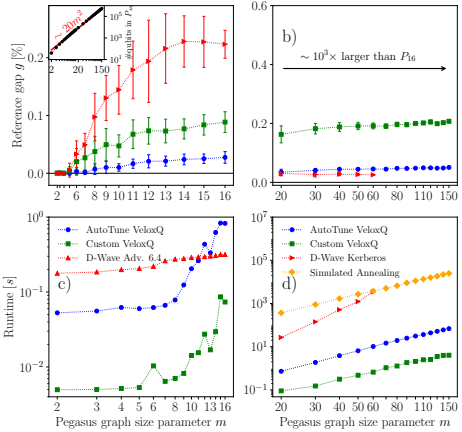
Comparison on Pegasus-native problems



- Random problems **native** to D-Wave Advantage (Pegasus graph)

D-Wave time: full QPU access time, not only the anneal.

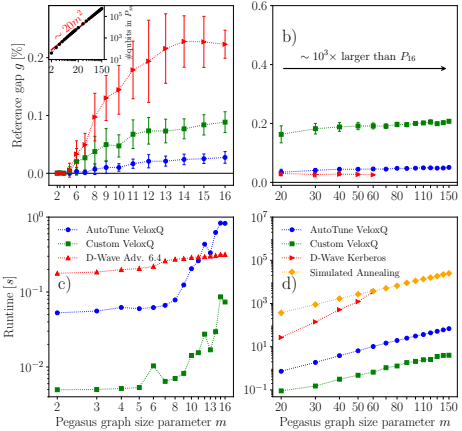
Comparison on Pegasus-native problems



- Random problems **native** to D-Wave Advantage (Pegasus graph)
- **(a)** Solution quality: VeloxQ comparable or better

D-Wave time: full QPU access time, not only the anneal.

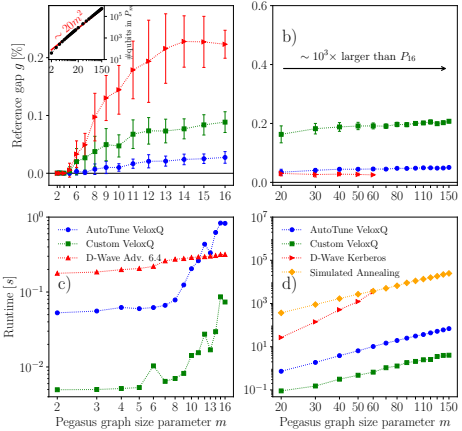
Comparison on Pegasus-native problems



- Random problems **native** to D-Wave Advantage (Pegasus graph)
- **(a)** Solution quality: VeloxQ comparable or better
- **(c)** Time: the QPU is faster than default VeloxQ on the full chip, tuned VeloxQ is faster

D-Wave time: full QPU access time, not only the anneal.

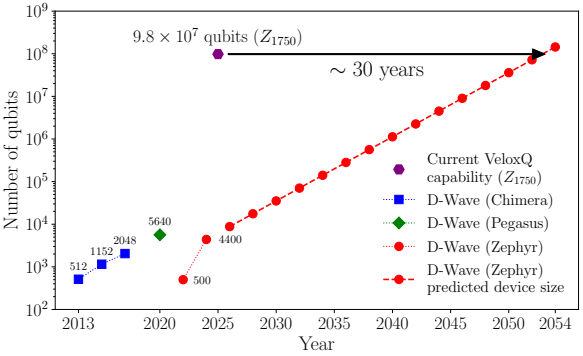
Comparison on Pegasus-native problems



- Random problems **native** to D-Wave Advantage (Pegasus graph)
- **(a)** Solution quality: VeloxQ comparable or better
- **(c)** Time: the QPU is faster than default VeloxQ on the full chip, tuned VeloxQ is faster
- **(b, d)** Beyond the QPU: up to $10^3\times$ larger problems; D-Wave's hybrid solver stops at $\sim 8.5 \times 10^4$ variables, VeloxQ is up to $100\times$ faster

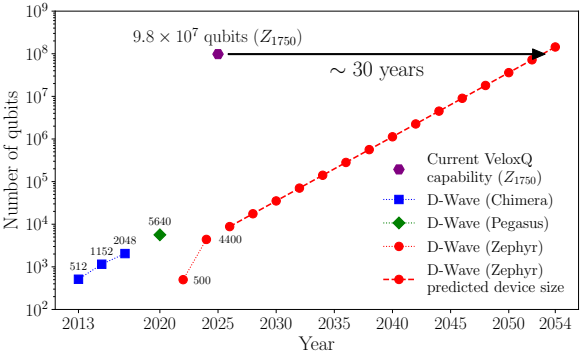
D-Wave time: full QPU access time, not only the anneal.

When could quantum hardware catch up on size?



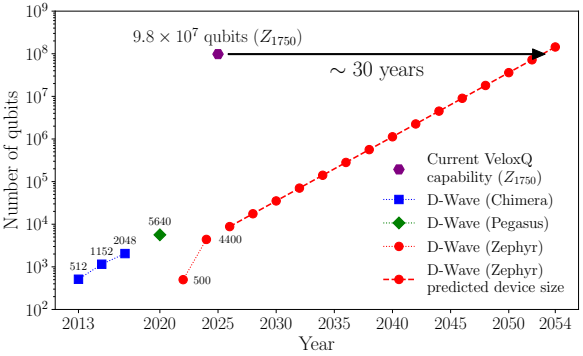
- D-Wave chips: 512 qubits (2013) to a few thousand today

When could quantum hardware catch up on size?



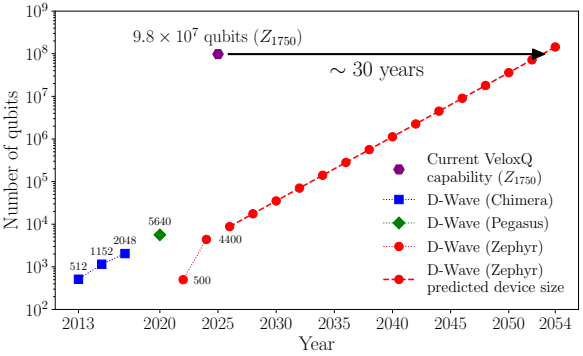
- D-Wave chips: 512 qubits (2013) to a few thousand today
- VeloxQ ran a 9.8×10^7-variable problem on D-Wave's newest (Zephyr) graph (runtime only, quality not checked)

When could quantum hardware catch up on size?



- D-Wave chips: 512 qubits (2013) to a few thousand today
- VeloxQ ran a 9.8×10^7-variable problem on D-Wave's newest (Zephyr) graph (runtime only, quality not checked)
- Doubling every two years: a native device around **2054**

When could quantum hardware catch up on size?



- D-Wave chips: 512 qubits (2013) to a few thousand today
- VeloxQ ran a 9.8×10^7-variable problem on D-Wave's newest (Zephyr) graph (runtime only, quality not checked)
- Doubling every two years: a native device around **2054**
- Simple extrapolation, ignoring engineering limits

Classical solvers already work at sizes quantum chips may need decades to reach.

Using VeloxQ from Python

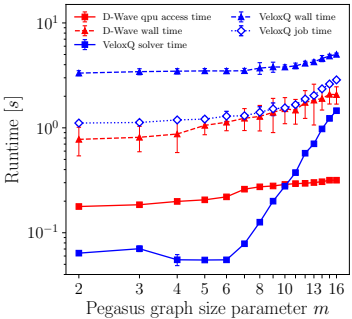
```python
import dimod
from dwave.system import DWaveSampler, EmbeddingComposite
from veloxq_sdk import VeloxQSolver

# a tiny Ising problem: fields h_i, couplings J_ij
bqm = dimod.BinaryQuadraticModel(
    {0: 1.0, 2: -1.0},
    {(0, 1): -1.0, (1, 2): 0.5},
    0.0, dimod.SPIN)

velox = VeloxQSolver().sample(bqm)       # GPU, in the cloud
print('VeloxQ:', velox.first)

qpu = EmbeddingComposite(DWaveSampler())  # D-Wave QPU
print('D-Wave:', qpu.sample(bqm).first)
```

Also accepts NumPy arrays, dictionaries or files. github.com/quantumz-io/veloxq_sdk



On small problems the cloud interface dominates, on both platforms.

Thank you

Questions and discussion

Closing the gap
Phys. Rev. Applied 26, 014024 (2026)

VeloxQ white paper
arXiv:2501.19221

VeloxQ SDK
github.com/quantumz-io/veloxq_sdk

We thank H. Muñoz-Bauza and D. A. Lidar for making their instances and data available.

Supported by NCN Poland, Sonata Bis 10, No. 2020/38/E/ST3/00269 (B.G.), and NCBR Poland, POIR.01.01.01-00-0061/22 (Quantumz.io).

Thank you for your attention!