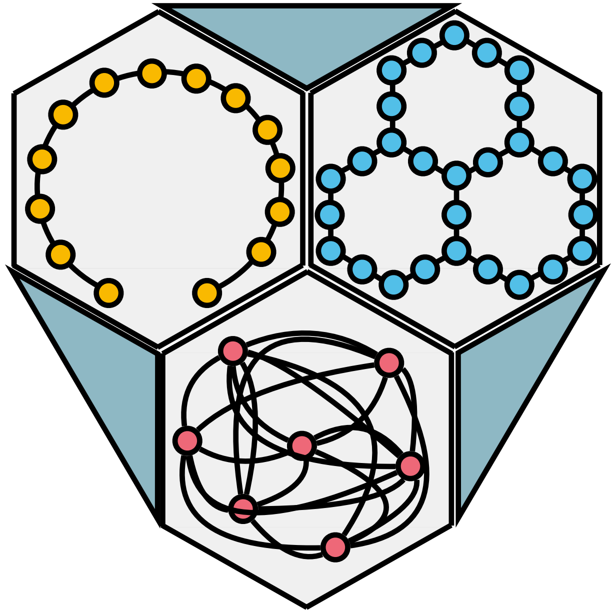


Evaluating the performance of quantum processing units (QPUs) at large width and depth

TQCI seminar dedicated to benchmarks for quantum computers

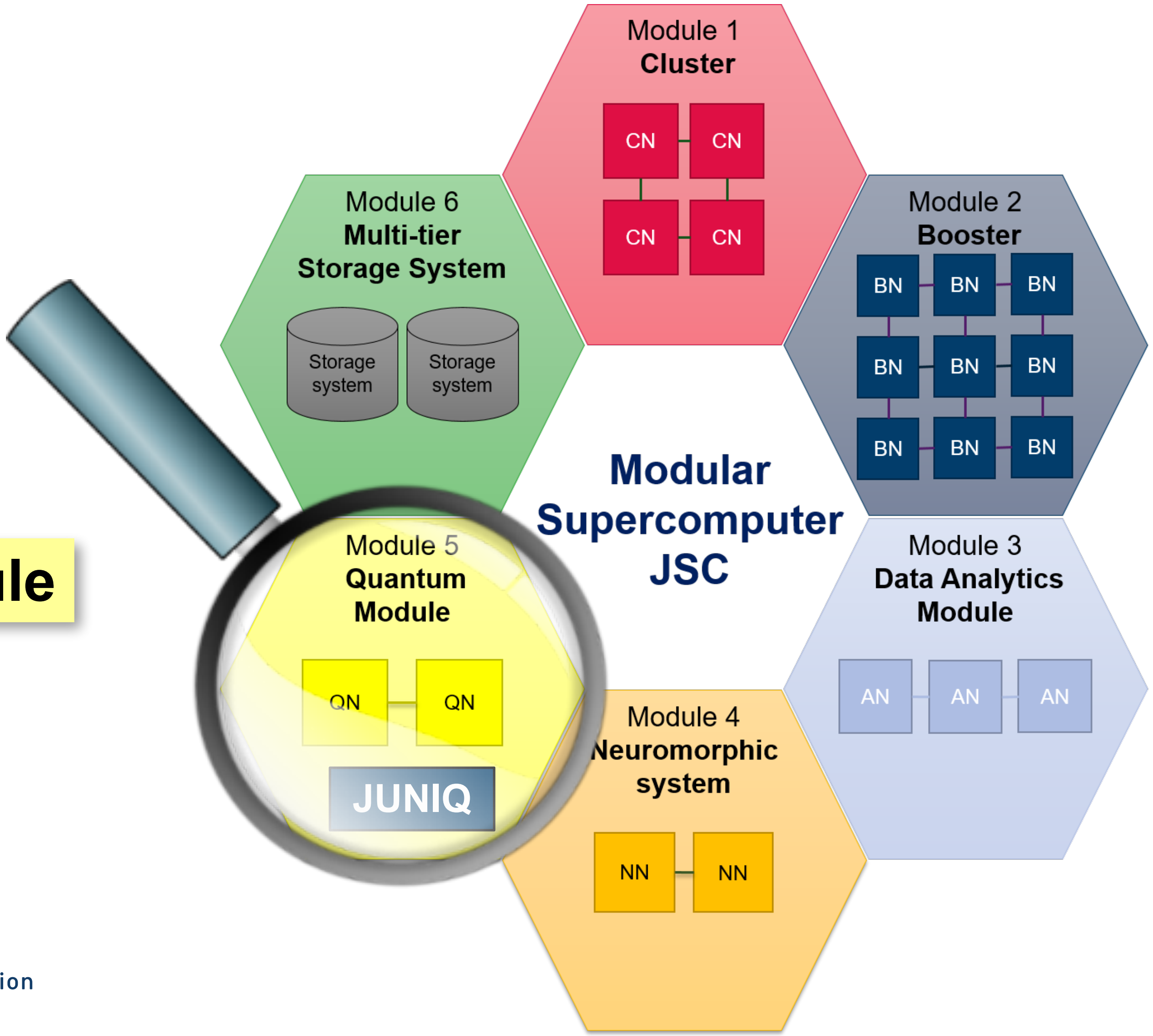
June 16, 2025 I Alejandro Montanez-Barrera | Forschungszentrum Jülich - Jülich Supercomputing Center

https://arxiv.org/abs/2502.06471

JUNIQ - Jülich UNified Infrastructure for Quantum computing

A European quantum computer user facility at the Jülich Supercomputing Centre

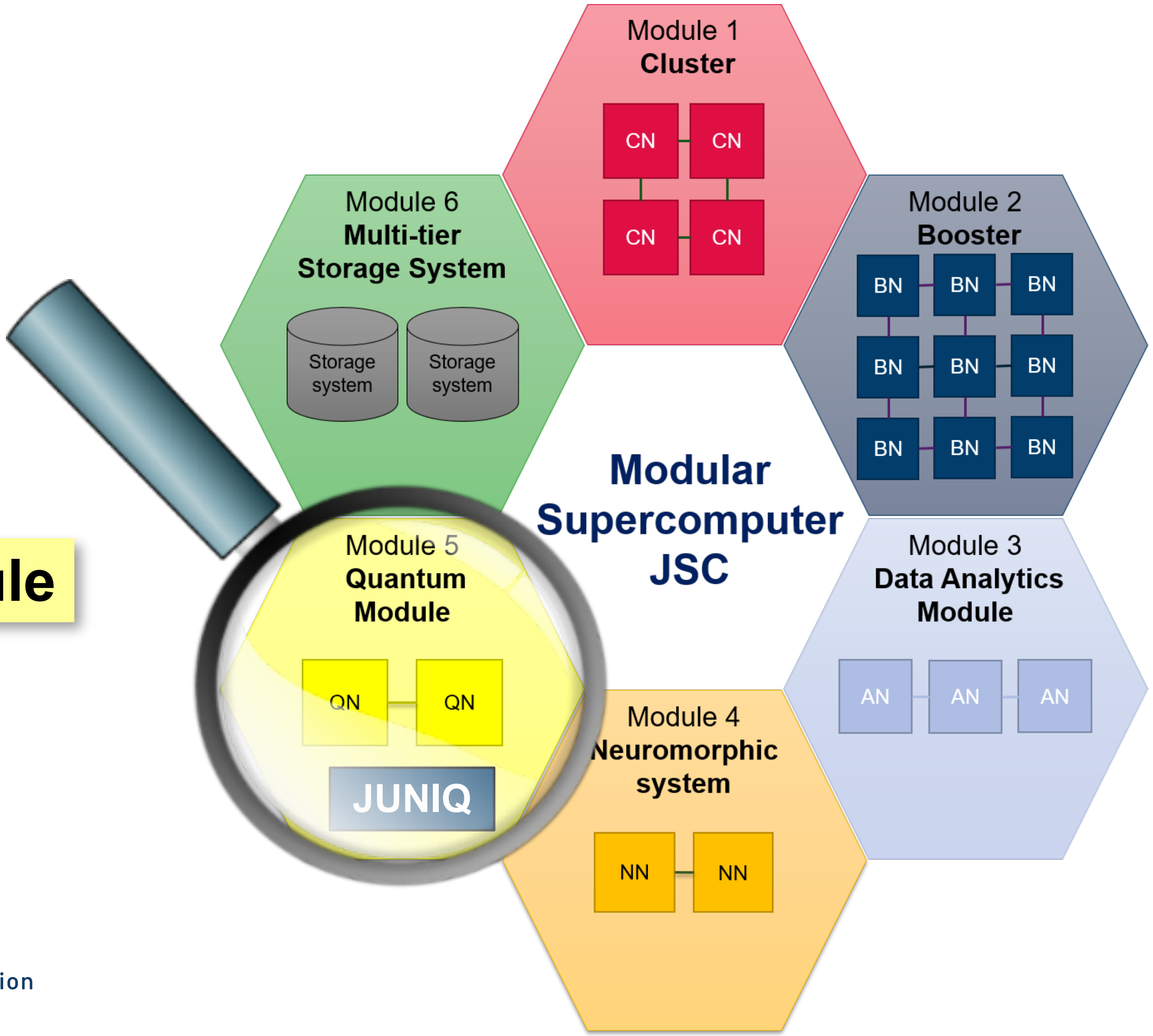
Quantum Module



JUNIQ - Jülich UNified Infrastructure for Quantum computing

A European quantum computer user facility at the Jülich Supercomputing Centre

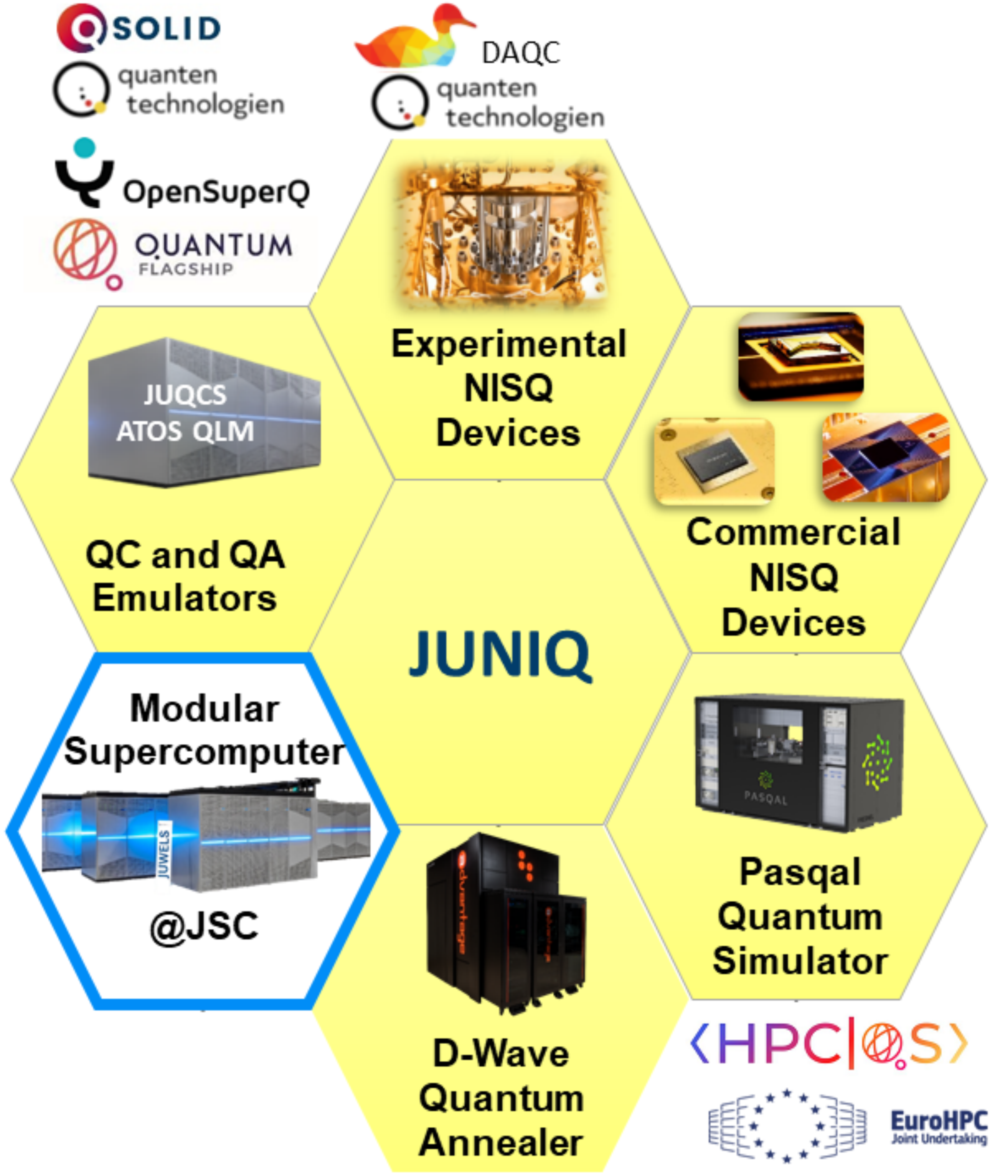
Quantum Module



Plan: Seamless QC-HPC integration

Perspective:
Quantum-Classical
Hybrid Computing Models

JUNIQ - Jülich UNified Infrastructure for Quantum computing

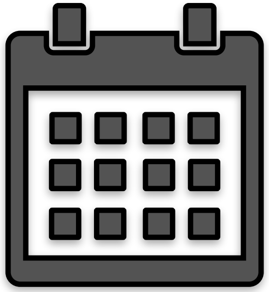


1. QC user facility for science and industry
2. Installation, operation and provision of QCs
3. Unified portal for access to QC emulators and to QC devices at different levels of technological maturity.
4. Development of algorithms and prototype applications
5. Services, training and user support
6. Modular quantum-HPC hybrid computing

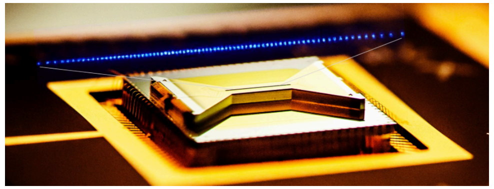
Why do we do quantum benchmarking?

Quantum benchmarking is essential because it provides **quantitative, standardized ways to evaluate and compare quantum devices** and **algorithms**. In short, it tells us **how good a quantum computer really is** and whether it's improving.

Measure Device Performance

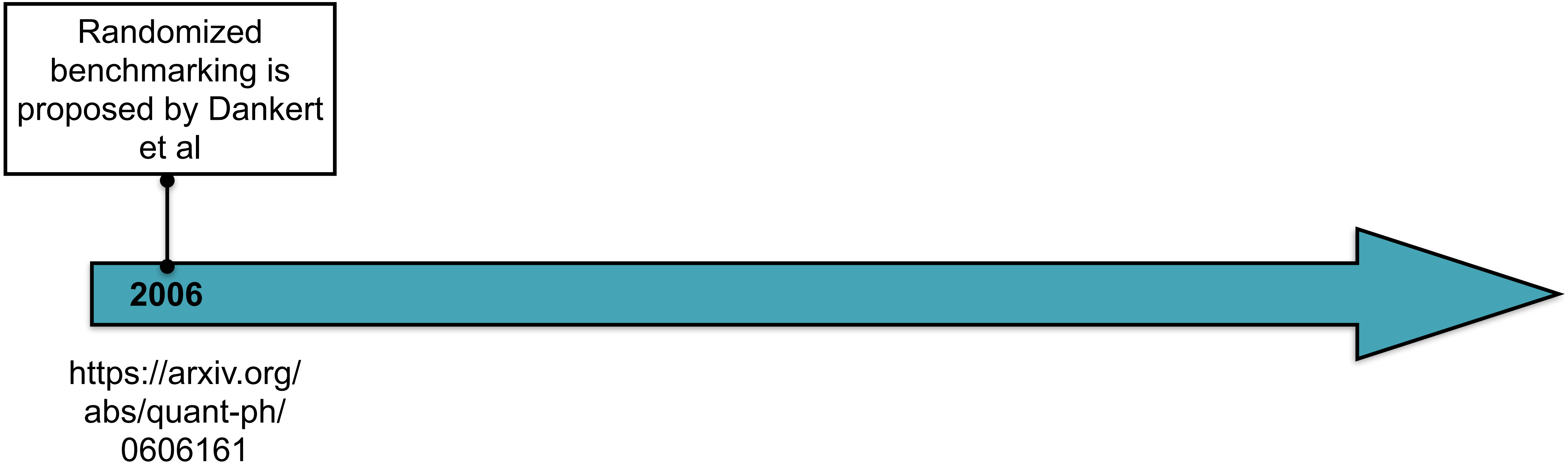


Track Progress Over Time



Compare Across Platforms

How to measure the progress of quantum processing units?



How to measure the progress of quantum processing units?

aws

Average 2Q fidelity (%) Info
98.720

Average 2Q fidelity (%)

Average gate fidelities for a maximally entangling two-qubit native gate. It is regularly measured by IonQ on the device using Direct Randomized Benchmarking

Randomized benchmarking is proposed by Dankert et al

2006

https://arxiv.org/abs/quant-ph/0606161

How to measure the progress of quantum processing units?

aws

Average 2Q fidelity (%) Info
98.720

Average 2Q fidelity (%)

Average gate fidelities for a maximally entangling two-qubit native gate. It is regularly measured by IonQ on the device using Direct Randomized Benchmarking

CZ fidelity (%)

IQM 99.232

CZ gate fidelities for the qubit pair measured by interleaved randomized benchmarking on 2 or 3 pairs simultaneously.

Randomized benchmarking is proposed by Dankert et al

2006

https://arxiv.org/abs/quant-ph/0606161

How to measure the progress of quantum processing units?

aws

Average 2Q fidelity (%) Info

98.720

Average 2Q fidelity (%)

Average gate fidelities for a maximally entangling two-qubit native gate. It is regularly measured by IonQ on the device using Direct Randomized Benchmarking

CZ fidelity (%)

IQM 99.232

CZ gate fidelities for the qubit pair measured by interleaved randomized benchmarking on 2 or 3 pairs simultaneously.

IBM Quantum **Platform**

Median CZ error:

2.696e-3 99.73 %

2Q error (best): The lowest two-qubit error rate from all edges measured by isolated randomized benchmarking.

Randomized benchmarking is proposed by Dankert et al

2006

https://arxiv.org/abs/quant-ph/0606161

How to measure the progress of quantum processing units?

aws

Average 2Q fidelity (%) Info

98.720

Average 2Q fidelity (%)

Average gate fidelities for a maximally entangling two-qubit native gate. It is regularly measured by IonQ on the device using Direct Randomized Benchmarking

CZ fidelity (%)

IQM 99.232

CZ gate fidelities for the qubit pair measured by interleaved randomized benchmarking on 2 or 3 pairs simultaneously.

QUANTINUUM SYSTEMS

2-qubit gate fidelity

The 2-qubit gate fidelity is measured with 2-qubit RB

$H2-1: 99.89\,\%$

Randomized benchmarking is proposed by Dankert et al

IBM Quantum **Platform**

Median CZ error:

2.696e-3 $99.73\,\%$

2Q error (best): The lowest two-qubit error rate from all edges measured by isolated randomized benchmarking.

https://arxiv.org/abs/quant-ph/0606161

JÜLICH Forschungszentrum

How to measure the progress of quantum processing units?

aws

Average 2Q fidelity (%) Info

98.720

Average 2Q fidelity (%)

Average gate fidelities for a maximally entangling two-qubit native gate. It is regularly measured by IonQ on the device using Direct Randomized Benchmarking

CZ fidelity (%)

IQM 99.232

CZ gate fidelities for the qubit pair measured by interleaved randomized benchmarking on 2 or 3 pairs simultaneously.

QUANTINUUM SYSTEMS

2-qubit gate fidelity

The 2-qubit gate fidelity is measured with 2-qubit RB

H2 – 1 : 99.89 %

Randomized benchmarking is proposed by Dankert et al

IBM Quantum **Platform**

Median CZ error

2.696e-3 99.73 %

2Q error (best): The lowest two-qubit error rate from all edges measured by isolated randomized benchmarking.

rigetti

ISWAP gate fidelity (%)

Average	Median
94.431	98.700

2006

https://arxiv.org/abs/quant-ph/0606161

How to measure the progress of quantum processing units?

aws

Average 2Q fidelity (%) Info

98.720

Average 2Q fidelity (%)

Average gate fidelities for a maximally entangling two-qubit native gate. It is regularly measured by IonQ on the device using Direct Randomized Benchmarking

CZ fidelity (%)

IQM 99.232

CZ gate fidelities for the qubit pair measured by interleaved randomized benchmarking on 2 or 3 pairs simultaneously.

QUANTINUUM SYSTEMS

2-qubit gate fidelity

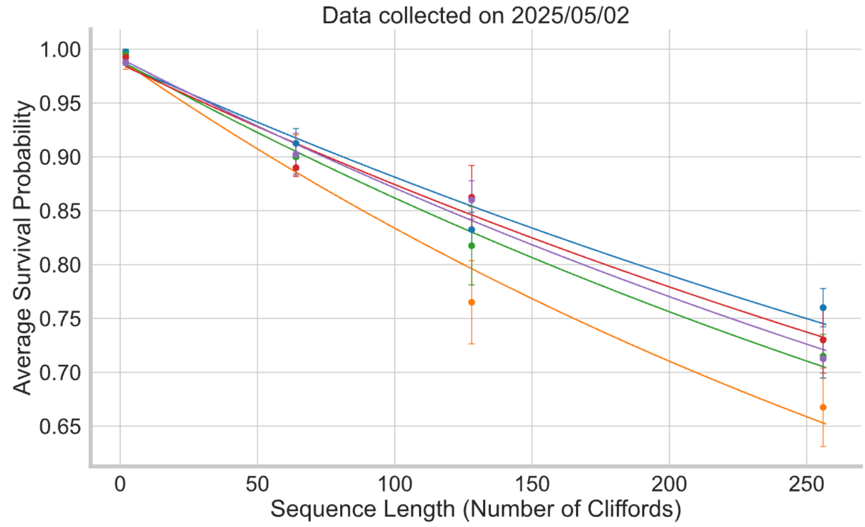
The 2-qubit gate fidelity is measured with 2-qubit RB

H2 – 1 : 99.89 %

rigetti

ISWAP gate fidelity (%)

Average	Median
94.431	98.700



Randomized benchmarking is proposed by Dankert et al

IBM Quantum **Platform**

Median CZ error:

2.696e-3   99.73 %

2Q error (best): The lowest two-qubit error rate from all edges measured by isolated randomized benchmarking.

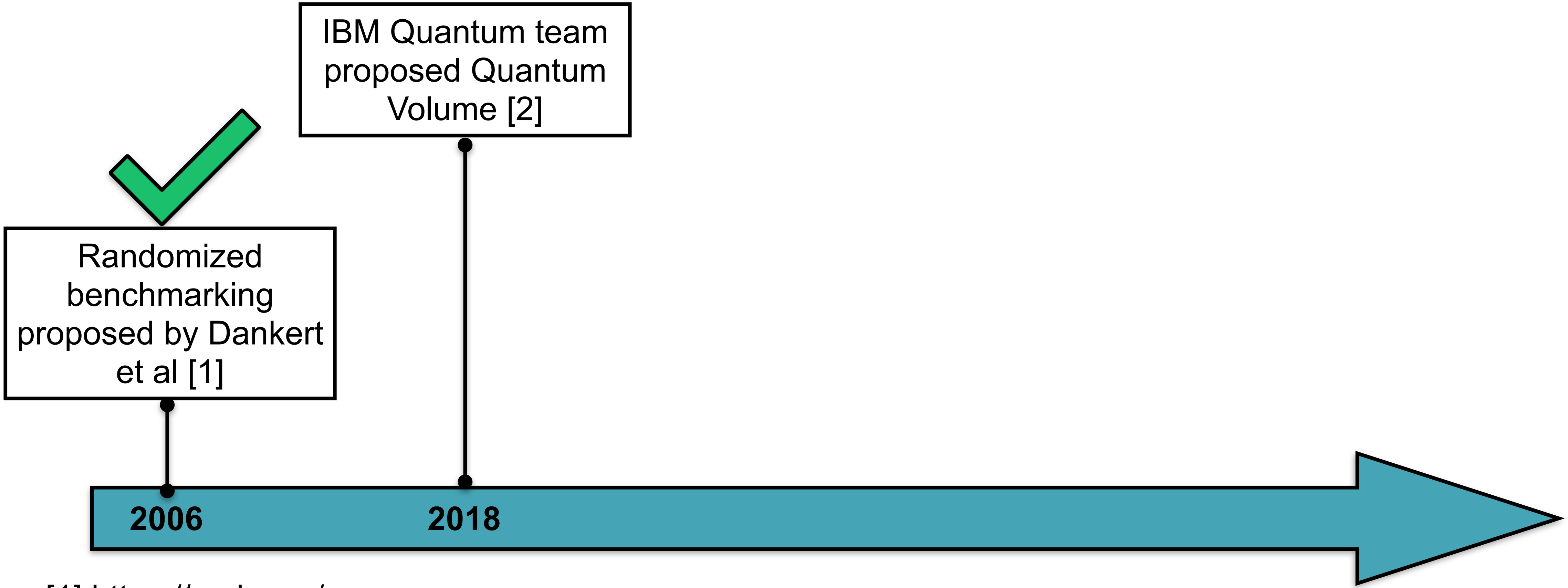
2006

https://arxiv.org/abs/quant-ph/0606161

How to measure the progress of quantum processing units?

https://arxiv.org/abs/quant-ph/0606161

How to measure the progress of quantum processing units?



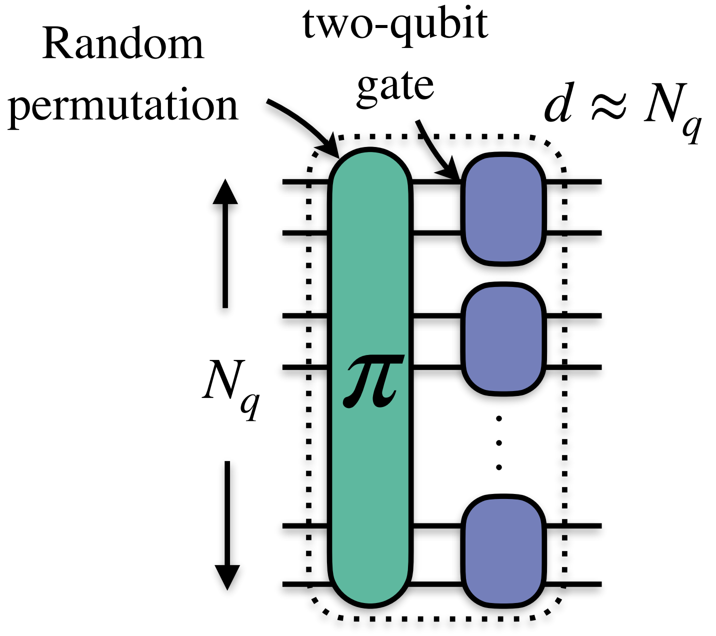
[1] https://arxiv.org/abs/quant-ph/0606161

[2] https://arxiv.org/pdf/1811.12926

How to measure the progress of quantum processing units?

Randomized benchmarking proposed by Dankert et al [1]

IBM Quantum team proposed Quantum Volume [2]



$$\log_2 V_Q = \underset{N_q}{\operatorname{argmax}} \min(N_q, d(N_q))$$

2006

2018

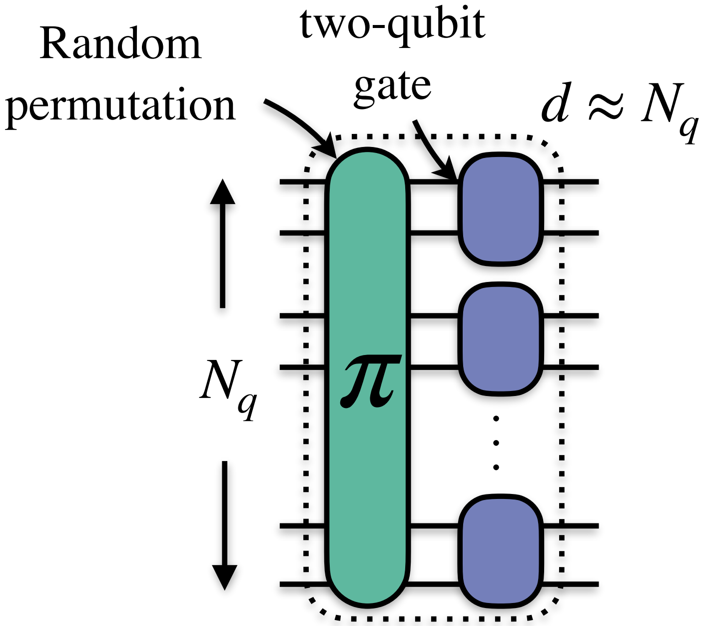
[1] https://arxiv.org/abs/quant-ph/0606161

[2] https://arxiv.org/pdf/1811.12926

How to measure the progress of quantum processing units?

Randomized benchmarking proposed by Dankert et al [1]

IBM Quantum team proposed Quantum Volume [2]



$$\log_2 V_Q = \underset{N_q}{\operatorname{argmax}} \min(N_q, d(N_q))$$

2006

2018

[1] https://arxiv.org/abs/quant-ph/0606161

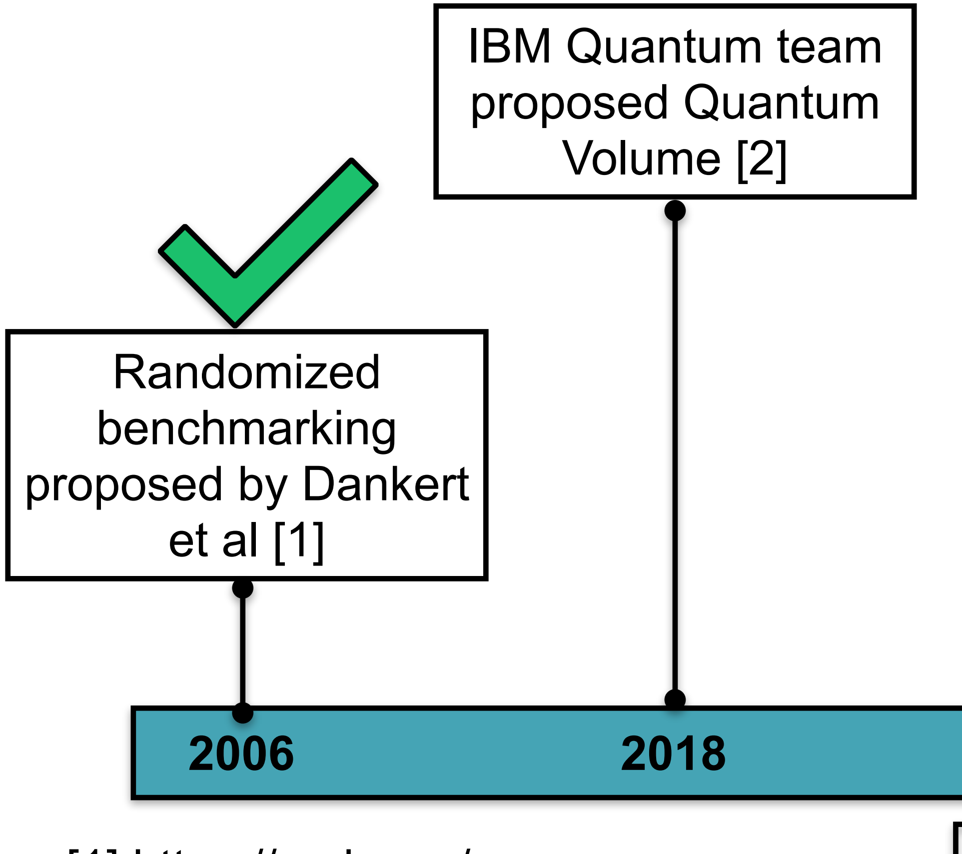
[2] https://arxiv.org/pdf/1811.12926

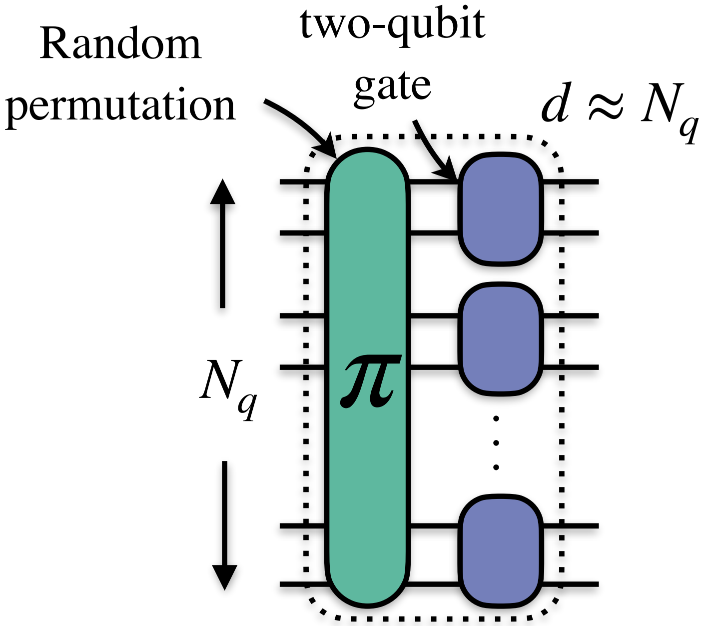
$h_U \approx 0.85$ ideal

$h_U > 2/3$ pass test

$h_U = 0.5$ fully mixed

How to measure the progress of quantum processing units?





$$\log_2 V_Q = \underset{N_q}{\operatorname{argmax}} \min(N_q, d(N_q))$$

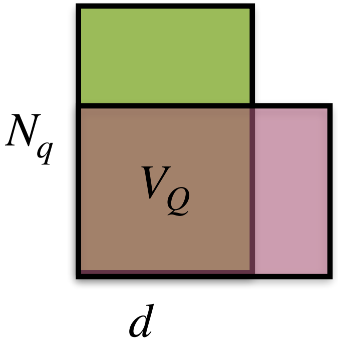
[1] https://arxiv.org/abs/quant-ph/0606161

[2] https://arxiv.org/pdf/1811.12926

$h_U \approx 0.85$ ideal

$h_U > 2/3$ pass test

$h_U = 0.5$ fully mixed



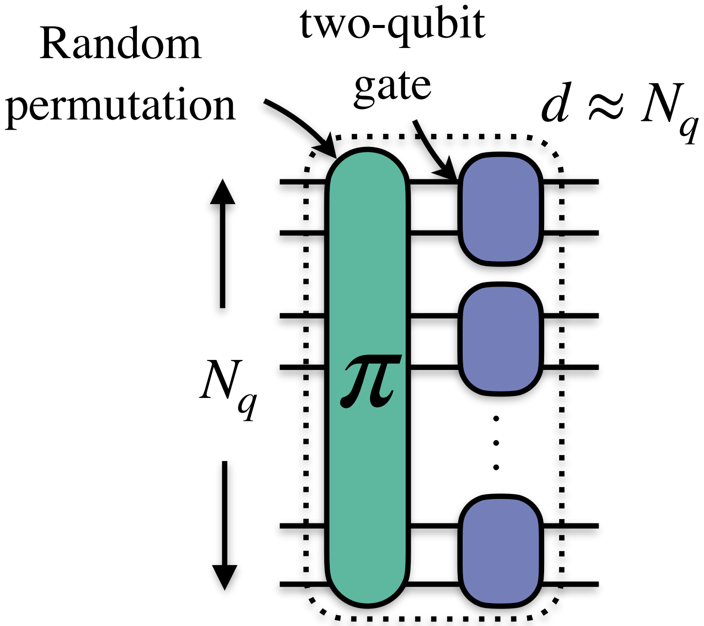
How to measure the progress of quantum processing units?

Randomized benchmarking proposed by Dankert et al [1]

IBM Quantum team proposed Quantum Volume [2]

2006

2018



$$\log_2 V_Q = \underset{N_q}{\operatorname{argmax}} \min(N_q, d(N_q))$$

Define a clear set of rules!

"The transpiler is free to use all available tricks and hardware resources to implement U (e.g., taking great computational effort in finding an optimized U, using extra qubits for gate teleportation or temporary storage, etc.). It may optimize over qubit placements by choosing the best region of the device. If it is practical to calibrate a very large gate set, and it happens to include an accurate implementation of U, the transpiler is free to use it."[2]

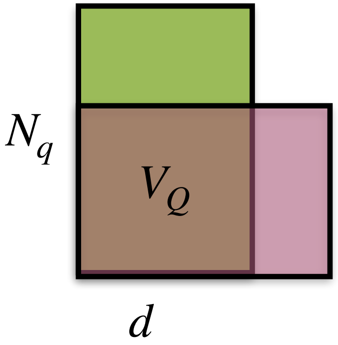
[1] https://arxiv.org/abs/quant-ph/0606161

[2] https://arxiv.org/pdf/1811.12926

$h_U \approx 0.85$ ideal

$h_U > 2/3$ pass test

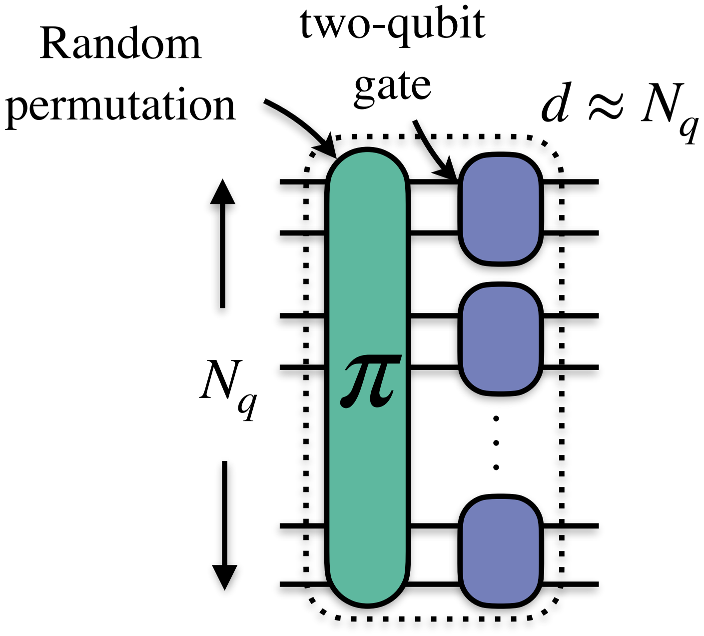
$h_U = 0.5$ fully mixed



How to measure the progress of quantum processing units?

Randomized benchmarking proposed by Dankert et al [1]

IBM Quantum team proposed Quantum Volume [2]



$$\log_2 V_Q = \underset{N_q}{\operatorname{argmax}} \min(N_q, d(N_q))$$

Define a clear set of rules!

"The transpiler is free to use all available tricks and hardware resources to implement U (e.g., taking great computational effort in finding an optimized U, using extra qubits for gate teleportation or temporary storage, etc.). It may optimize over qubit placements by choosing the best region of the device. If it is practical to calibrate a very large gate set, and it happens to include an accurate implementation of U, the transpiler is free to use it."[2]

Limitation

"To determine if a given output is heavy, we compute H_U directly from U using a method that scales exponentially with Nq".

2006 **2018**

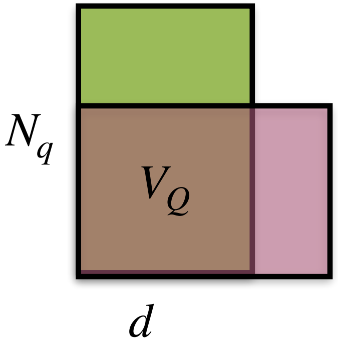
[1] https://arxiv.org/abs/quant-ph/0606161

[2] https://arxiv.org/pdf/1811.12926

$h_U \approx 0.85$ ideal

$h_U > 2/3$ pass test

$h_U = 0.5$ fully mixed

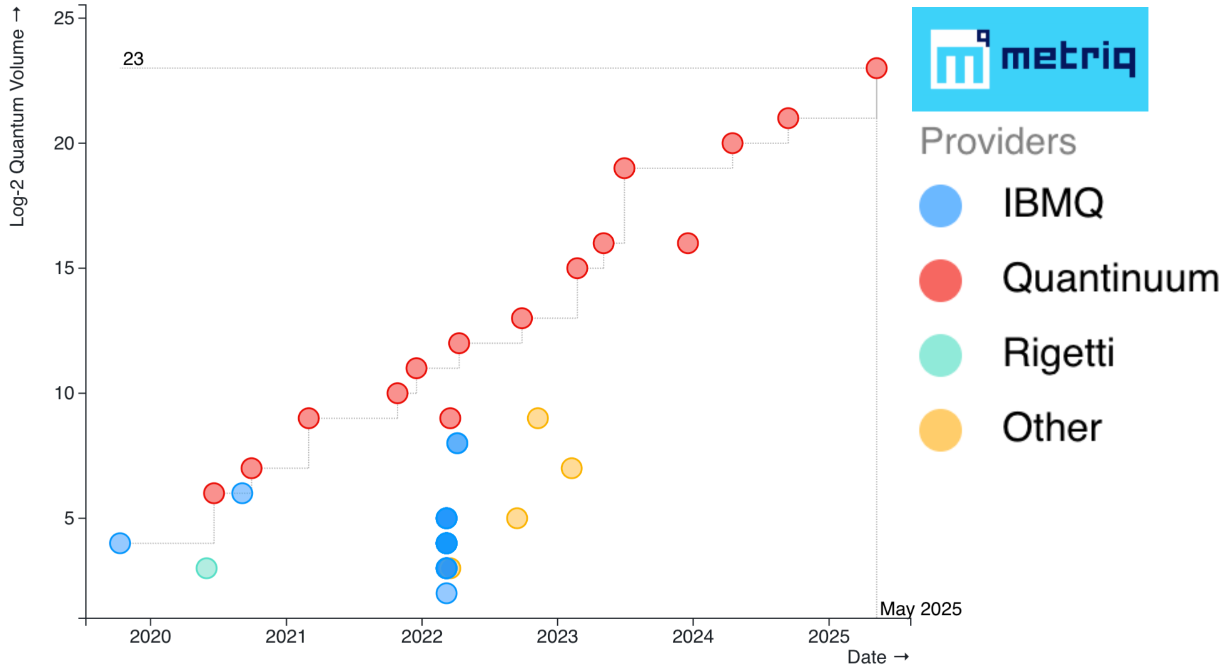


JÜLICH Forschungszentrum

How to measure the progress of quantum processing units?

IBM Quantum team proposed Quantum Volume [2]

Randomized benchmarking proposed by Dankert et al [1]



2006

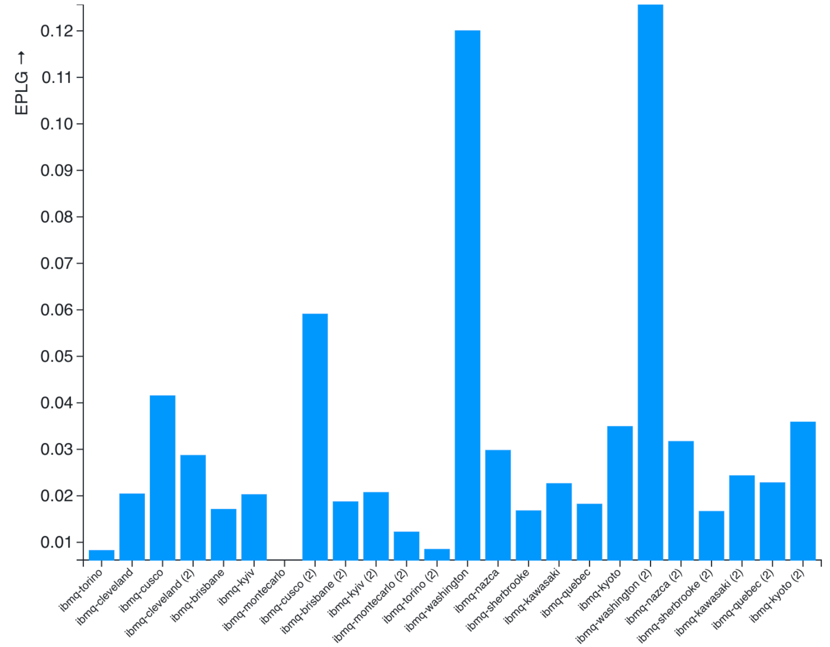
2018

[1] https://arxiv.org/abs/quant-ph/0606161

[2] https://arxiv.org/pdf/1811.12926

How to measure the progress of quantum processing units?

Error per layered gate



Randomized benchmarking proposed by Dankert et al [1]

IBM Quantum team proposed Quantum Volume [2]

IBM Quantum team proposed Error per layered gate (EPLG) [3]

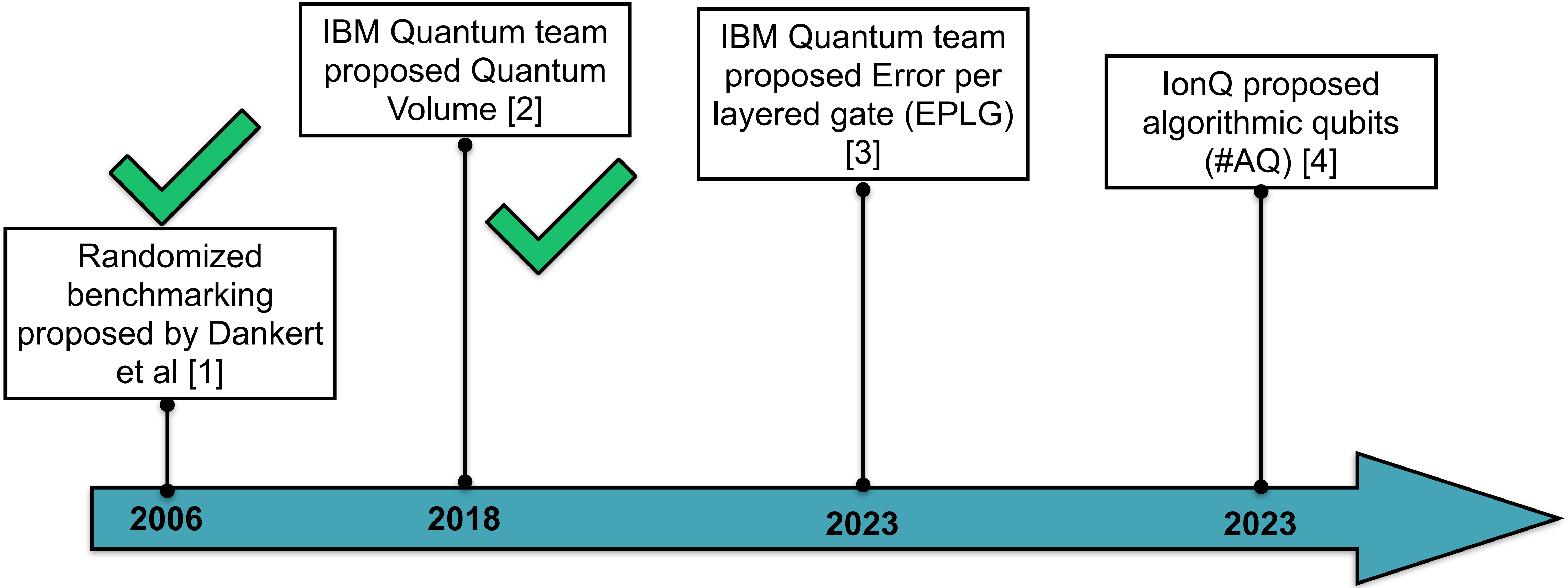
2006

2018

2023

[1] arXiv:0606161 [2] arXiv:1811.12926 [3] arXiv:2311.05933

How to measure the progress of quantum processing units?



[1] arXiv:0606161 [2] arXiv:1811.12926 [3] arXiv:2311.05933 [4] arXiv:2308.05071

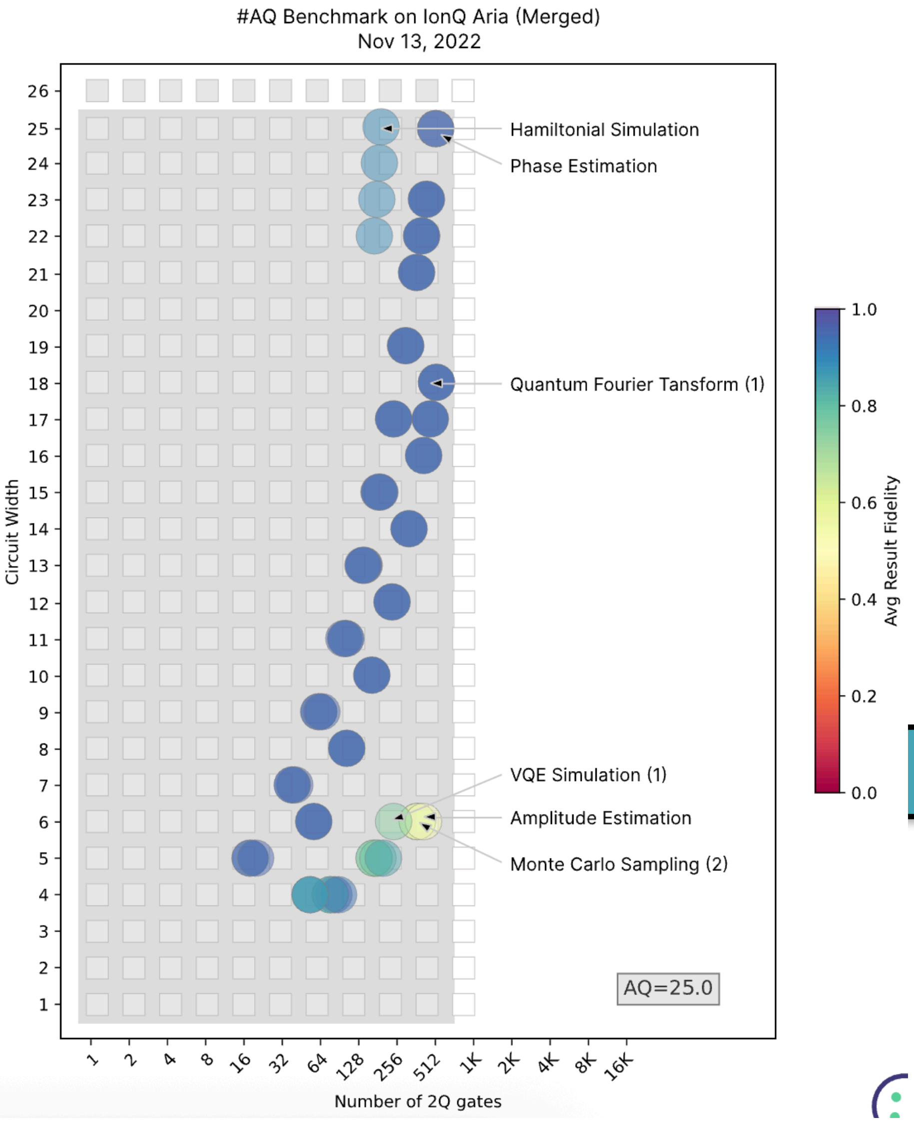
How to measure the progress of quantum processing units?

IBM C
propo
V

Randomized benchmarking proposed by Dankert et al [1]

2006

[1] arXiv:0606161 [2] ar



IonQ proposed algorithmic qubits (#AQ) [4]

2023

[4] arXiv:2308.05071

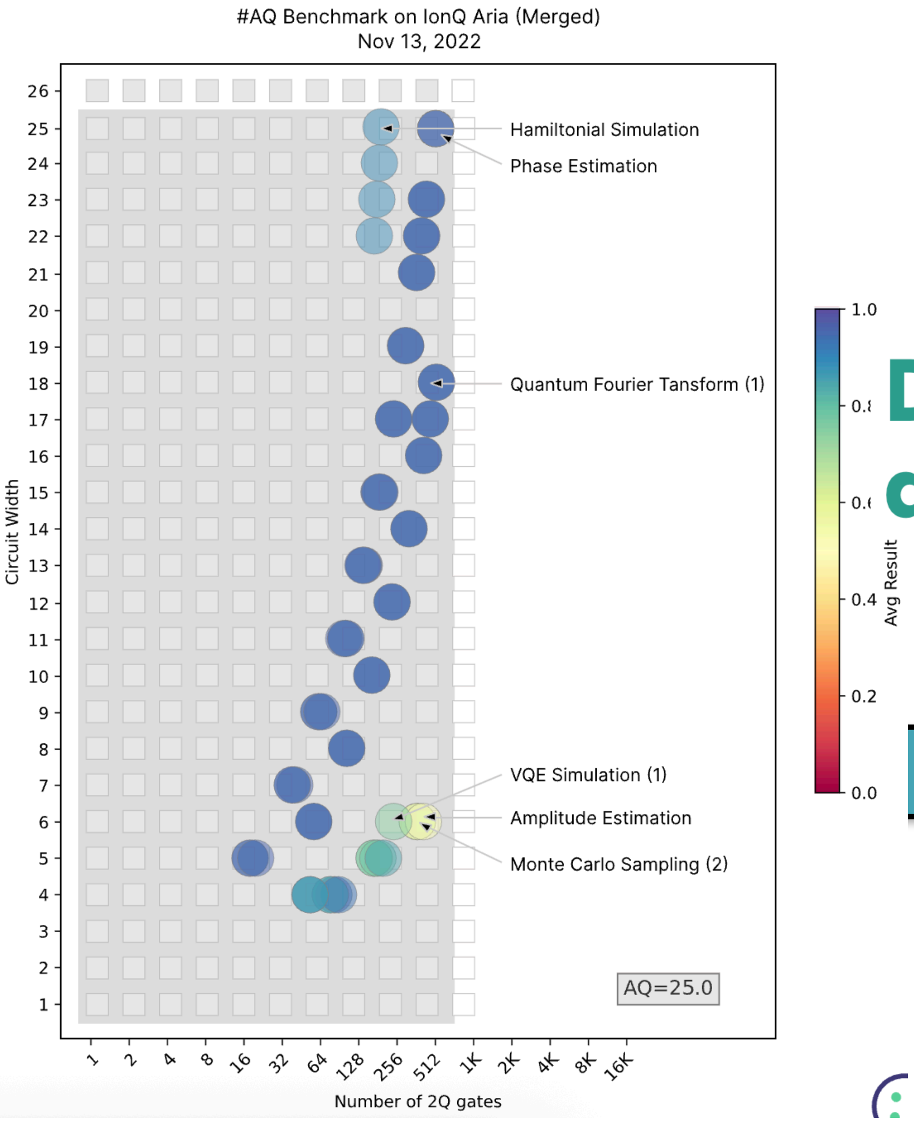
How to measure the progress of quantum processing units?

IBM C
propo
V

Randomized benchmarking proposed by Dankert et al [1]

2006

[1] arXiv:0606161 [2] ar



IonQ proposed algorithmic qubits (#AQ) [4]

Debunking algorithmic qubits

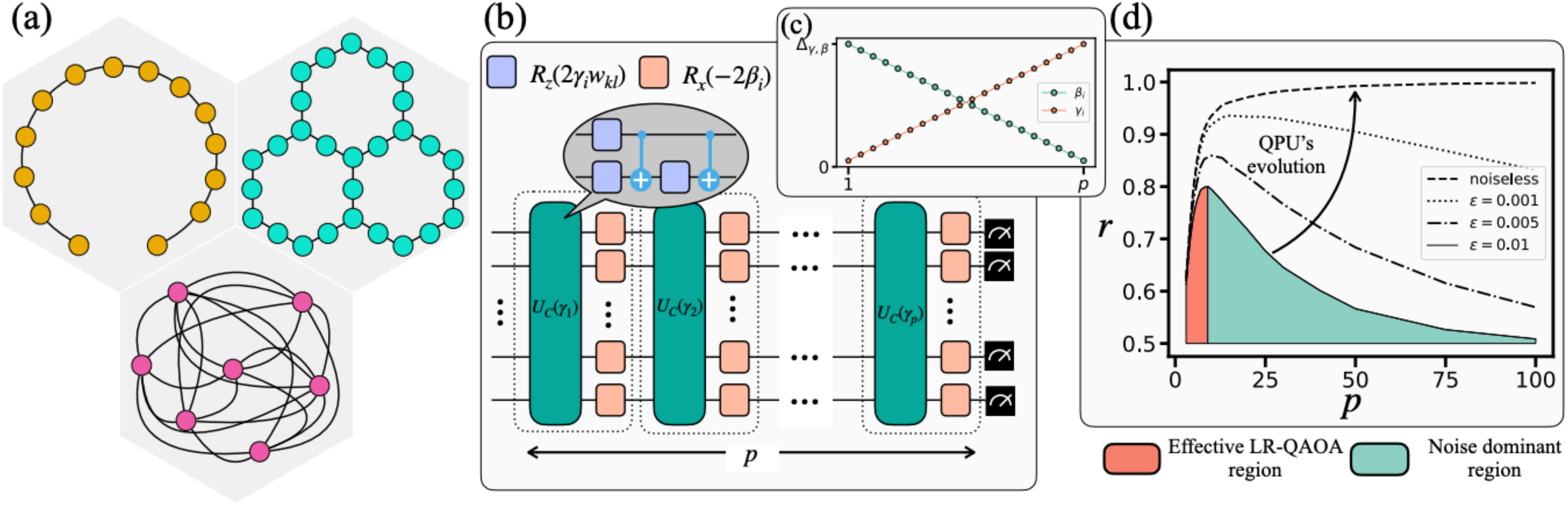
QUANTINUUM

2023

[4] arXiv:2308.05071

JÜLICH Forschungszentrum

The Linear Ramp Quantum Approximate Optimization Algorithm (LR-QAOA)

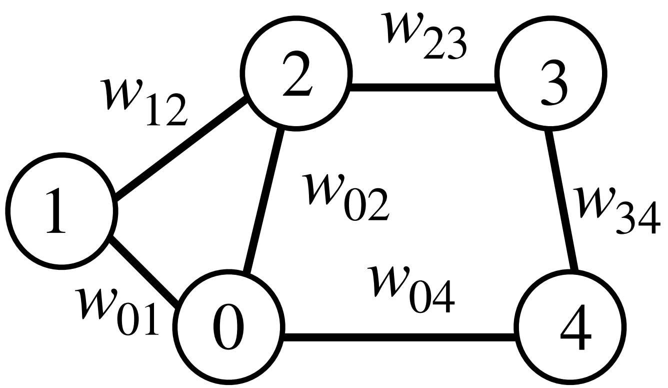


(a) Graph topologies
(b) QAOA algorithm
(c) Linear ramp protocol for QAOA
(d) Expected performance with noise

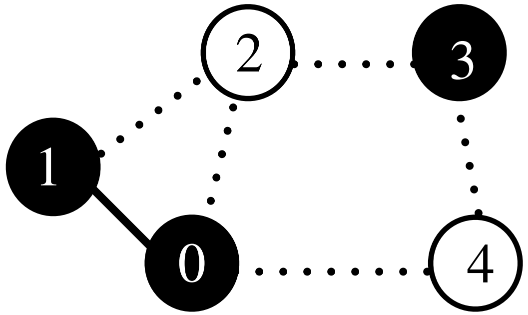
We applied this benchmark methodology to 24 different QPUs from 6 different vendors, IQM, IBM, Rigetti, IonQ, Quantinuum, and OriginQ using 5 to 156 qubits and up to p=10,000.

The problem behind LR-QAOA

The weighted maxcut (WMC) problem involves determining the partition of the vertices in an undirected graph so that the total weight of the edges between the two sets is maximized.



$$x^* = 00101$$



Cost function

$$C(x) = \sum_{(i,j)\in E} w_{ij}(x_i + x_j - 2x_i x_j)$$

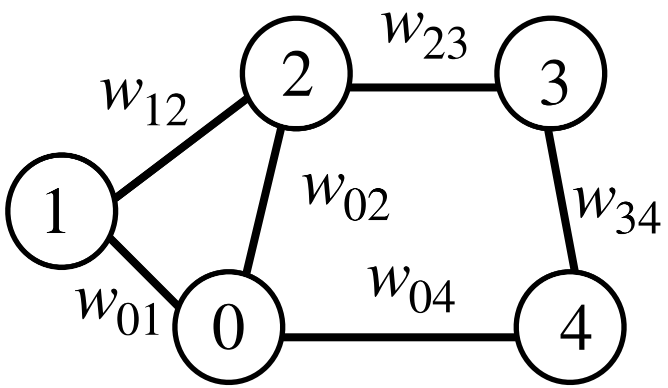
Approximation ratio

$$r = \frac{\sum_{k=1}^{n} C(x^k)/n}{C(x^*)}$$

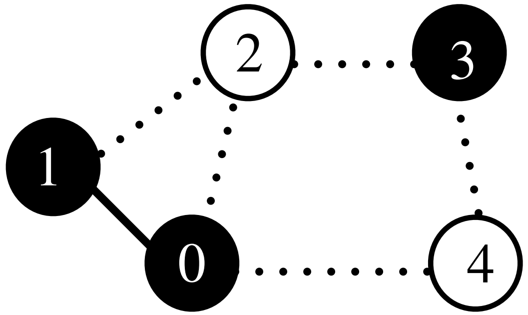
x_k	sample solution
x^*	optimal solution
n	Samples

The problem behind LR-QAOA

The weighted maxcut (WMC) problem involves determining the partition of the vertices in an undirected graph so that the total weight of the edges between the two sets is maximized.



$$x^* = 00101$$



Cost function

$$C(x) = \sum_{(i,j)\in E} w_{ij}(x_i + x_j - 2x_i x_j)$$

Approximation ratio $$\boxed{r = \frac{\sum_{k=1}^{n} C(x^k)/n}{C(x^*)}}$$

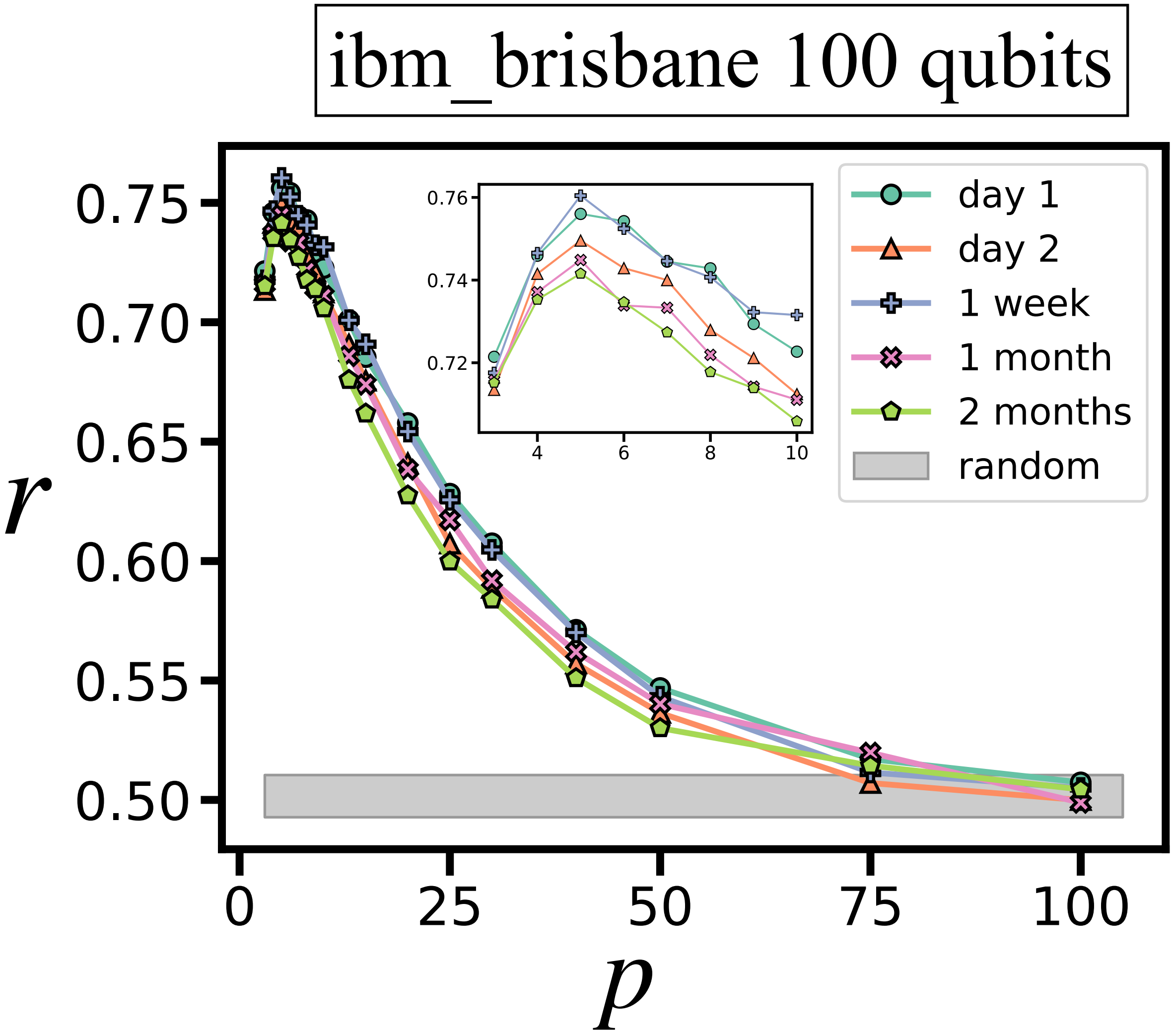
x_k sample solution

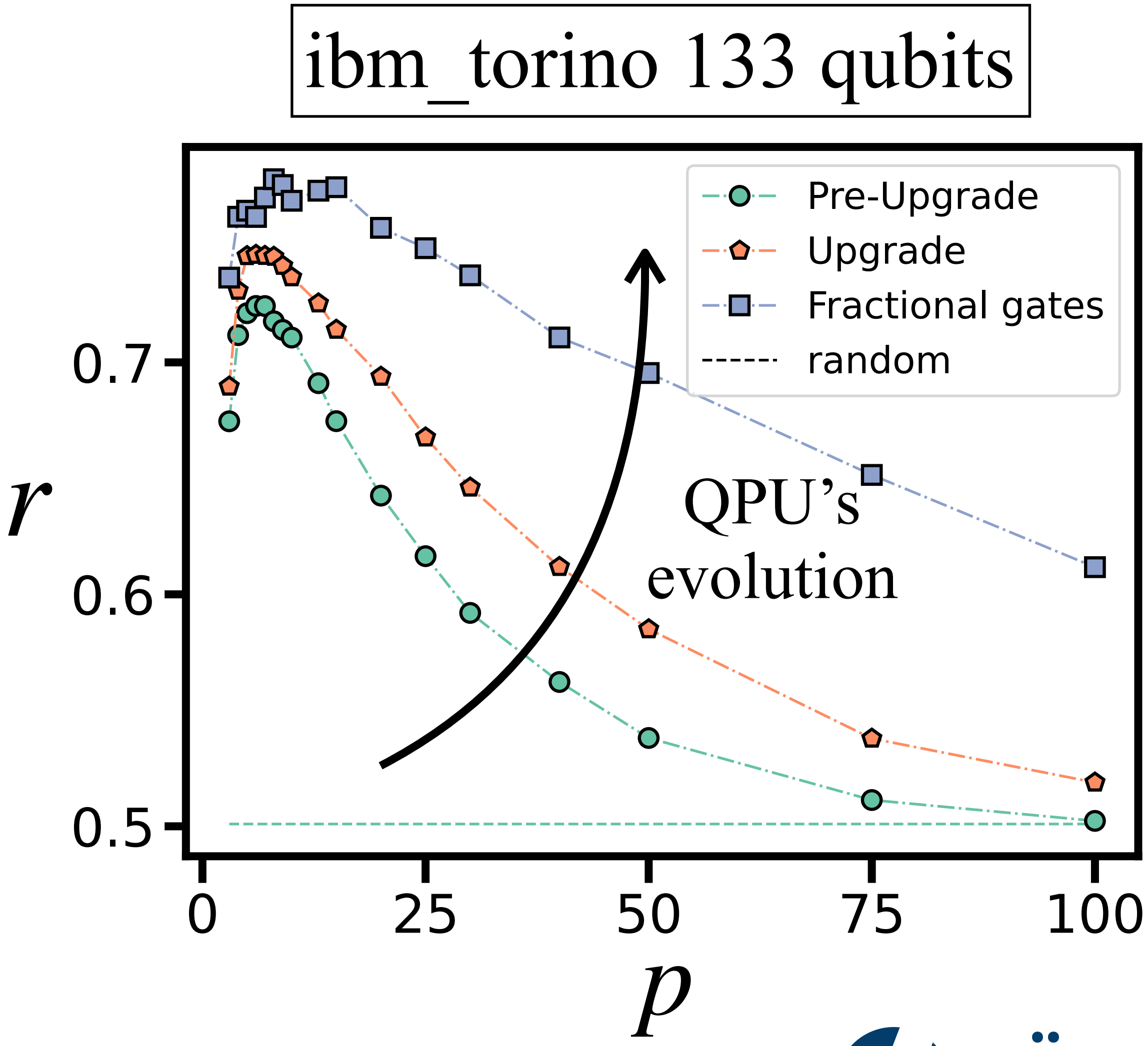
x^* optimal solution

n Samples

Performance metric

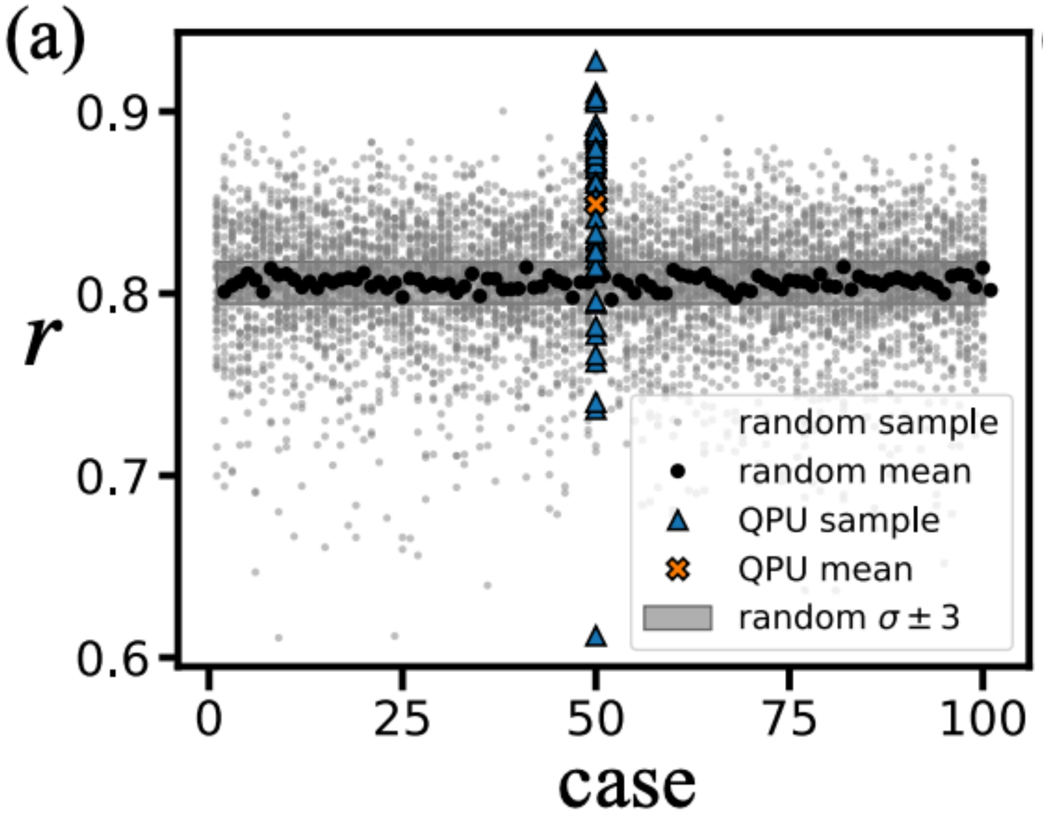
Tracking the evolution of real QPUs



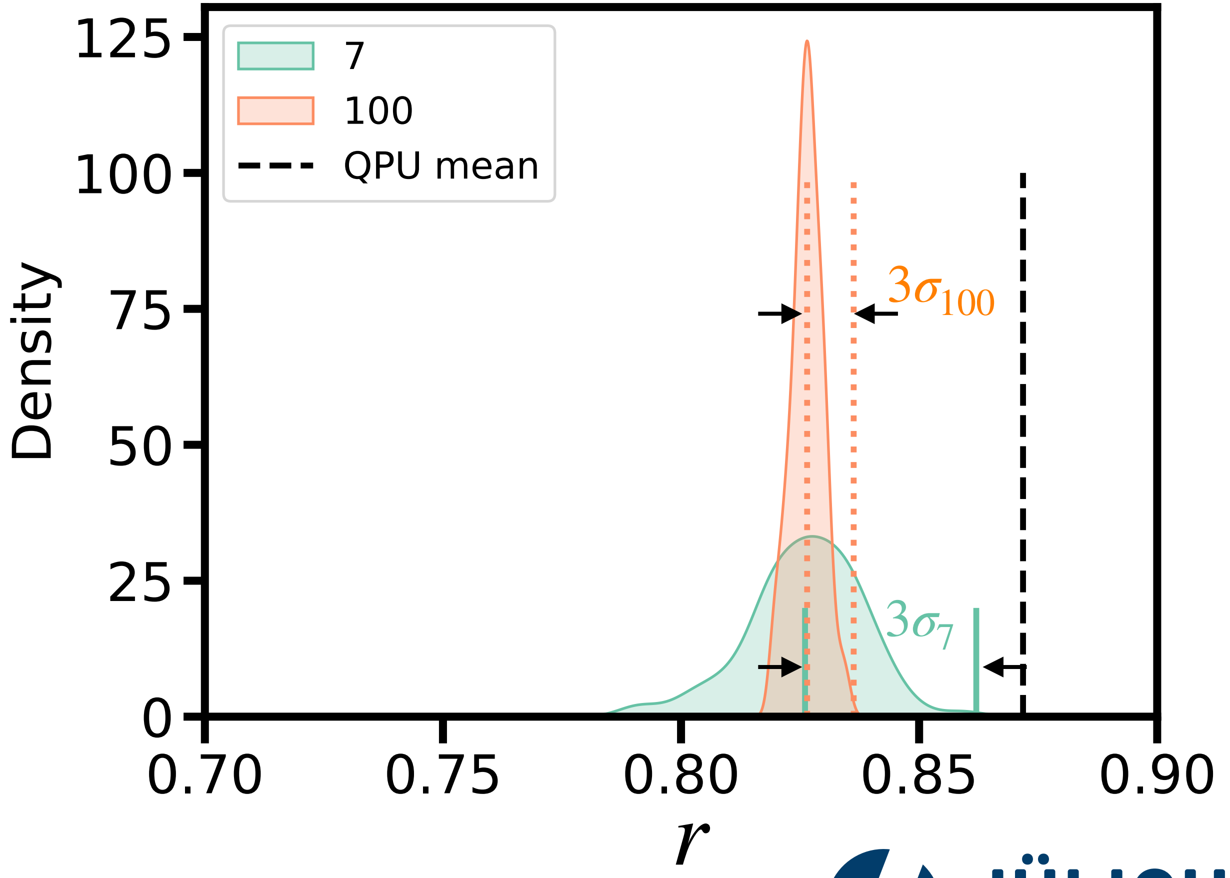


Distinguishing successful results

To certify if the result of a QPU is still meaningful, we compare the approximation ratio for the LR-QAOA WMC problem given by the samples of the QPU to those coming from a random sampler.

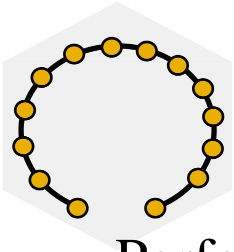


(a) H2-1 50-qubit, 50 samples, and p=4.

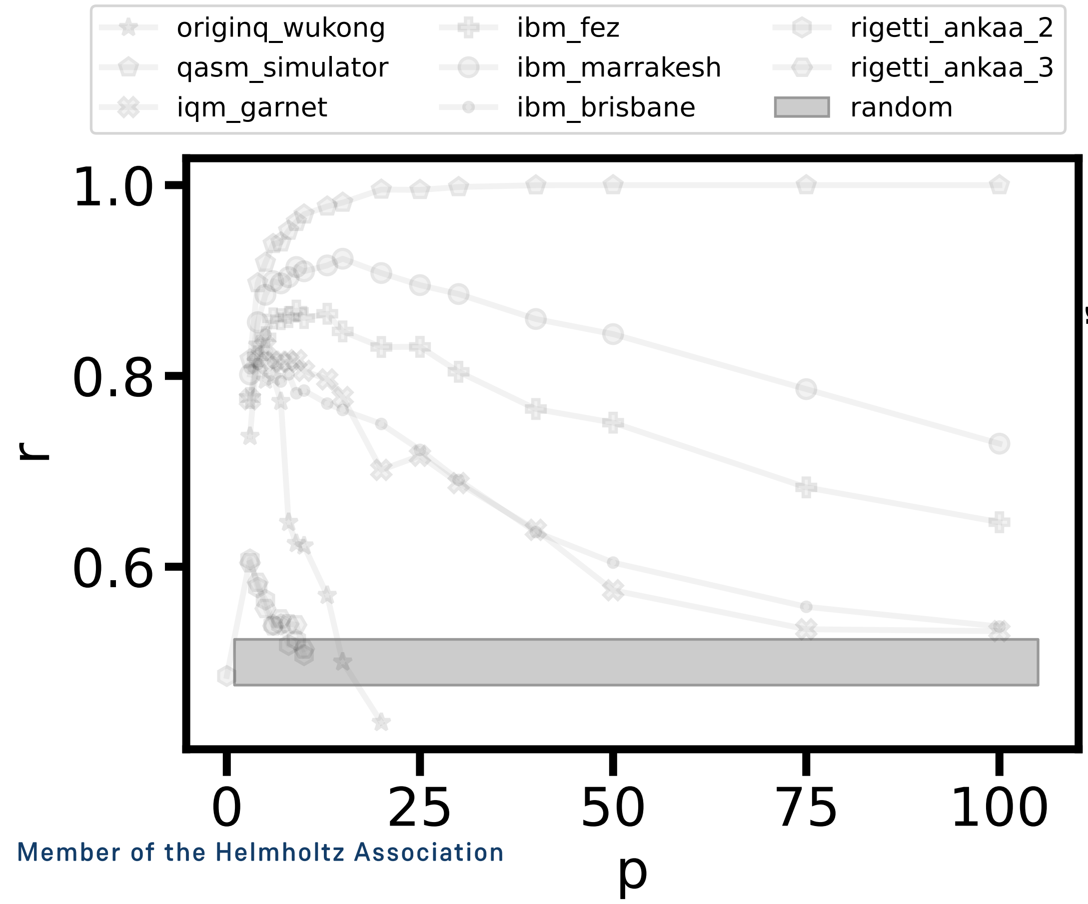


JÜLICH Forschungszentrum

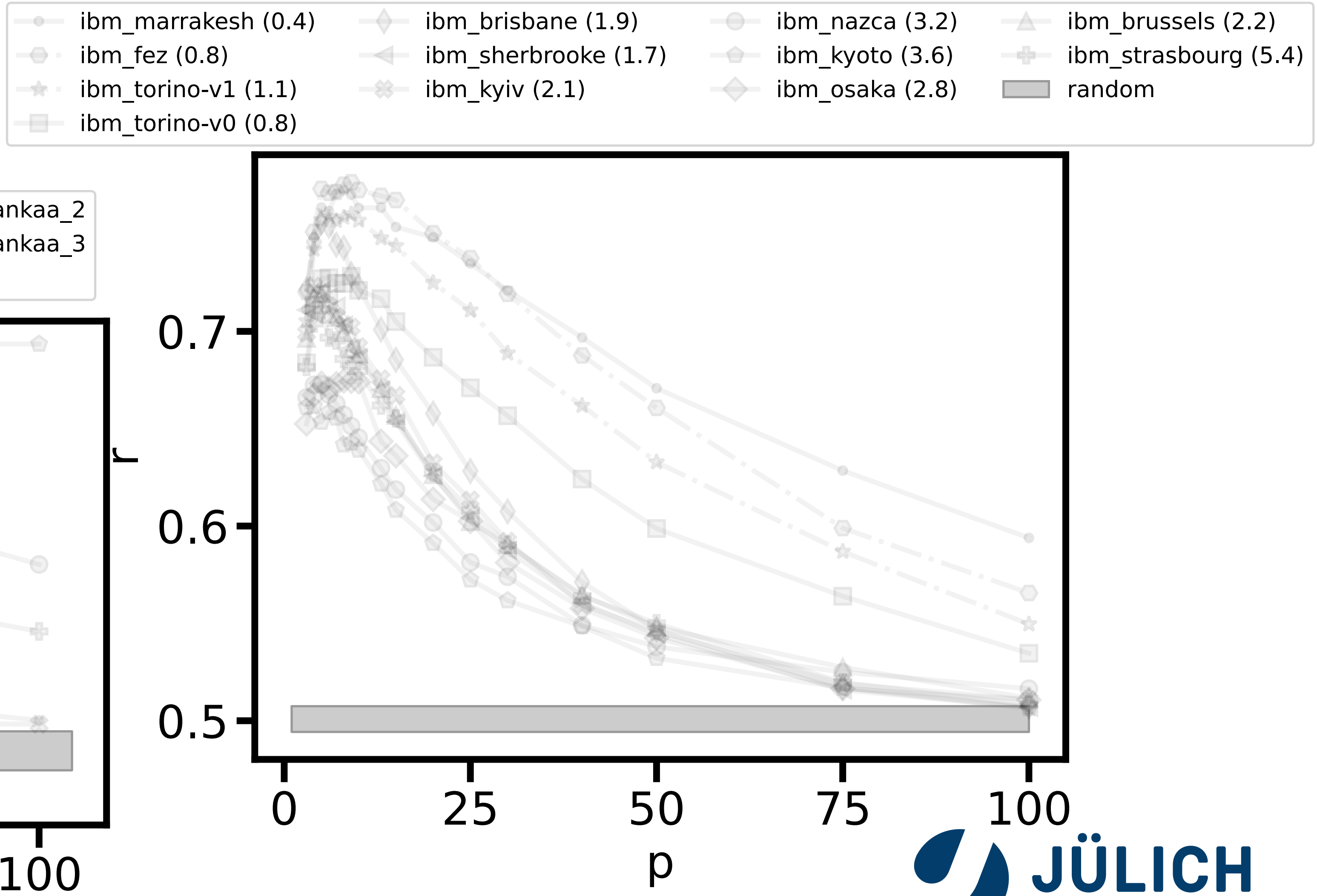
LR-QAOA on a 1D-Chain graph



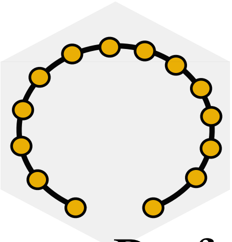
Performance on the best 5-qubit 1D chain experiment, different QPUs



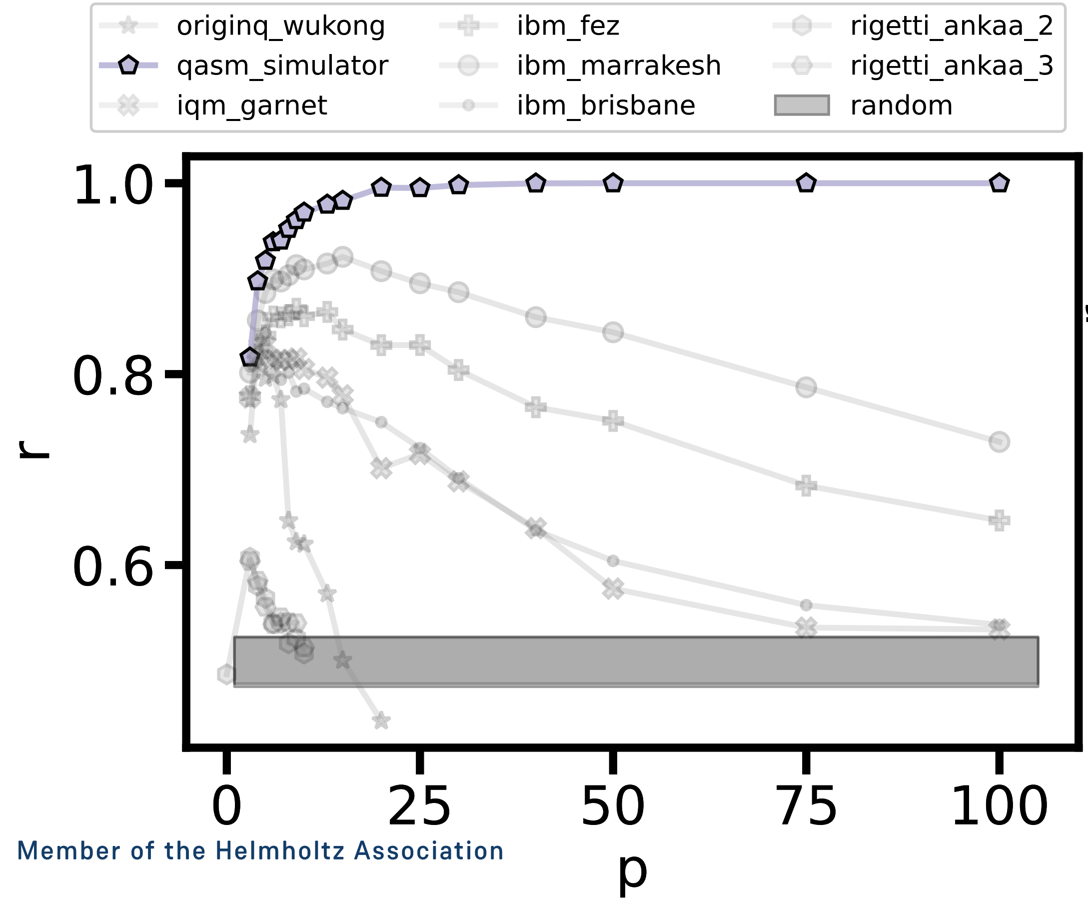
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



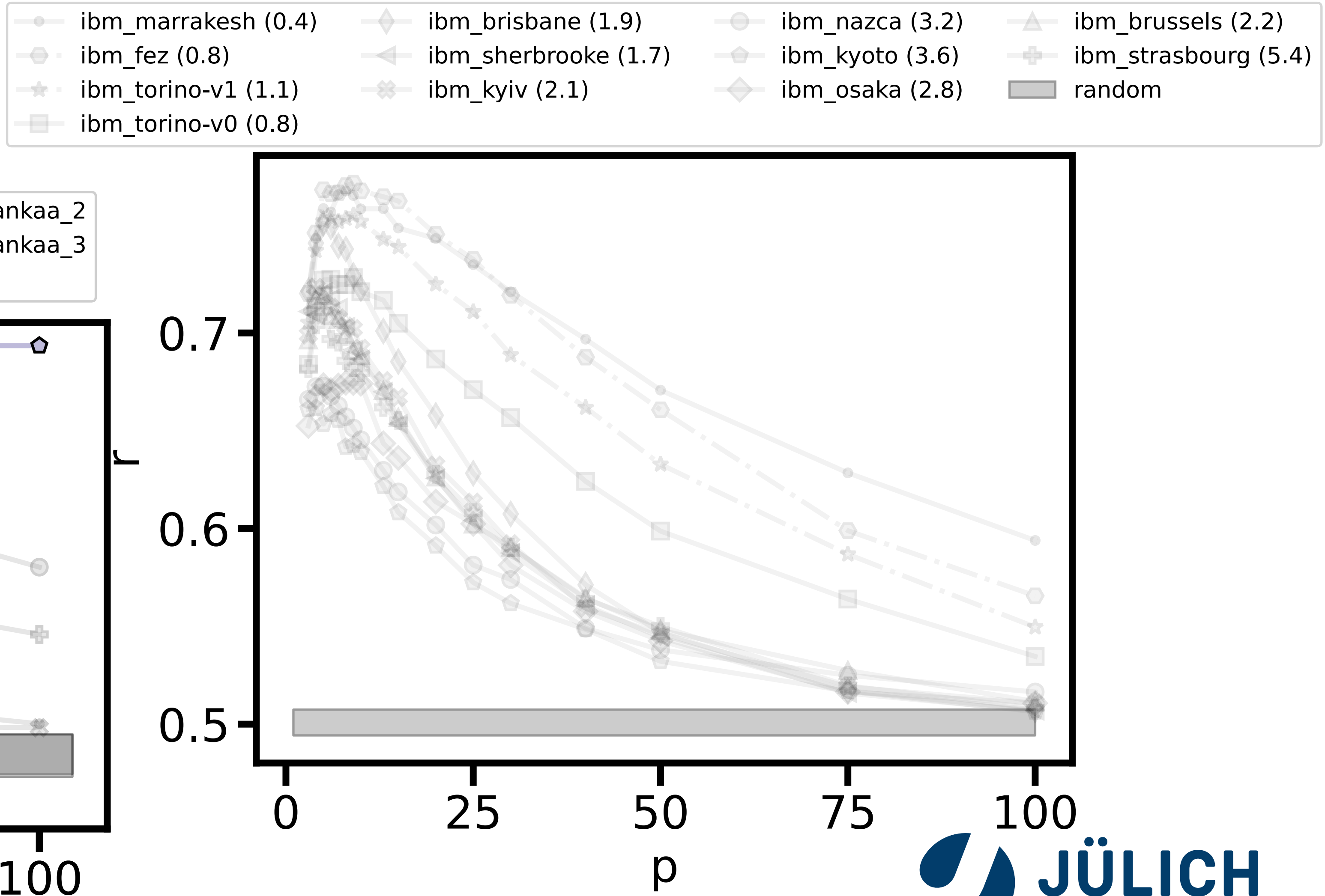
LR-QAOA on a 1D-Chain graph



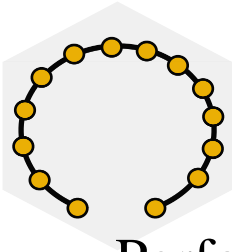
Performance on the best 5-qubit 1D chain experiment, different QPUs



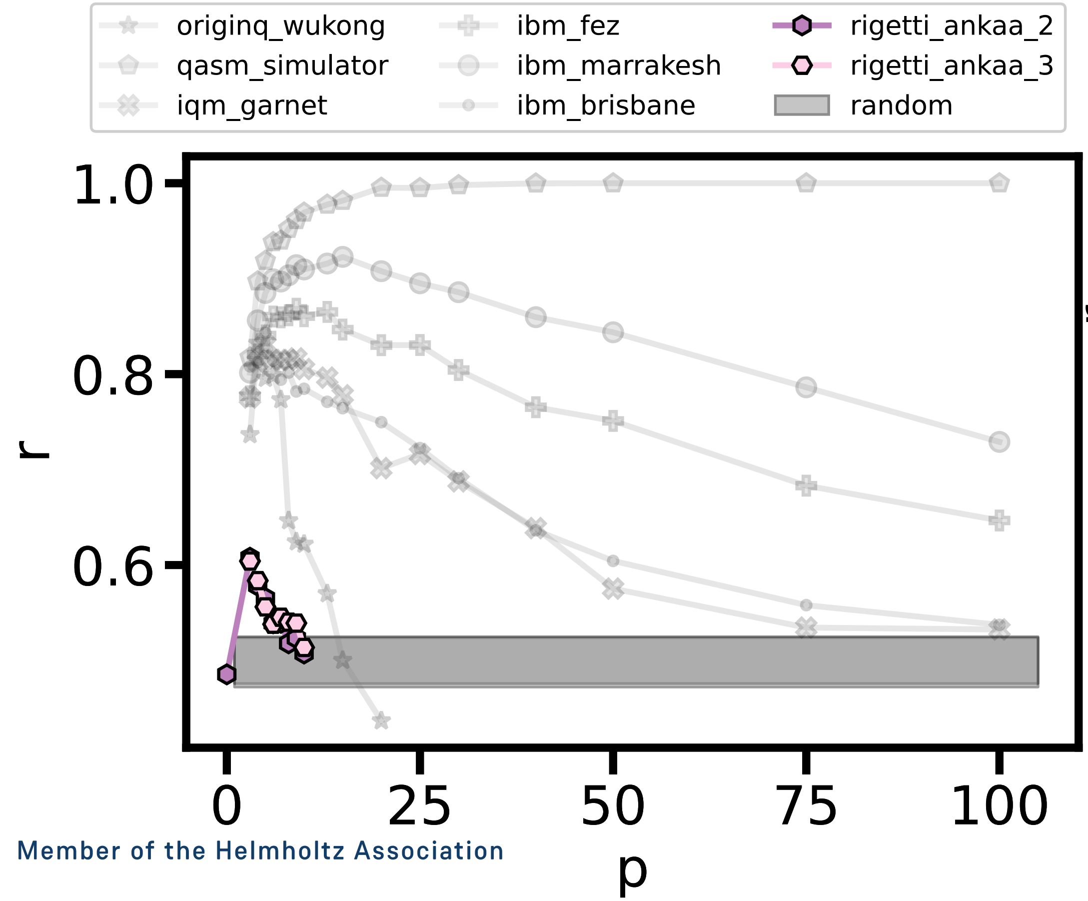
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



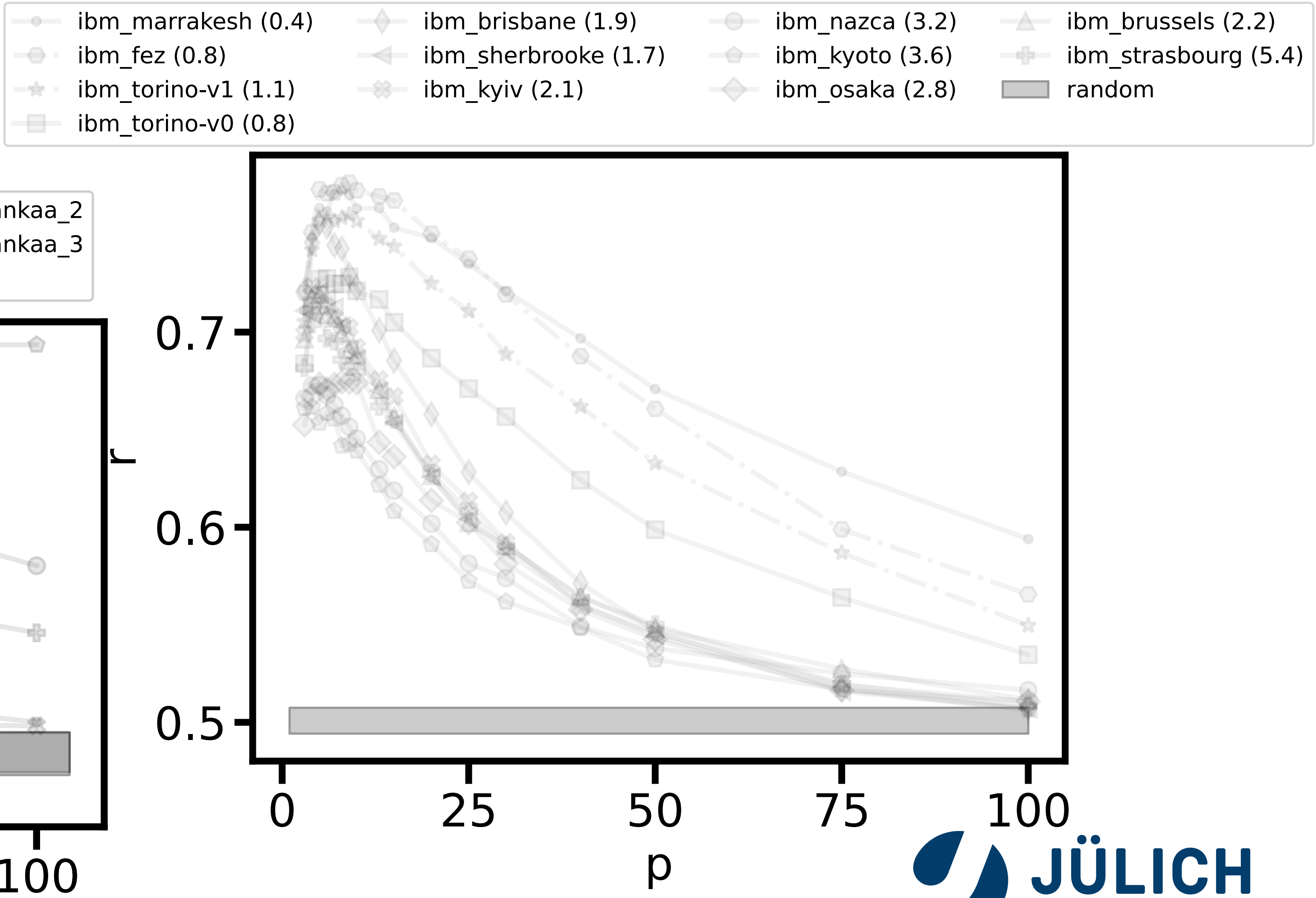
LR-QAOA on a 1D-Chain graph



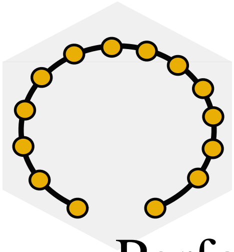
Performance on the best 5-qubit 1D chain experiment, different QPUs



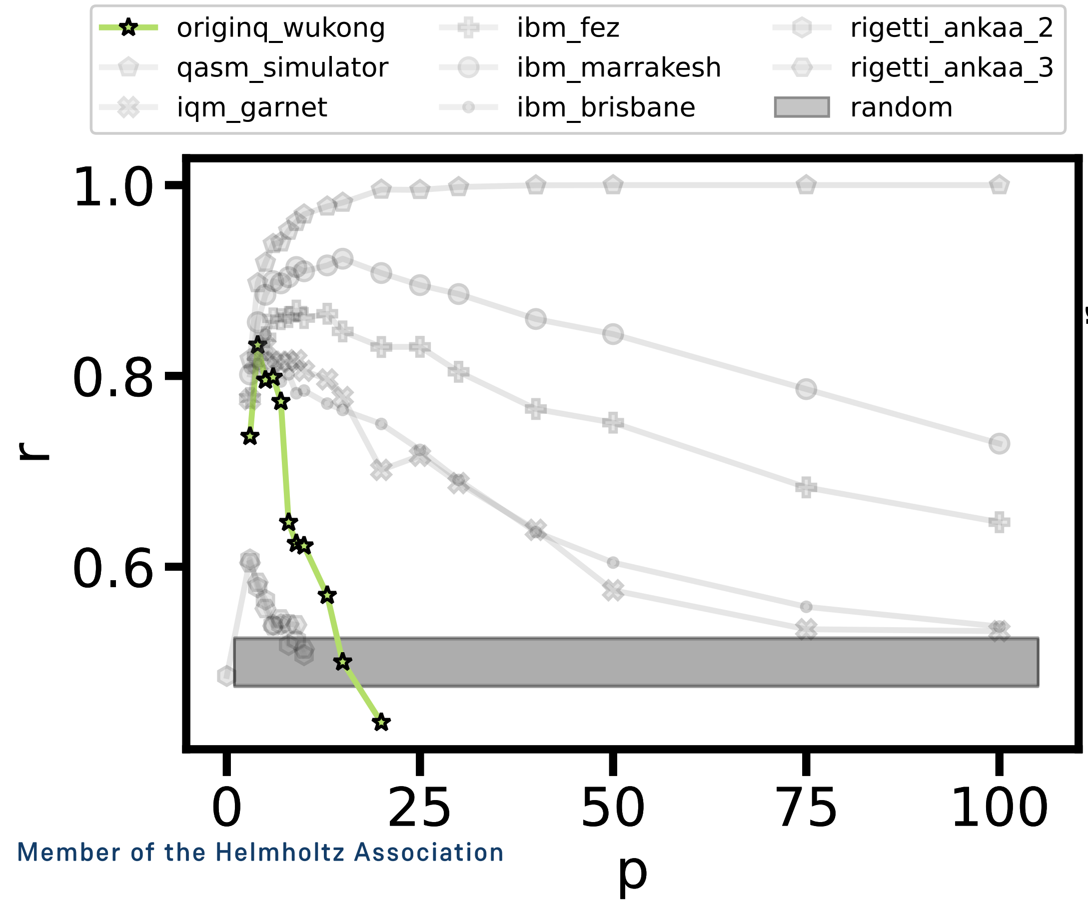
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



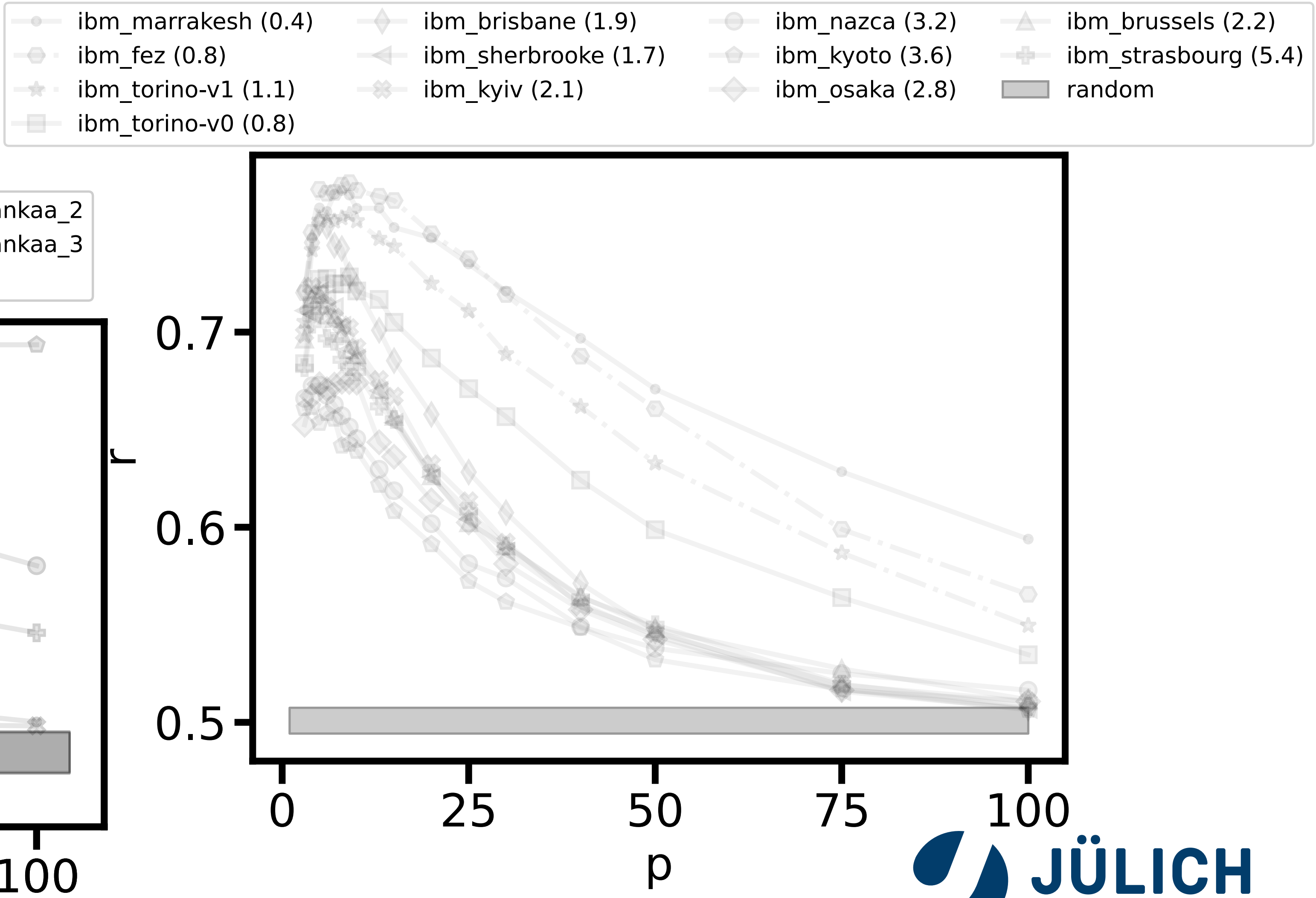
LR-QAOA on a 1D-Chain graph



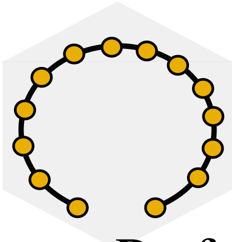
Performance on the best 5-qubit 1D chain experiment, different QPUs



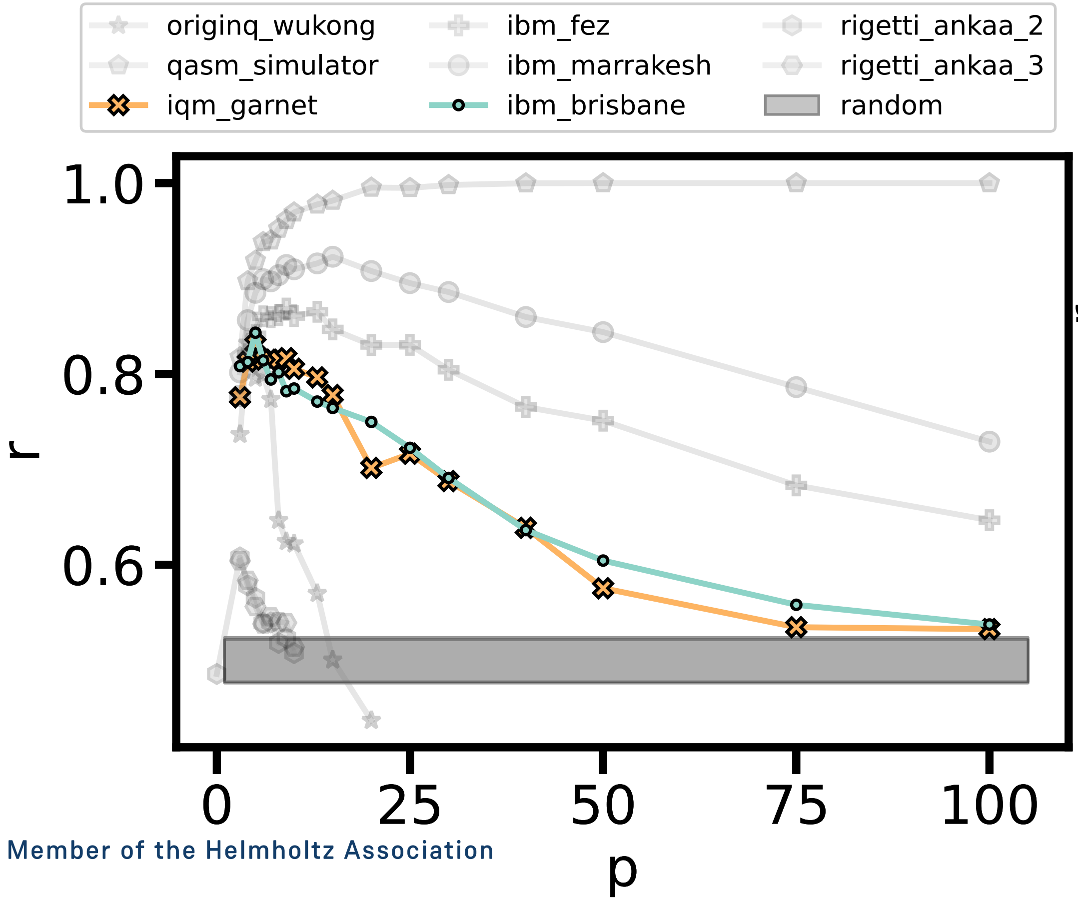
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



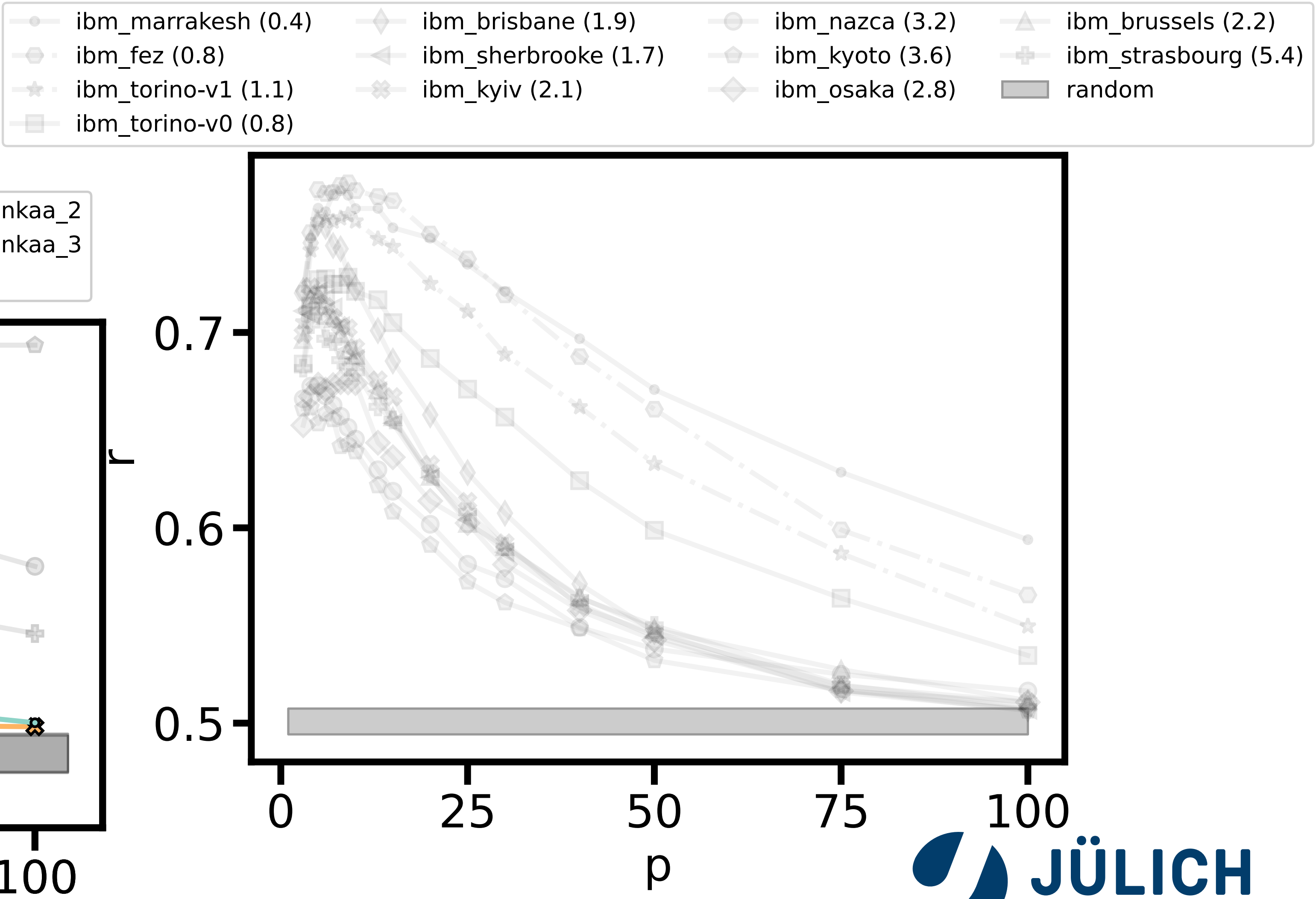
LR-QAOA on a 1D-Chain graph



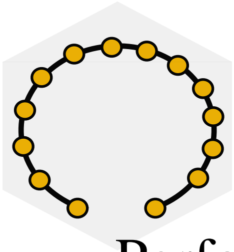
Performance on the best 5-qubit 1D chain experiment, different QPUs



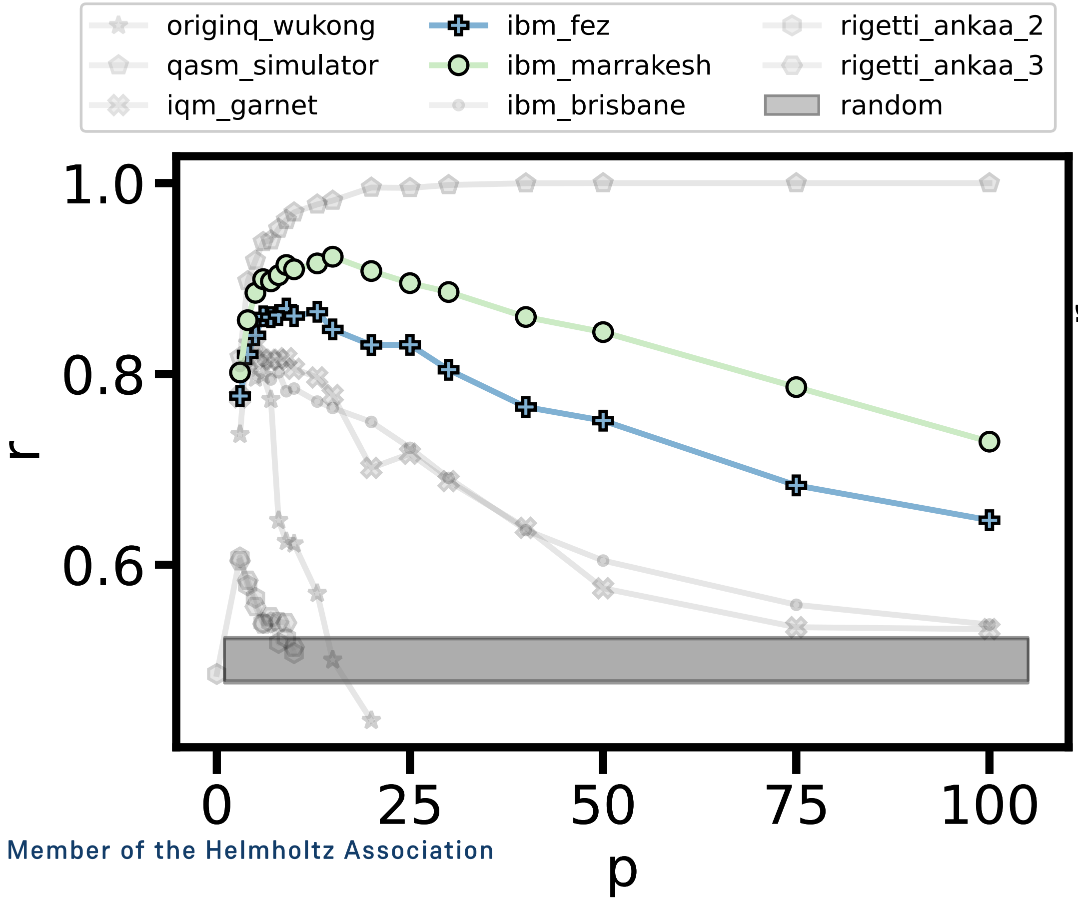
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



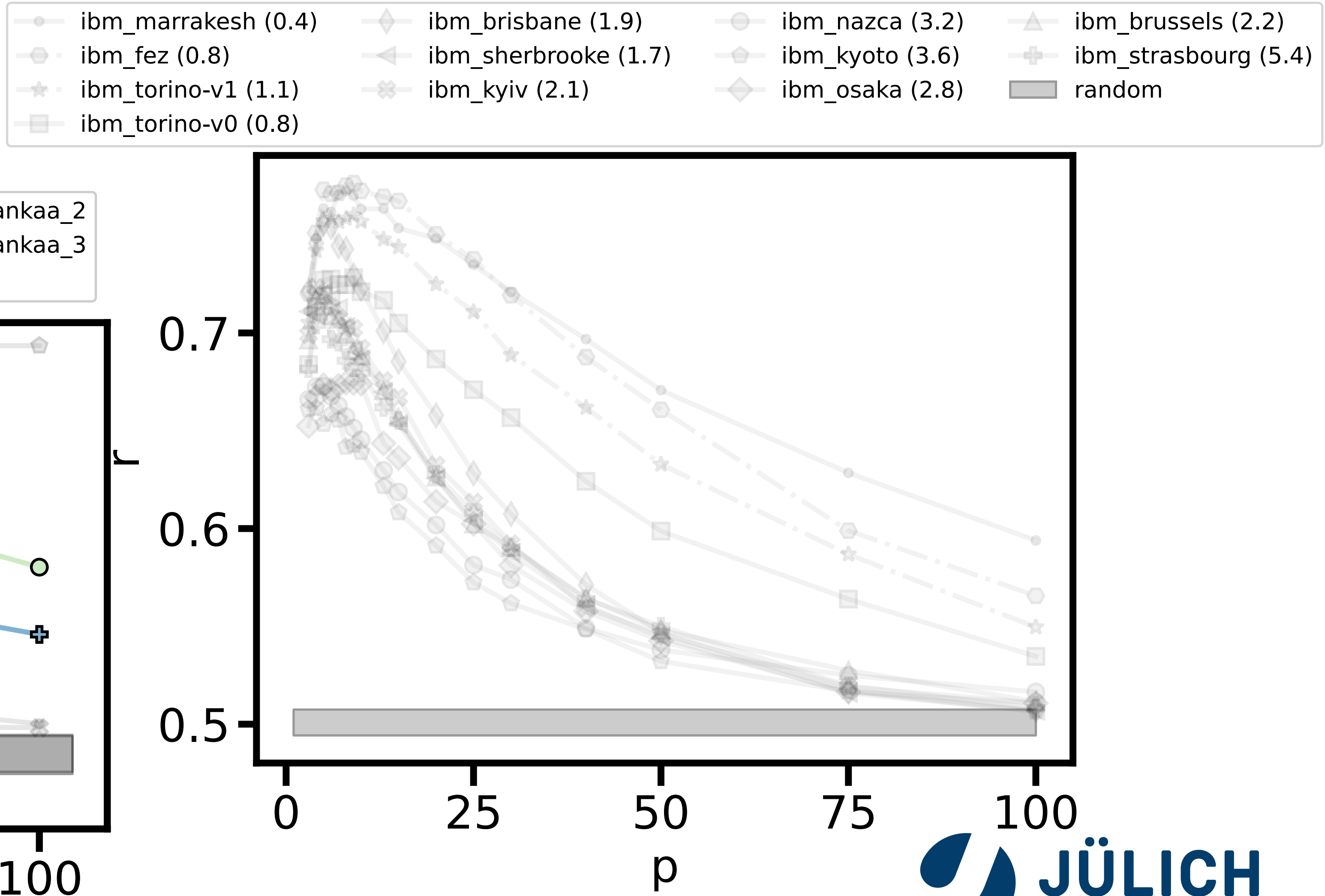
LR-QAOA on a 1D-Chain graph



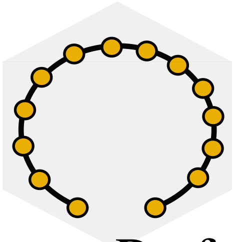
Performance on the best 5-qubit 1D chain experiment, different QPUs



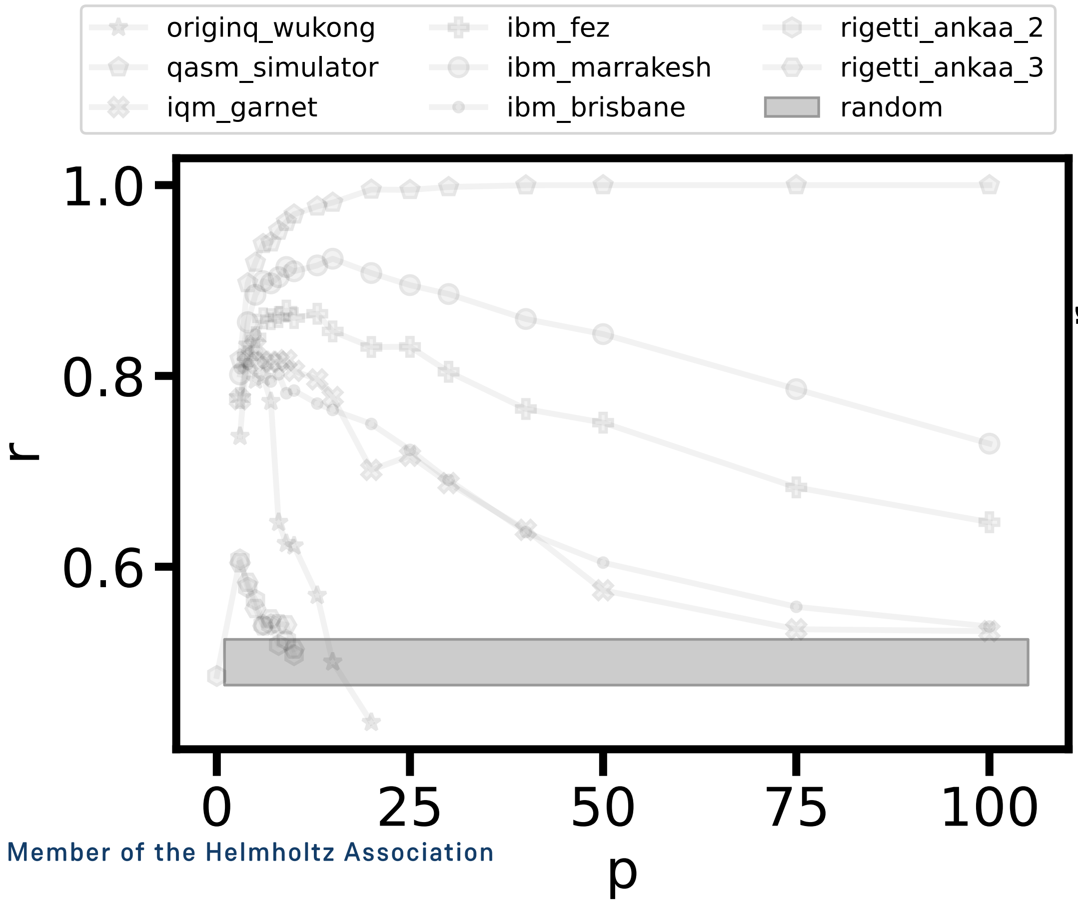
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



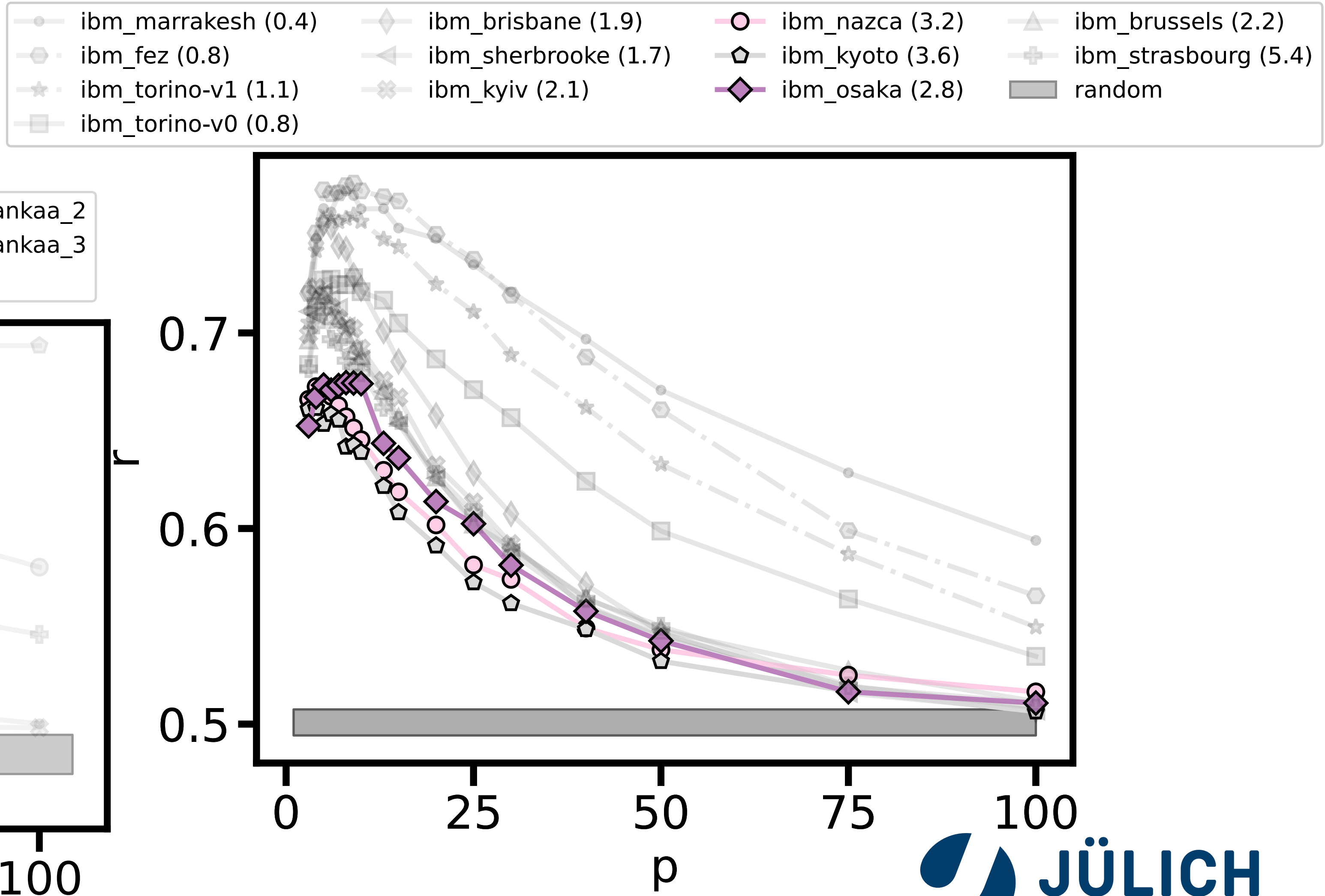
LR-QAOA on a 1D-Chain graph



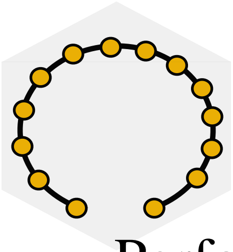
Performance on the best 5-qubit 1D chain experiment, different QPUs



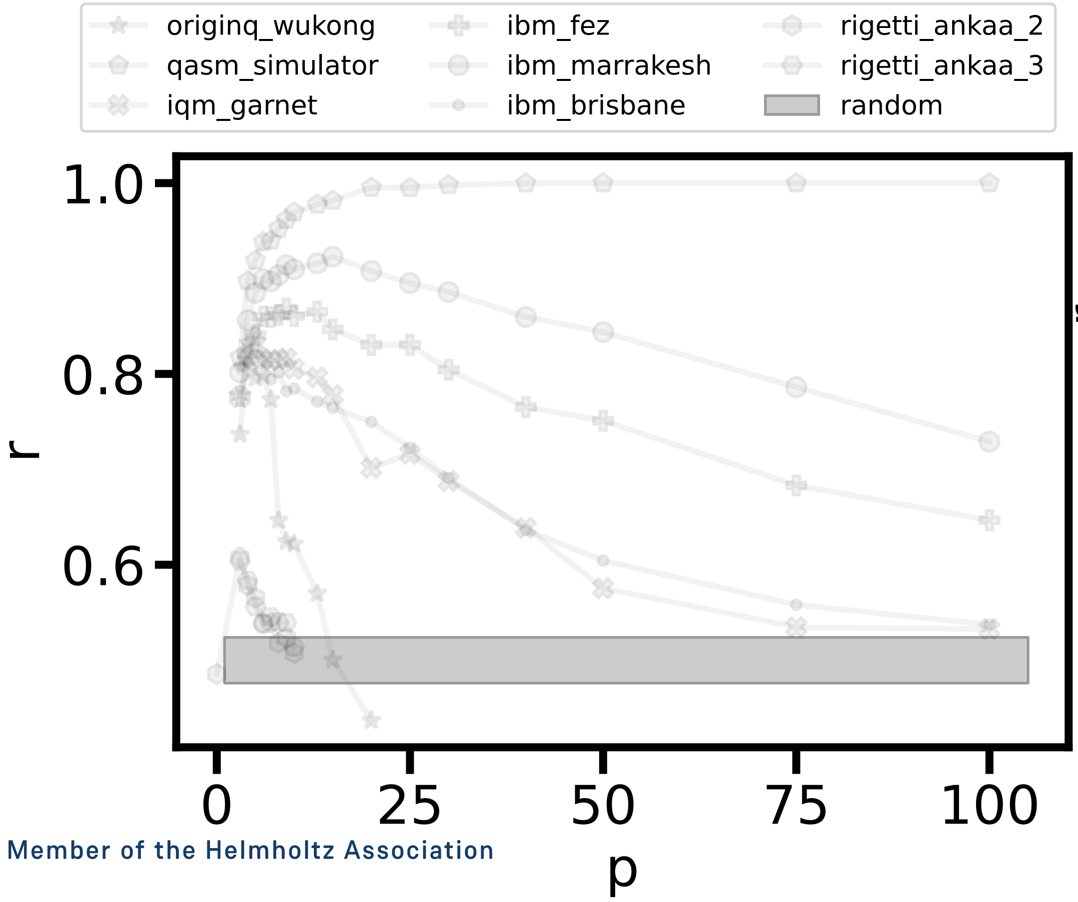
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



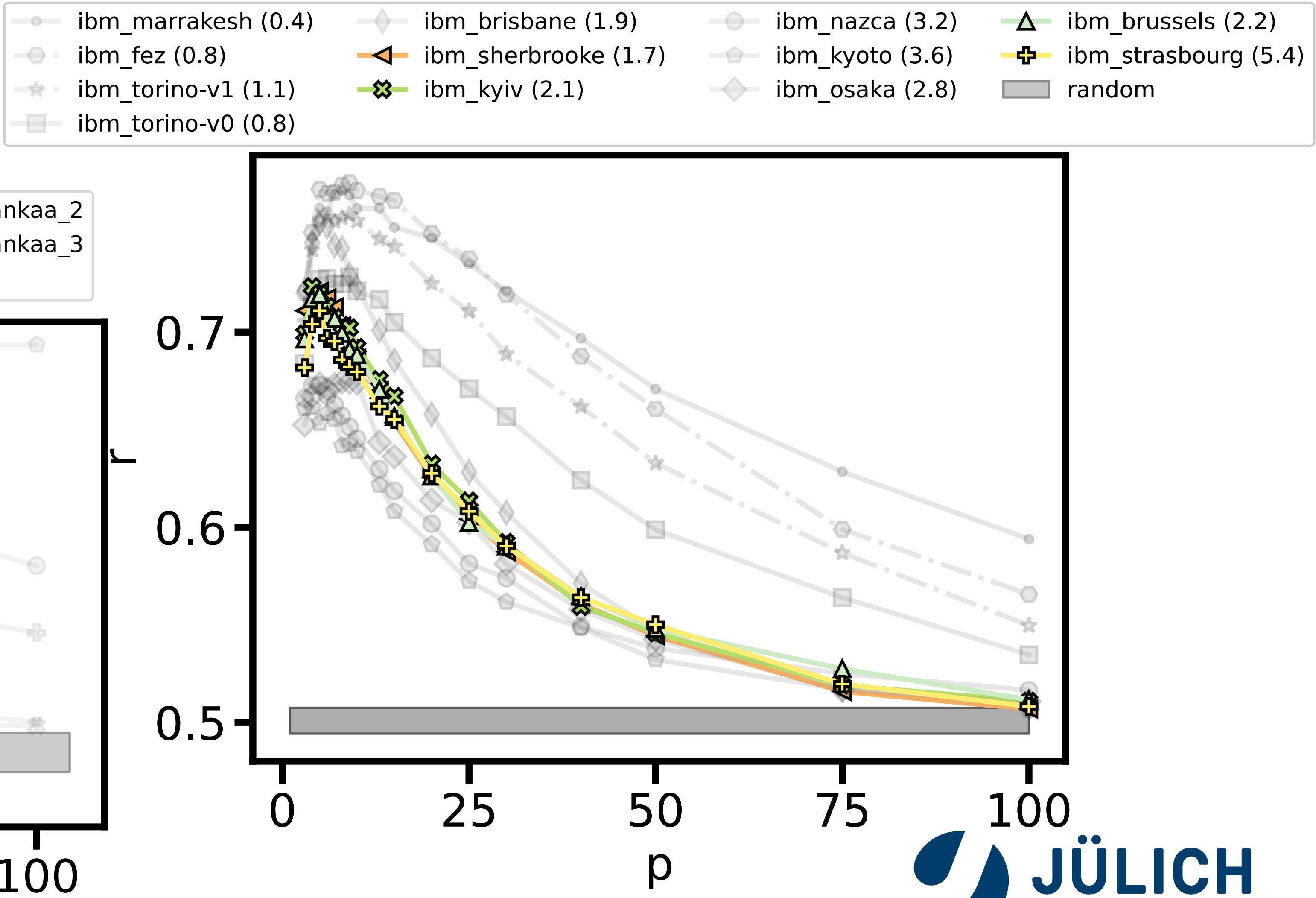
LR-QAOA on a 1D-Chain graph



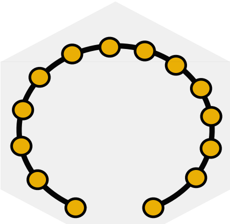
Performance on the best 5-qubit 1D chain experiment, different QPUs



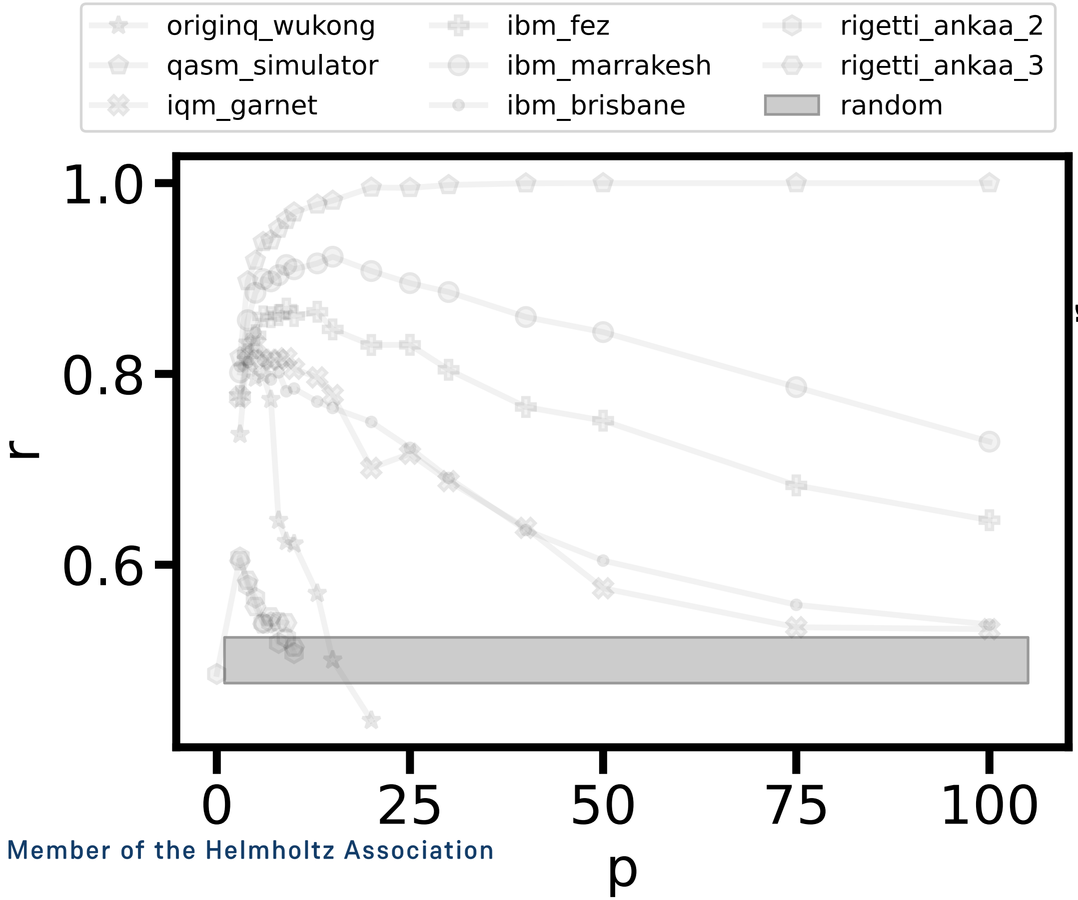
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



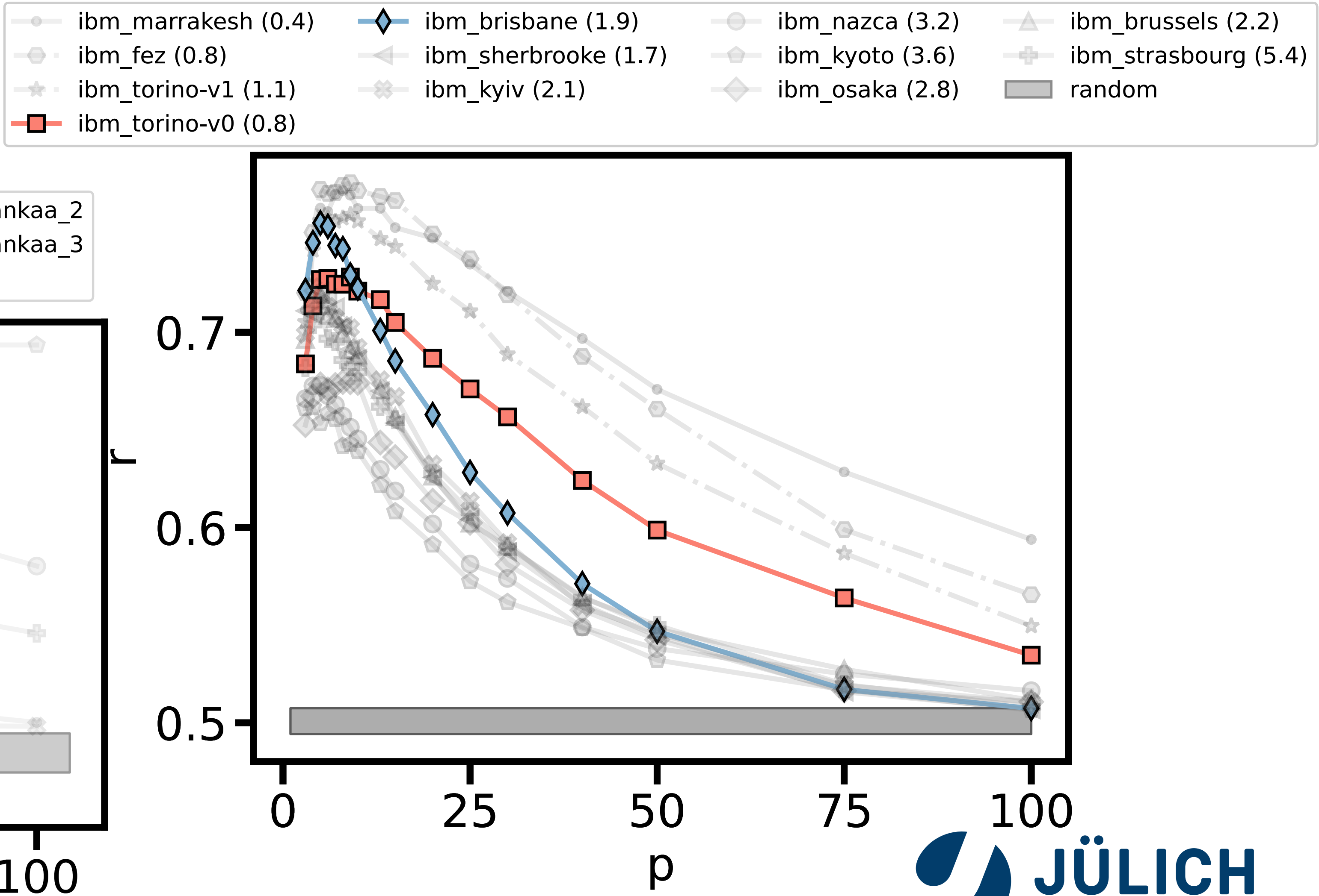
LR-QAOA on a 1D-Chain graph



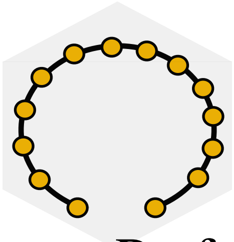
Performance on the best 5-qubit 1D chain experiment, different QPUs



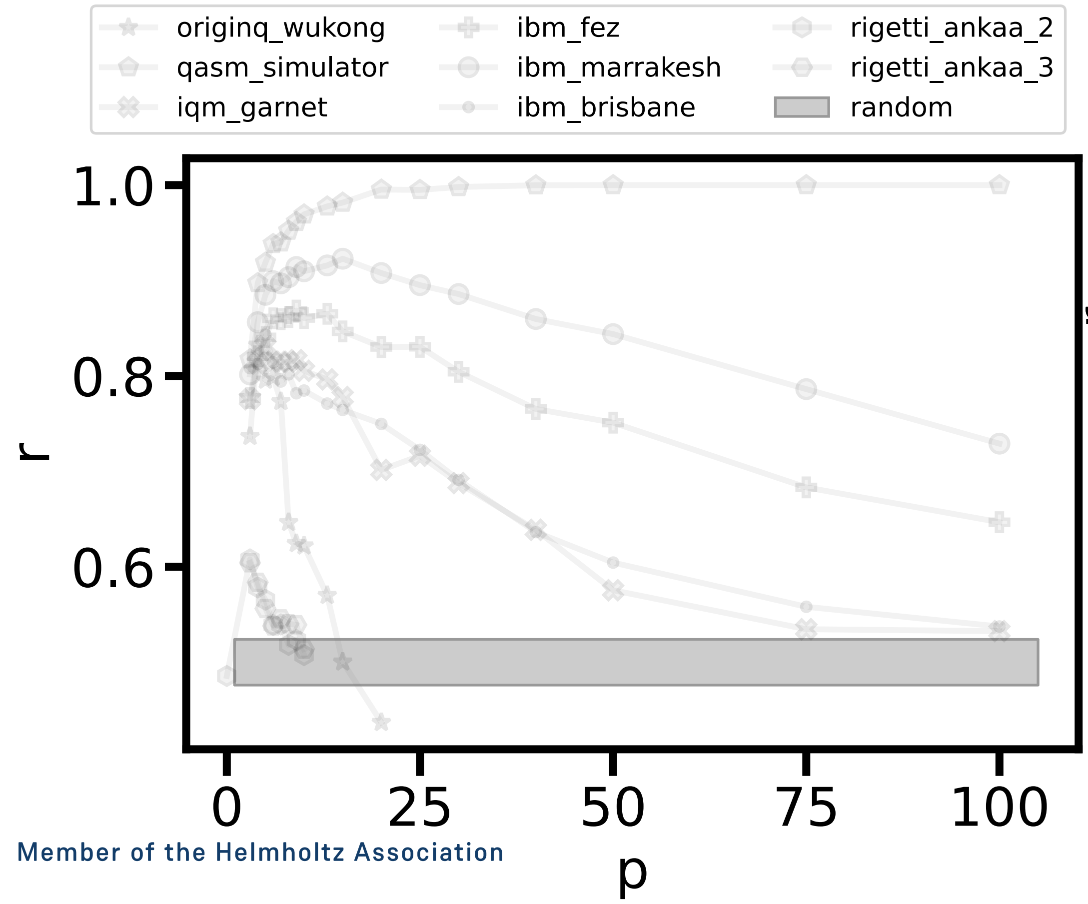
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



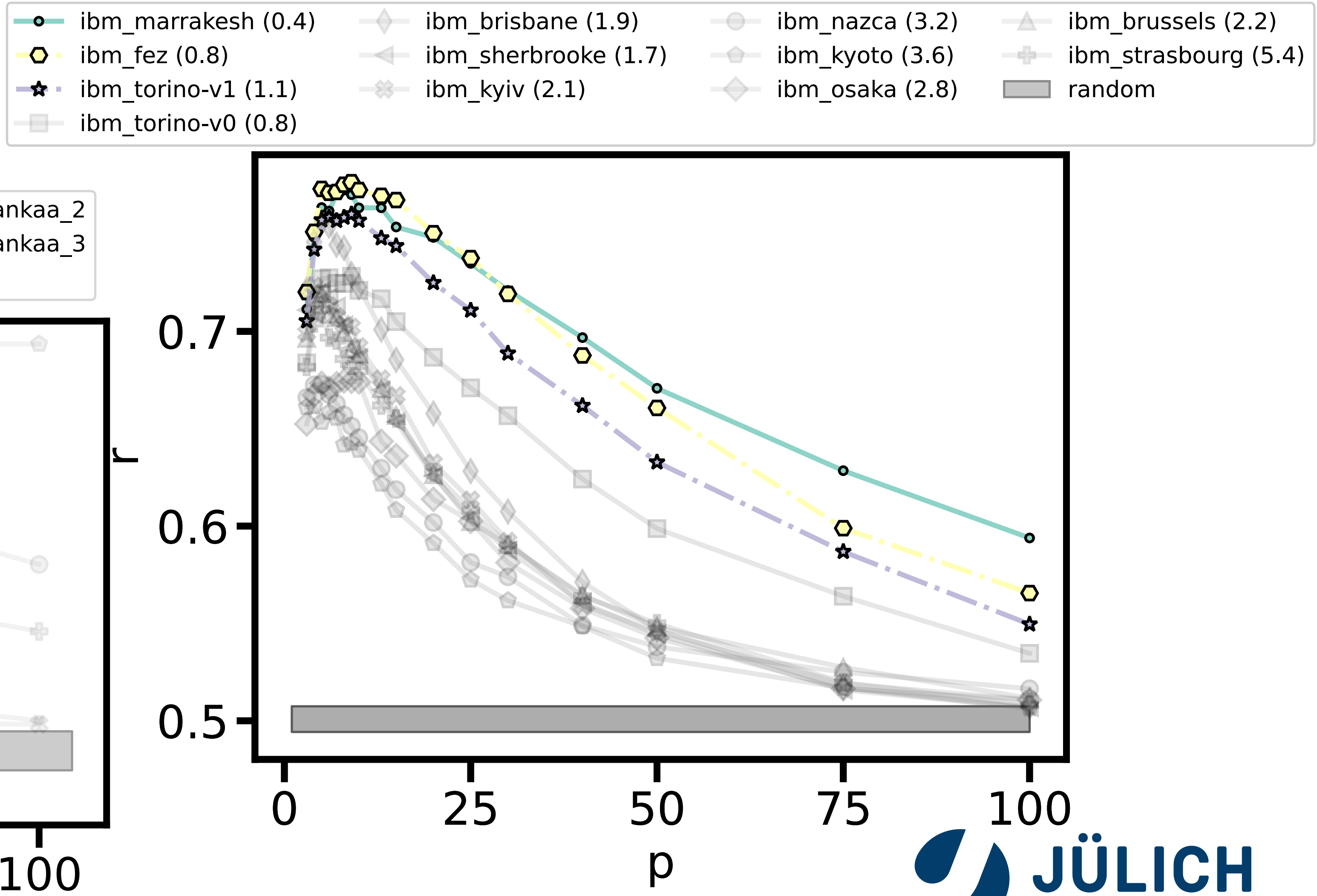
LR-QAOA on a 1D-Chain graph



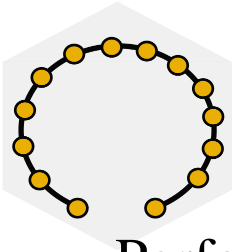
Performance on the best 5-qubit 1D chain experiment, different QPUs



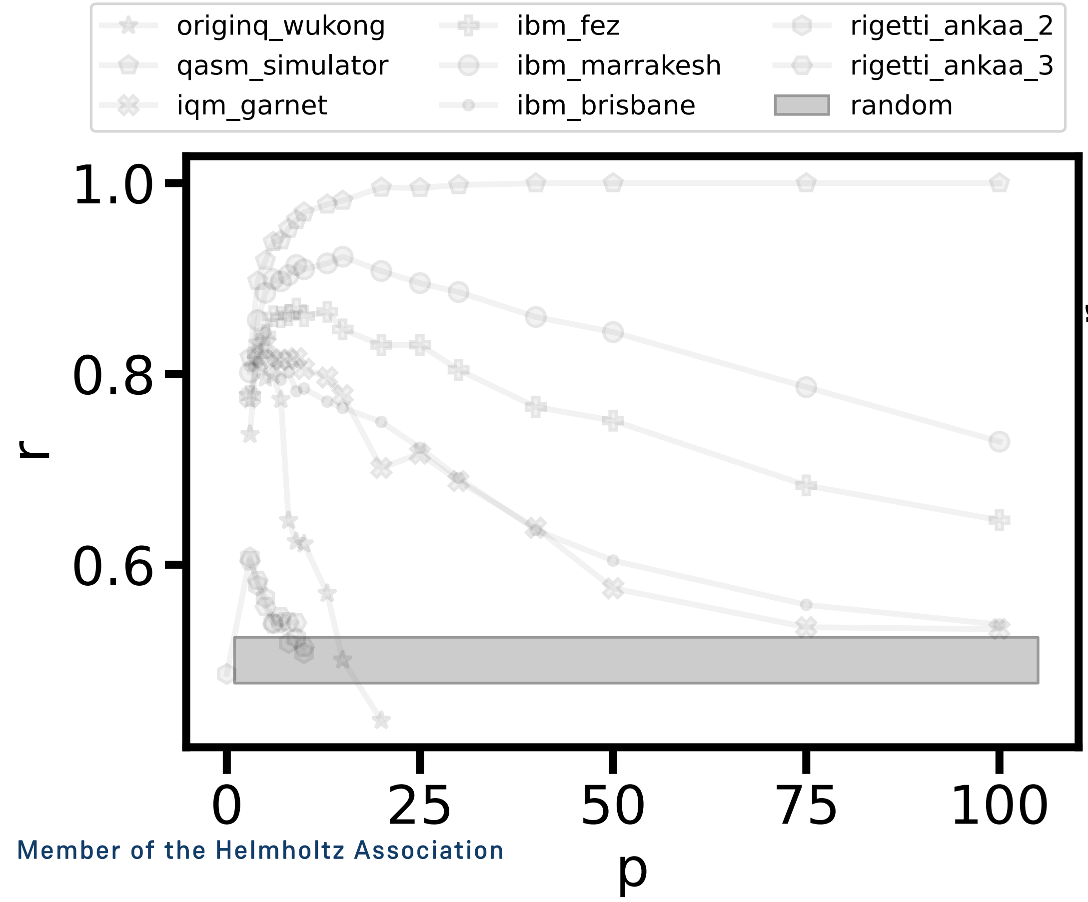
Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



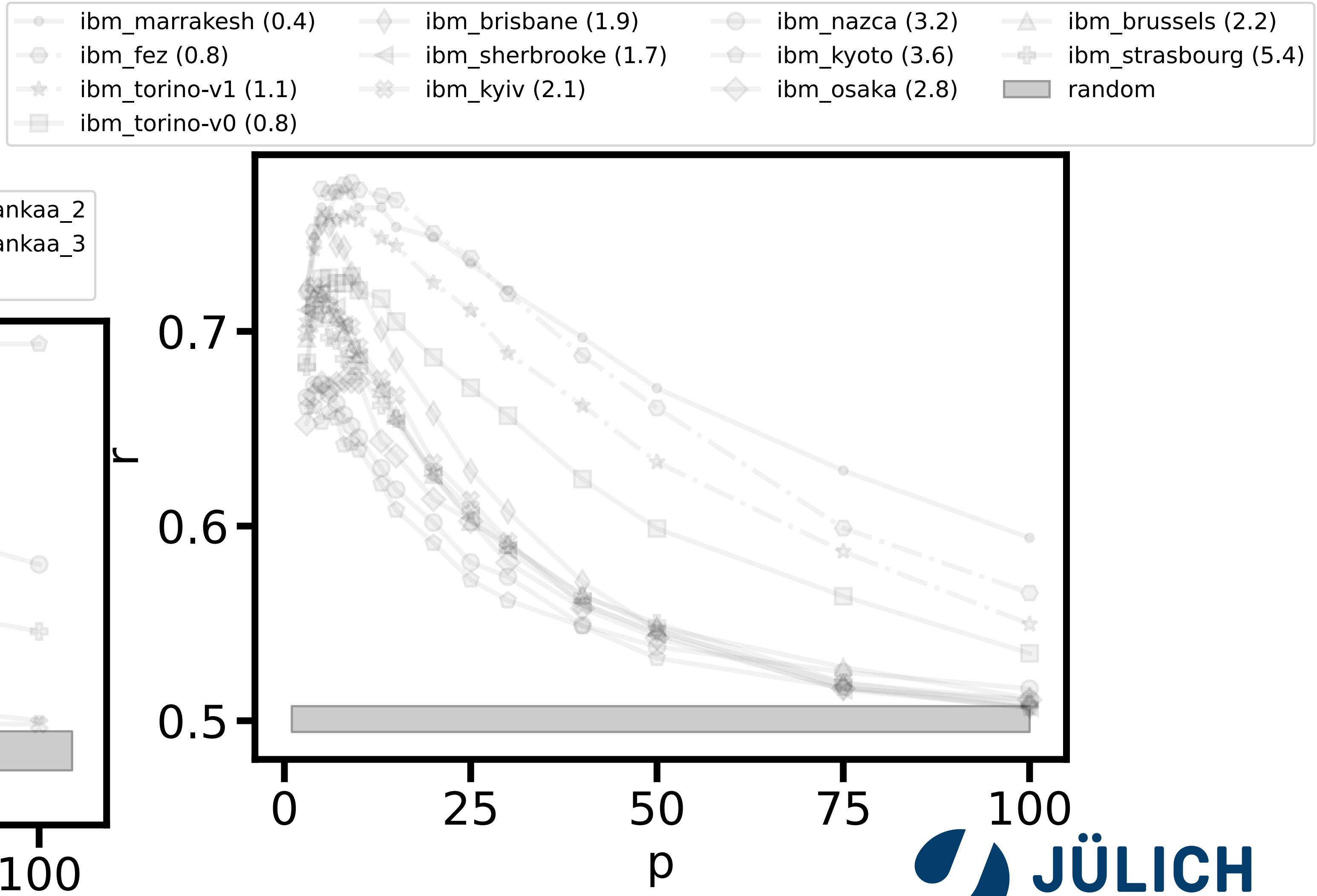
LR-QAOA on a 1D-Chain graph



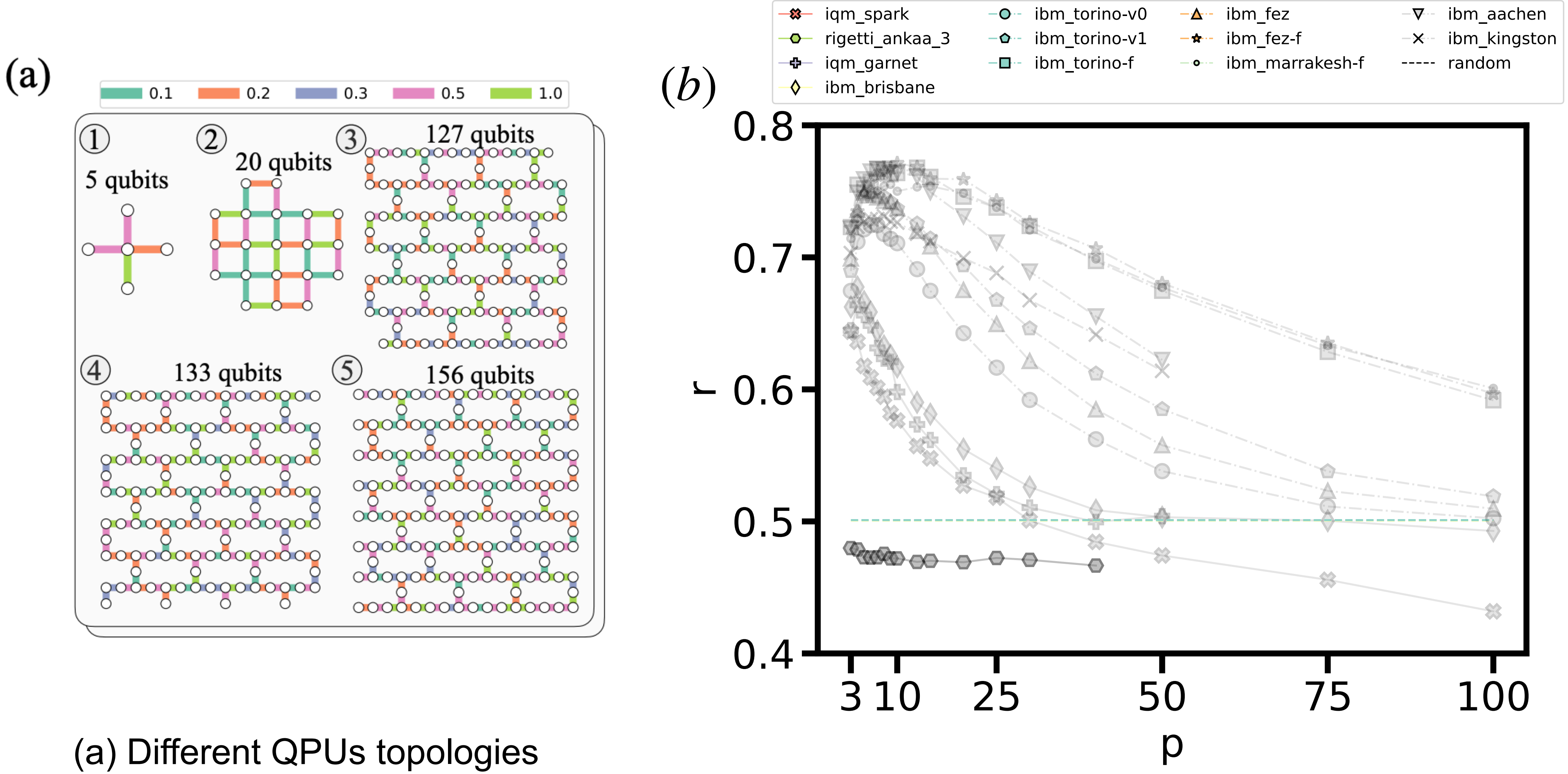
Performance on the best 5-qubit 1D chain experiment, different QPUs



Performance on a 100-qubit 1D chain experiment. IBM QPUs (EPLG)



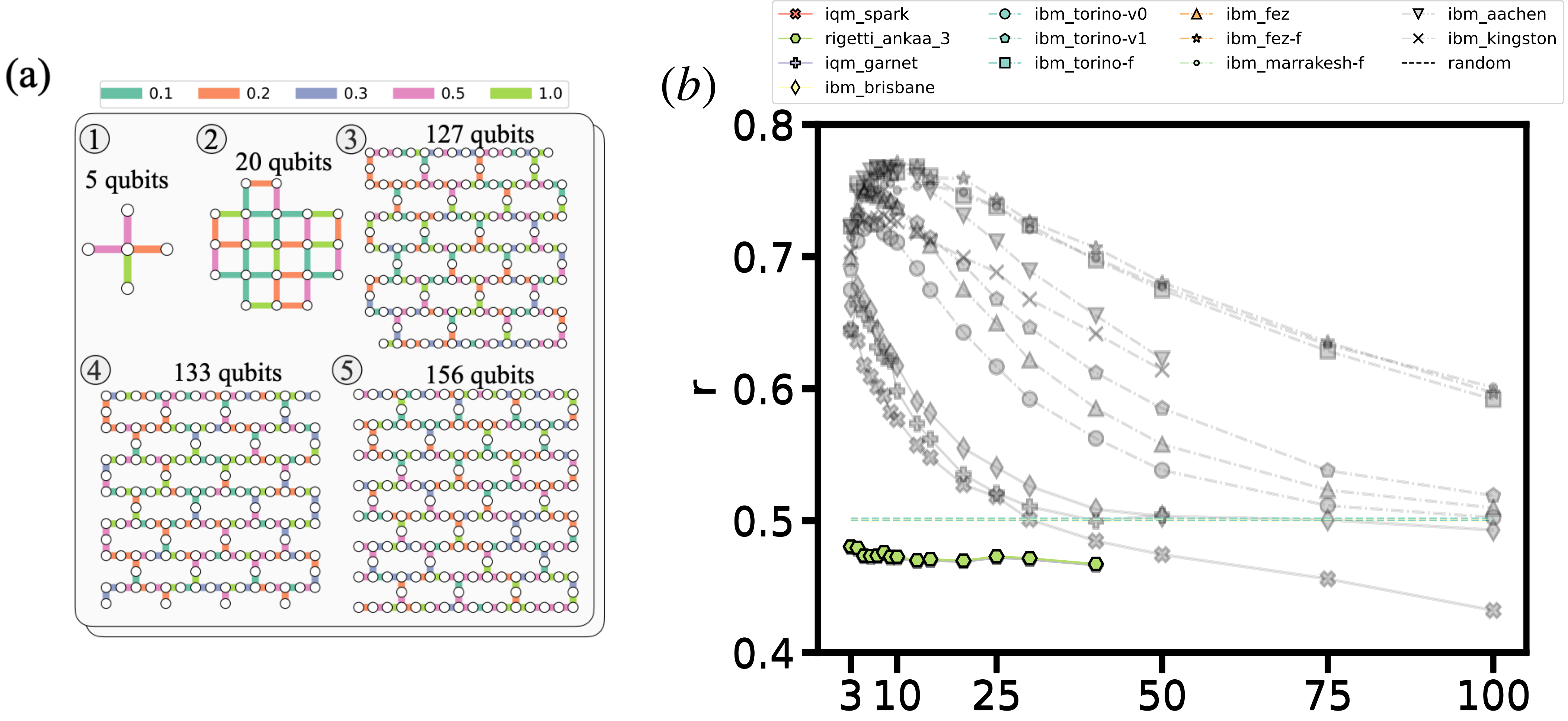
LR-QAOA on a Native layout graph



(a) Different QPUs topologies
(b) Performance on different devices

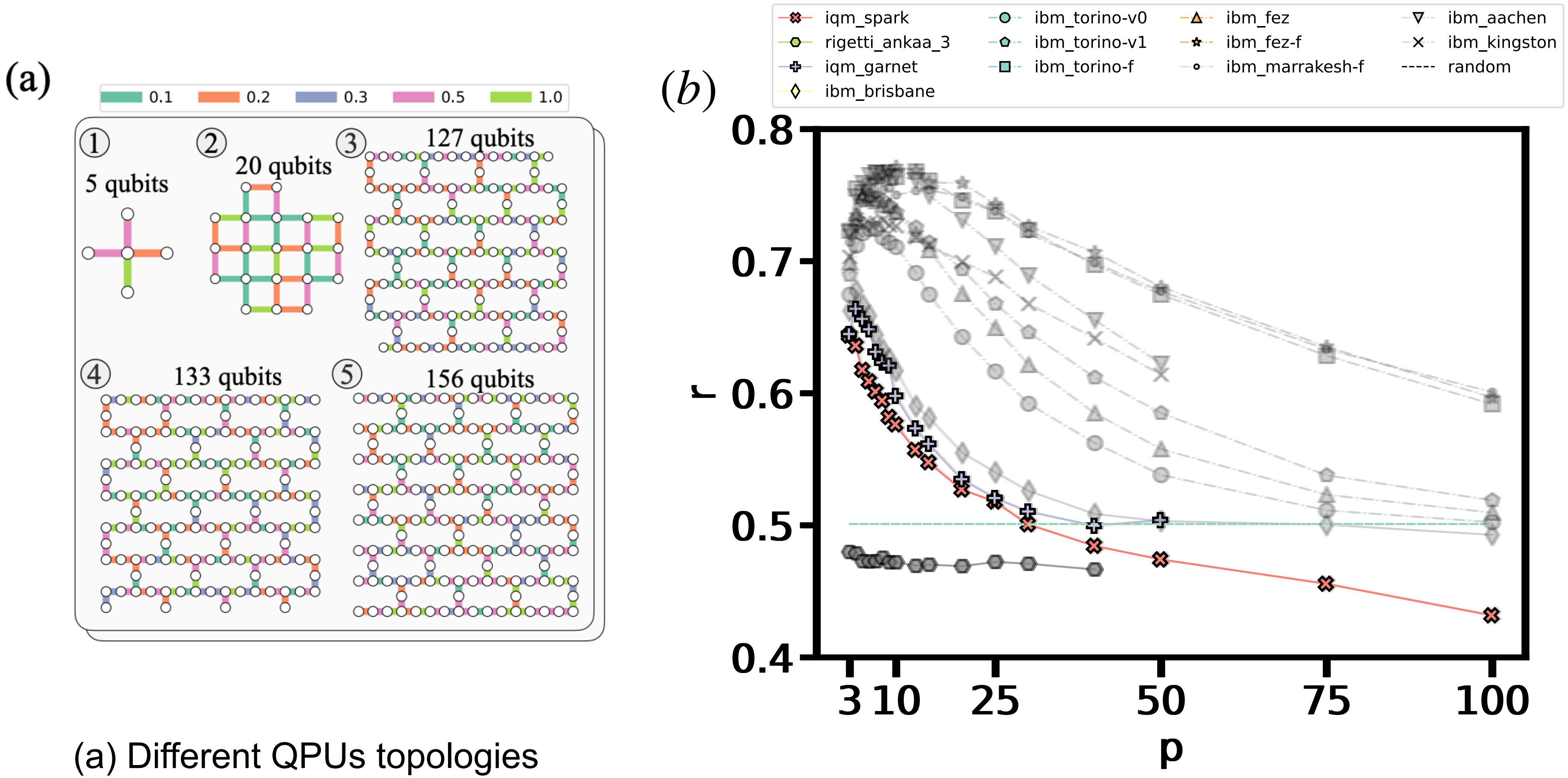
JÜLICH Forschungszentrum

LR-QAOA on a Native layout graph



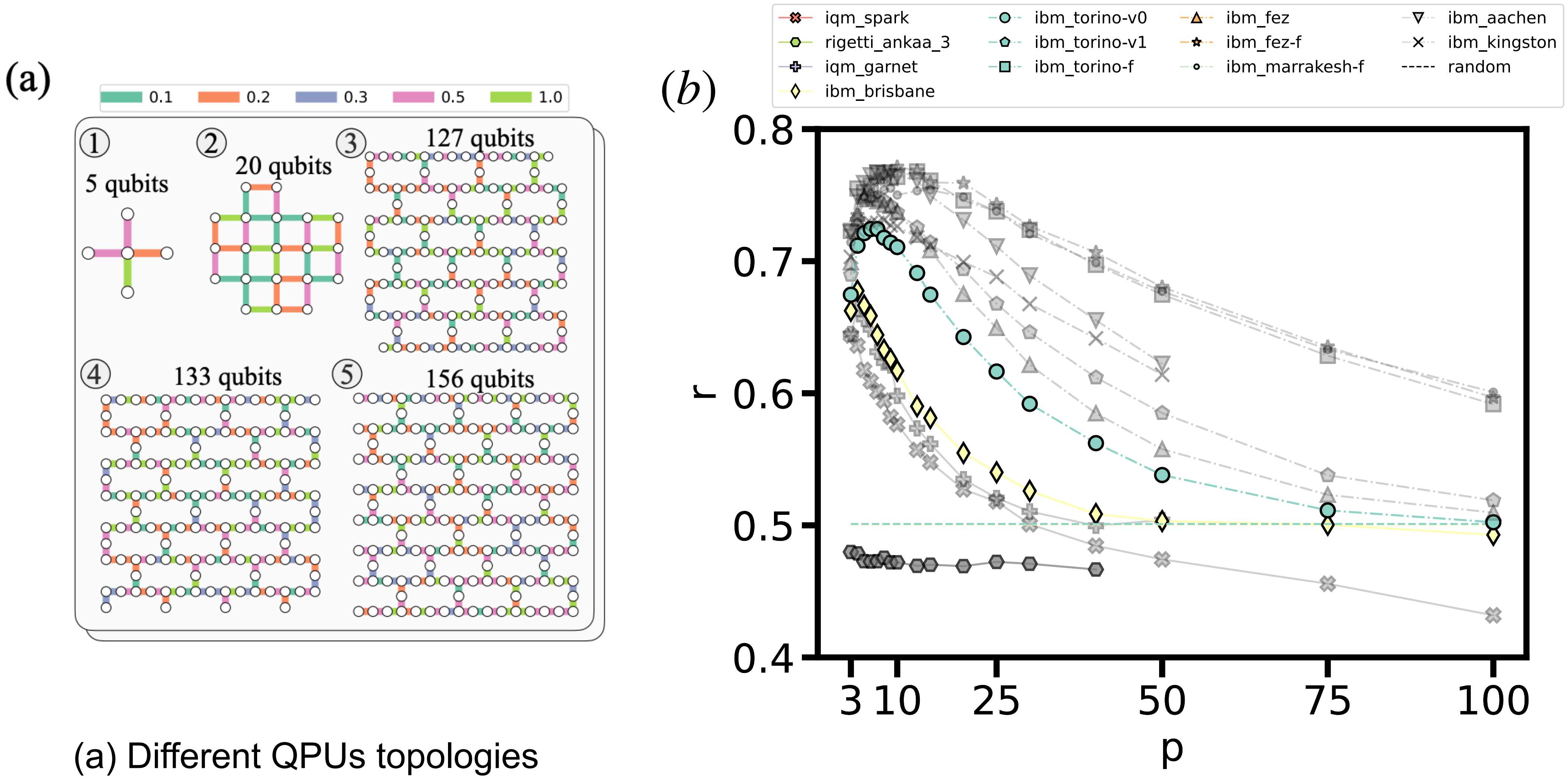
(a) Different QPUs topologies
(b) Performance on different devices

LR-QAOA on a Native layout graph



(a) Different QPUs topologies
(b) Performance on different devices

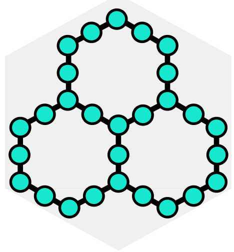
LR-QAOA on a Native layout graph

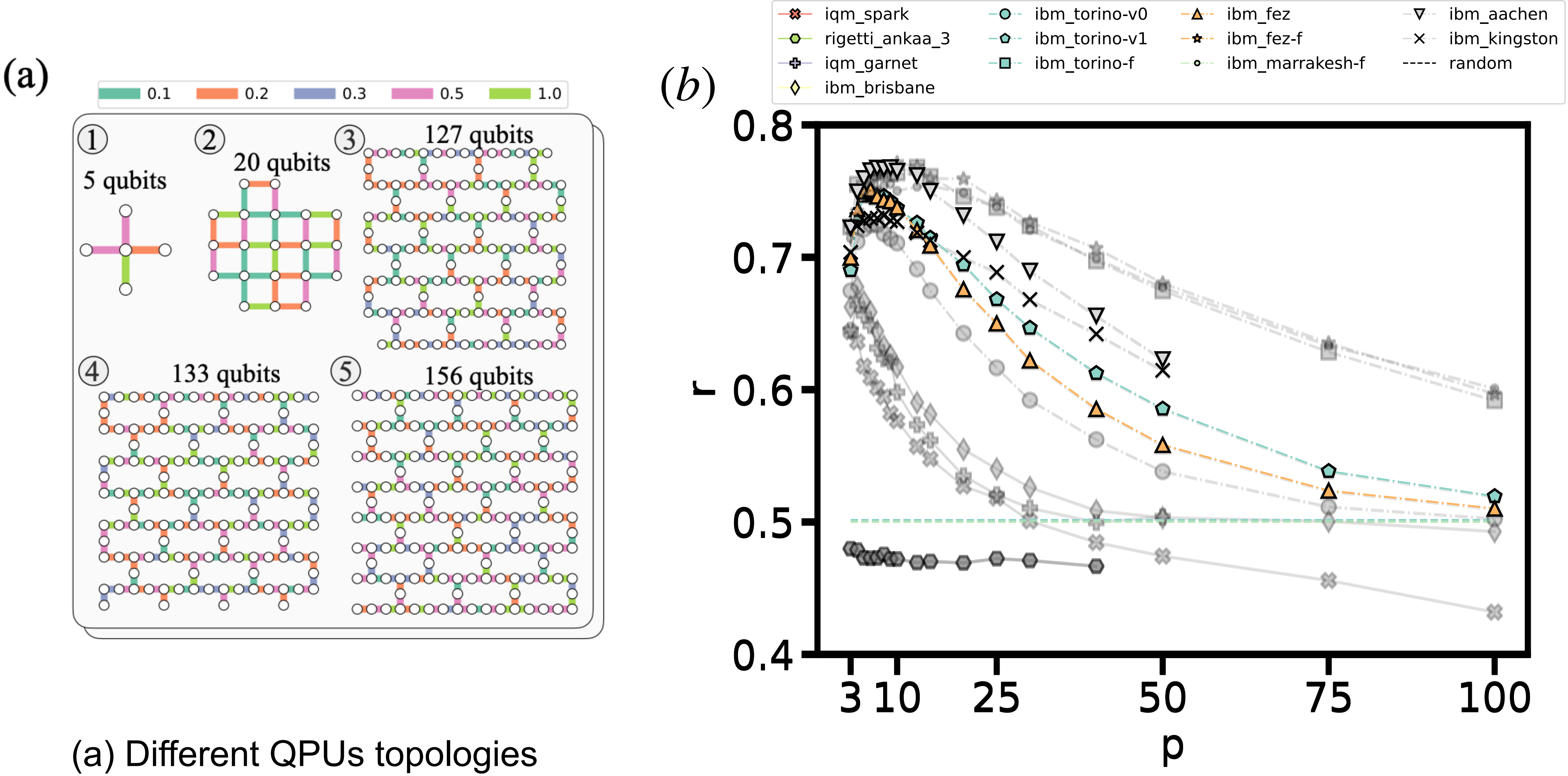


(a) Different QPUs topologies
(b) Performance on different devices

JÜLICH Forschungszentrum

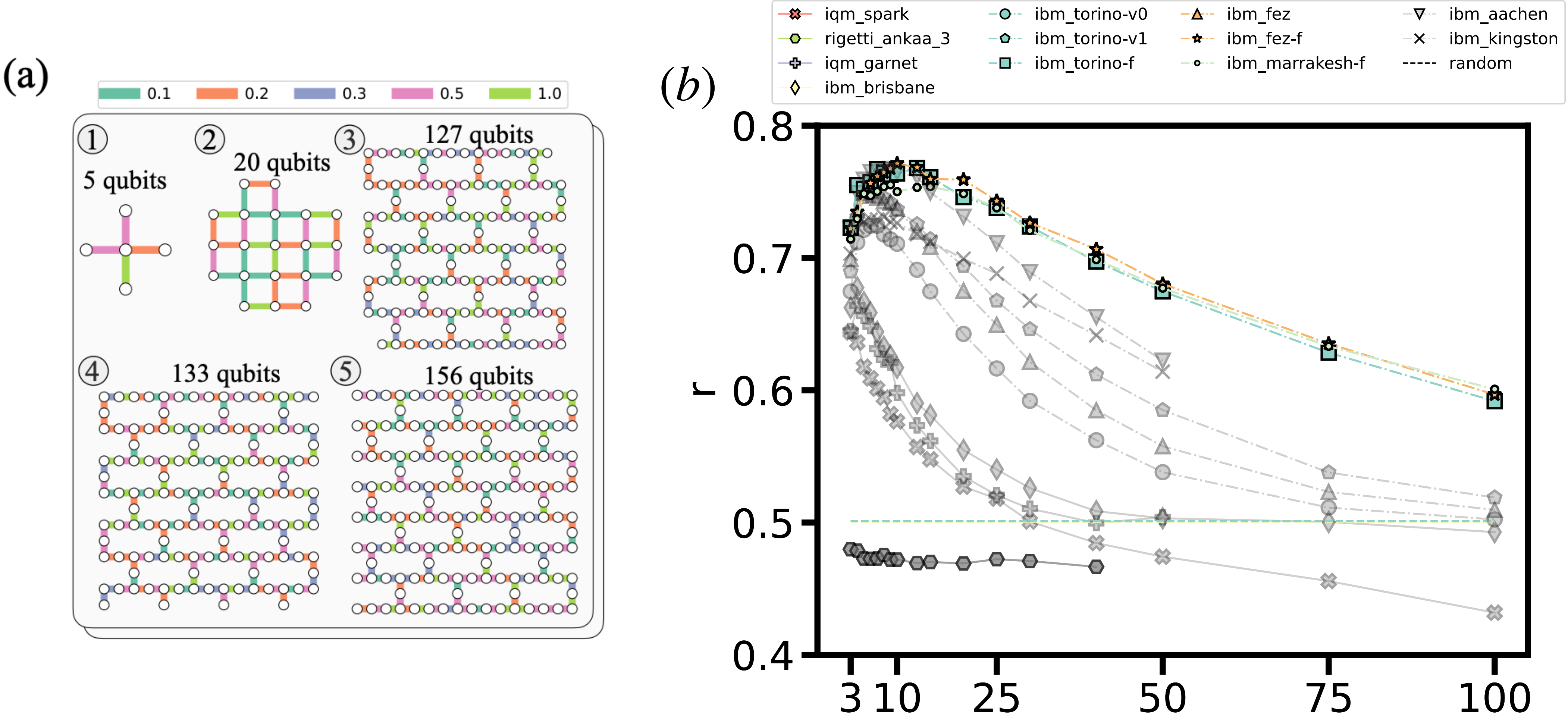
LR-QAOA on a Native layout graph





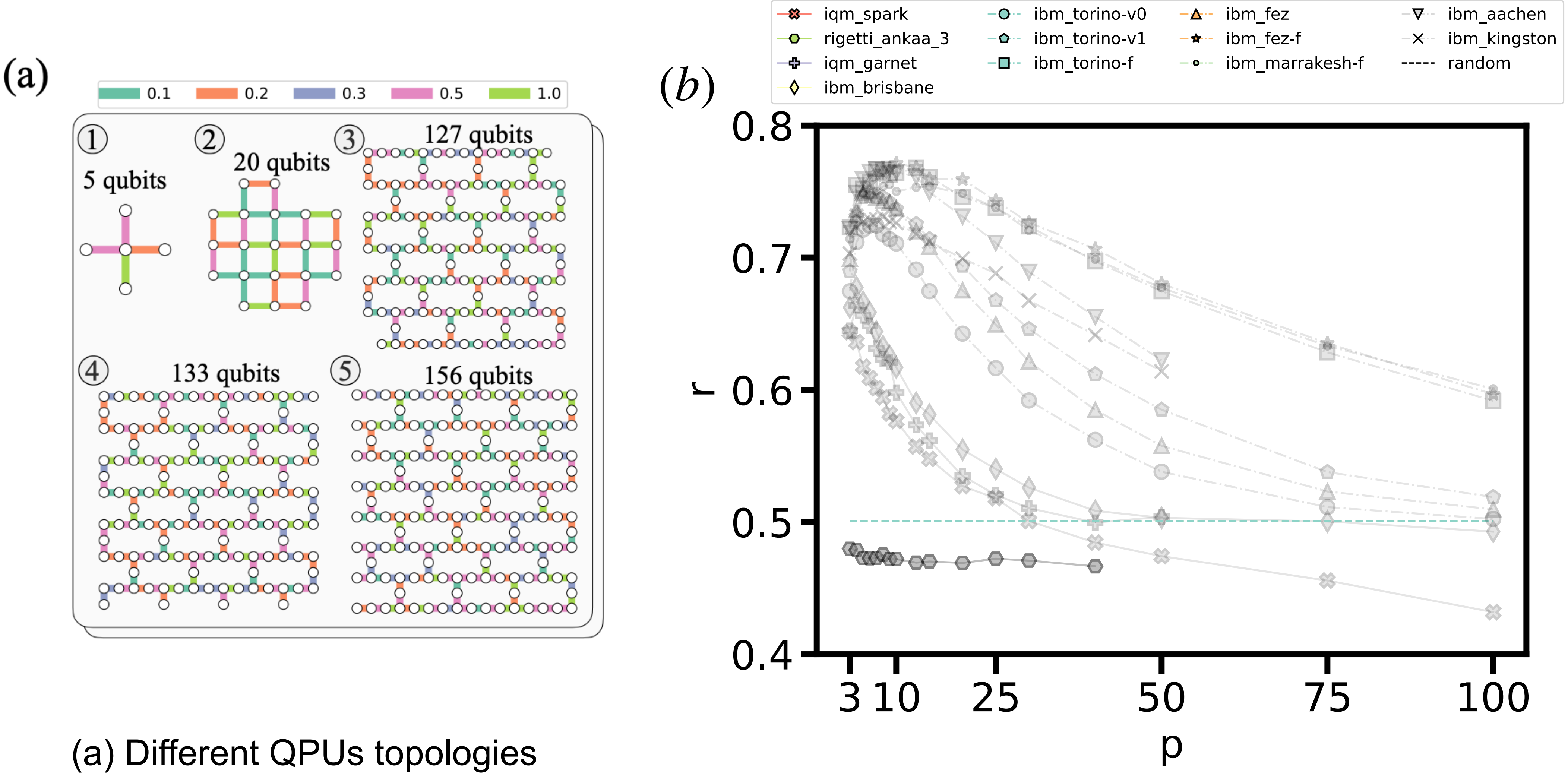
(a) Different QPUs topologies
(b) Performance on different devices

LR-QAOA on a Native layout graph



(a) Different QPUs topologies
(b) Performance on different devices

LR-QAOA on a Native layout graph

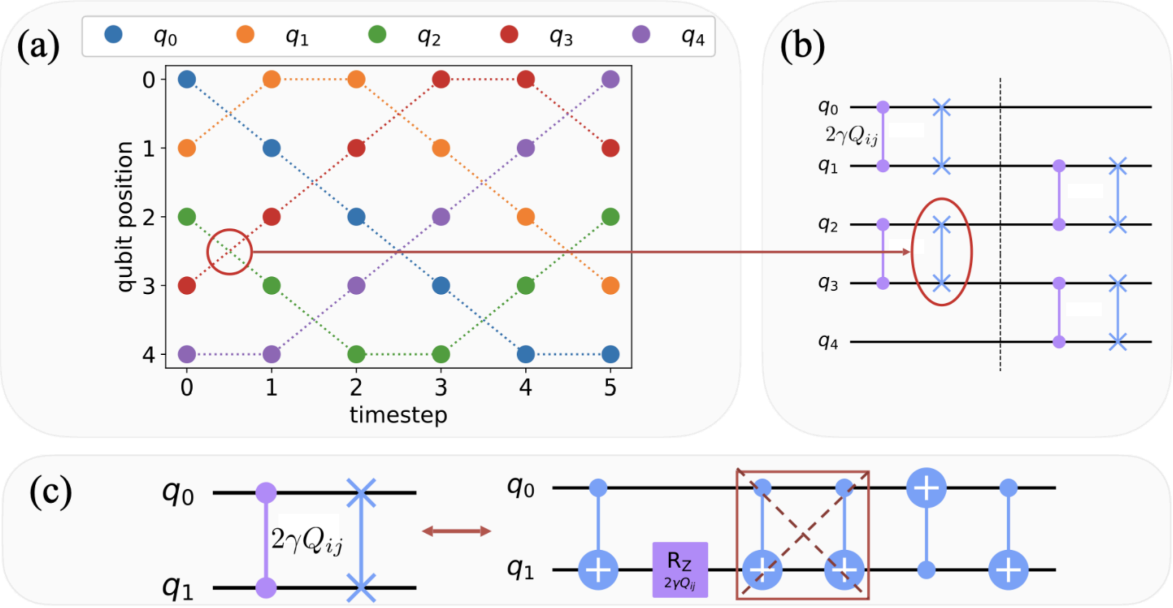


(a) Different QPUs topologies
(b) Performance on different devices

JÜLICH Forschungszentrum

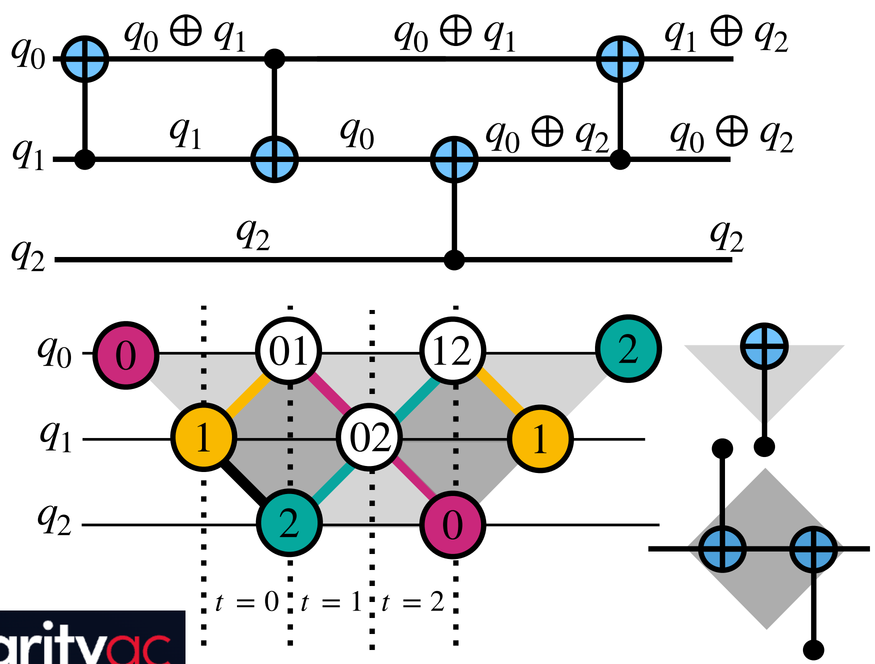
From a fixed layout to a fully connected QPUs

SWAP networks



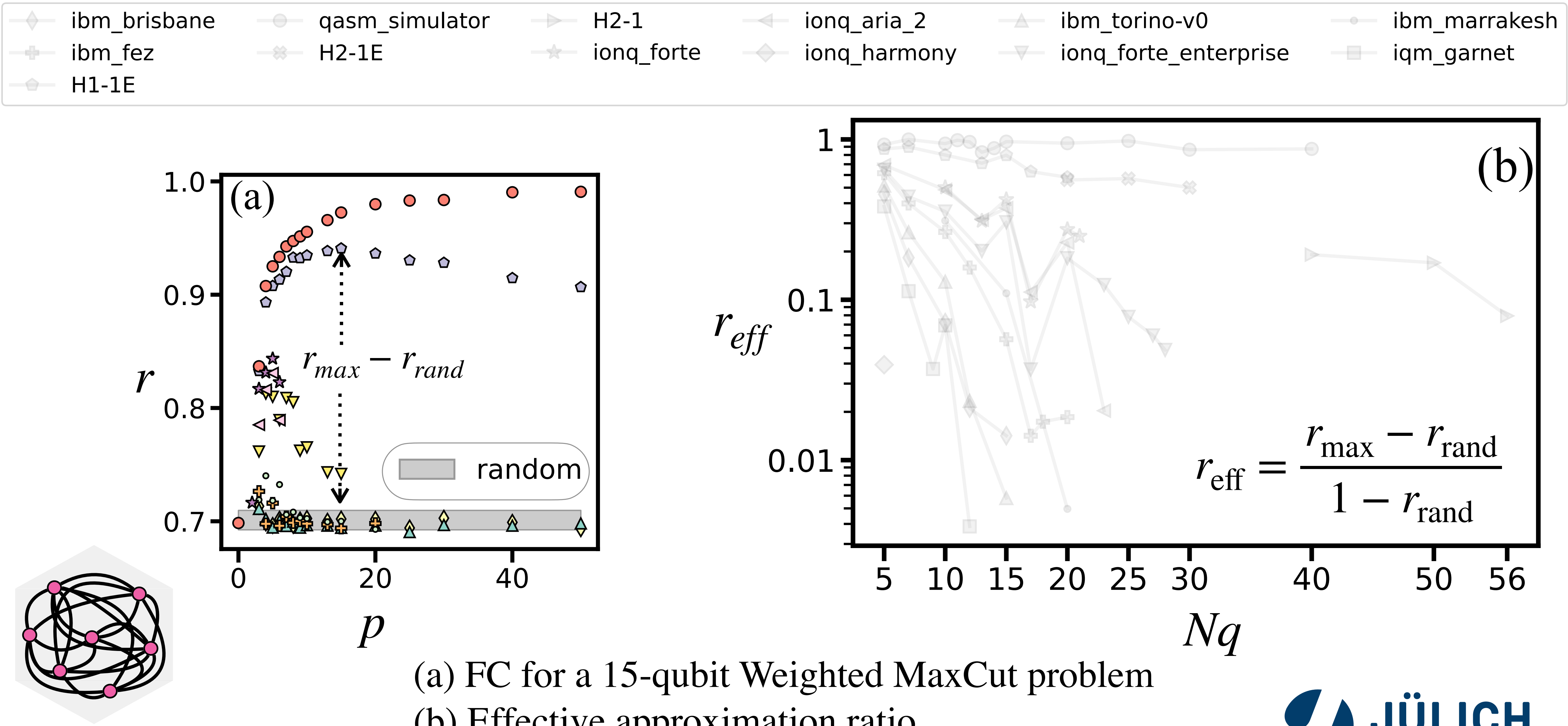
Using a swap strategy we can convert a 1D-Chain graph into a fully connected graph. We need 3 times more 2-qubit gates to implement this protocol.

Parity Twine Chain



Schematic representation of the PTC encoding. (a) Circuit model to get different parities. (b) Graphic representation of the PTC in a 1D chain. Using this diagram, the equivalent building blocks for the CNOT gates are shown at the right.

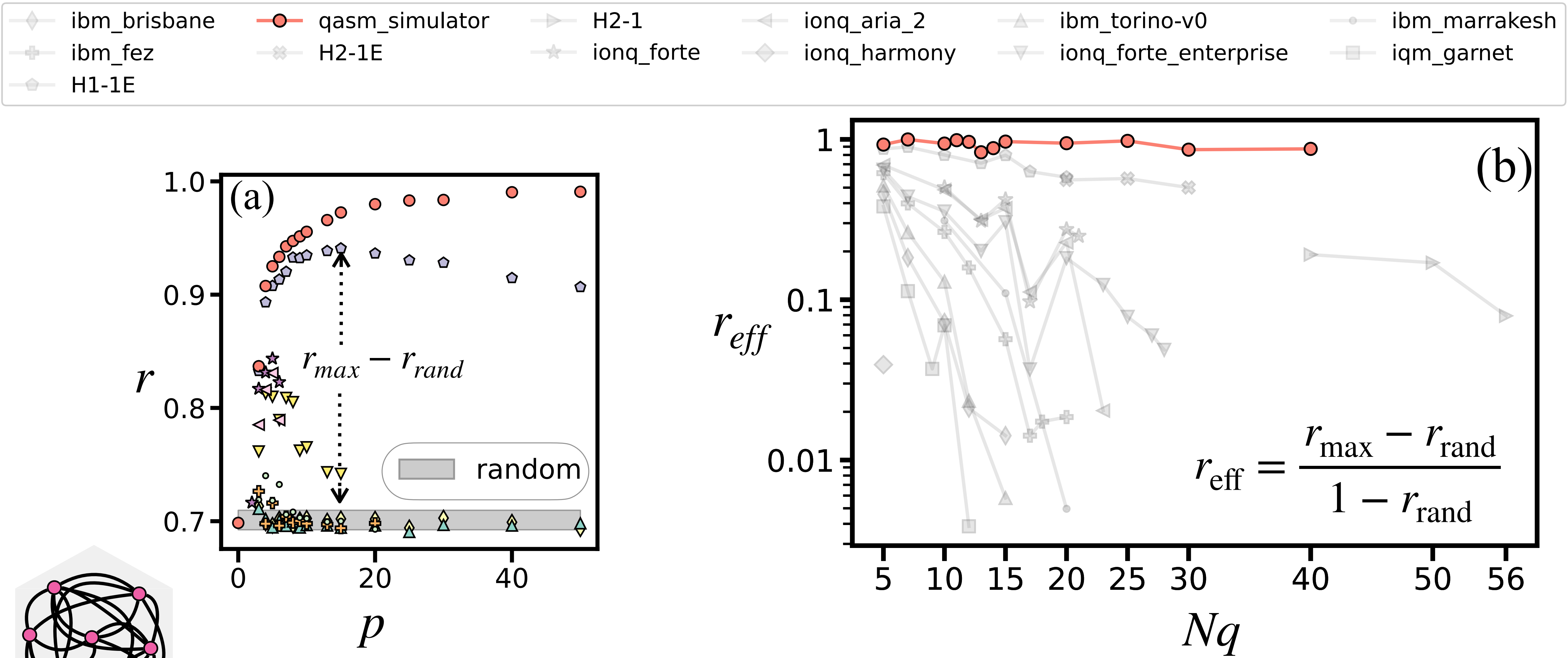
LR-QAOA on Fully Connected (FC) problems



(a) FC for a 15-qubit Weighted MaxCut problem

(b) Effective approximation ratio

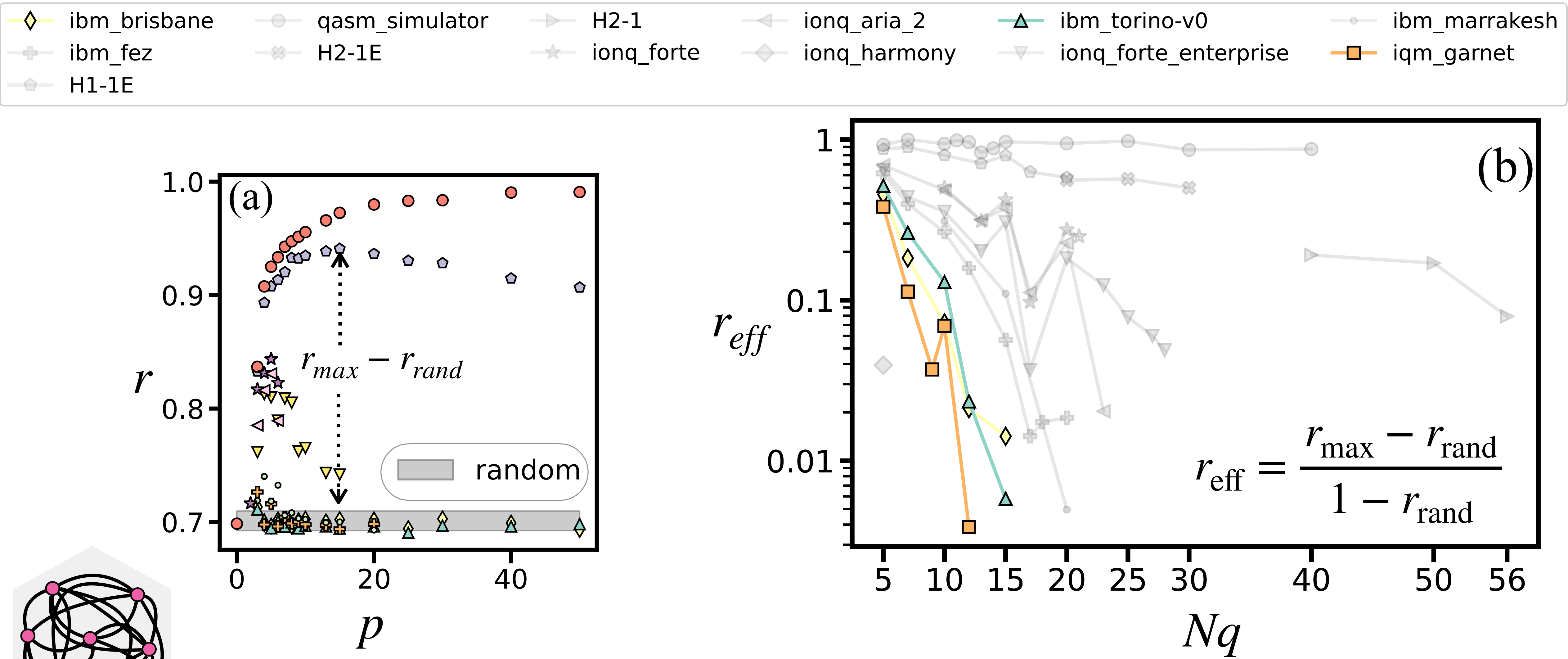
LR-QAOA on Fully Connected (FC) problems



(a) FC for a 15-qubit Weighted MaxCut problem

(b) Effective approximation ratio

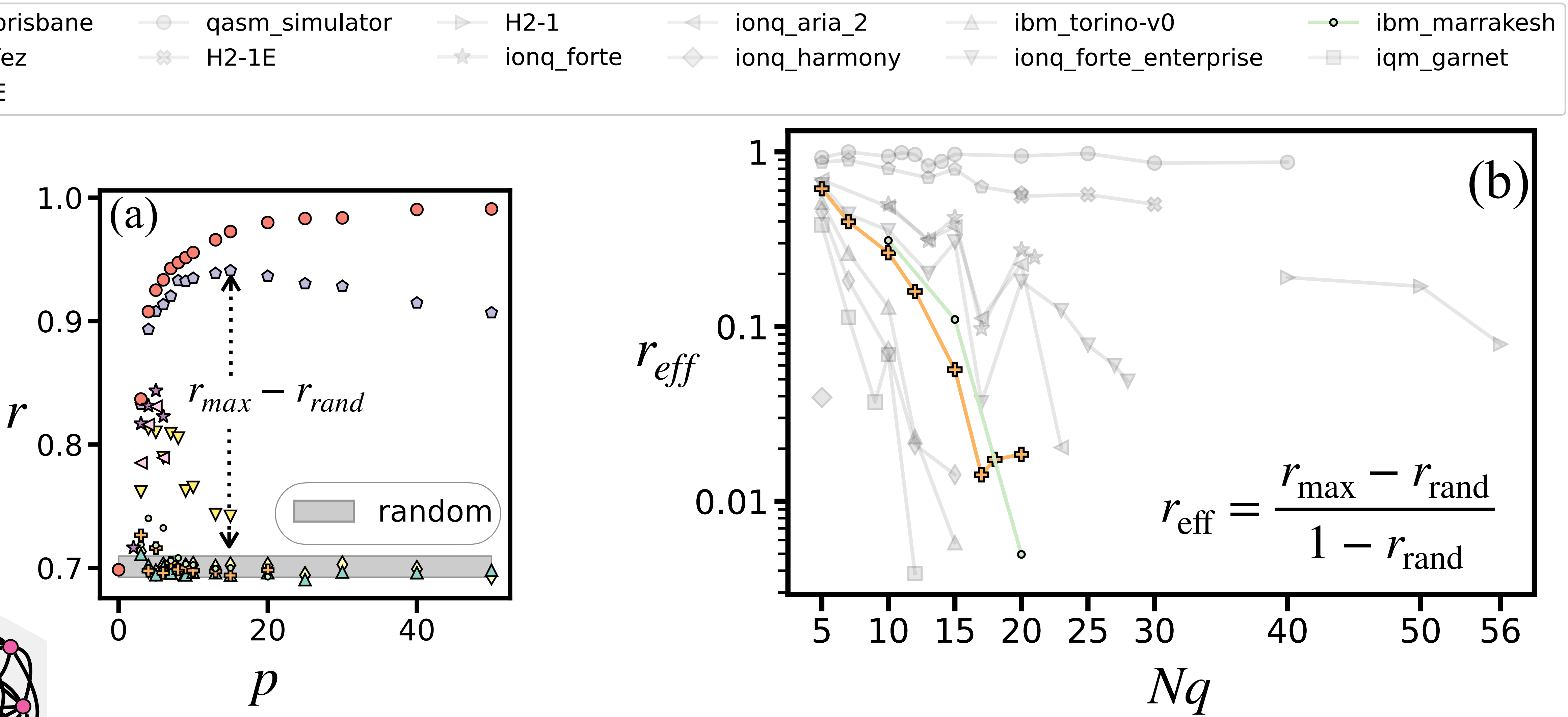
LR-QAOA on Fully Connected (FC) problems

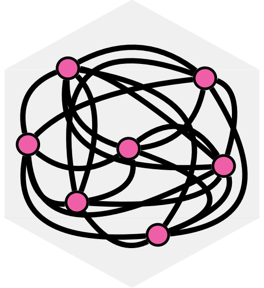


(a) FC for a 15-qubit Weighted MaxCut problem

(b) Effective approximation ratio

LR-QAOA on Fully Connected (FC) problems

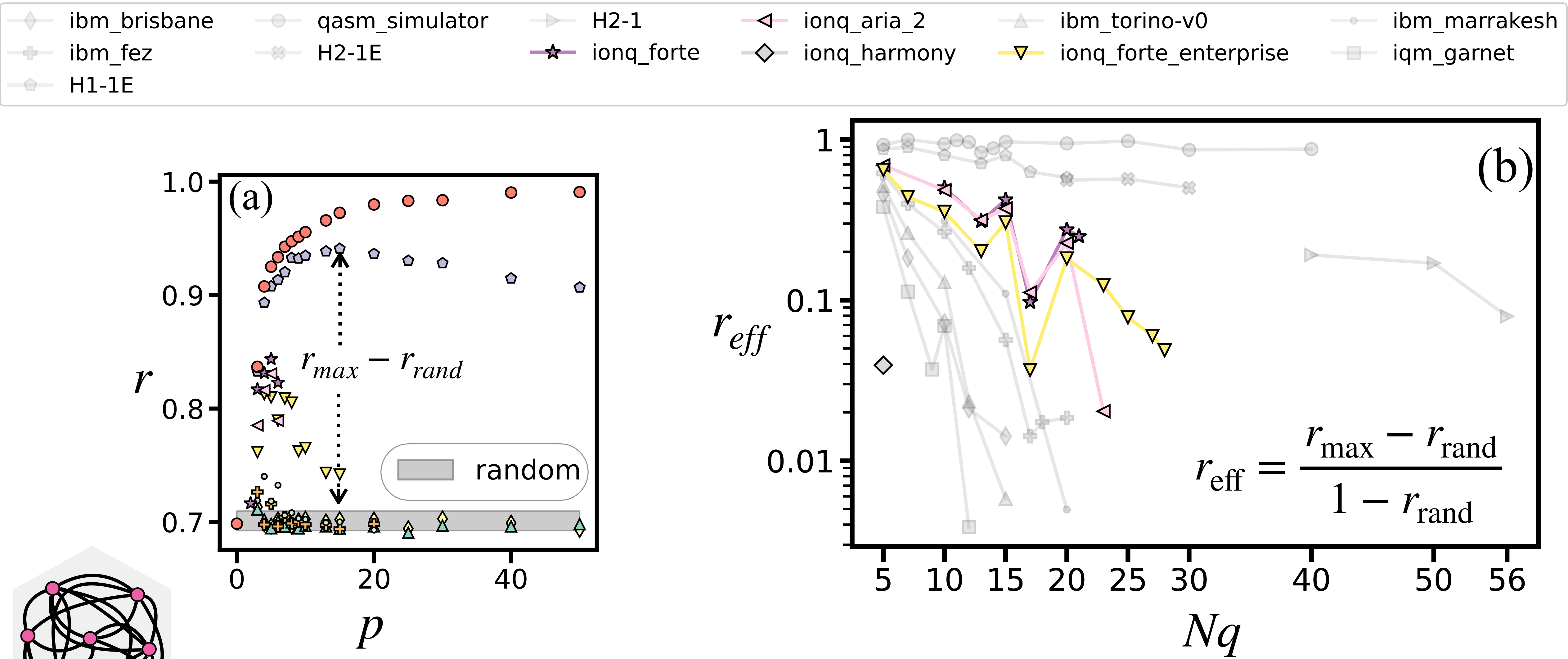




(a) FC for a 15-qubit Weighted MaxCut problem

(b) Effective approximation ratio

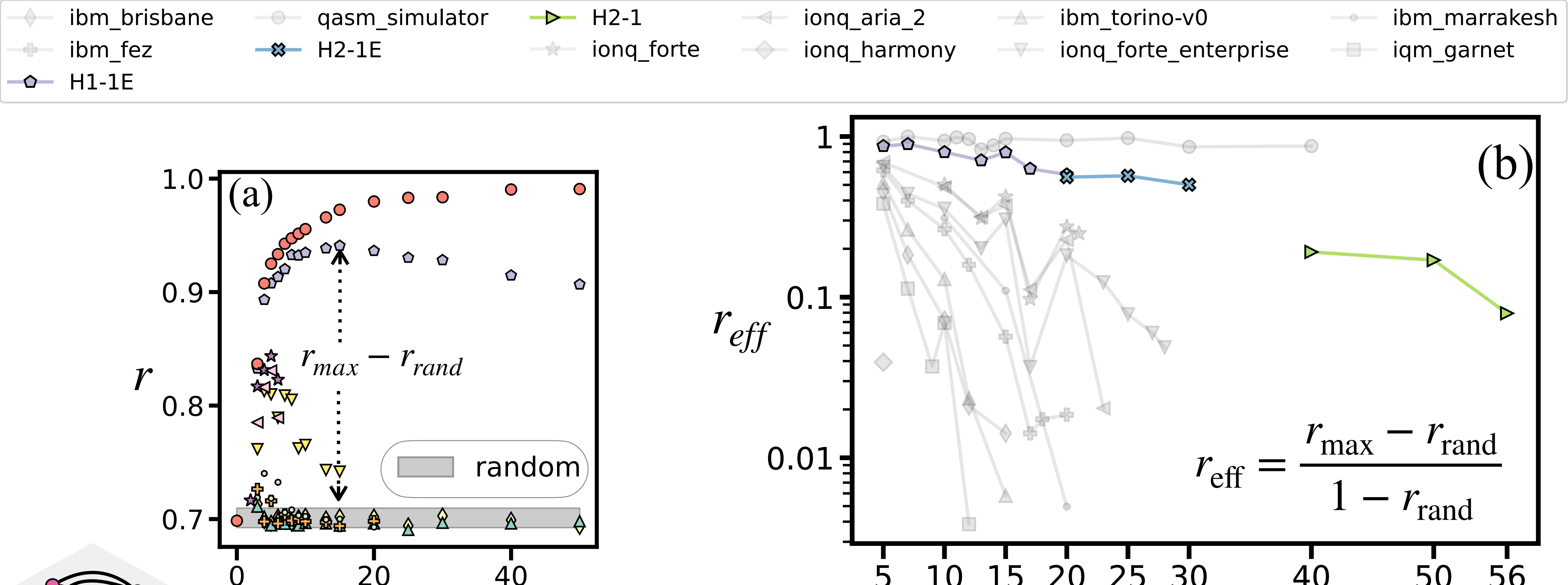
LR-QAOA on Fully Connected (FC) problems



(a) FC for a 15-qubit Weighted MaxCut problem

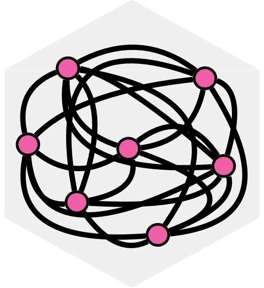
(b) Effective approximation ratio

LR-QAOA on Fully Connected (FC) problems



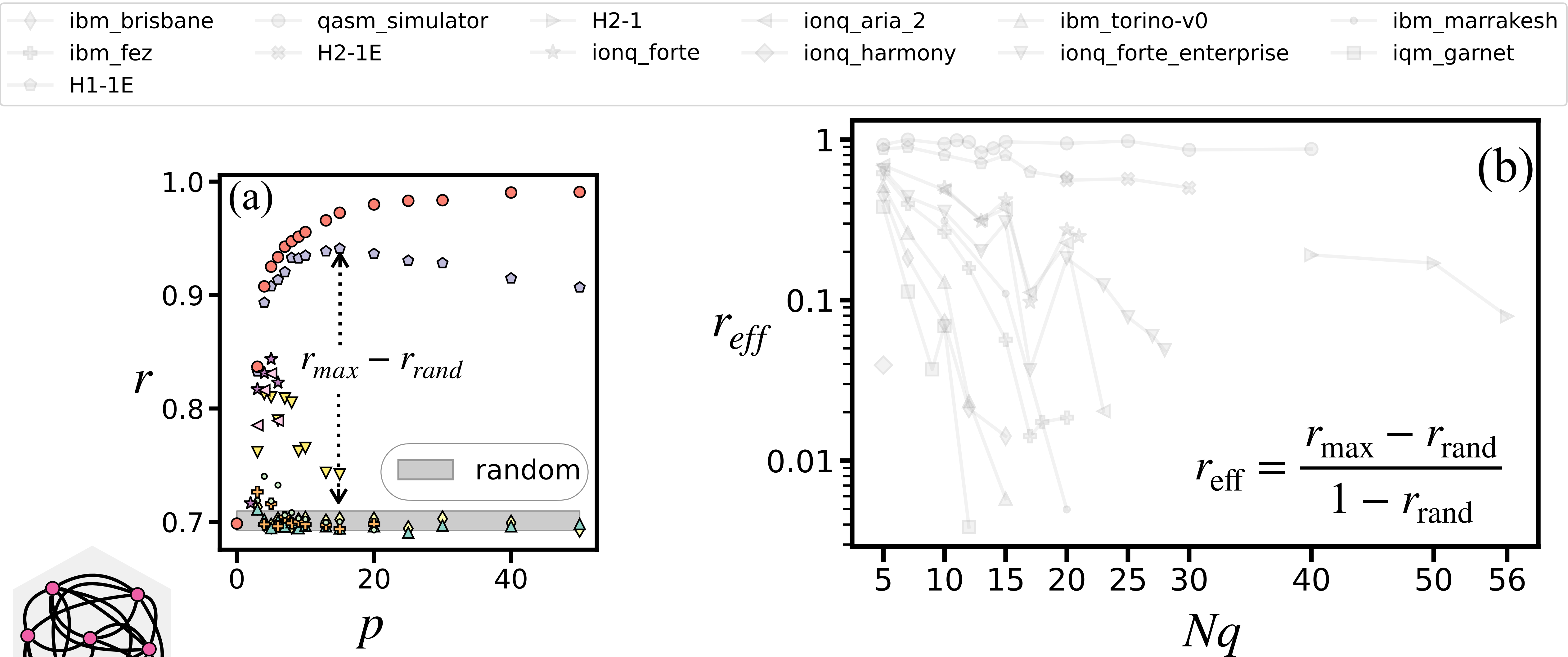
(a) FC for a 15-qubit Weighted MaxCut problem

(b) Effective approximation ratio



JÜLICH Forschungszentrum

LR-QAOA on Fully Connected (FC) problems



(a) FC for a 15-qubit Weighted MaxCut problem

(b) Effective approximation ratio

Integration LR-QAOA as an open source benchmark

Unitary Foundation

metriq

metriq-gym

Supported By Unitary Foundation | Discord 376 online. | Contributor Covenant 2.1

`metriq-gym` is a Python framework for implementing and running standard quantum benchmarks on different quantum devices by different providers.

- *Open* – Open-source since its inception and fully developed in public.
- *Transparent* – All benchmark parameters are defined in a schema file and the benchmark code is reviewable by the community.
- *Cross-platform* – Supports running benchmarks on multiple quantum hardware providers (*integration powered by qBraid-SDK*)
- *User-friendly* – Provides a simple command-line interface for dispatching, monitoring, and polling benchmark jobs (you can go on with your life while your job waits in the queue).

Conclusions

- We holistically benchmarked 24 QPUs from five vendors using LR-QAOA, evaluating their performance on different graph topologies and testing scalability in qubit count and circuit depth.
- IBM QPUs show significant improvements from Eagle to Heron generations, while IonQ and Quantinuum maintain performance through generations and offer better gate fidelity but suffer from slow execution times.
- Our results highlight key bottlenecks in quantum hardware, emphasizing the need for advancements in circuit depth, execution speed, and gate fidelity to support large-scale quantum algorithms.

Thank you