Error reduction



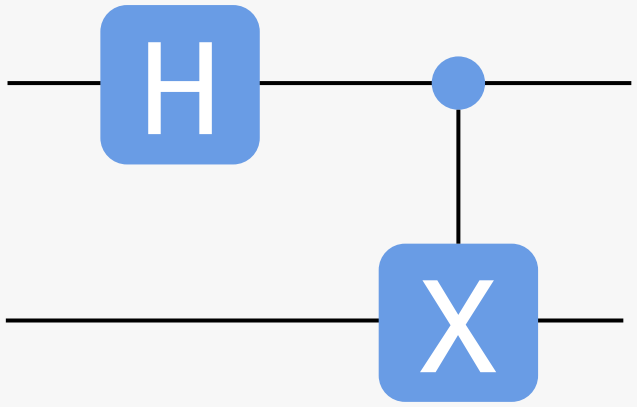
A lecture series by IQM

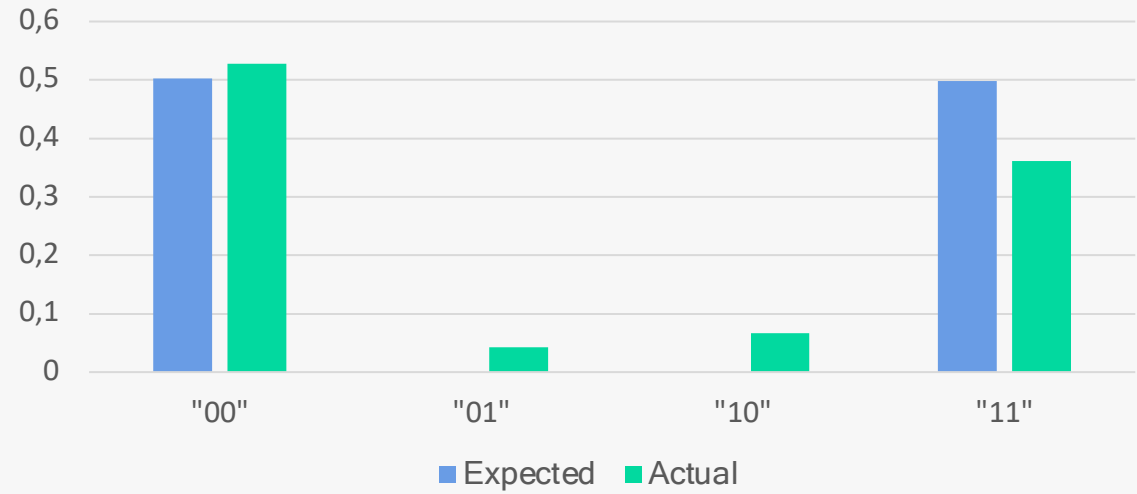
Authors: Alessio Calzona, Amin Hosseinkhani, Miha Papic, Pedro Figueroa-Romero, Manish Thapa, Raphael Amadeus Brieger & Stefan Seegerer

Last Updated 07/2025

www.meetiqm.com

Real quantum hardware vs. simulator





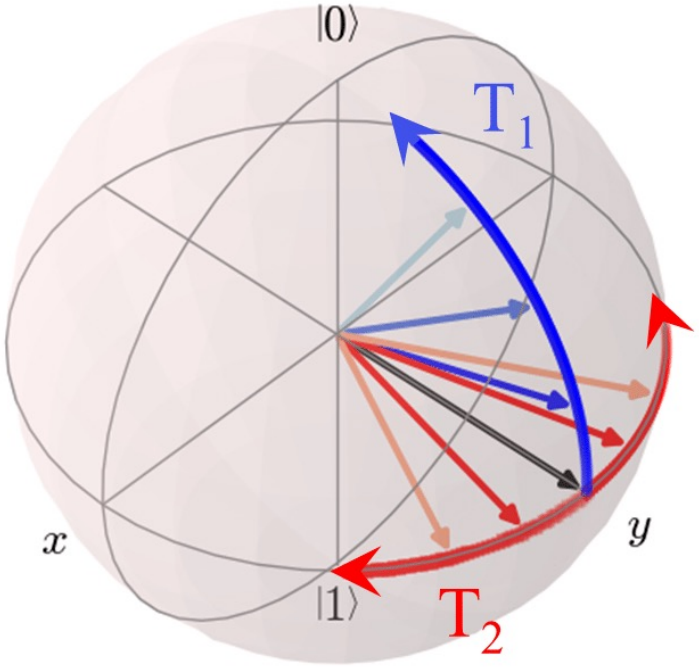
- Noise is present in modern day quantum computers.
- Until large scale error correction is reached, the algorithms of today must be able to remain functional in the presence of noise.
- This makes strategies of reducing errors an important building block for creating quantum algorithms on today's quantum computers.

Incoherent Noise

Incoherent noise arises from the **interaction** of the quantum system **with its environment**, leading to a loss of quantum information (decoherence) and entanglement. These errors are stochastic and non-unitary, causing a quantum state to transition into a mixed state.

- **Energy Relaxation (T1):** Loss of energy from the qubit to the environment, causing the qubit to decay from an excited state to its ground state.
- **Dephasing (T2):** Loss of phase coherence due to environmental fluctuations, leading to a loss of the qubit's ability to interfere with itself.

Irreversible Process: Information lost due to incoherent noise cannot be fully recovered.



Clement Gódfrin, Quantum information processing using a molecular magnet single nuclear spin qudit, retrieved from ResearchGate

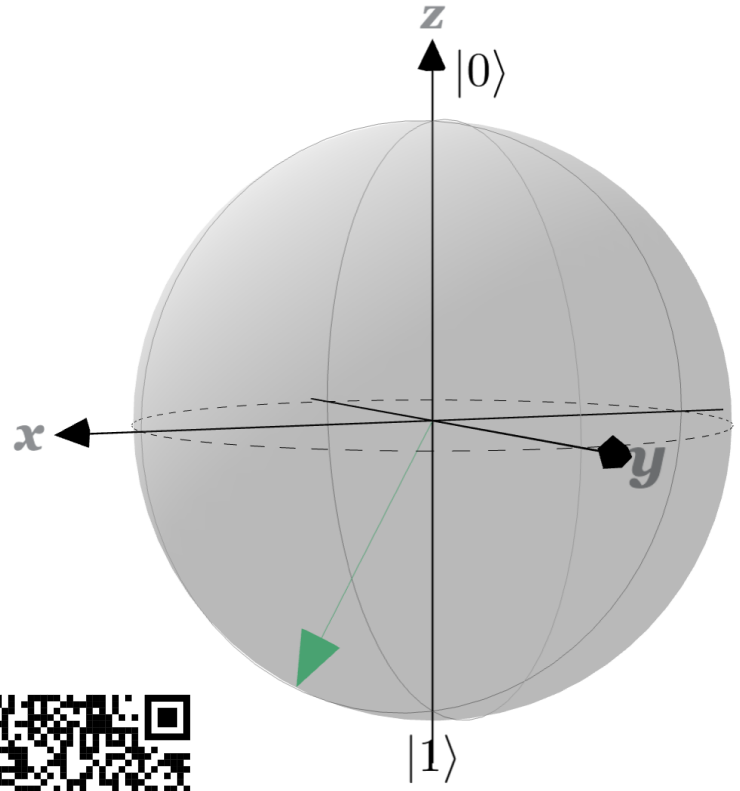
Coherent Noise

Coherent noise refers to errors that are **deterministic** and **unitary** in nature, often arising from miscalibrations or imprecise control operations. Unlike incoherent noise, coherent errors preserve the purity of the quantum state and can, in principle, be reversed or compensated for.

- **Systematic Deviations:** Constant offsets in control pulse amplitudes, phases, or frequencies.
- **Under/Over-Rotations:** Gates applying slightly more or less than the intended rotation angle.

Controllable & Correctable: Unlike random incoherent noise, coherent errors arise from *tunable* aspects of our system. This opens the door for:

- **Precision Calibration:** Dialing in control parameters for optimal performance.
- **Tailored Pulse Design:** Engineering specific control sequences to cancel out systematic deviations.



Try it yourself and apply the same operation (e.g. the X gate) multiple times.

https://www.iqmacademy.com/curriculum/assets/bloch-viz/index.html?gates=H,X,T,Y&noKet&noise

Some Potential Sources of Error in Superconducting Circuits

Energy Relaxation (T1)

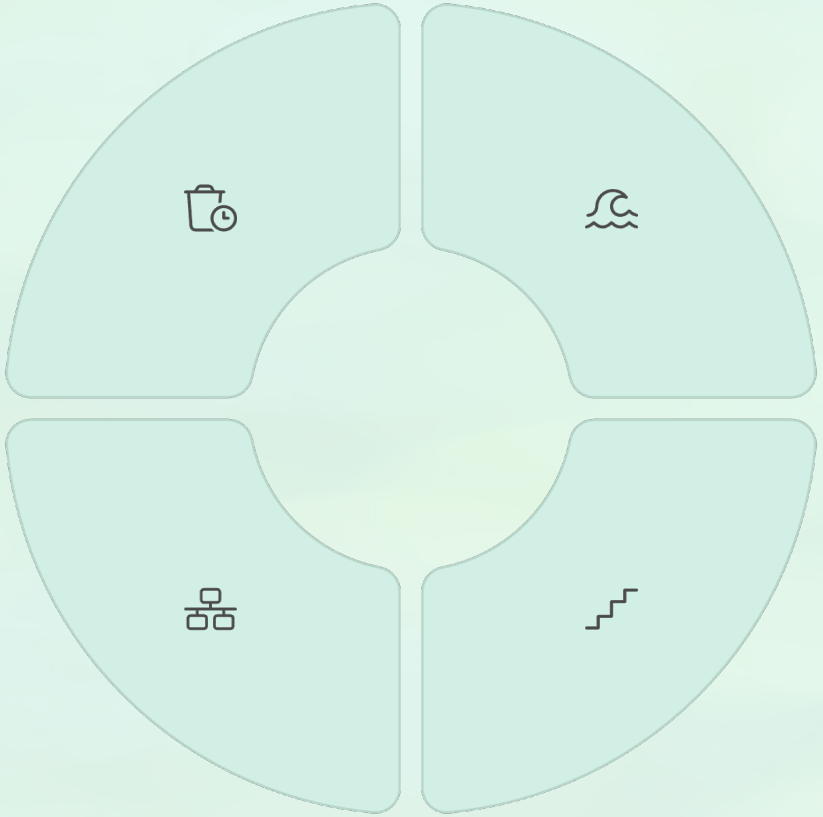
Photon loss to environment through:

- Dielectric loss in materials
- Quasi-particle excitations
- Radiative loss to control lines

Crosstalk & Spectator Errors

Unwanted interactions from:

- Dense wiring and control line coupling
- Frequency collisions between qubits
- Shared readout resonators



Dephasing (T2)

Phase coherence loss due to:

- Flux noise from trapped vortices
- Charge fluctuations
- Environmental temperature fluctuations

Leakage Errors

Population escaping computational subspace:

- Excitation to $|2\rangle$ states in transmons
- Creates non-correctable, correlated errors
- Persists across multiple gate operations

How to fight noise?

Perform classical computations for

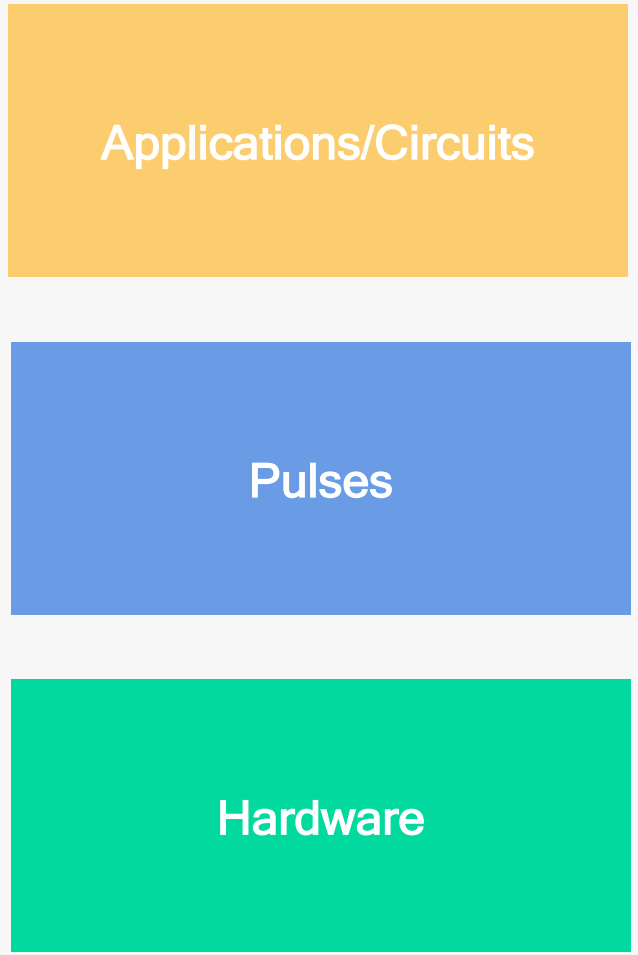
- noise-aware compilation (e.g. qubit selection)
- Correcting errors by post-processing classically

Improving pulse signals

- Optimal control
- Compensating pulses

Improving hardware

- chip design, shielding, …



Modeling noise

Quantum Evolution

What we've covered so far:

1. Description of a quantum state: $|\psi(t)\rangle$
2. The state evolves according to the Schrödinger equation:

$$i\hbar \frac{\partial|\psi(t)\rangle}{\partial t} = H|\psi(t)\rangle.$$

The state after the evolution is therefore:

$$|\psi(t)\rangle = e^{-iHt/\hbar}|\psi(0)\rangle,$$

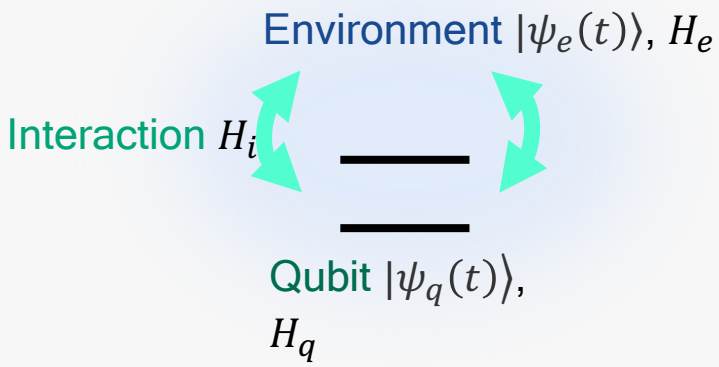
i.e. the matrix $U = e^{-iHt/\hbar}$ describes how an initial state $|\psi(0)\rangle$ evolves.

Works very well for an ISOLATED system!

But what is the alternative?

Imagine a qubit interacting with some unknown environment, e.g.
The combined system (qubit + environment) still evolves according to the Schrödinger equation:

$$|\xi(t)\rangle = e^{-i(H_q \otimes I + I \otimes H_e + H_i)t/\hbar} |\psi_q(0)\rangle \otimes |\psi_e(0)\rangle$$



Quantum Evolution

In the real world the "environment" can be anything with the following properties:
- We do not have control over the environment (we cannot perform measurements etc.).
- We do not really care about the state of the environment; we are only interested in the qubit.

The dynamics of a qubit interacting with some environment = **Open** quantum system evolution!

Property	Isolated system	Open system
Quantum State	$\|\psi(t)\rangle = \sum_i c_i \|\varphi_i\rangle$	$\rho = \sum_i p_i \|\psi_i\rangle\langle\psi_i\|, \rho^\dagger = \rho$ Statistical mixture of "pure" states!
Conservation of probability	$\sum_i \|c_i\|^2 = 1$	$\mathrm{tr}\{\rho\} = 1$ $p_i \geq 0$, i.e. ρ is positive semi-definite
Unitary time evolution, $U^\dagger U = I$	$\|\psi(t)\rangle = U\|\psi(0)\rangle$	$\rho(t) = U\rho(0)U^\dagger$
Expectation value	$\langle O\rangle = \langle\psi\|O\|\psi\rangle$	$\langle O\rangle = \mathrm{tr}\{O\rho\}$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

Density matrix ρ

can describe open quantum systems

$$\begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix}$$
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

Density matrix ρ

can describe open quantum systems

$$\begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix}$$
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

Density matrix ρ

can describe open quantum systems

$$\begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix}$$
$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

amplitudes

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

Density matrix ρ

can describe open quantum systems

$$\begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

amplitudes

$$\begin{pmatrix} a_{00} \\ 0 \\ 0 \\ a_{11} \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

Superposition of $|00\rangle$ and $|11\rangle$

Density matrix ρ

can describe open quantum systems

$$\begin{matrix} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{matrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}$$

Density matrix ρ

can describe open quantum systems

$$\begin{pmatrix} |a_{00}|^2 & a_{00}a_{01}^* & a_{00}a_{10}^* & a_{00}a_{11}^* \\ a_{01}a_{00}^* & |a_{01}|^2 & a_{01}a_{10}^* & a_{01}a_{11}^* \\ a_{10}a_{00}^* & a_{10}a_{00}^* & |a_{10}|^2 & a_{10}a_{11}^* \\ a_{11}a_{00}^* & a_{11}a_{00}^* & a_{11}a_{10}^* & |a_{11}|^2 \end{pmatrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

Probabilities

Density matrix ρ

can describe open quantum systems

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}$$

$$\begin{pmatrix} |a_{00}|^2 & a_{00}a_{01}^* & a_{00}a_{10}^* & a_{00}a_{11}^* \\ a_{01}a_{00}^* & |a_{01}|^2 & a_{01}a_{10}^* & a_{01}a_{11}^* \\ a_{10}a_{00}^* & a_{10}a_{00}^* & |a_{10}|^2 & a_{10}a_{11}^* \\ a_{11}a_{00}^* & a_{11}a_{00}^* & a_{11}a_{10}^* & |a_{11}|^2 \end{pmatrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}$$

Density matrix ρ

can describe open quantum systems

Probabilities

‘Coherences’

$$\begin{pmatrix} |a_{00}|^2 & a_{00}a_{01}^* & a_{00}a_{10}^* & a_{00}a_{11}^* \\ a_{01}a_{00}^* & |a_{01}|^2 & a_{01}a_{10}^* & a_{01}a_{11}^* \\ a_{10}a_{00}^* & a_{10}a_{00}^* & |a_{10}|^2 & a_{10}a_{11}^* \\ a_{11}a_{00}^* & a_{11}a_{00}^* & a_{11}a_{10}^* & |a_{11}|^2 \end{pmatrix}$$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}$$

Density matrix ρ

can describe open quantum systems

Probabilities

‘Coherences’

$$\begin{pmatrix} |a_{00}|^2 & 0 & 0 & a_{00}a_{11}^* \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_{11}a_{00}^* & 0 & 0 & |a_{11}|^2 \end{pmatrix}$$

Superposition of $|00\rangle$ and $|11\rangle$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}$$

Probabilities

Density matrix ρ

can describe open quantum systems

$$\begin{pmatrix} |a_{00}|^2 & 0 & 0 & \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ & 0 & 0 & |a_{11}|^2 \end{pmatrix}$$

Classical mixed state of $|00\rangle$ and $|11\rangle$

State vector $|\psi\rangle$

describes closed quantum systems

$$\begin{pmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{pmatrix}$$

Density matrix ρ

can describe open quantum systems

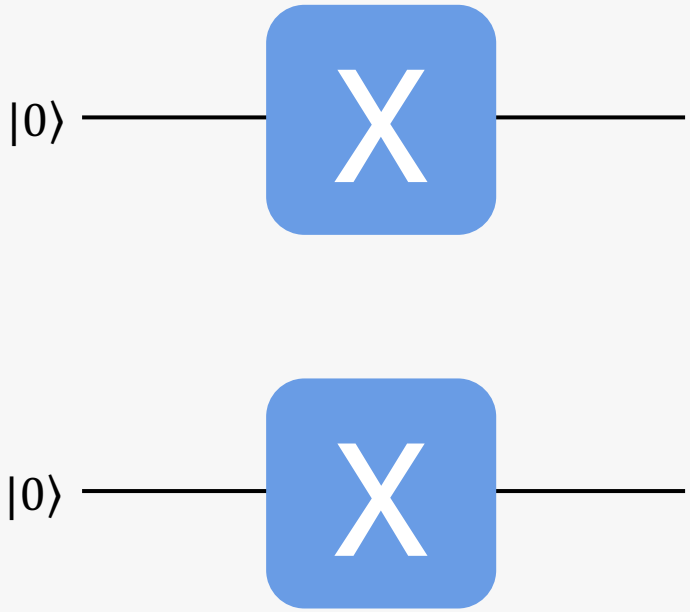
Probabilities

‘Coherences’

$$\begin{pmatrix} |a_{00}|^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & |a_{11}|^2 \end{pmatrix}$$

$\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|$ where p_j gives the probability for the system being in state $|\psi_j\rangle$

Describing noiseless quantum computers



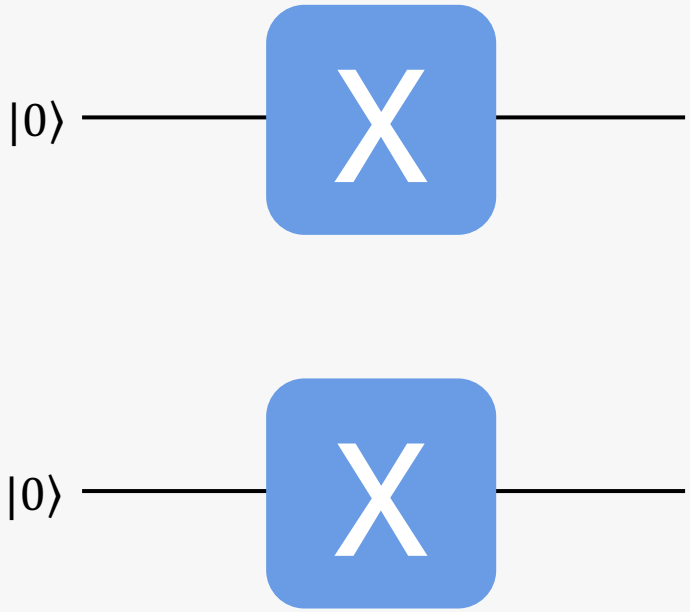
Tensor product of unitary matrices of X gates

$$\left[\begin{pmatrix}0&1\\1&0\end{pmatrix}\otimes\begin{pmatrix}0&1\\1&0\end{pmatrix}\right]\begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$|00\rangle$

$$\begin{pmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

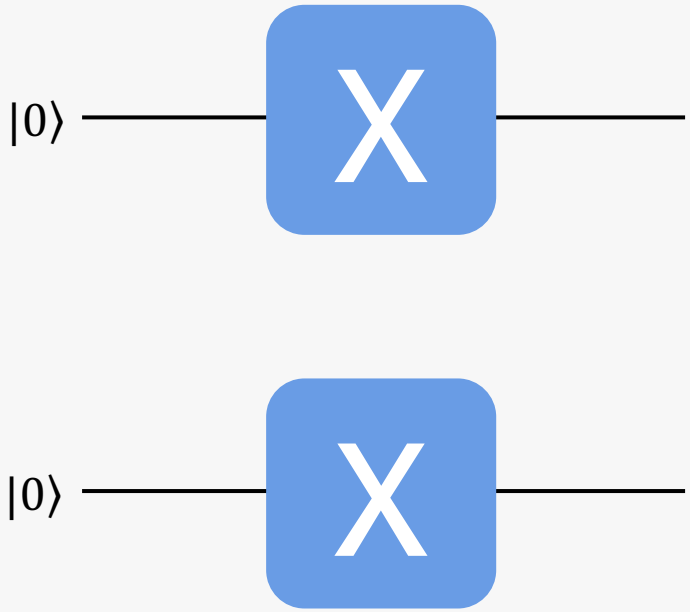
Describing noiseless quantum computers



$|11\rangle$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

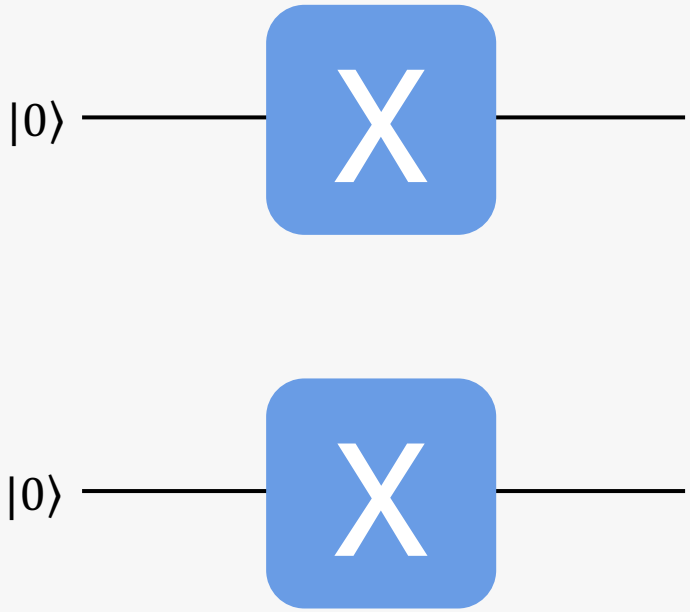
Describing noiseless quantum computers



Density matrix corresponding to $|00\rangle$

$$|00\rangle\langle 00|$$

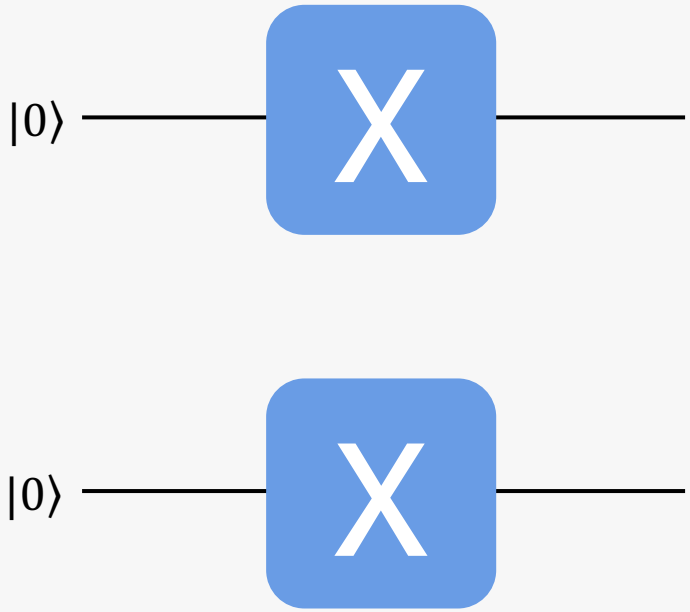
Describing noiseless quantum computers



Density matrix corresponding to $|00\rangle$

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} (1\ 0\ 0\ 0)$$

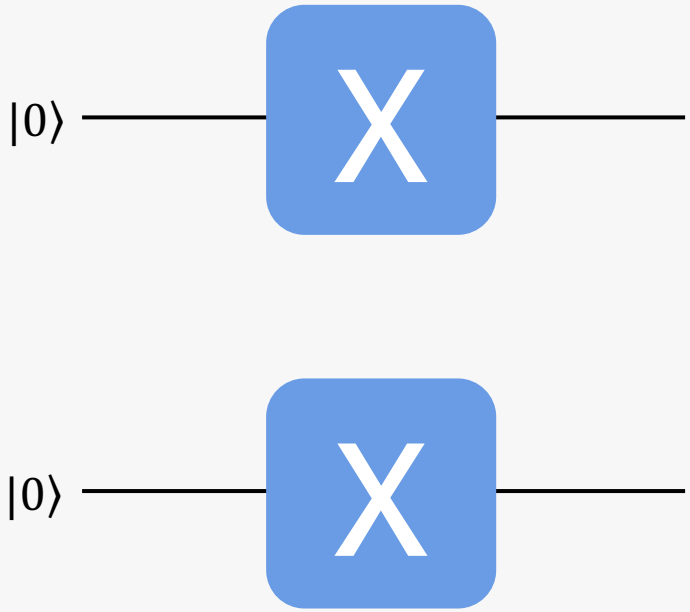
Describing noiseless quantum computers



Density matrix corresponding to $|00\rangle$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Describing noiseless quantum computers



Tensor product of unitary matrices of X gates

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

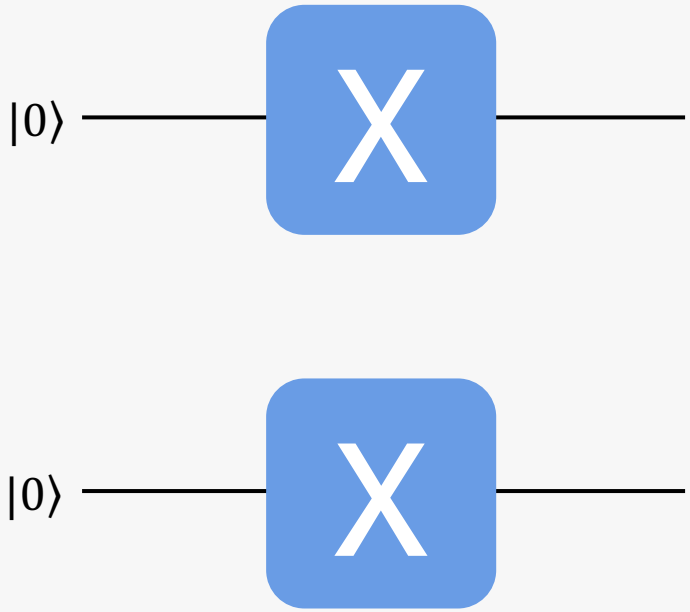
Density matrix corresponding to $|00\rangle$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Tensor product of unitary matrices of X gates

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Describing noiseless quantum computers

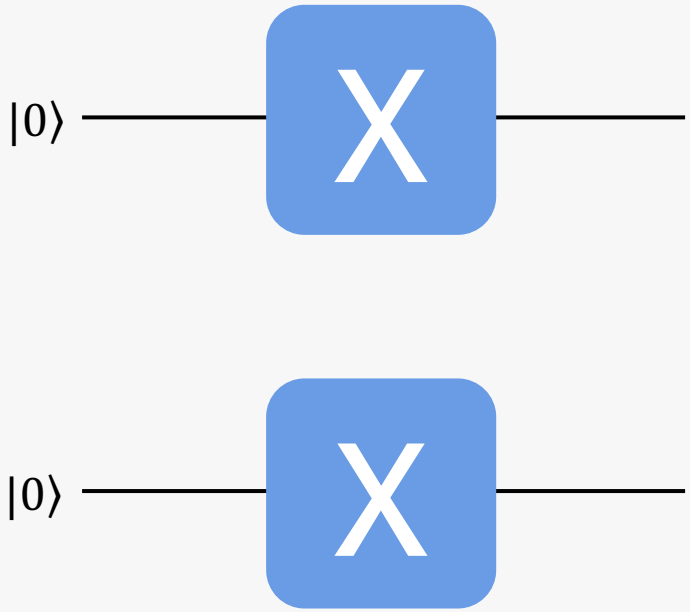


Density matrix corresponding to $|00\rangle$

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$(X \otimes X)|00\rangle\langle 00|(X^\dagger \otimes X^\dagger)$$

Describing noiseless quantum computers



Density matrix corresponding to $|11\rangle$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Quantum Evolution

Dynamical map $\mathcal{E}$: $\rho(t) = \mathcal{E}(t,0)[\rho(0)]$

What are the *properties* of $\mathcal{E}$?

1. A valid description of the time evolution must map a valid quantum state to a valid quantum state!
2. If $\mathrm{tr}\{\rho(0)\}$ then for any *valid* $\mathcal{E}$, $\mathrm{tr}\{\mathcal{E}(t,0)[\rho(0)]\} = 1 \Rightarrow \mathcal{E}$ is TRACE-PRESERVING (TP)
3. If $\rho(0)$ is positive semi-definite then $\rho(t)$ is positive semi-definite $\Rightarrow \mathcal{E}$ is POSITIVE (P)

Turns out that POSITIVITY is not strict enough! …but why?

Assume we have:

- an initial state $\rho(0)$,
- a dynamical map $\mathcal{E}$, acting on a part of the whole Hilbert space (e.g. one qubit in a quantum computer)

For the total evolution $\mathcal{E}\otimes I$ to be a valid physical map, $\mathcal{E}\otimes I$ must therefore be POSITIVE!

We will refer to this property as COMPLETE POSITIVITY!

A valid physical map $\mathcal{E}$ is therefore COMPLETELY POSITIVE and TRACE-PRESERVING (CPTP)

Kraus representation of dynamical maps

Any valid open quantum system evolution is described by a CPTP map.

But what does such a map look like?

Choi-Kraus theorem: A linear map $\mathcal{E}$ is completely positive if and only if it admits a Kraus representation.

$$\rho(t) = \mathcal{E}(t,0)[\rho(0)] = \sum_i E_i \rho(0) E_i^\dagger, \text{with} \sum_i E_i^\dagger E_i = I$$

Trace-preservation

Example 1: Amplitude damping channel

- Each CPTP quantum evolution is described by a set of Kraus operators
- The amplitude damping channel models the decay of an excitation into a thermalized environment

Kraus operators: $E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}$ $E_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}$

Transfers the population from $|1\rangle$ to $|0\rangle$

Does not affect the $|0\rangle$-state
Decreases the population of the $|1\rangle$-state

This channel is TP:

$$E_0^\dagger E_0 + E_1^\dagger E_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1-p \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & p \end{pmatrix} = I$$

- **The amplitude damping channel induces decays from the $|1\rangle$ to the $|0\rangle$-state:**

$$\begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} \rightarrow E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger = \begin{pmatrix} \rho_{00} + p\rho_{11} & \sqrt{1-p}\rho_{01} \\ \sqrt{1-p}\rho_{10} & (1-p)\rho_{11} \end{pmatrix}$$

Coherence times

- People often report coherence times when discussing QPU performance
- Thus far we have characterized our channels based on the probability to apply an error (p)
- Consider amplitude damping:
    1. p is the probability for a decay of state $|1\rangle$ to state $|0\rangle$
    2. these decays are independent and uncorrelated ⇒ Poisson process
    3. the probability of no errors occurring in a time interval Δt is determined by the decay time T
    4. For any Poisson process, we can equate ⇒ $1 - p = e^{-\Delta t/T}$

T_1 experiment

1. **Prepare initial state** $\rho(0) = |1\rangle\langle 1|$
2. **Let state evolve for time** Δt

$$\rho(0) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \rightarrow \rho(\Delta t) = \begin{pmatrix} 1 - e^{-\Delta t/T} & 0 \\ 0 & e^{-\Delta t/T} \end{pmatrix}$$

3. **Plot probability of measuring** $|1\rangle$ **at different values of** Δt
4. **Fit exponential decay to extract coherence time -** T_1

Assume noise is described by amplitude damping!

T_1 measures the timescale of the decay of the **diagonal** elements!

Example 2: Pure dephasing channel

- **Amplitude damping "kills" the off-diagonal elements of the DM**
- **In experiment, we often observe processes which affect the off-diagonal elements, but not the diagonals**

Kraus operators: $E_0 = \sqrt{1-p}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$ $E_1 = \begin{pmatrix}0 & 0\\ 0 & \sqrt{p}\end{pmatrix}$ $E_2 = \begin{pmatrix}\sqrt{p} & 0\\ 0 & 0\end{pmatrix}$

Does nothing

Destroy coherences

- **The pure dephasing channel decays the value of the coherences = off-diagonal elements:**

$$\begin{pmatrix}\rho_{00} & \rho_{01}\\ \rho_{10} & \rho_{11}\end{pmatrix} \rightarrow \begin{pmatrix}\rho_{00} & (1-p)\rho_{01}\\ (1-p)\rho_{10} & \rho_{11}\end{pmatrix} = \begin{pmatrix}\rho_{00} & e^{-\Delta t/T}\rho_{01}\\ e^{-\Delta t/T}\rho_{10} & \rho_{11}\end{pmatrix}$$

- **The "strength" of this channel is experimentally characterized by the pure dephasing time T_φ**

T_2 experiment

1. **Prepare initial state** $\rho(0) = |+\rangle\langle+|$
2. **Let state evolve for time** Δt
3. **Plot probability of measuring** $|+\rangle$ **at different values of** Δt
4. **Fit exponential decay to extract coherence time -** T_2

If only pure dephasing is present:
$T_2 = T_\varphi$

$$\rho(0) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \rightarrow \rho(\Delta t) = \frac{1}{2}\begin{pmatrix} 1 & e^{-\Delta t/T_\varphi} \\ e^{-\Delta t/T_\varphi} & 1 \end{pmatrix}$$

But we cannot "turn off" amplitude damping (with a decay time of T_1 **)!**

$$\rho(\Delta t) = \frac{1}{2}\begin{pmatrix} 1 - e^{-\Delta t/T} & e^{-\Delta t/T_\varphi}\, e^{-\Delta t/2T_1} \\ e^{-\Delta t/T_\varphi}\, e^{-\Delta t/2T_1} & 1 \end{pmatrix}$$

Pure dephasing + Amplitude damping

$$\frac{1}{T_2} = \frac{1}{2T_1} + \frac{1}{T_\varphi}$$

T_2 measures the timescale of the decay of the **off-diagonal** elements!

Example 3: Measurement

- **The Kraus formalism enables us to model the effects of a projective measurement!**
- **The ”environment” in this case is the measurement apparatus interacting with the qubit.**

Kraus operators: $E_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

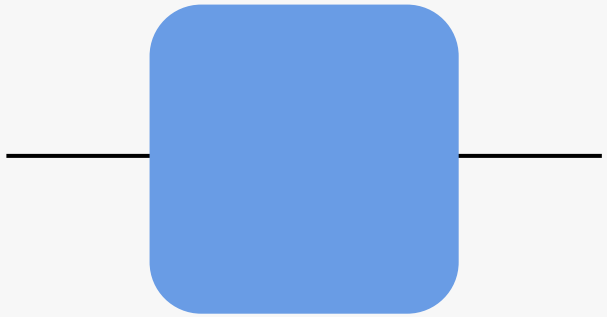
Projects onto $|0\rangle\langle 0|$

$E_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

Projects onto $|1\rangle\langle 1|$

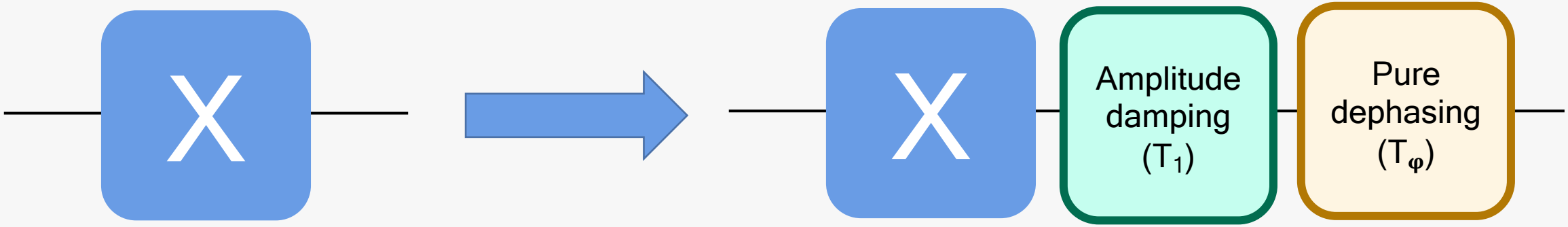
- **The system after measurement is in a statistical mixture of the two measurement eigenstates:**

$$\begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} \rightarrow E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger = \begin{pmatrix} \rho_{00} & 0 \\ 0 & \rho_{11} \end{pmatrix}$$



Modelling circuit operations with errors

Modelling Noisy Gates



Noiseless evolution: $\rho \rightarrow X\rho X$

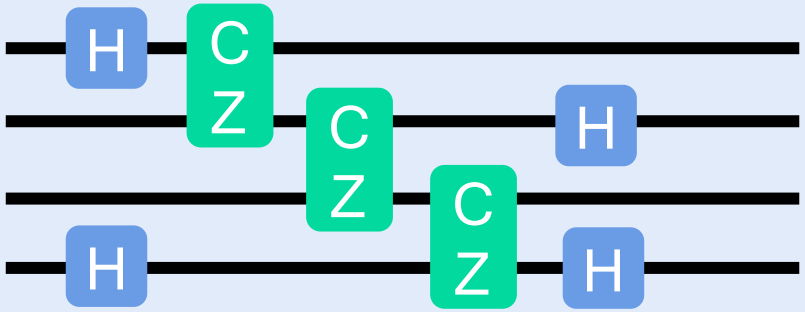
Noisy evolution:

1. **Apply gate.**
2. **Apply noise channels.**

$$\rho \rightarrow X\rho X \rightarrow \sum_i E_i X\rho X E_i^\dagger \rightarrow \sum_j F_j \left(\sum_i E_i X\rho X E_i^\dagger \right) F_j^\dagger$$

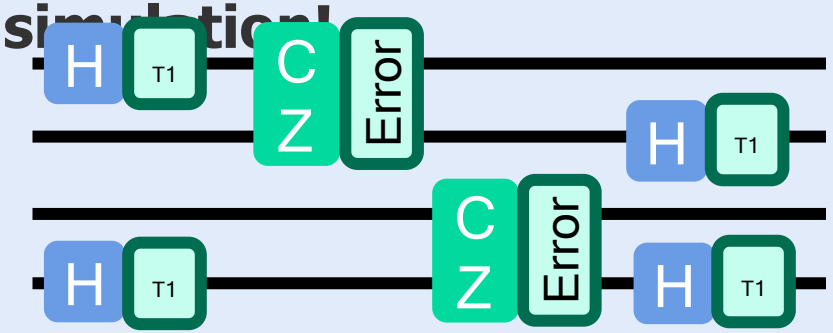
Modelling Noisy Circuits

Noiseless evolution



Noisy evolution

- Append noise channel after each gate
- Can be more demanding computationally than noiseless simulation!



We have implicitly assumed no time-correlations in this model!

Error supression

What is error suppression?

Error suppression encompass a very wide set of techniques aiming at reducing the amount errors happening during the execution of a desired algorithm.

Error suppression is therefore deeply connected to the hardware properties of the QPU in use and it's intertwined with topic such as QPU calibration, error characterization and even noise-aware circuit transpilation.

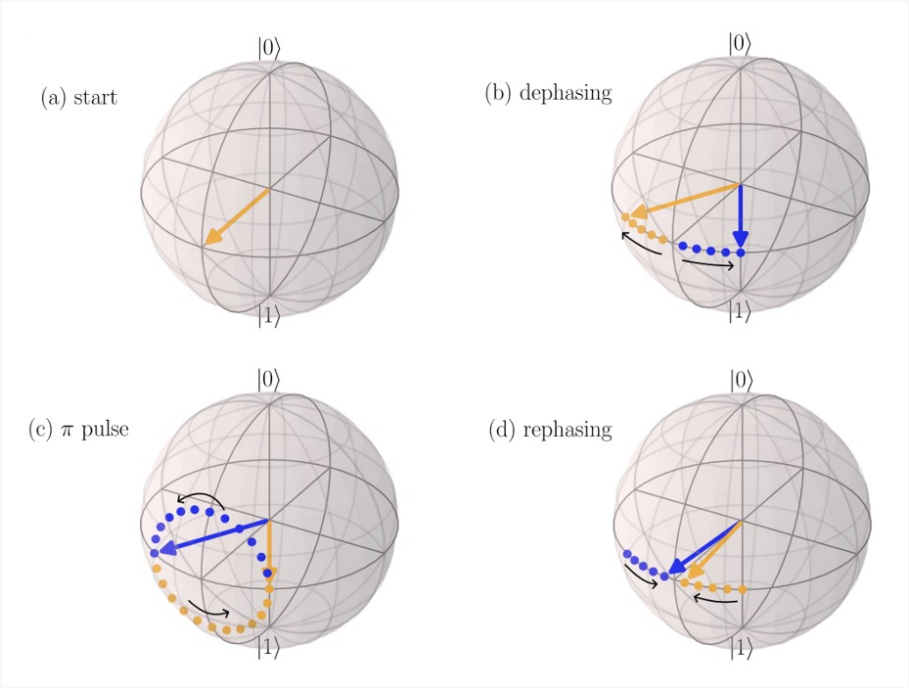
Importantly, error suppression does not rely on heavy post-processing and does not come with (significant) runtime overhead. Often, suppression techniques are applied unbeknownst to the general user.

In this set of slides, we'll focus on two error suppression techniques:

- Dynamical decoupling
- Pauli Twirling

Dynamical Decoupling (DD)

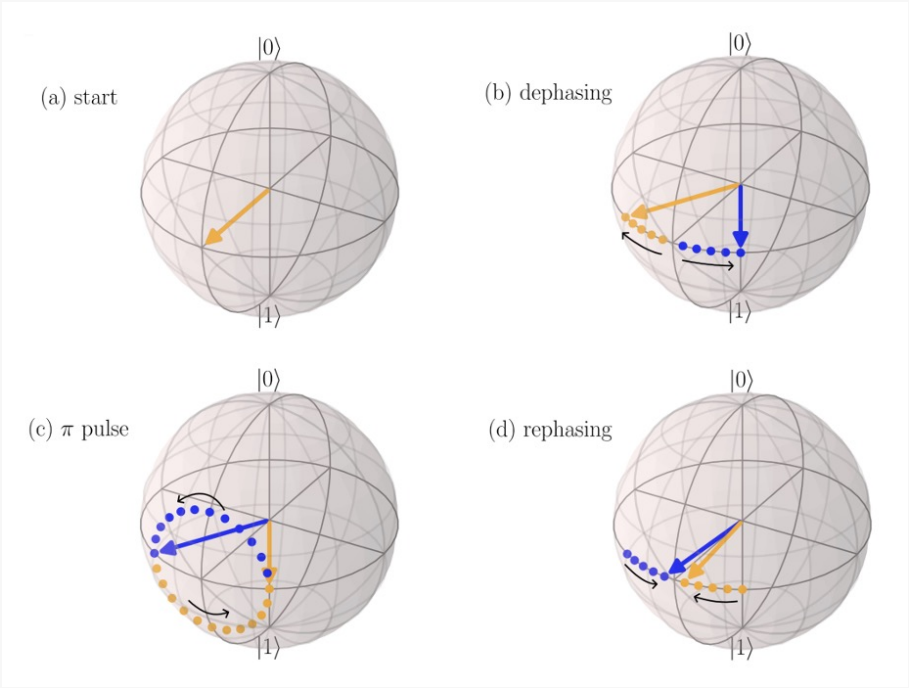
- DD is a quantum control technique.
- Meaning, the goal is to drive the dynamics in a particular and targeted way.
- It is typically known to kill coherent (covered in the next slides) and non-Markovian errors in qubits.



AWS Blog:
https://aws.amazon.com/blogs/quantum-computing/suppressing-errors-with-dynamical-decoupling-using-pulse-control-on-amazon-braket/

Dynamical Decoupling (DD)

- DD is a quantum control technique.
- Meaning, the goal is to drive the dynamics in a particular and targeted way.
- It is typically known to kill coherent (covered in the next slides) and non-Markovian errors in qubits.

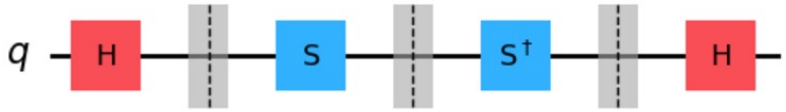


AWS Blog:
https://aws.amazon.com/blogs/quantum-computing/suppressing-errors-with-dynamical-decoupling-using-pulse-control-on-amazon-braket/

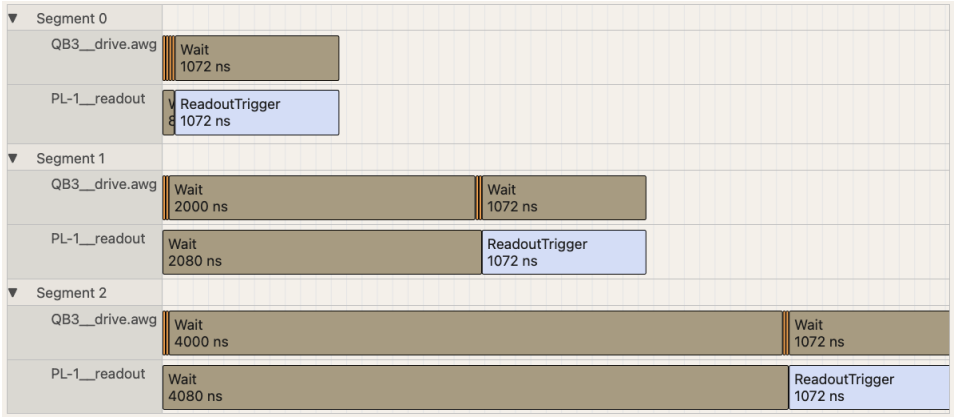
1. **XX Sequence Properties**:
   - Perfect suppression of Z/ZZ errors:
     - XZX = -Z ⇒ ⟨Z⟩_DD ≈ Z + XZX = 0 over one cycle
     - Similarly, XZZX = -ZZ ⇒ ⟨ZZ⟩_DD = 0
   - No X-error suppression:
     - XXX = X ⇒ ⟨X⟩_DD = X remains unchanged
2. **XYXY Universal Sequence**:
   - Higher-order cancellation via nested anti-commutation:
     - XXYXXXY = -X (using {X,Y}=0)
     - Similarly cancels Y and Z errors to 1st order

IQM

Single qubit case



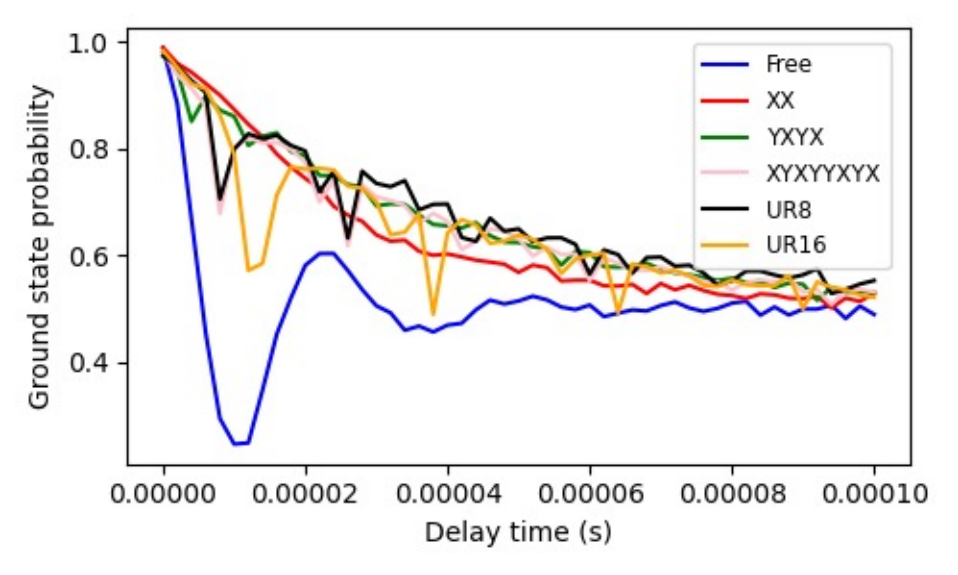
Let us apply some gates, and do the inversion.
Ideally this operation should do nothing! But due to errors, this is not the case.



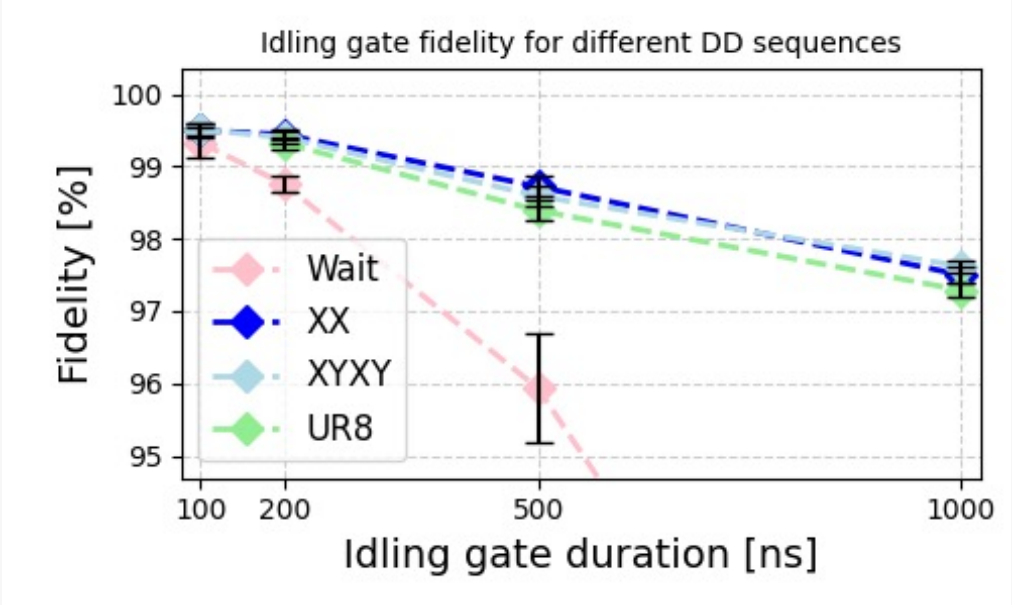
- Let's artificially add delay in the middle using IQM Pulla.

- Check capacity of each sequence to preserve $|-i\rangle$ state.

DD with IQM Quantum Computers



Delay experiment performed on one of the qubits



IRB of single qubit idling gate of different durations with and without DD sequences

- These are measurements performed at one of our internal stations.
- Depending on the calibration set and error profile on a given day, DD offers error suppression to varying degrees.
- DD is seen to significantly boost the performance, and especially so when the idles are long.

Pauli twirling (randomized compiling)

GOAL

To tailor coherent errors into a stochastic noise channel

HOW

By combining the results of many logically equivalent circuits (where random gates are locally added)

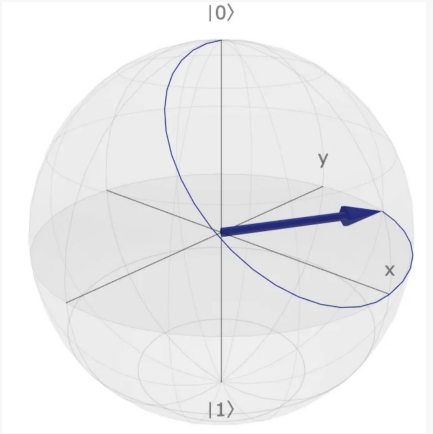
In the next slides, we cover:

- What are coherent errors?
- How to randomize a given circuit in practice?
- What do we gain from the randomization?

Reminder: coherent errors

- A coherent error is a unitary error with noise map

$$\Lambda(\rho) = U\rho U^{\dagger}$$



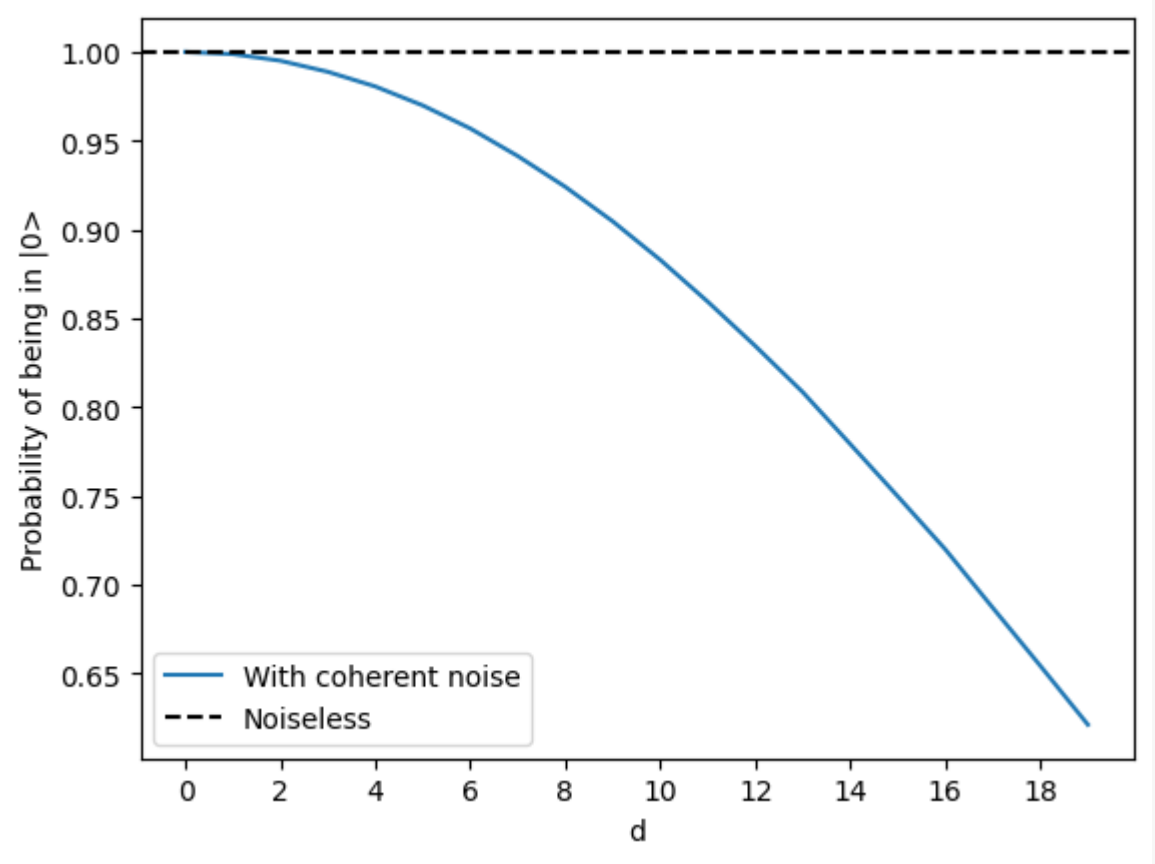
- Can be small extra over/underrotations, spurious couplings, ...
- Can not be completely removed (e.g., via better calibration)
- Effects of coherent errors accumulate quadratically fasters than stochastic errors!
- At the same time: average gate fidelity is insensitive to coherent errors:

$$\mathrm{U} = e^{i\epsilon\sigma_x}\colon\ F(\Lambda_U) \approx 1 - \epsilon^2$$

Simple example: single-qubit X overrotation

Let's consider a single-qubit noise map associated with a small X rotation: $U = \exp(-i\, X\, \theta/2)$ with, say, $\theta = 4°$.

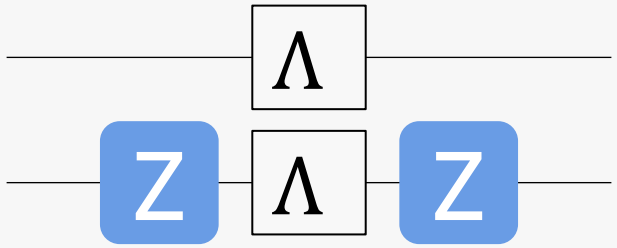
Let's apply $\Lambda(\cdot) = U \cdot U^{\dagger}$ on a qubit initialized in the $|0\rangle$ state d times and plot the probability of measuring the qubit in the $|0\rangle$ state.



Simple example: single-qubit X overrotation

What if we randomly add pairs of (noiseless Z gates) before and after each Λ? (Twirled noise)

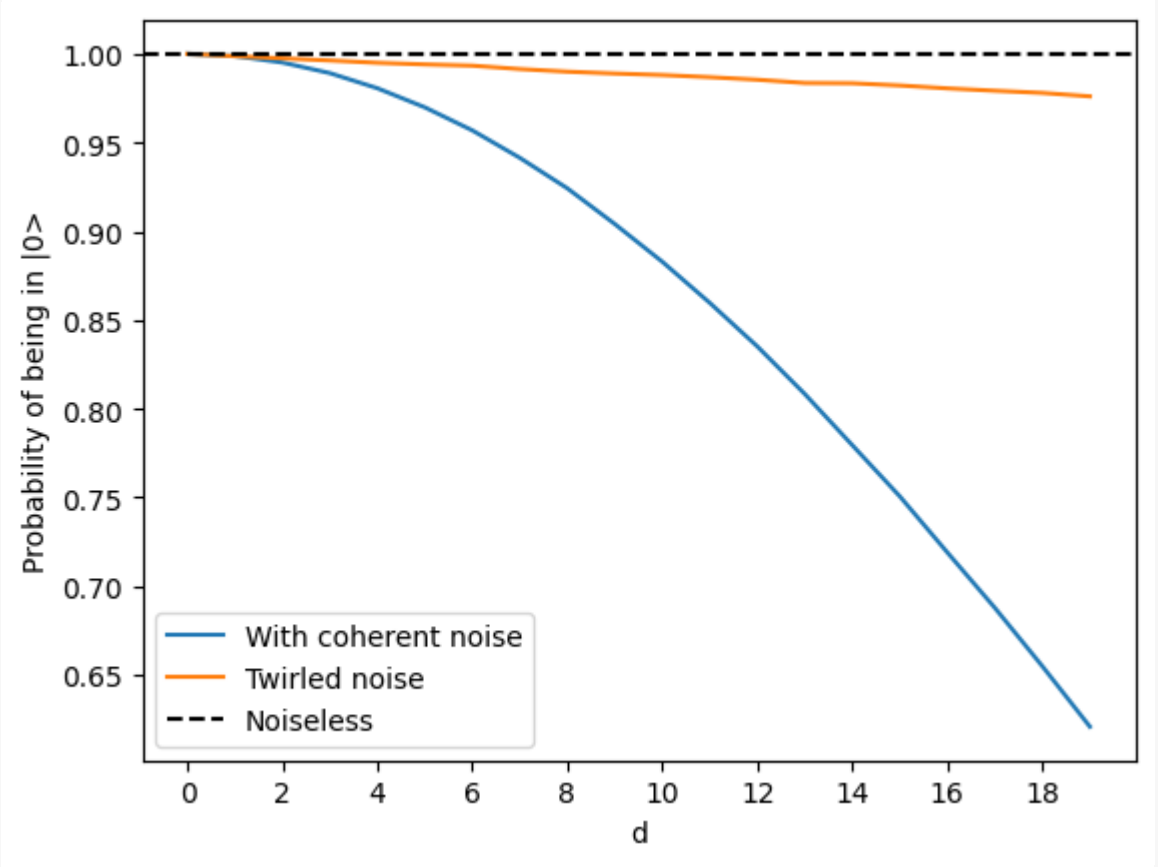
- 50% of the times:
- 50% of the times:



Adding pairs of Z gates does not change the noiseless circuit, as $ZZ = I$

However, when we sandwich the noisy map with Z, we flip the sign of the coherent X rotation.

By averaging over many (logically equivalent) randomized circuits, we get a much better result!



IQM

How to do that in practice?

The example before is too simple for several reasons… but we can take care of those!

- The noise map Λ always comes with a gate attached: ΛG

  W can use $G = PGG^{\dagger}P^{\dagger}G$ and use the pair of twirling gates P and $P' = G^{\dagger}P^{\dagger}G$ to twirl the noise map and still obtain a logically equivalent circuit!

- There's no such a thing as "noiseless gates" that one can add.

  Typically, single-qubit gates are an order of magnitude better than two-qubit gates. → Use single qubit gates to twirl two qubit gates!

- What about coherent errors different from X over rotation?

  If the twirling gates are sampled form the Pauli group, all possible coherent errors are twirled into stochastic Pauli errors!

How to do that in practice?

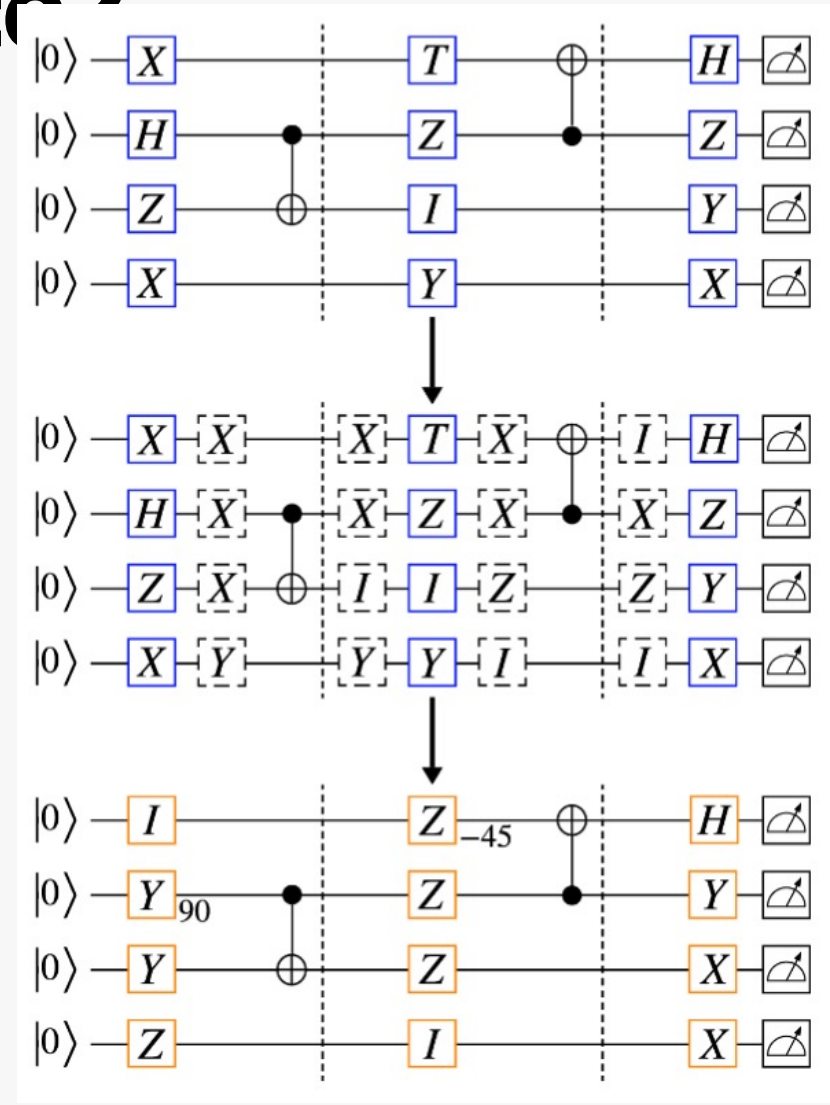
Example from arXiv:2010.00215

Original circuit

One instance of a randomized version of the original circuit, with random single-qubit Paulis (dashed black boxes) added before and after each CNOT layer

Exercise: verify that this circuit is logically equivalent to the original one!

The added random Paulis are **compiled together with the existing single qubit gates** into new single-qubit rotations (orange boxes)



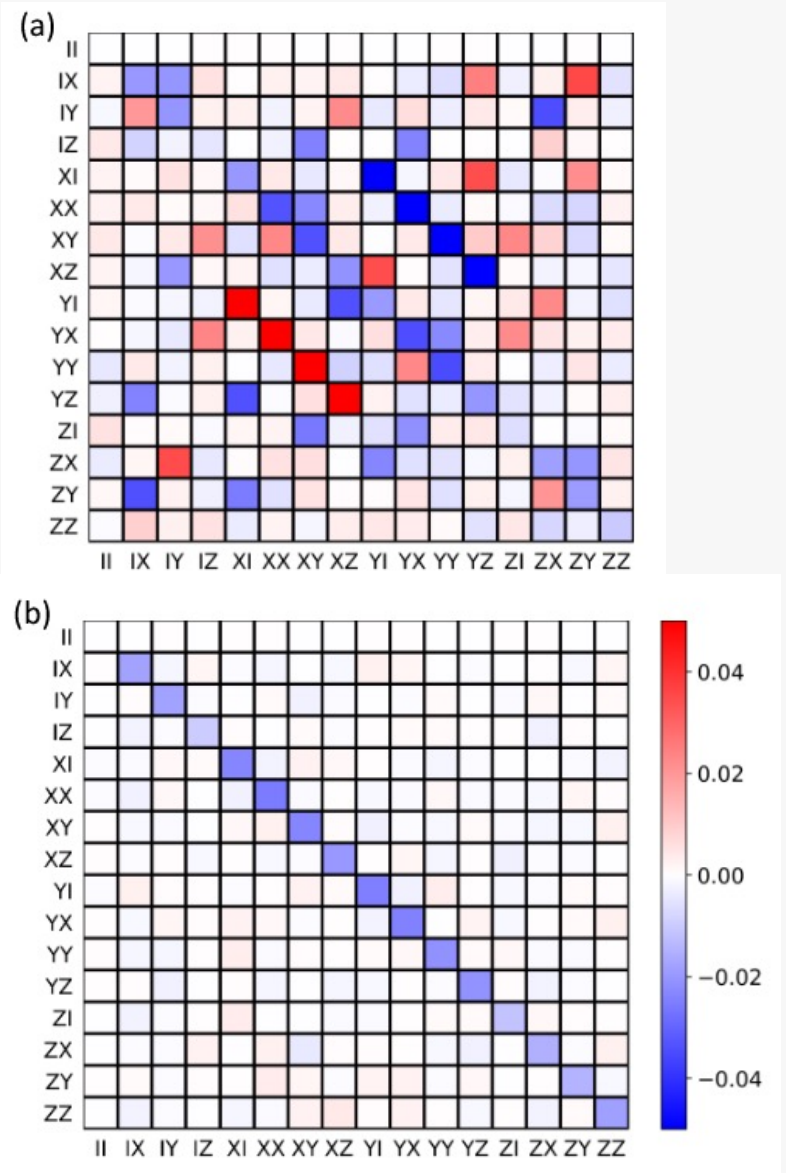
Effects of Pauli Twirling

Figure (a): Experimental noise $\Lambda = e^{\mathcal{L}}$ of a CZ gate, visualized with error generator $\mathcal{L}$ in the Pauli basis

Without Pauli twirling (a), there are many off-diagonal terms. Those indicate for example:

- Non-unital errors (like T1 decay)
- Coherent errors

Figure (b): With Pauli twirling the noise generator is approximately diagonal, showing that twirling successfully twirled the CZ noise into stochastic Pauli noise.

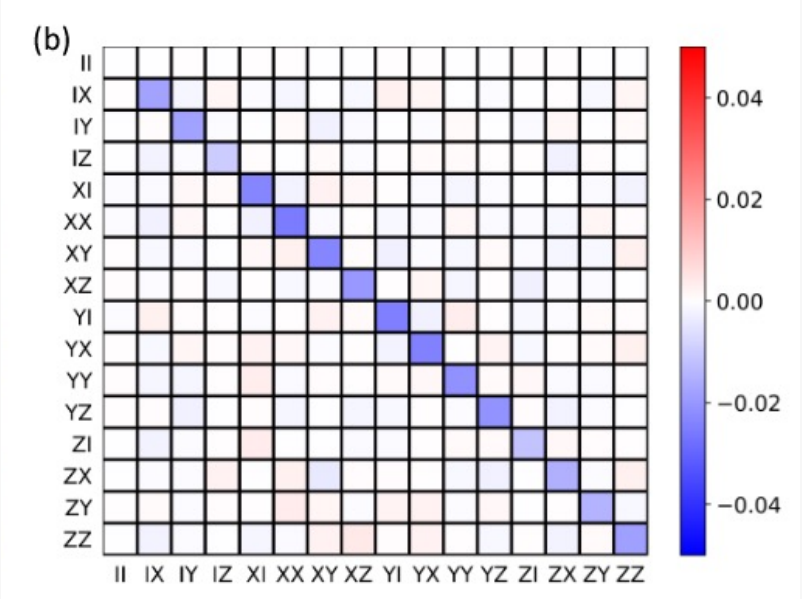


arXiv: 2406.13967

Effects of Pauli Twirling

- The noise generator $\mathcal{L}$ is now much easier to measure and understand: 16 diagonal elements instead of 16x16 elements of the full matrix!
- The *average gate fidelity* now gives a much more accurate picture, since it *can only see the diagonal:*

$$F(\Lambda) = \frac{Tr\left(e^{\mathcal{L}}\right) + d}{d(d+1)}$$



arXiv: 2406.13967

Error mitigation

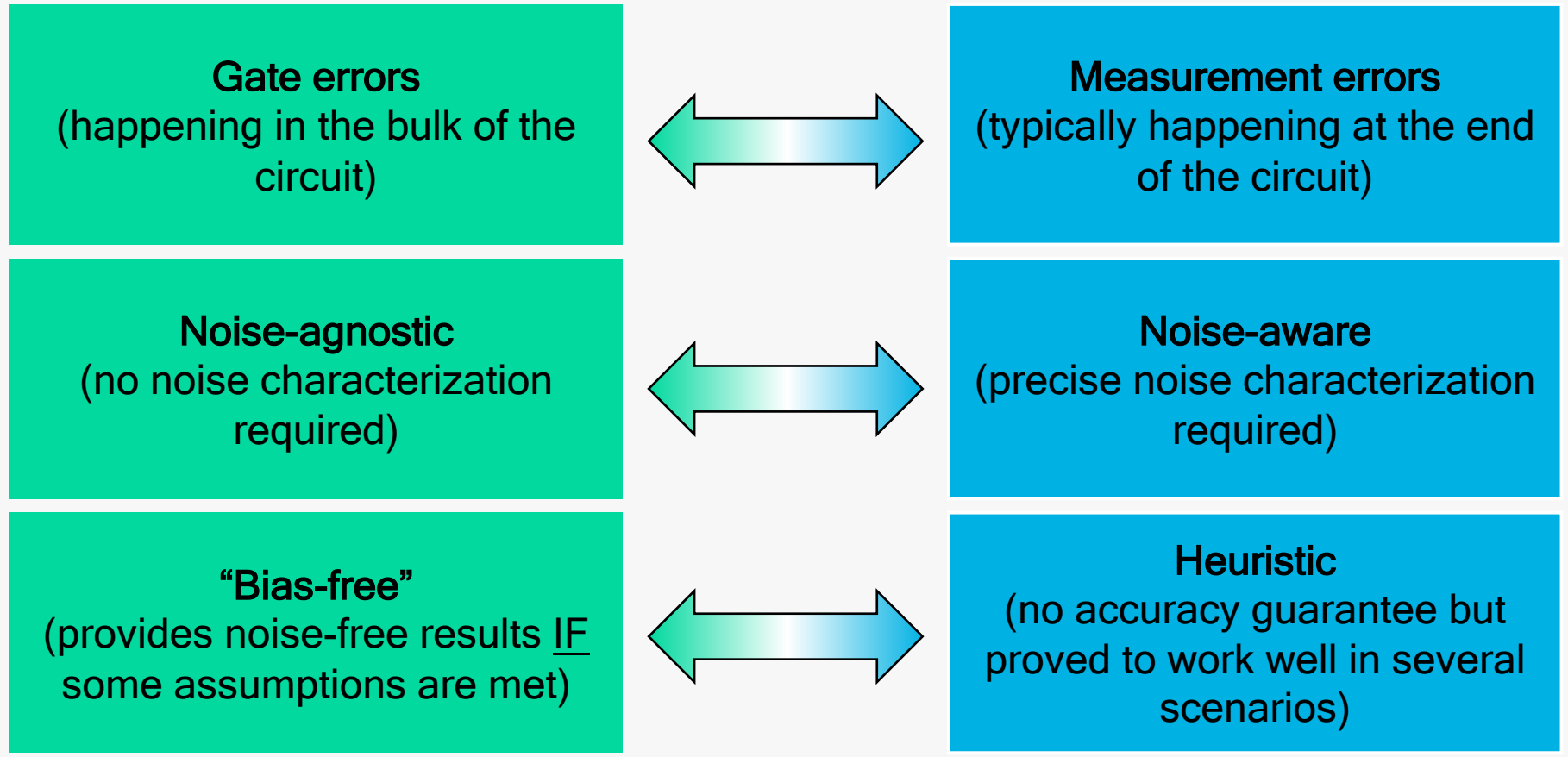
What is error mitigation?

- Unlike error suppression, error mitigation **accepts the presence of errors during the execution** of the desired algorithm and aims at mitigating their effects on the desired outcome.
- This is typically done via (potentially heavy) **post-processing**, often combining together the outcomes of **several non-logically equivalent circuits** executed on the QPU.
- In this sense, error mitigation is a second layer of error reduction which comes after the implementation of all possible error suppression techniques.
- Successful error mitigation allows us to recover almost-error-free results even from very noisy circuits and it is therefore a key ingredient when running large and complicated algorithms before fully-fledged fault tolerant computers will be available.
- The price to pay for “cancelling” large part of the noise is a **significant sampling overhead**, which scales exponentially in the noise level and circuit size. This means that, if the circuit is too noisy, the runtime required to execute the error mitigated algorithm explodes and becomes impractical. Typically, error mitigation is feasible when the number of expected detrimental errors per circuit run is up to $O(1)$.

Types of error mitigation?

In the literature there are dozens of different proposed error mitigation techniques!

One simple (but sometimes too simplistic) approach to classify them is to distinguish them along three axes:



Readout error mitigation

We consider an approach that targets error happening in **the final round of measurements**
(mitigation of mid-circuit measurement is also possible but requires a slightly different approach)

The idea is to:

1. Define a simple (but often times effective) model for the noise process
2. Characterize the readout error rates for each qubit of the QPU (it's a noise-aware technique!)
3. Use this information about readout noise to "undo" it in a post-processing stage.

At first, let cover these three important steps with a simple single-qubit example.

Simple example: 1. Noise process

In general, measuring the state of a qubit is a complicated process where the qubit is coupled to a readout resonator and gradually collapse to one of the two computational states during the measurement time.

In practice, when it comes to readout errors, it turns out that it's often enough to describe them in terms of classical bitflip probabilities. This means that:

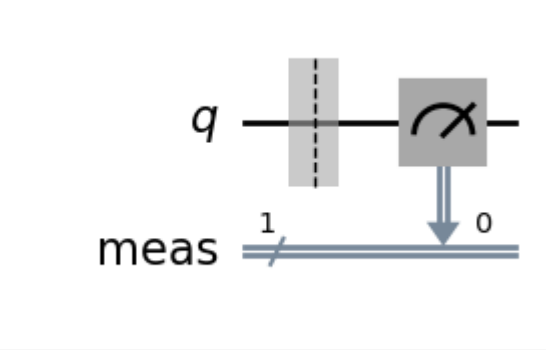
- With a probability $P(0|1)$ a qubit is measured in $|0\rangle$ even if it should have been measured in $|1\rangle$;
- With a probability $P(1|0)$ a qubit is measured in $|1\rangle$ even if it should have been measured in $|0\rangle$.

In general, those two probabilities are different and typically $P(0|1) > P(1|0)$ given the fact that an excited qubit is more likely to decade to its ground state during the measurement time.

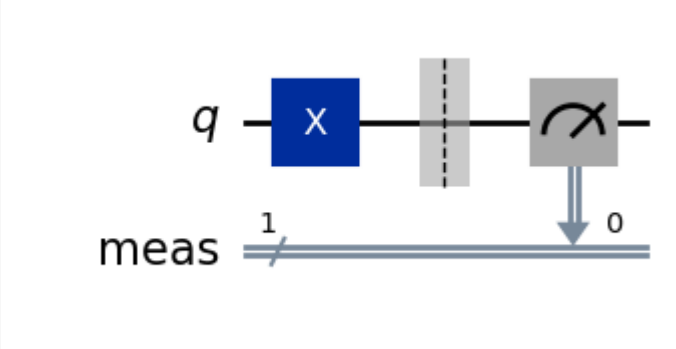
Simple example: 2. Noise charact.

The goal is to measure the two probabilities $P(0|1)$ and $P(1|0)$.

To this end, we run two characterization circuits, each with N_{shots} shots:



In the noiseless case, we expect to measure 0.
Because of errors, we measure 1 for a total of N_1 times.
We can thus estimate $P(0|1) = \frac{N_1}{N_{shots}}$



Similarly, by running this other circuit,
we can estimate $P(1|0) = \frac{N_0}{N_{shots}}$

Note: it is convenient to use a not-too-small N_{shots} as the statistical uncertainty scales as $\sqrt{P(1-P)/N_{shots}}$ (Bernoulli distribution) with $\mathrm{P} \in \{P(0|1), P(1|0)\}$.

Simple example: 3. Post processing

We are now ready to run our desired circuit (using n_{shots}) and measure the outcomes. We measure 0 (1) a total of n_0 (n_1) times. It is convenient to organize the noisy result in a vector $\vec{p}_{noisy} = \begin{pmatrix} p_0 \\ p_1 \end{pmatrix}$, with $p_j = n_j / n_{shots}$.

Based on the characterization, we can construct a classical assignment matrix which relates the noisy output to the exact one $\vec{q}_{exact}$:

$$\vec{p}_{noisy} = \begin{pmatrix} 1 - P(1|0) & P(0|1) \\ P(1|0) & 1 - P(0|1) \end{pmatrix} \vec{q}_{exact}$$

We can thus invert the matrix to recover the exact results from the noisy one:

$$\vec{q}_{exact} = \frac{1}{1 - P(0|1) - P(1|0)} \begin{pmatrix} 1 - P(0|1) & -P(0|1) \\ -P(1|0) & 1 - P(1|0) \end{pmatrix} \vec{p}_{noisy}$$

IF the simple model we used to describe readout noise is correct and IF the characterization is perfect, we are therefore able to “undo” the effect of the readout noise and recover the **noiseless result** (it’s a “bias-free” method)!

Simple example: Sampling overhead

This very simple example is also extremely useful to highlight the sampling overhead associated with (generic) error mitigation strategies.

To this end, let's focus on the statistical uncertainty (due to the finite number of shots) and assume, for simplicity, $P(1|0) = P(0|1) = \epsilon$.

- For the noisy result, the statistical uncertainty on $\vec{p}_{\text{noise}}$ simply read $\sqrt{p_0 p_1 / n_{shots}}$ (Bernoulli distribution).
- The mitigated result reads

$$\vec{q}_{exact} = \begin{pmatrix} (1-\epsilon)p_0 - \epsilon p_1 \\ (1-\epsilon)p_1 - \epsilon p_0 \end{pmatrix} \frac{1}{1-2\epsilon}$$

- The uncertainty on the mitigated result is thus amplified by the factor $(1-2\epsilon)^{-1} \sim (1+2\epsilon)$
- To recover the statistical uncertainty of the noisy result, it is therefore needed to increase the number of shots by the square of the amplification factor.

NOTE: For a system where we measure N_q qubits, the (exponential) overhead is $(1+4\epsilon))^{N_q}$.

Simple example: Summary

We just saw an example of a readout error mitigation technique, which is noise-aware, bias-free and suffer from an overhead which is exponential in the error rate ϵ and in the number of qubits!

Note that, in practice, the sampling overhead of readout error mitigation is not a limiting factor. This is because, in many circuits of practical interest, the number of measurements (relevant to determine the observable or the probability distribution of interest) is much less than the number of gates. The exponential overhead of techniques that mitigate gate errors is therefore larger, typically.

In the following, we will see how to extend the simple example described so far to the multi-qubit case and how to use available tools to perform readout error mitigation in practice!

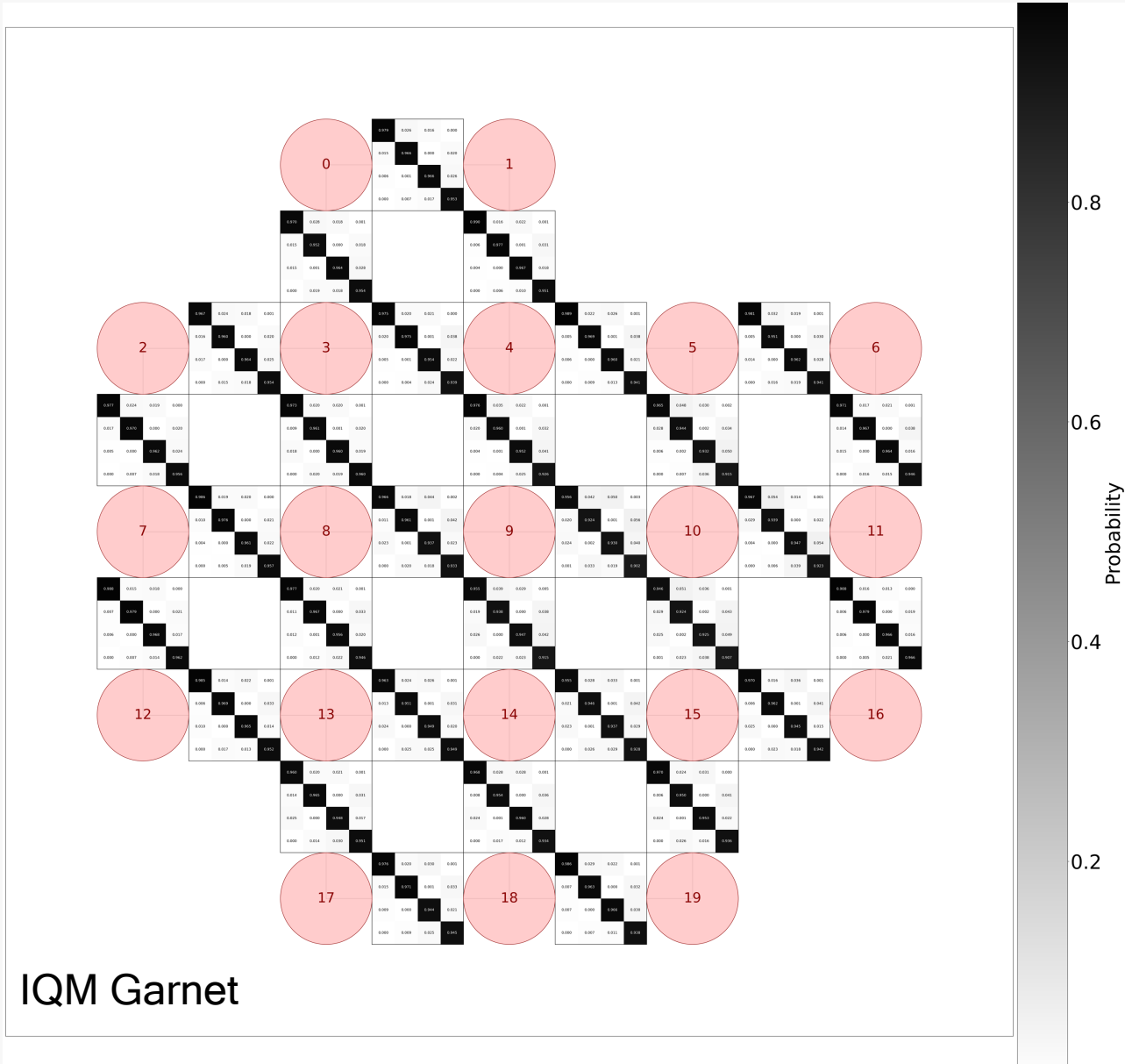
Readout errors are not negligible

Zoom in to see actual values.
Confusion matrices indicate the error probabilities:

M =

P(00|00)	P(00|01)	P(00|10)	P(00|11)
P(01|00)	P(01|01)	P(01|10)	P(01|11)
P(10|00)	P(10|01)	P(10|10)	P(10|11)
P(11|00)	P(11|01)	P(11|10)	P(11|11)

P(Y|X) is the probability of observing state Y knowing that we prepared state X.



IQM Garnet

Readout error mitigation

1. **Calibration stage**: Obtain confusion matrix M.
2. Run the quantum circuit in the noisy QPU and obtain $\vec{p}_{noisy}$.
3. **Correction stage**: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

[1] Nation, Paul D., et al. "Scalable mitigation of measurement errors on quantum computers." *PRX Quantum* 2.4 (2021): 040326.

[2] Yang, Bo, Rudy Raymond, and Shumpei Uno. "Efficient quantum readout-error mitigation for sparse measurement outcomes of near-term quantum devices." *Physical Review A* 106.1 (2022): 012423.

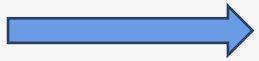
Readout error mitigation

1. **Calibration stage**: Obtain confusion matrix *M*.

Let us assume that single-qubit readout errors dominate, then confusion matrix *M* reads,

$M_j =$

P(0|0)	P(0|1)
P(1|0)	P(1|1)



$$M = M_1 \otimes \cdots \otimes M_n$$

Readout error mitigation

1. **Calibration stage**: Obtain confusion matrix *M*.

Let us assume that single-qubit readout errors dominate, then confusion matrix *M* reads,

$M_j =$

P(0|0)	P(0|1)
P(1|0)	P(1|1)

$$M = M_1 \otimes \cdots \otimes M_n$$

Many-qubit independent errors are still captured!

e.g. P(11|00) = $P(1|0)_1$ x $P(1|0)_2$

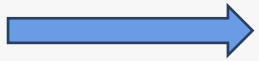
Readout error mitigation

1. **Calibration stage**: Obtain confusion matrix *M*.

Let us assume that single-qubit readout errors dominate, then confusion matrix *M* reads,

$M_j =$

P(0|0)	P(0|1)
P(1|0)	P(1|1)



$$M = M_1 \otimes \cdots \otimes M_n$$

How can we obtain all these matrices in a QPU?

"Marginal" method: 2 calibration circuits, where all qubits are either at state 0 or 1.
Independent method: *2n* calibration circuits, where each qubit is independently prepared in state 0 or 1.
Balanced method: *2n* calibration circuits, where each qubit state, 0 or 1 is prepared *n* times.

Readout error mitigation

1. **Calibration stage**: Obtain confusion matrix M.
2. Run the quantum circuit in the noisy QPU and obtain $\vec{p}_{noisy}$.
3. **Correction stage**: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

(Non-twirling) techniques differ in the way they solve this equation

[1] Nation, Paul D., et al. "Scalable mitigation of measurement errors on quantum computers." *PRX Quantum* 2.4 (2021): 040326.

[2] Yang, Bo, Rudy Raymond, and Shumpei Uno. "Efficient quantum readout-error mitigation for sparse measurement outcomes of near-term quantum devices." *Physical Review A* 106.1 (2022): 012423.

With twirling

[3] Gupta, Riddhi S., et al. "Probabilistic error cancellation for dynamic quantum circuits." *Physical Review A* 109.6 (2024): 062617.

[4] Koh, Jin Ming, Dax Enshan Koh, and Jayne Thompson. "Readout error mitigation for mid-circuit measurements and feedforward." *arXiv preprint arXiv:2406.07611* (2024).

[5] Hashim, Akel, et al. "Quasiprobabilistic Readout Correction of Midcircuit Measurements for Adaptive Feedback via Measurement Randomized Compiling." *PRX Quantum* 6.1 (2025): 010307.

Readout error mitigation

Correction stage: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

- NNLS is expensive and requires the construction of the 2^n x 2^n confusion matrix M. Therefore, techniques like M3 [1] or Bo et. al. [2], do not used constrained optimization.

Readout error mitigation

Correction stage: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

- NNLS is expensive and requires the construction of the 2^n x 2^n confusion matrix M. Therefore, techniques like M3 [1] or Bo et. al. [2], do not used constrained optimization.
- They first reduce the confusion matrix dimensionality by assuming that $\vec{p}_{ideal} \in \vec{p}_{noisy}$ → limited by number of shots S.

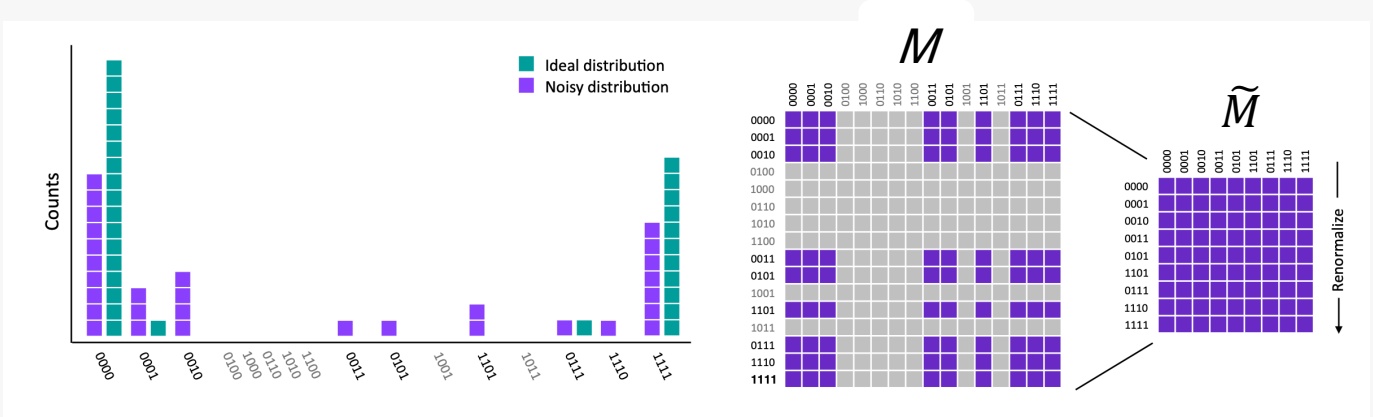


Image source: Qiskit mthree package

Readout error mitigation

Correction stage: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

- NNLS is expensive and requires the construction of the 2^n x 2^n confusion matrix M. Therefore, techniques like M3 [1] or Bo et. al. [2], do not used constrained optimization.
- They first reduce the confusion matrix dimensionality by assuming that $\vec{p}_{ideal} \in \vec{p}_{noisy}$ → limited by number of shots *S*.
- LU factorization (direct) or iterative (matrix-free) methods available in open-source packages e.g. [1].

S<20000
$O(S^3)$ time

Buggy
(lower memory requirements and most likely faster; approximate variance calculation only)

Readout error mitigation

Correction stage: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

- NNLS is expensive and requires the construction of the 2^n x 2^n confusion matrix M. Therefore, techniques like M3 [1] or Bo et. al. [2], do not used constrained optimization.
- They first reduce the confusion matrix dimensionality by assuming that $\vec{p}_{ideal} \in \vec{p}_{noisy}$ → limited by number of shots S.
- LU factorization (direct) or iterative (matrix-free) methods available in open-source packages e.g. [1]
- Bo et al. [2] can achieve it in $O(nS^2)$ time and memory scaling as O(S). Parallel computation is possible.

Readout error mitigation

Correction stage: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

- NNLS is expensive and requires the construction of the 2^n x 2^n confusion matrix M. Therefore, techniques like M3 [1] or Bo et. al. [2], do not used constrained optimization.
- They first reduce the confusion matrix dimensionality by assuming that $\vec{p}_{ideal} \in \vec{p}_{noisy}$ → limited by number of shots *S*.
- LU factorization (direct) or iterative (matrix-free) methods available in open-source packages e.g. [1].
- Bo et al. [2] can achieve it in $O(nS^2)$ time and memory scaling as O(S). Parallel computation is possible.
- [2]: Then (numerically) make the sum of the output vector elements be equal to 1 in O(S) time to get quasi-probability vector $\vec{q}_{corrected}$.

Readout error mitigation

Correction stage: find $\vec{p}_{corrected}$ via the following equation:

$$\vec{p}_{noisy} = M\,\vec{p}_{corrected}$$

- NNLS is expensive and requires the construction of the 2^n x 2^n confusion matrix M. Therefore, techniques like M3 [1] or Bo et. al. [2], do not used constrained optimization.
- They first reduce the confusion matrix dimensionality by assuming that $\vec{p}_{ideal} \in \vec{p}_{noisy}$ → limited by number of shots *S*.
- LU factorization (direct) or iterative (matrix-free) methods available in M3 [1].
- Bo et al. [2] can achieve it in $O(nS^2)$ time and memory scaling as O(S). Parallel computation is possible.
- [2]: Then (numerically) make the sum of the output vector elements be equal to 1 in O(S) time to get quasi-probability vector $\vec{q}_{corrected}$.
- Lastly, find the nearest probability distribution $\vec{p}_{corrected}$ from $\vec{q}_{corrected}$ in O(S log(S)) time [1, 2].

Readout error mitigation

Consider the following test:

- Apply $R_x(\theta)$ to all qubits in the Emerald QPU.
- Obtain the average Z (or ZZ) observable outcome over all (nearest-neighbour) qubits and compute the mean absolute observable error, as follows,

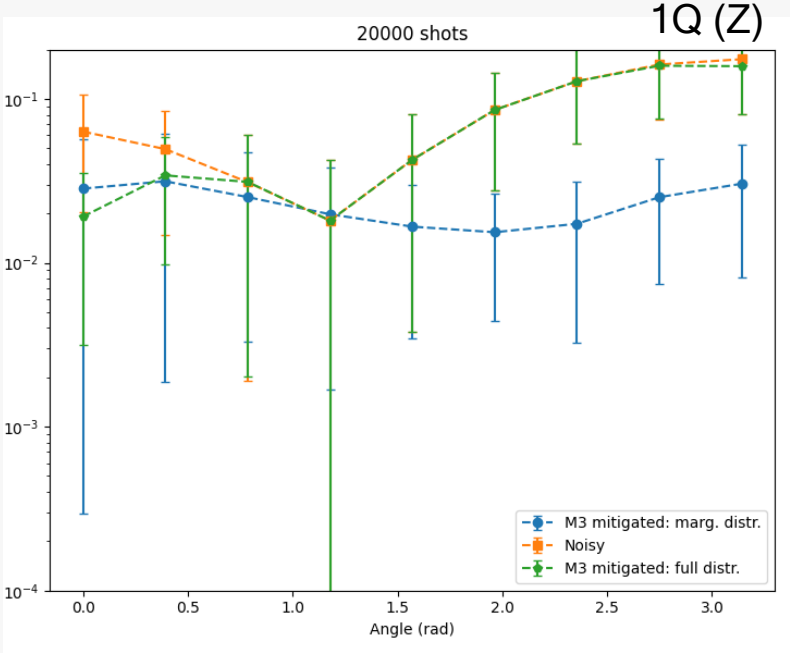
$$Error = \sum_i |\langle O_i \rangle_{ideal} - \langle O_i \rangle_{ideal}| / N.$$

Readout error mitigation

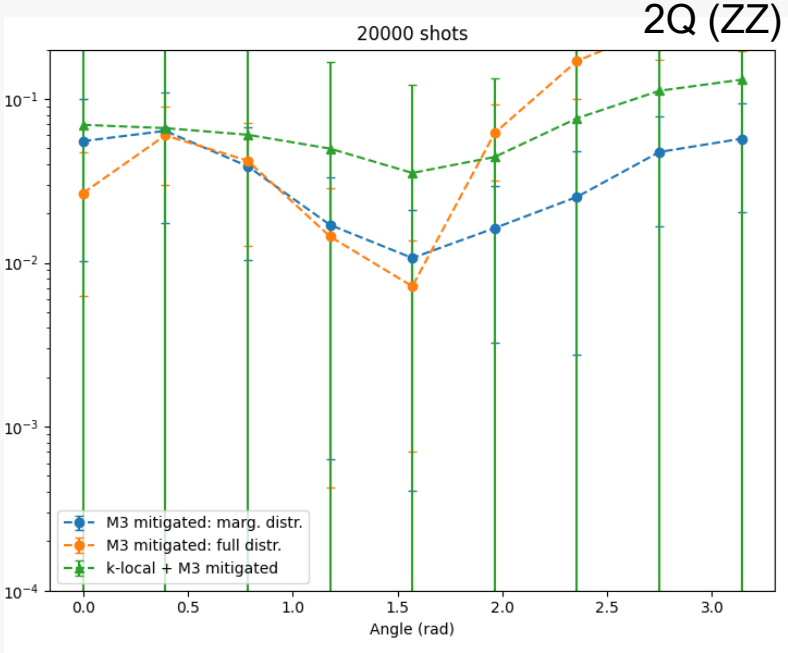
Consider the following test:

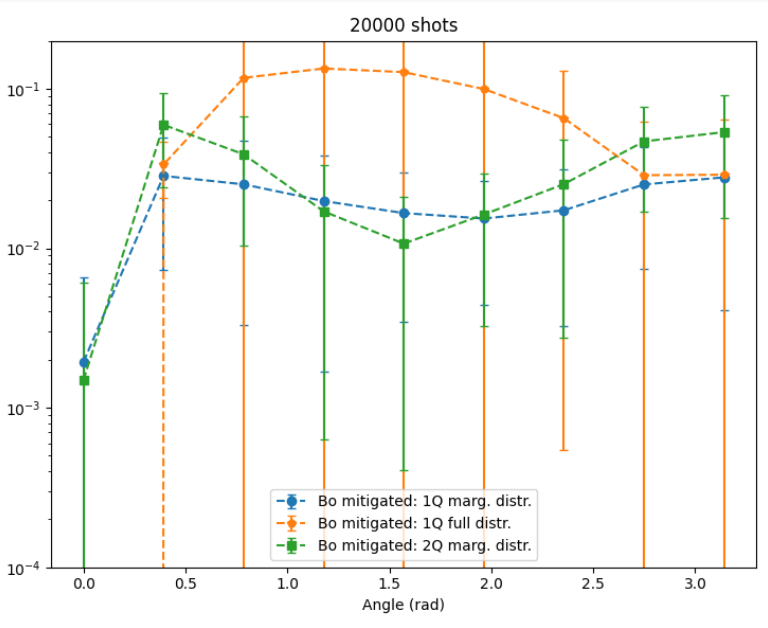
- Apply $R_x(\theta)$ to all qubits in the Emerald QPU.
- Obtain the average Z (or ZZ) observable outcome over all (nearest-neighbour) qubits and compute the mean absolute error.

1Q (Z)



2Q (ZZ)





Noise Robust Estimation (NRE) of Quantum Observables.

arXiv:2503.06695

Noise-aware methods (PEC):

Requirement: Precise noise characterization.
Promise: Theoretically a bias-free estimation
Challenges: Pauli noise learnability, and parameter drifts

→ **Finite bias in the final estimation**

Noise-agnostic methods (ZNE, CDR, vnCDR, MDN):

Requirement: Noise-amplification, or training circuits, or both
Promise: Estimation of the noise-free observable (no guarantee)
Challenges: Various forms of **model mismatch** for different methods

→ **Bias in the final estimation**

Noise-Robust Estimation (NRE): Noise-agnostic and reliable

- **Using a novel correlation to address (and suppress) the 'model mismatch' problem.**
- **Tested on IQM superconducting QPUs with circuits up to 20 qubits and 240 CZ gates.**
- **Achieved highly accurate estimations, significantly outperforming other noise-agnostic methods.**

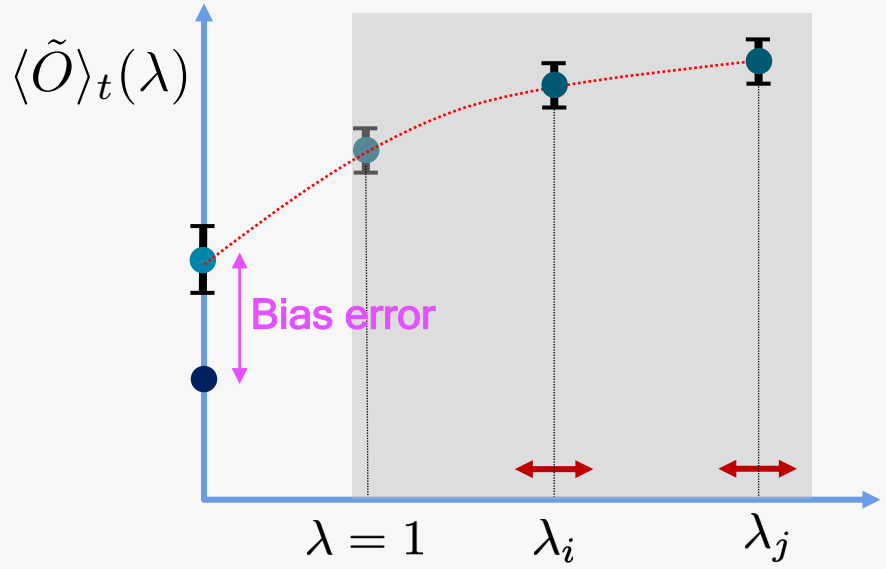
- Let's take a closer look at ZNE first!

- **PEC:** Nature Physics **19,** 1116 (2023)
- **ZNE:** PRL **119**, 180509 (2017)
- **CDR:** Quantum **5**, 592 (2021)
- **vnCDR**: PRResearch **3**, 033098 (2021)
- **MDN**: **M**itigating **D**epolarizing **N**oise on Quantum Computers with Noise-Estimation Circuits, PRL. **127**, 270502 (2021)

Overview of Zero Noise Extrapolation (ZNE):

- Assume the noise of the system is characterized by a parameter ε
- In practice, ε cannot be set to zero.
  However, the noise can be amplified: $\varepsilon \rightarrow \lambda\varepsilon$
- **ZNE workflow:**

  i. Measure $\langle O \rangle$ for different values of λ
  ii. Having a set of $\langle O \rangle(\lambda)$, extrapolate the data to the case where $\lambda \rightarrow 0$



λ : Noise scale factor

Challenge:	Example of model mismatch:
Knowing the precise scaling of $\langle \tilde{O} \rangle(\lambda)$ by the noise scale factor λ	Assuming a certain mathematical function for $\langle \tilde{O} \rangle(\lambda)$ (e.g. quadratic) which does not hold true.
Precise implementation of λ_i	Assuming certain values for λ_i that is different than the (unknown) truly implemented values.

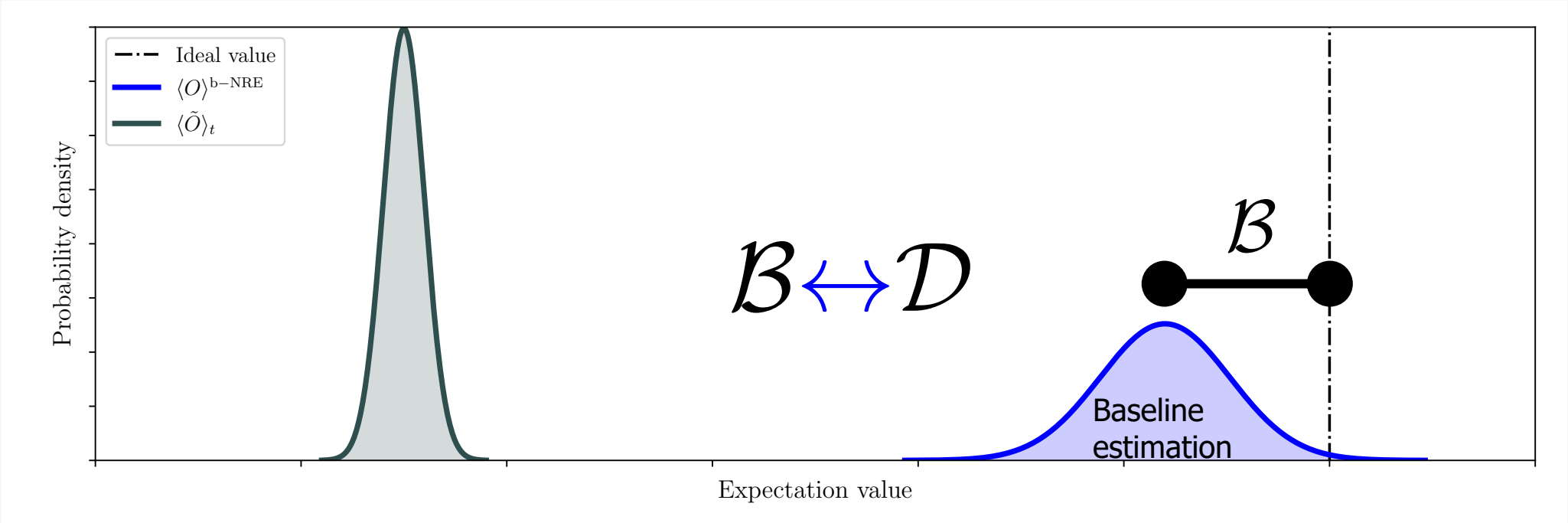
→ Bias error

NRE framework

NRE *Ansatz* : An auxiliary quantity with ***reduced*** noise-sensitivity $\mathcal{A}(\lambda)$ compared with $\langle \tilde{O} \rangle_t(\lambda)$

1st step post-processing: Baseline error-mitigated estimation (b-NRE) includes an ***unknown*** **bias** error, $\mathcal{B}$

A key insight: Identification of a correlation between the unknown bias and a **computable** metric, **normalized dispersion**, $\mathcal{D}$



IQM

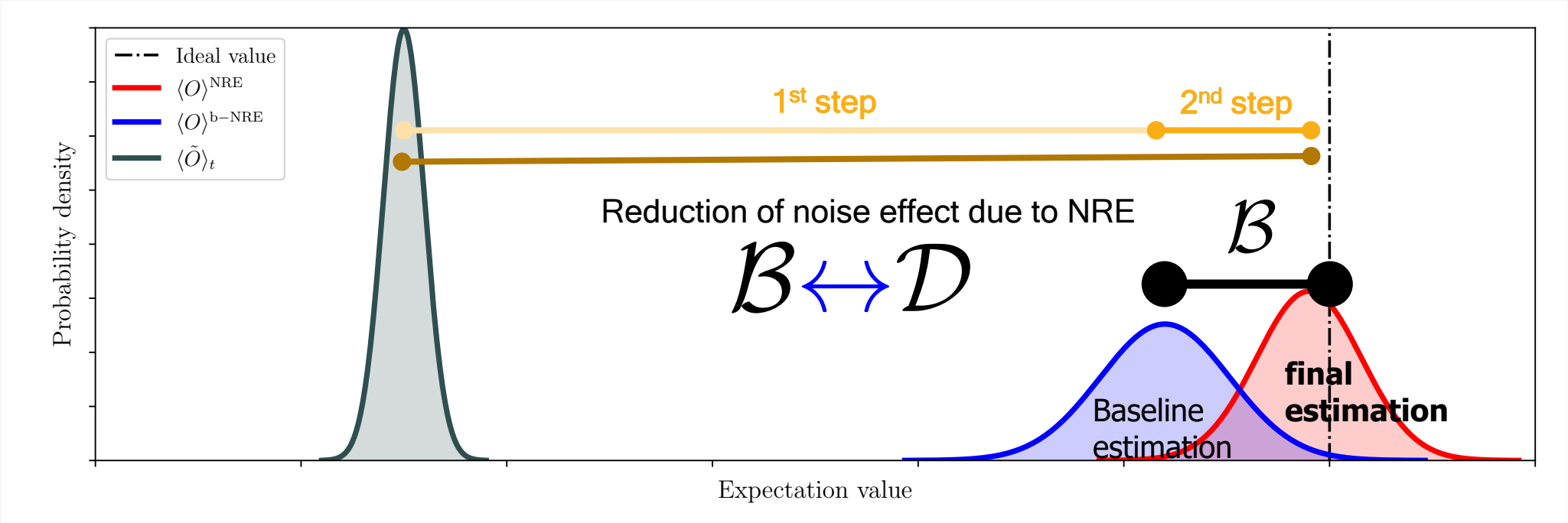
NRE framework

NRE *Ansatz* : An auxiliary quantity with ***reduced*** noise-sensitivity $\mathcal{A}(\lambda)$ compared with $\langle \tilde{O} \rangle_t(\lambda)$

1st step post-processing: Baseline error-mitigated estimation (b-NRE) includes an ***unknown*** **bias** error, $\mathcal{B}$

A key insight: Identification of a correlation between the unknown bias and a **computable** metric, **normalized dispersion**, $\mathcal{D}$

2nd step post-processing: → **final NRE estimation with systematically reduced bias, using the** $\mathcal{B} \leftrightarrow \mathcal{D}$ **correlation**



IQM

NRE framework

$$\mathcal{A}_{ncc}(n, \lambda_i) = \mathcal{P}_1(\lambda_i) + n\mathcal{P}_2(\lambda_i)$$

Minimize noise-sensitivity
via tuning control parameter, n

Competition of noise effects!

$$\mathcal{P}_1(\lambda_i) = \langle \tilde{O} \rangle_t(\lambda_i) \left(\frac{\langle O \rangle_{ncc}}{\langle \tilde{O} \rangle_{ncc}(\lambda_i)} \right),$$

$$\mathcal{P}_2(\lambda_i) = \log \left(\frac{\langle O \rangle_{ncc}}{\langle \tilde{O} \rangle_{ncc}(\lambda_i)} \right)$$

$\mathcal{P}_1 \sim$ (multi-)exponential with *small* exponent
$\sim$ Linear in λ

$P_2 \sim$ Linear in λ

Noise-canceling circuit (ncc): use to cancel the effect of the noise on $\mathcal{A}$

- Structurally similar to the target circuit with known noiseless expectation value $\langle O \rangle_{ncc}$
- We consider the "closest" Clifford circuit to the target circuit in the following results

Control parameter, n : A real-valued parameter used in the first-step post-processing

NRE framework

$$\mathcal{A}_{ncc}(n,\lambda_i) = \mathcal{P}_1(\lambda_i) + n\mathcal{P}_2(\lambda_i) \qquad \mathcal{P}_1(\lambda_i) = \langle\tilde{O}\rangle_t(\lambda_i)\left(\frac{\langle O\rangle_{ncc}}{\langle\tilde{O}\rangle_{ncc}(\lambda_i)}\right), \qquad \mathcal{P}_2(\lambda_i) = \log\left(\frac{\langle O\rangle_{ncc}}{\langle\tilde{O}\rangle_{ncc}(\lambda_i)}\right)$$

$$\mathcal{A}_{ncc}(n,\lambda=0) = \langle O\rangle_t$$

t and ncc circuits are measured at M values for the noise scale factor, $\lambda_i, i = 1,2,\ldots,M$

Taylor expansion: $\langle O\rangle_t = \mathcal{A}_{ncc}(n,\lambda_1) + \sum_{j=1}^{M-1}\frac{(-\lambda_1)^j}{j!}\mathcal{A}^{[j]}_{\lambda_1}(n) + \mathcal{B}$ $\mathcal{B}$: Bias error

1st step post-processing: $\sum_{j=1}^{M-1}\frac{(-\lambda_1)^j}{j!}\mathcal{A}^{[j]}_{\lambda_1}(n_{op}) = 0 \rightarrow \langle O\rangle_t = \mathcal{A}_{ncc}(n_{op},\lambda_1) + \mathcal{B}$

baseline estimation: $\langle O\rangle_{\mathrm{b-NRE}} = \mathcal{A}_{ncc}(n_{op},\lambda_1)$

$$\langle O\rangle_t = \langle O\rangle_{\mathrm{b-NRE}} + \mathcal{B}$$

- The accuracy of the baseline estimation is limited by the *unknown* bias error.
- The 2nd post-processing step aims to systematically reduce the bias error.

IQM IQM

NRE framework

Origin of the bias error $\mathcal{B}$:

- **Truncation** → Higher-order *derivatives*
- **Discretization** → Higher-order *derivatives*
- **Imprecise noise amplification**

Smaller variation in $\mathcal{A}_{ncc}(\lambda_i)$ across different values for λ_i ⇒ **Smaller Bias!**

Normalized dispersion: $\mathcal{D} = \dfrac{\mathrm{MAD}[\{\mathcal{A}(\lambda_i)\}]}{\mathrm{MAD}[\{\langle\tilde{O}\rangle_t(\lambda_i)\}]}$

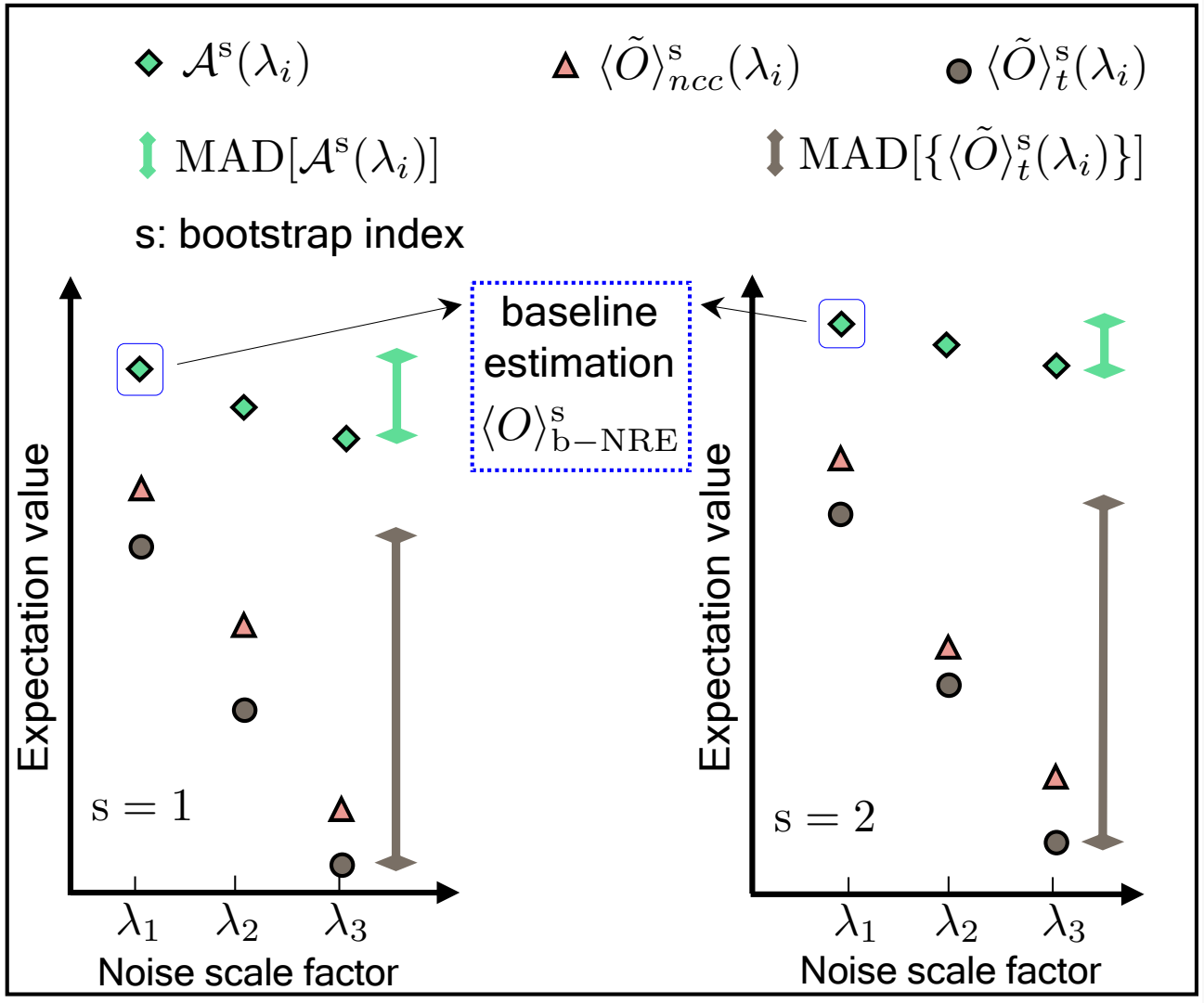
A proxy for bias

Bootstrapping → a data set $\langle O\rangle^{\mathrm{s}}_{\mathrm{b-NRE}}$, $\mathcal{D}^{\mathrm{s}}$

2nd post-processing step:

$$\langle O\rangle_{\mathrm{NRE}} \equiv \lim_{\mathcal{D}^{\mathrm{s}}\to 0} \langle O\rangle^{\mathrm{s}}_{\mathrm{b-NRE}}(\mathcal{D}^{\mathrm{s}}).$$

Baseline estimation obtained from 2 bootstrap instances:



MAD: Mean Absolute Deviation

NRE framework

Origin of the bias error $\mathcal{B}$:

- **Truncation** → **Higher-order *derivatives***
- **Discretization** → **Higher-order *derivatives***
- **Imprecise noise amplification**

Smaller variation in $\mathcal{A}_{ncc}(\lambda_i)$ across different values for λ_i → **Smaller Bias!**

Normalized dispersion: $\mathcal{D} = \dfrac{\mathrm{MAD}[\{\mathcal{A}(\lambda_i)\}]}{\mathrm{MAD}[\{\langle\tilde{O}\rangle_t(\lambda_i)\}]}$

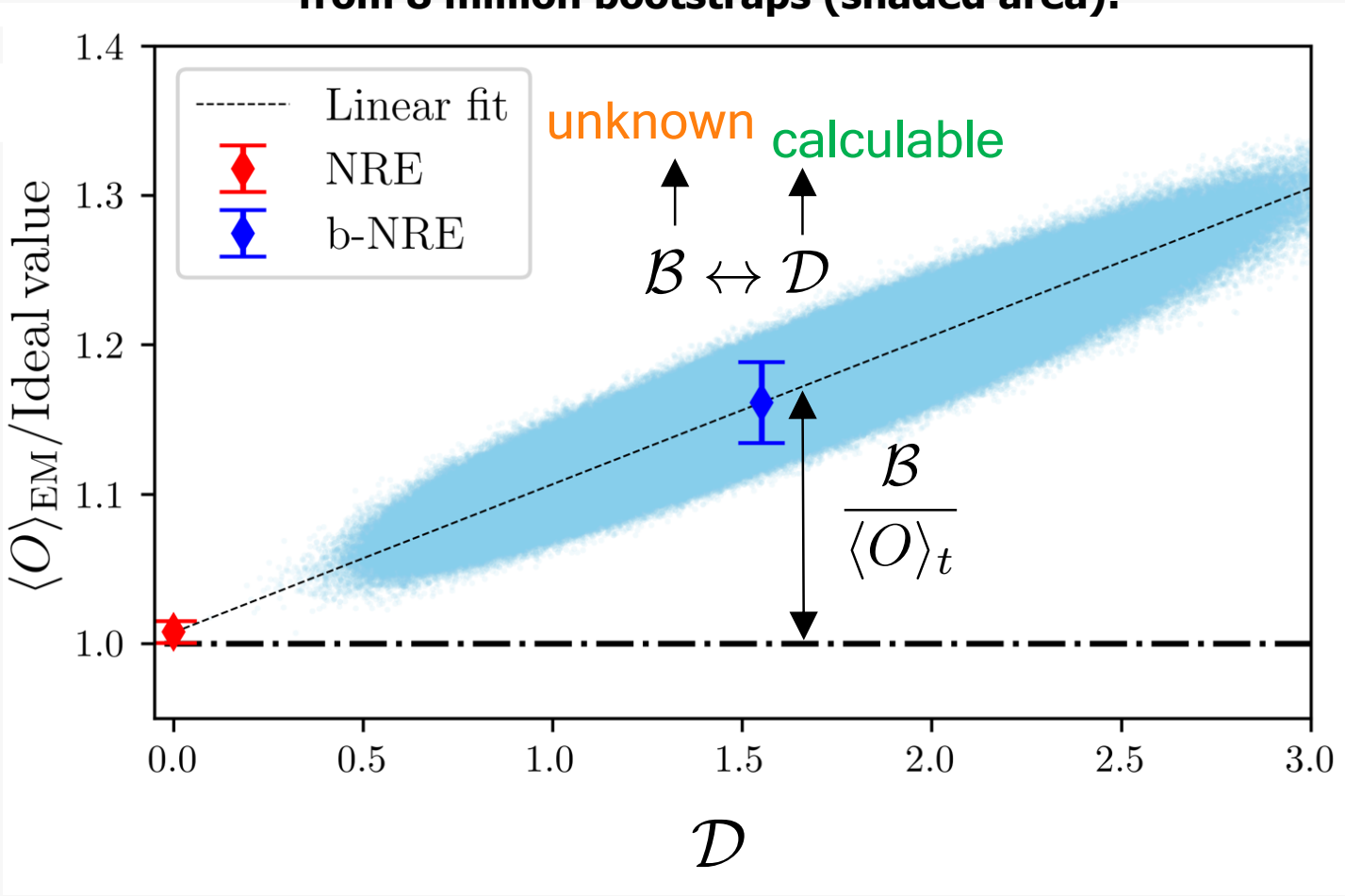
A proxy for bias

Bootstrapping → a data set $\langle O\rangle^{\mathrm{s}}_{\mathrm{b-NRE}}$, $\mathcal{D}^{\mathrm{s}}$

2nd post-processing step:

$$\langle O\rangle_{\mathrm{NRE}} \equiv \lim_{\mathcal{D}^{\mathrm{s}}\to 0} \langle O\rangle^{\mathrm{s}}_{\mathrm{b-NRE}}(\mathcal{D}^{\mathrm{s}}).$$

Experimentally obtained baseline estimations from 8 million bootstraps (shaded area):



MAD: Mean Absolute Deviation

NRE verification I: ground state energy of Transverse-Field Ising Model (TFIM)

$$H = -g\sum_j \sigma_X^j - \sum_{(i,j)} \sigma_Z^i \sigma_Z^j.$$

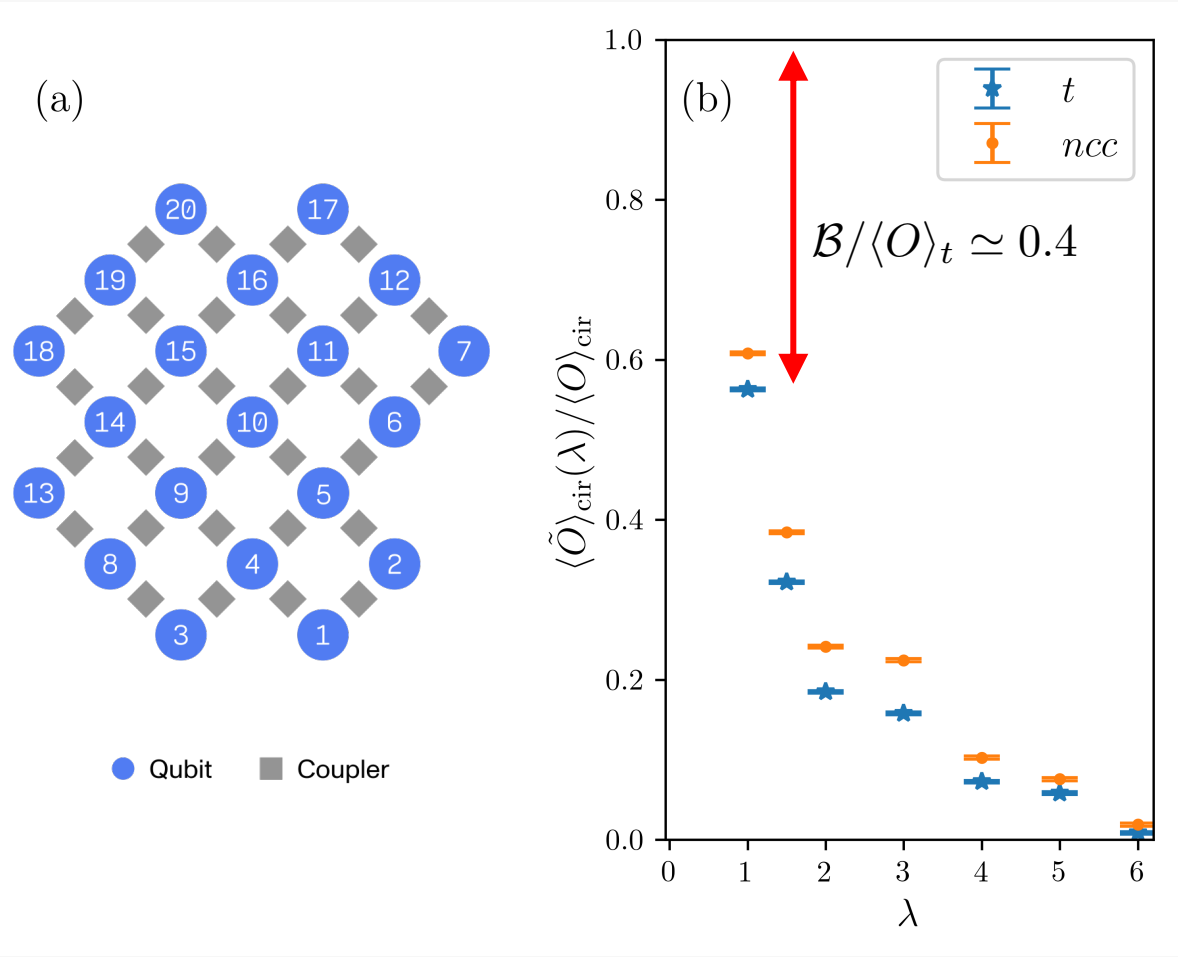
Transverse field strength, $g = 2$

Target circuit is obtained by a QAOA ansatz with 4 layers

QPU: IQM Garnet
Qubit count: **20**
CZ count: **240**
Depth: **77**
Observables of weight **1** and **2**

Noise-amplification method:
Global unitary folding

IQM Garnet is available on cloud:
https://resonance.meetiqm.com/



NRE verification I: ground state energy of Transverse-Field Ising Model (TFIM)

$$H = -g\sum_j \sigma_X^j - \sum_{(i,j)} \sigma_Z^i \sigma_Z^j.$$

Transverse field strength, $g = 2$

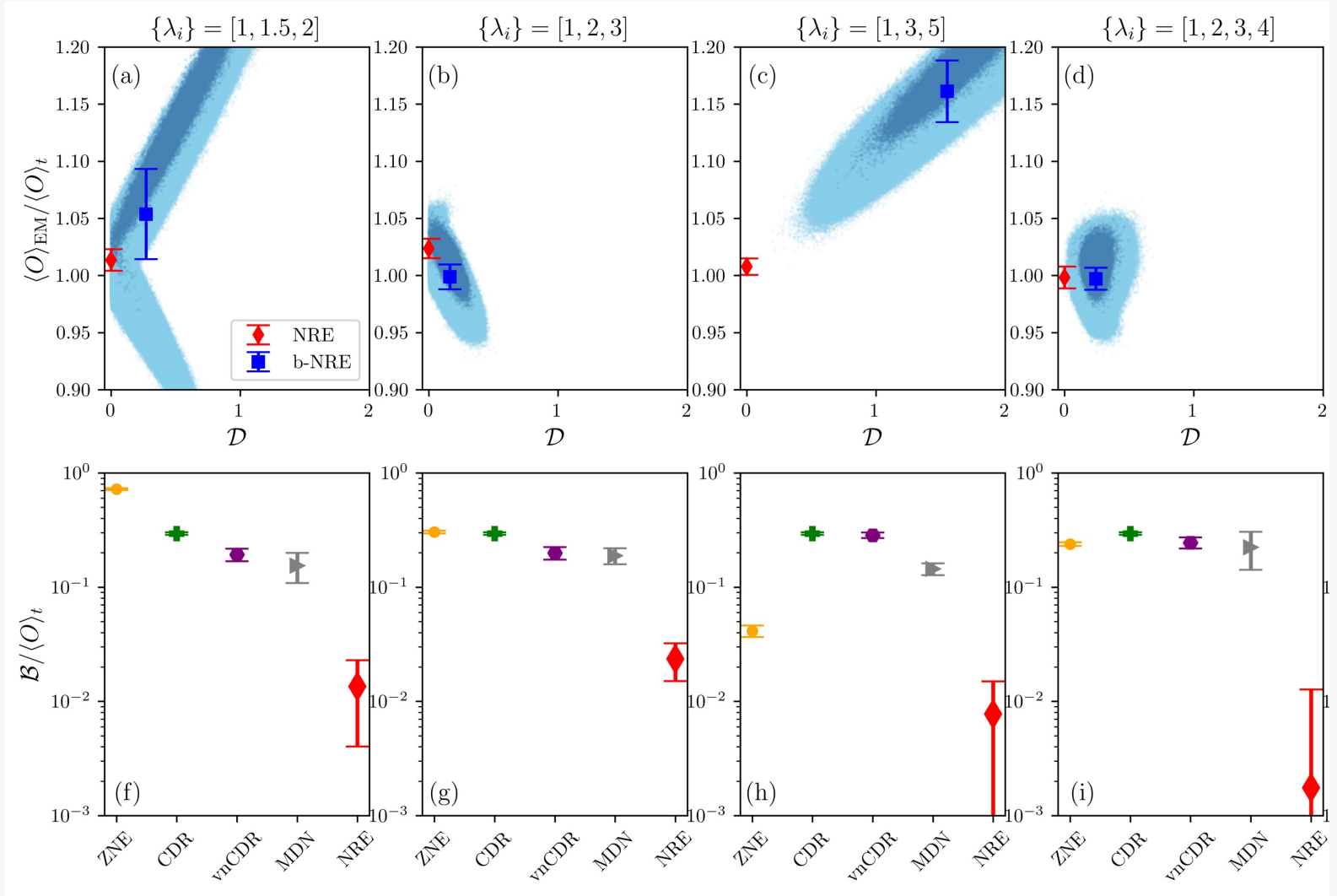
Target circuit is obtained by a QAOA ansatz with 4 layers

NRE: Averaged over 200 bootstrap sets
# of baseline estimations per single bootstrap index = 4×10^4

Total number of shots: 2.4×10^5

NRE:

- **Highly-accurate and stable predictions**
- **Persistently outperforming other methods**



Conclusion

Conclusion

Quantum computing holds big promise, but noise remains a fundamental challenge. In these slides, we explored:

Noise Processes – We discussed coherent errors (e.g., unitary over-rotations) and modelled incoherent (e.g., T_1/T_2 processes) errors, showing how they affect gate operations and idle qubits. Understanding these processes is crucial for developing effective error reduction techniques (error suppression and mitigation).

Error Suppression – Techniques like **dynamical decoupling** (to mitigate coherent errors and non-Markovian errors, for example) and **Pauli twirling** (to convert coherent noise into stochastic noise channels) help reduce errors in quantum circuits before they accumulate.

Error Mitigation – Methods such as **readout error mitigation** (correcting measurement inaccuracies) and **Noise Robust Estimation (NRE)** further improve results in the presence of unavoidable noise.