

Introduction to Quantum Error Mitigation

Part 1 - Noise and Error-Mitigation
Dan Mills



How to deal with noise?

Combination of software and hardware techniques!

- Physical layer (optimal control, compensating pulses, dynamical decoupling)
- Circuit level (noise aware compilation/routing, frame randomisation/randomised compilation)
- Application level
 - Error mitigation of outcome probabilities (e.g SPAM)
 - Error mitigation of observables
 - Error mitigation for specific algorithms (or problem instances)
- Logical level

How to deal with noise?

Combination of software and hardware techniques!

- Physical layer (optimal control, compensating pulses, dynamical decoupling)
- Circuit level (noise aware compilation/routing, frame randomisation/randomised compilation)
- Application level
 - Error mitigation of outcome probabilities (e.g SPAM)
 - Error mitigation of observables
 - Error mitigation for specific algorithms (or problem instances)
- Logical level



Reduce the approximation
error via classical post-
process

How to deal with noise?

Combination of software and hardware techniques!

- Physical layer (optimal control, compensating pulses, dynamical decoupling)
- Circuit level (noise aware compilation/routing, frame randomisation/randomised compilation)
- Application level
 - Error mitigation of outcome probabilities (e.g SPAM)
 - Error mitigation of observables
 - Error mitigation for specific algorithms (or problem instances)
- Logical level

Qermit can do that!
qerm.it

QERMIT in a nutshell

- QERMIT - an open-source python software package for design and execution of digital error mitigation
- Compositional approach – graph-based software design
- Embedded into TKET SDK compiler – runs with Pytket backend and supports most hardware providers

```
pip install qermit
```

Documentation and examples → www.qerm.it.

Repository and manual → <https://github.com/CQCL/Qermit>.

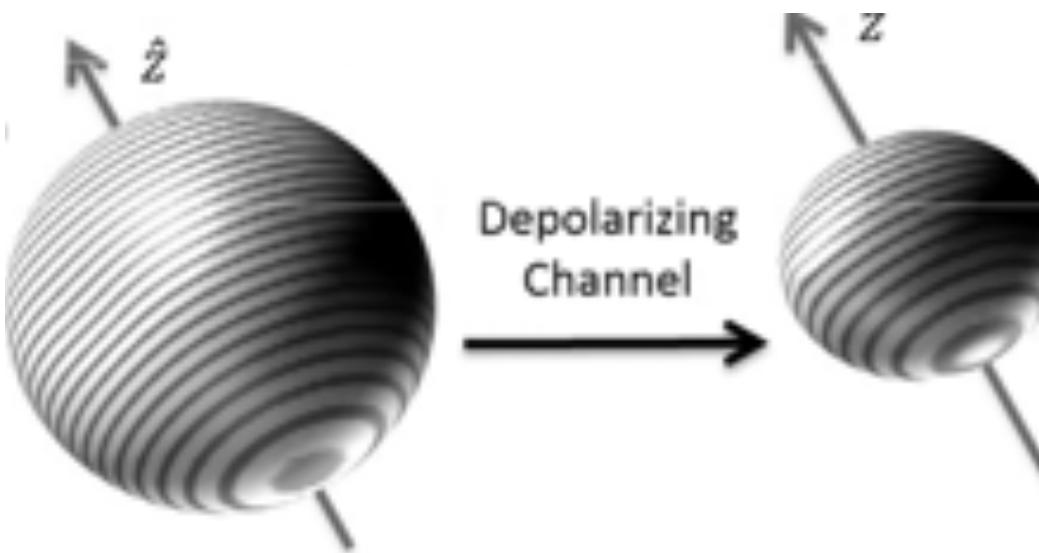
Noise models

Coherent vs Incoherent noise



$$\rho \rightarrow U\rho U^\dagger$$

Depolarisation Probabilistically mixes input state with the maximally mixed state

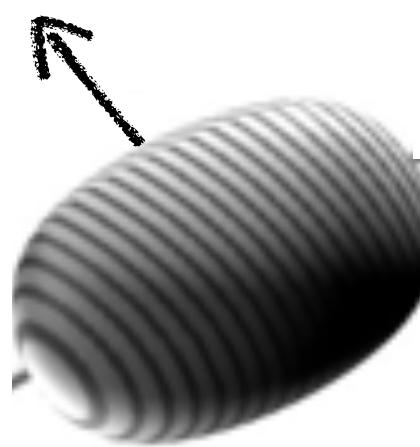


$$\rho \rightarrow (1-p)\rho + p\mathbb{I}/2$$

Bit-flip errors

$$|0\rangle \xrightarrow{p} |1\rangle$$

$$|1\rangle \xrightarrow{p} |0\rangle$$



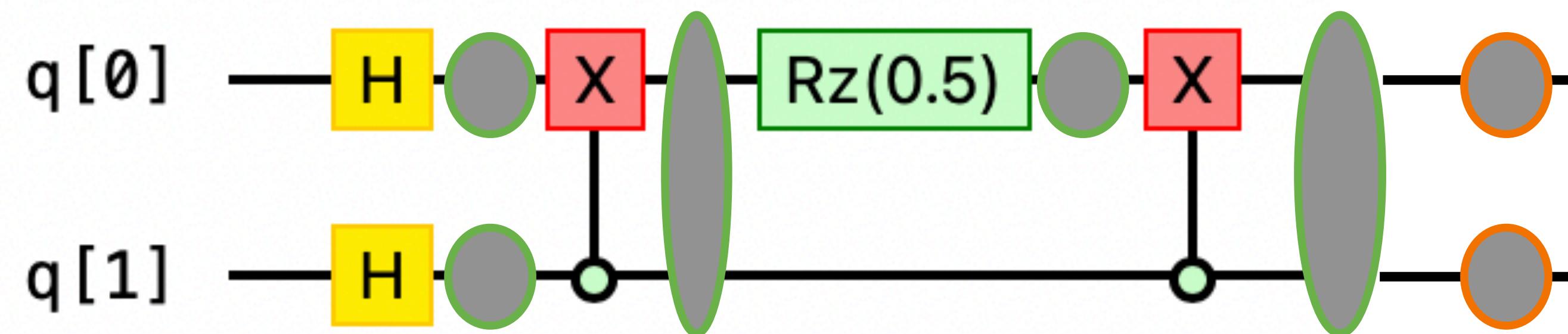
$$\rho \rightarrow (1-p)\rho + X\rho X$$

Noise models in practice – classical simulations of noisy hardware

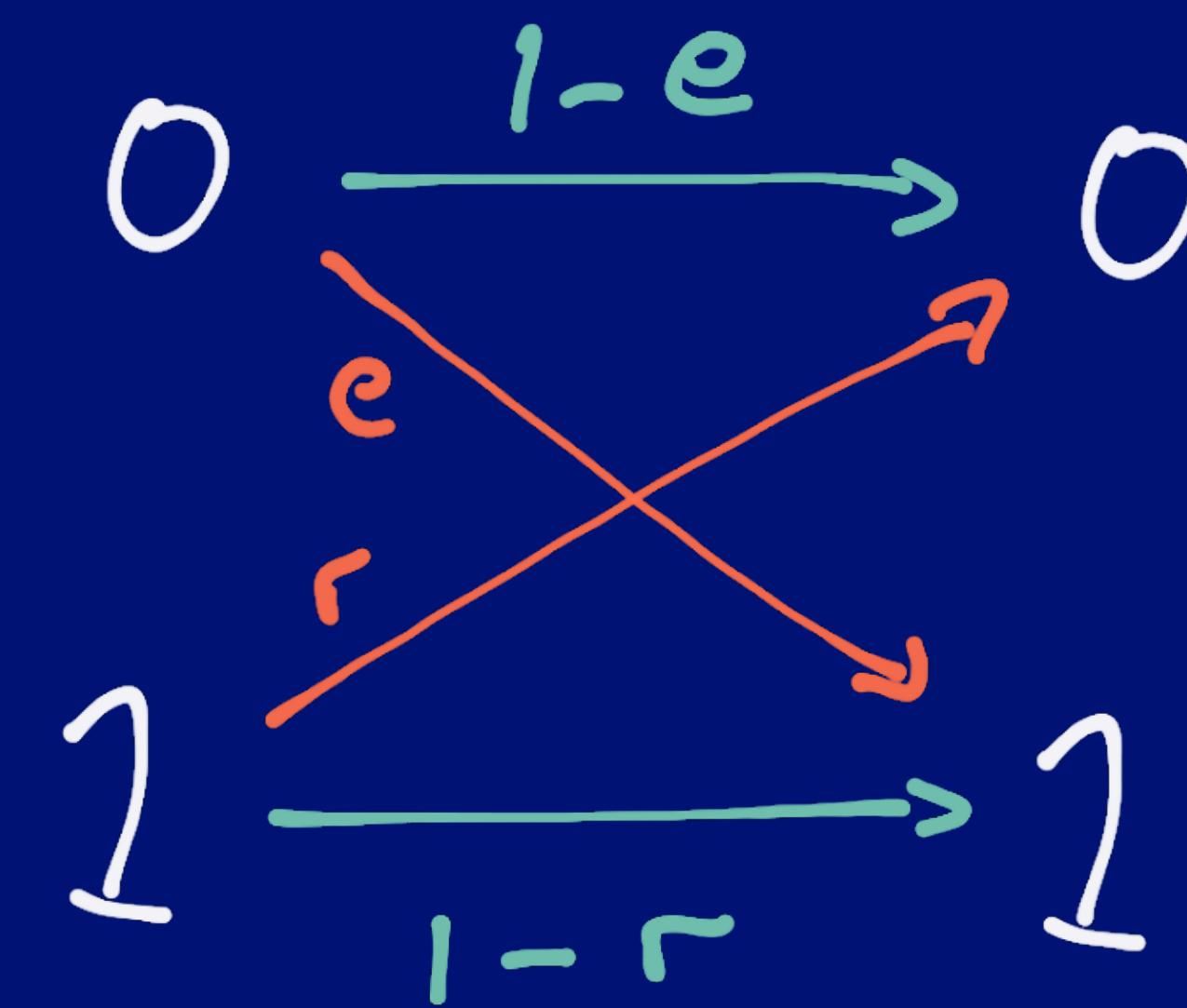
```
import qiskit.providers.aer.noise as noise
from pytket.extensions.qiskit import AerBackend
```

```
noise_model = depolarizing_noise_model(n_qubits, prob_2*0.1, prob_2, 3*prob_2)
noisy_backend = AerBackend(noise_model)
```

Pytket supports noisy simulators as backends



State Preparation and Measurement error mitigation



$$P_{\text{noise}} = N P_{\text{ideal}}$$

$$N = \begin{pmatrix} 1-e & r \\ e & 1-r \end{pmatrix}$$

State Preparation and Measurement error mitigation

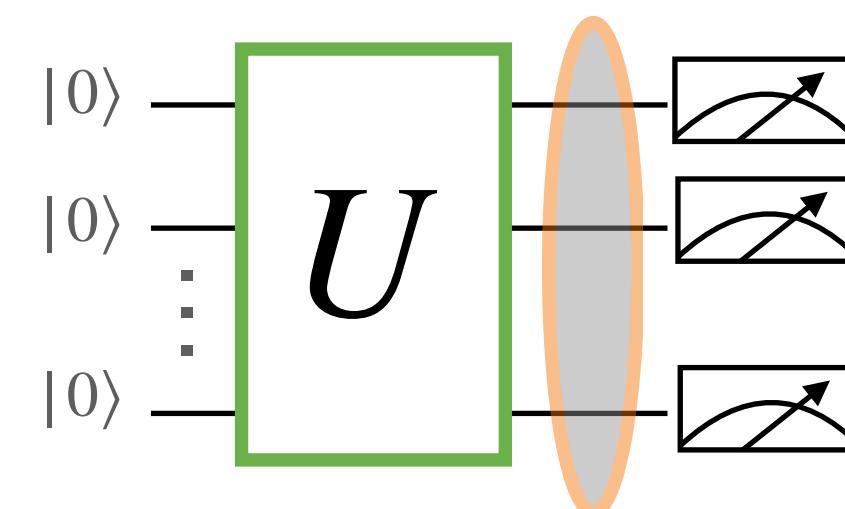
- Reduce the effect of SPAM on the outcome sample probability distribution

- Detector tomography

Construct process matrix S of transition probabilities

$$S_{zz'} = \text{prob}(z | z')$$

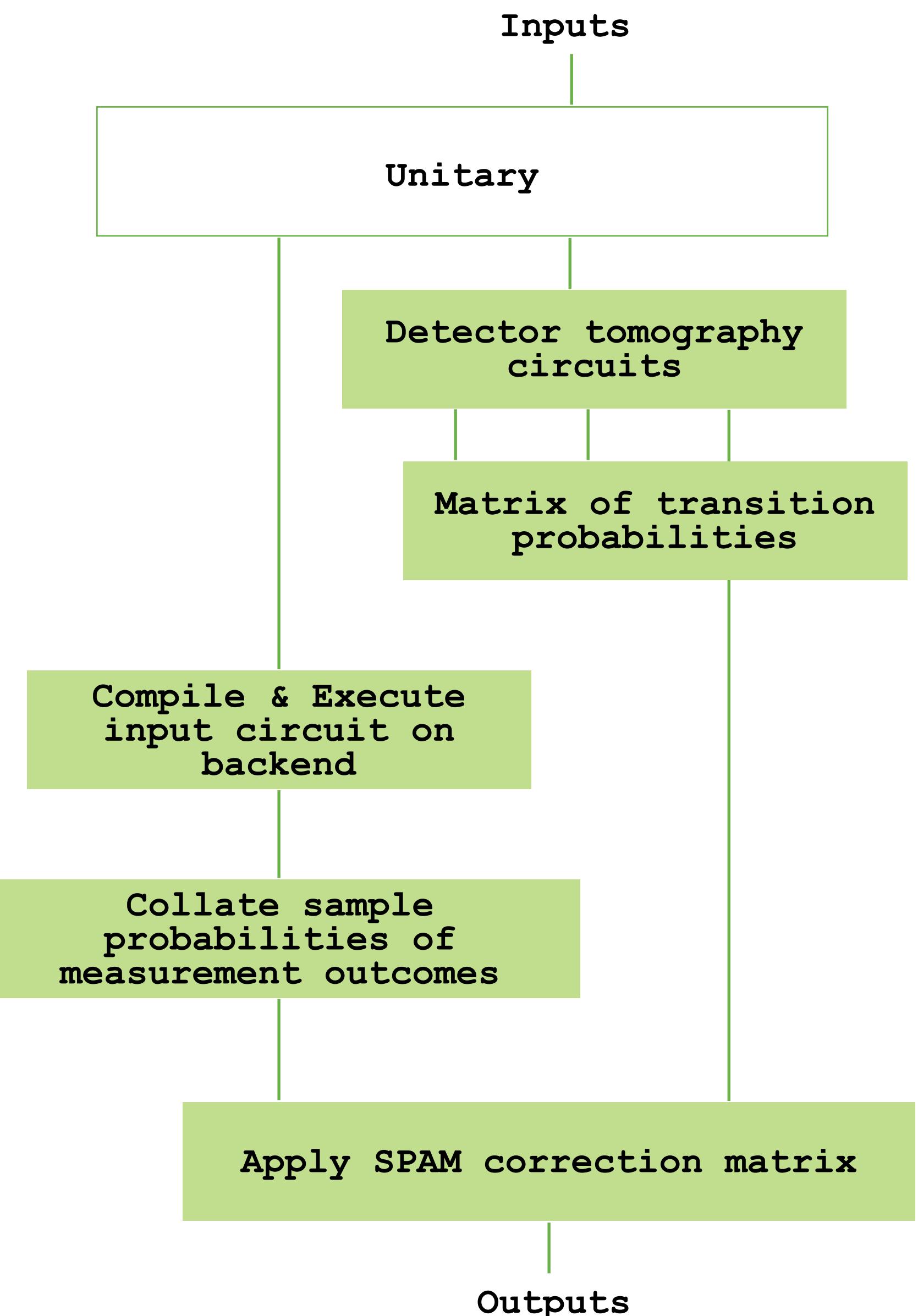
z - measurement outcome n bit-string



$$\mathbf{p}_{\text{noisy}} = p(z)$$

Error mitigated probability

$$\mathbf{p}_{EM} = S^{-1} \mathbf{p}_{\text{noisy}}$$



State Preparation and Measurement error mitigation

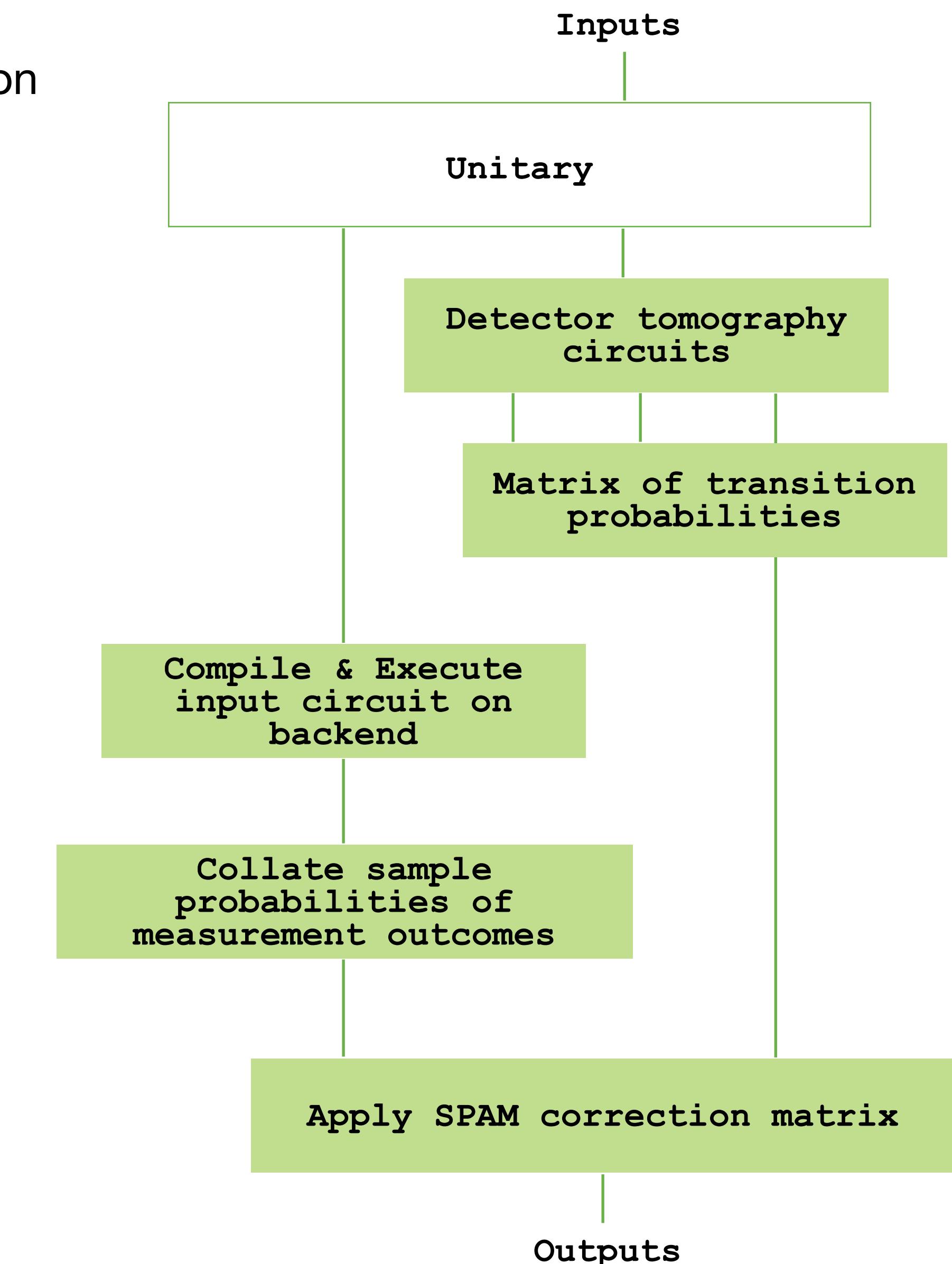
- Reduce the effect of SPAM on the outcome sample probability distribution

Full detector tomography – unscalable



Alternative approaches:

- Restrict locality
- Work with reduced noisy subspace of possible outcomes
- Overlapping tomography



Expectation value of observables — workflow

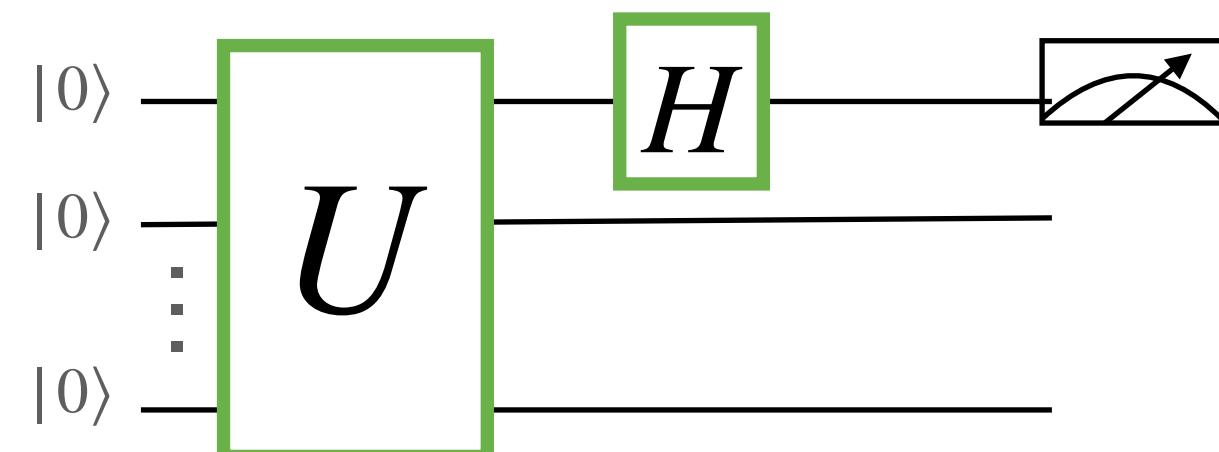
Input: Observable O , unitary $U = U_D \dots U_1$

Goal: Compute expectation $\langle O \rangle = \text{Tr}(O U_D \dots U_1 \rho_0 U_1^\dagger \dots U_D^\dagger)$

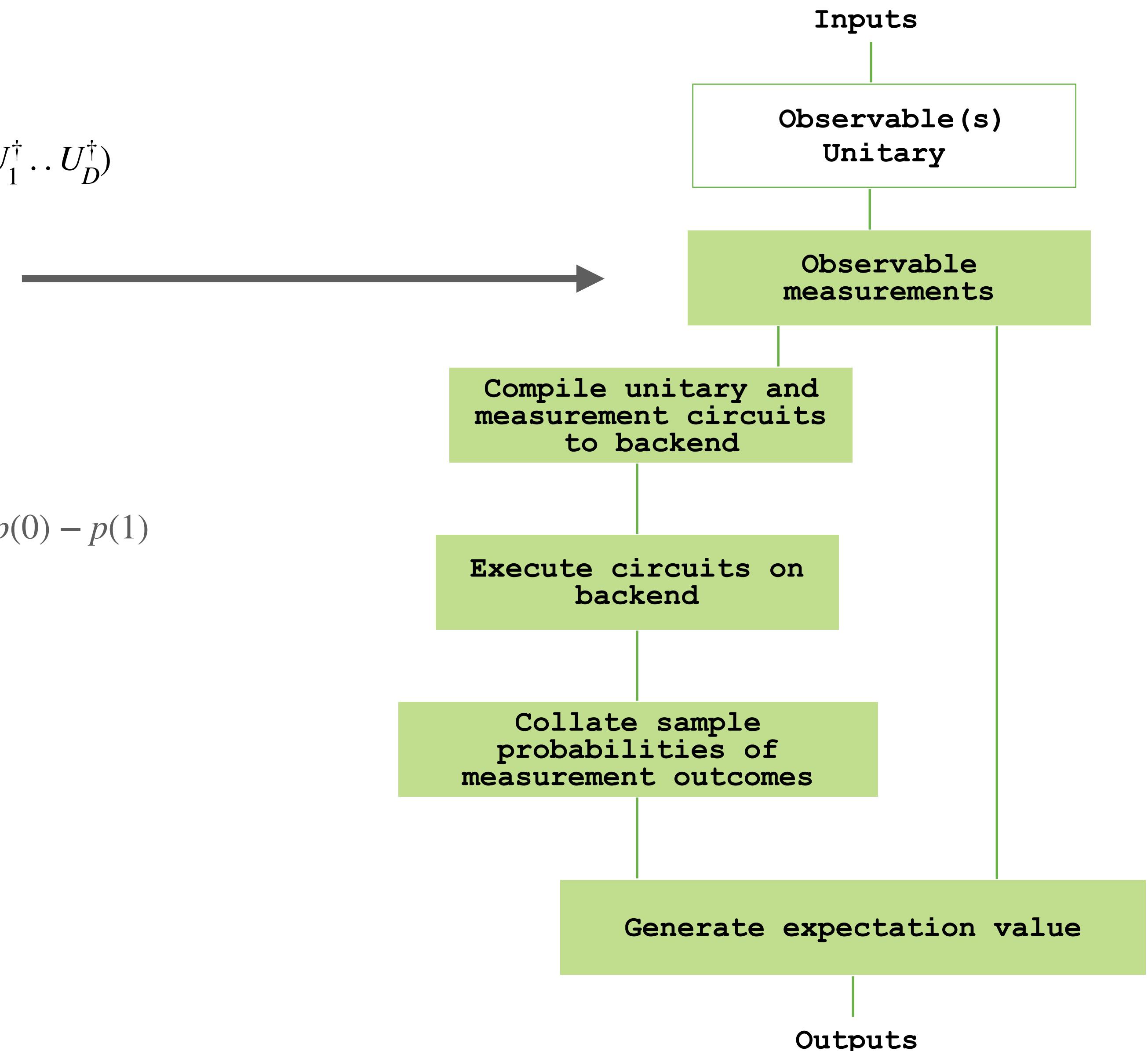
Decompose O into (projective) operator measurements

$$O = \sum o_R(z) P_z^R$$

Example: $X = |+\rangle\langle +| - |- \rangle\langle -|$



$$\langle X \rangle = p(0) - p(1)$$

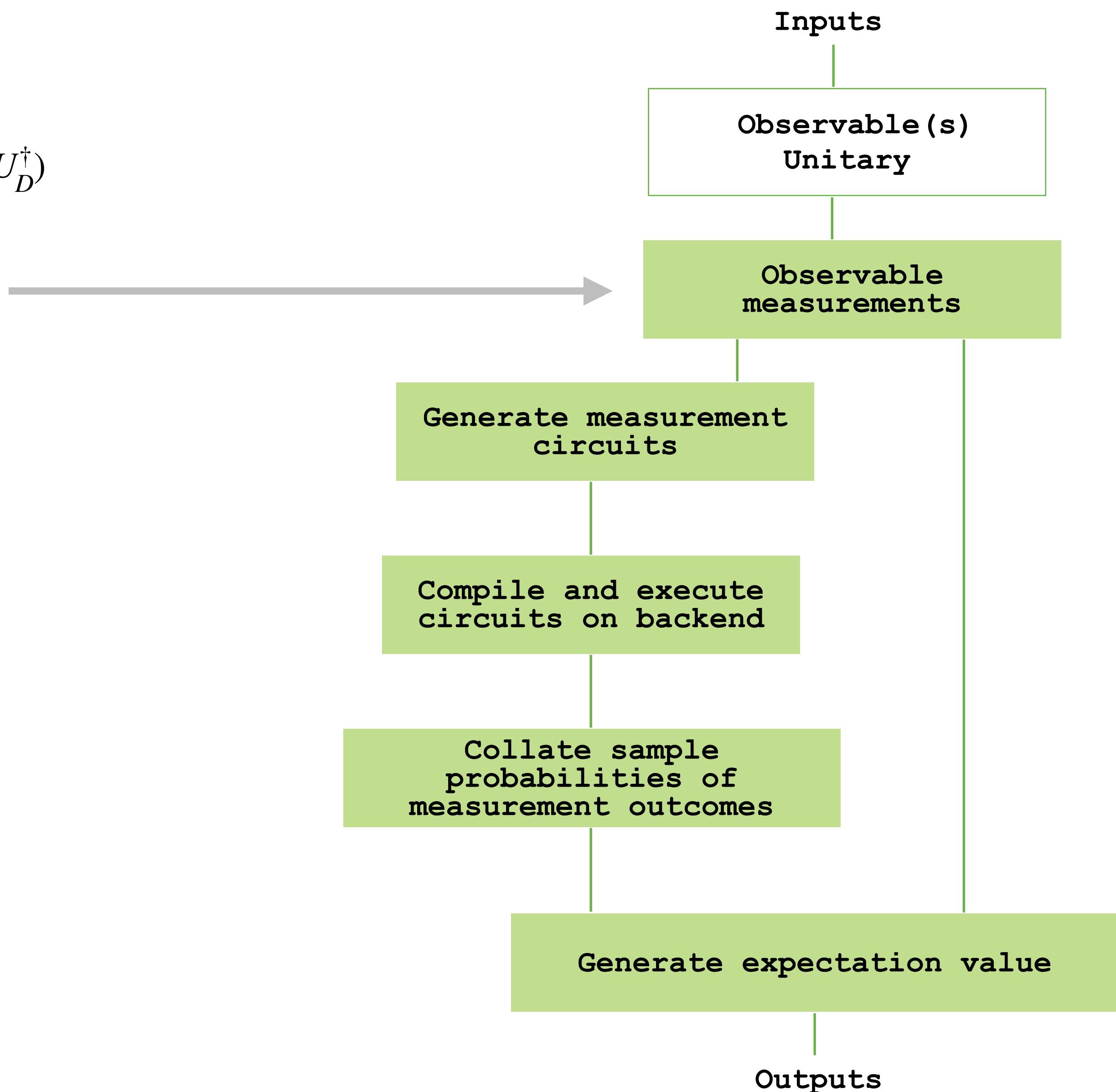


Expectation value of observables — workflow

Input: Observable O , unitary $U = U_D \dots U_1$

Goal: Compute expectation $\langle O \rangle = \text{Tr}(O U_D \dots U_1 \rho_0 U_1^\dagger \dots U_D^\dagger)$

Decompose O into (projective) operator measurements



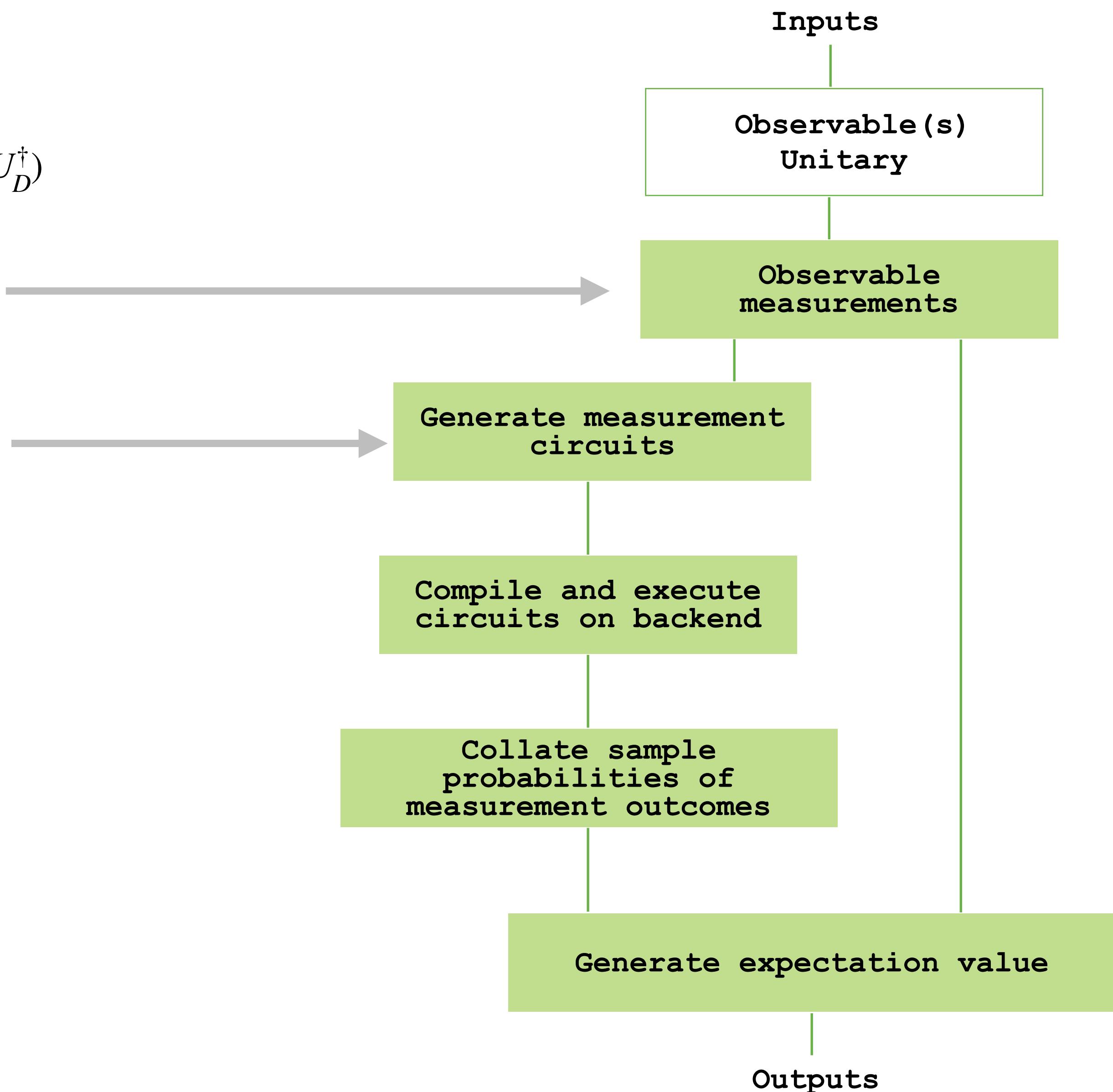
Expectation value of observables — workflow

Input: Observable O , unitary $U = U_D \dots U_1$

Goal: Compute expectation $\langle O \rangle = \text{Tr}(O U_D \dots U_1 \rho_0 U_1^\dagger \dots U_D^\dagger)$

Decompose O into (projective) operator measurements

Appends rotation gates and measurement in computational basis



Expectation value of observables — workflow

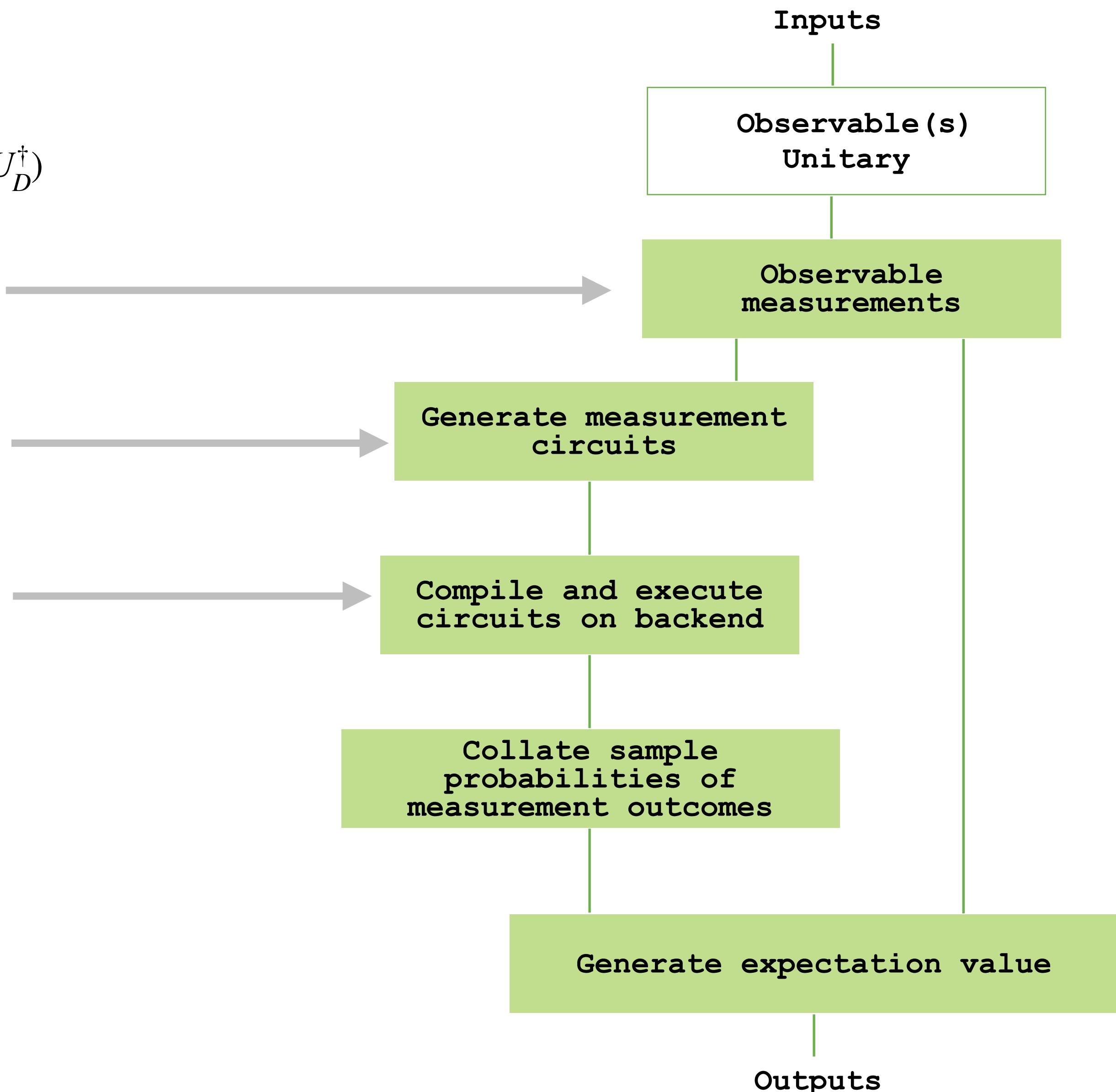
Input: Observable O , unitary $U = U_D \dots U_1$

Goal: Compute expectation $\langle O \rangle = \text{Tr}(O U_D \dots U_1 \rho_0 U_1^\dagger \dots U_D^\dagger)$

Decompose O into (projective) operator measurements

Appends rotation gates and measurement in computational basis

Compiles circuits to specific gate-set/topology and run
 n_{shots} independent measurement trials



Expectation value of observables — workflow

Input: Observable O , unitary $U = U_D \dots U_1$

Goal: Compute expectation $\langle O \rangle = \text{Tr}(O U_D \dots U_1 \rho_0 U_1^\dagger \dots U_D^\dagger)$

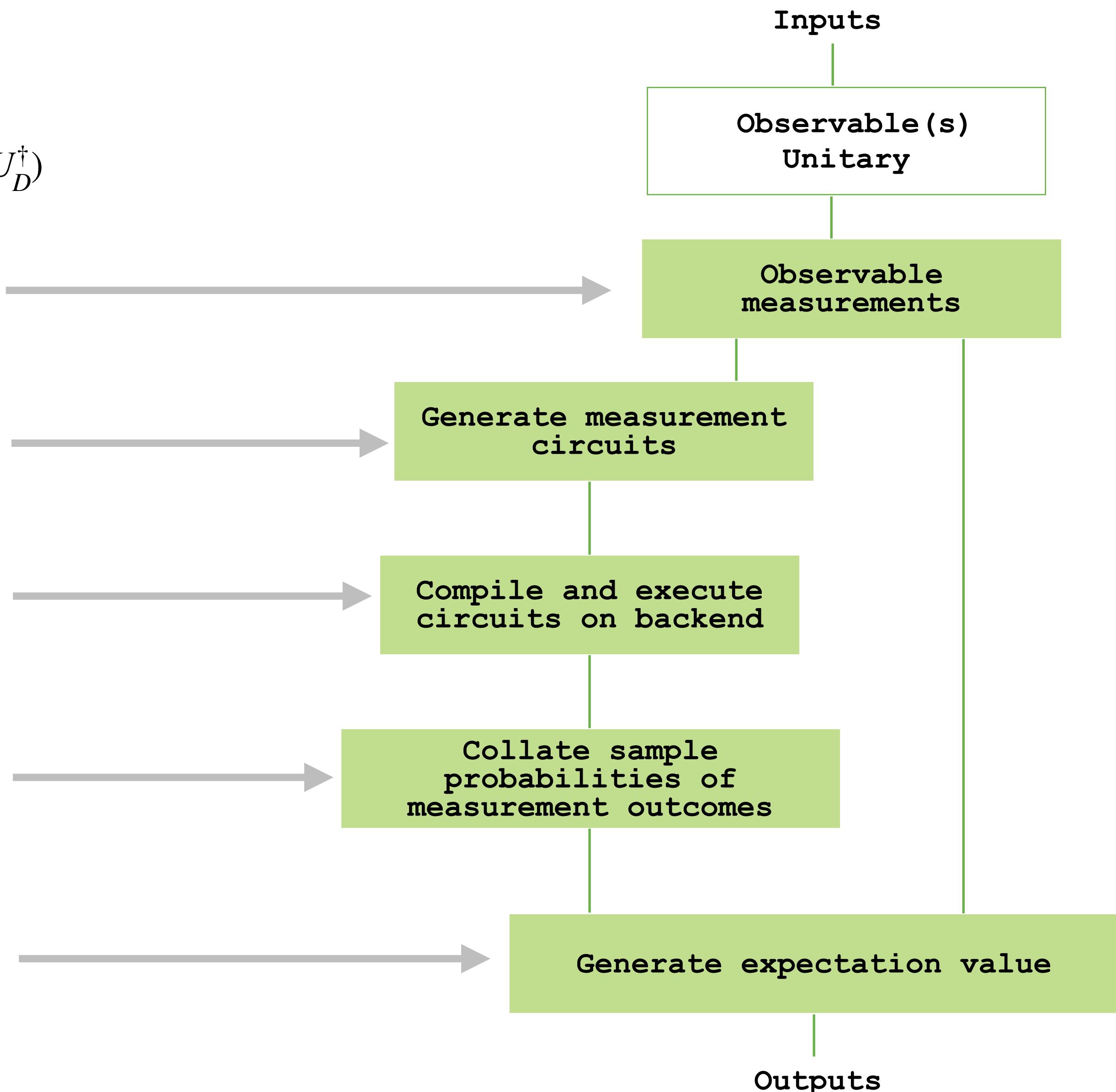
Decompose O into (projective) operator measurements

Appends rotation gates and measurement in computational basis

Compiles circuits to specific gate-set/topology and run
 n_{shots} independent measurement trials

Obtain sample probabilities $p^R(z)$ for each measurement
outcome z and setting R

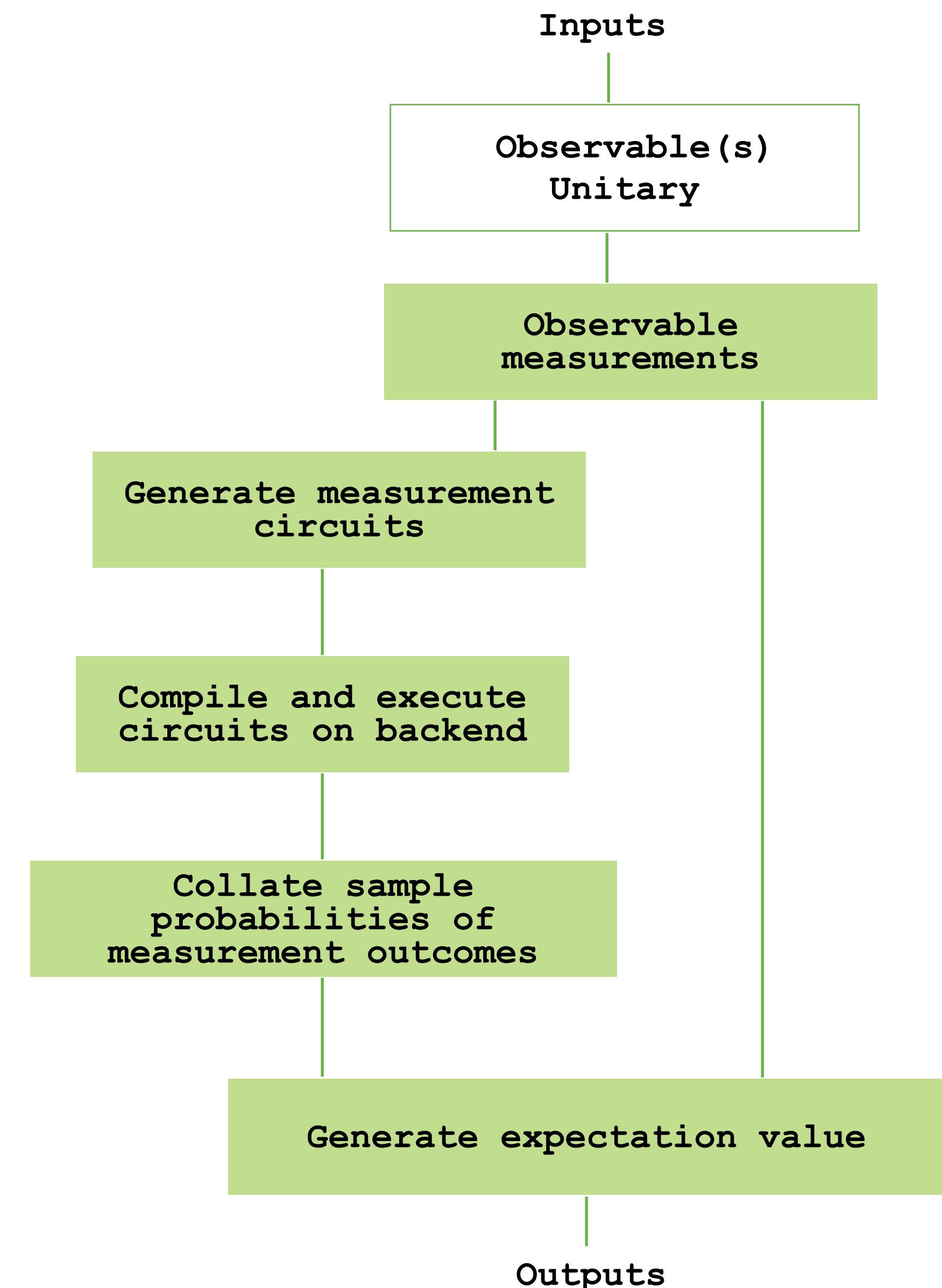
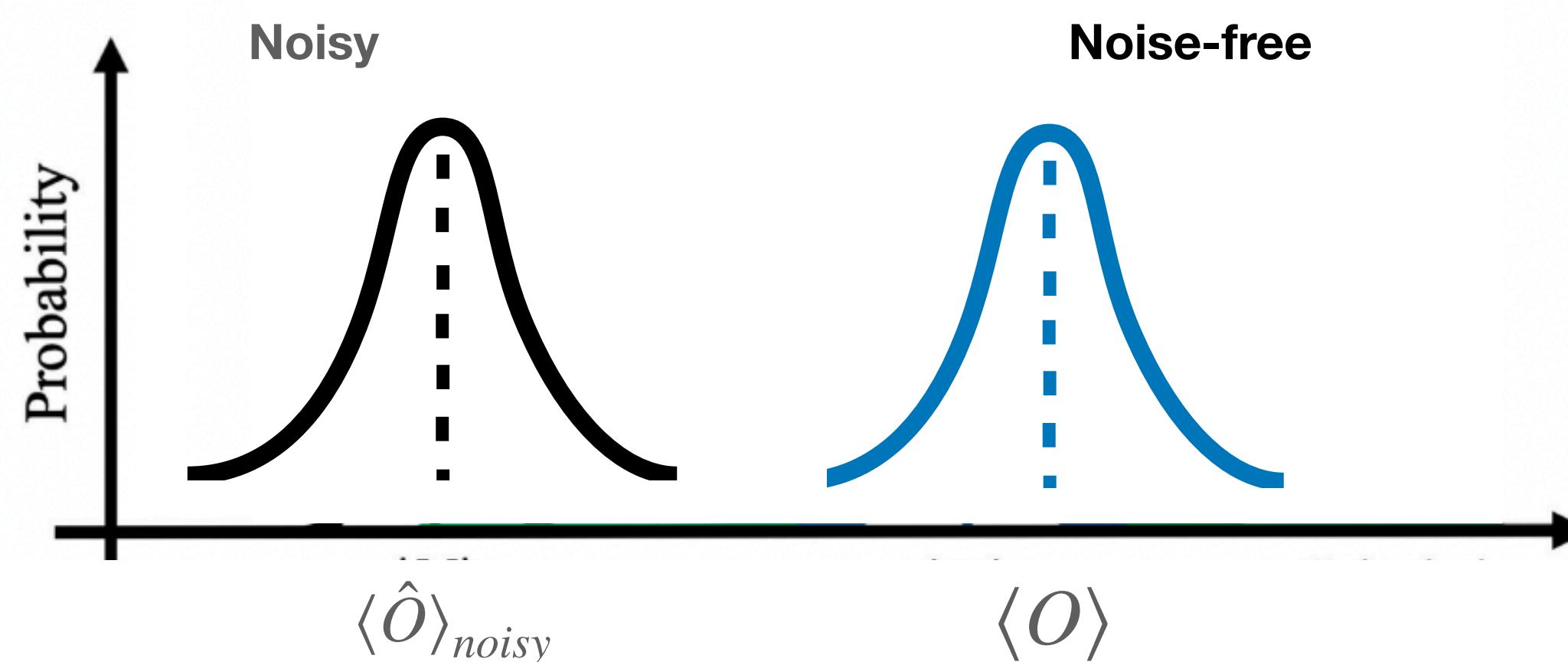
Produce sample mean estimator $\langle \hat{O} \rangle = \sum_{R,z} o_R(z)p^R(z)$



Noisy expectation value of observables – finite sampling

Input: Observable O , unitary $U = U_D \dots U_1$

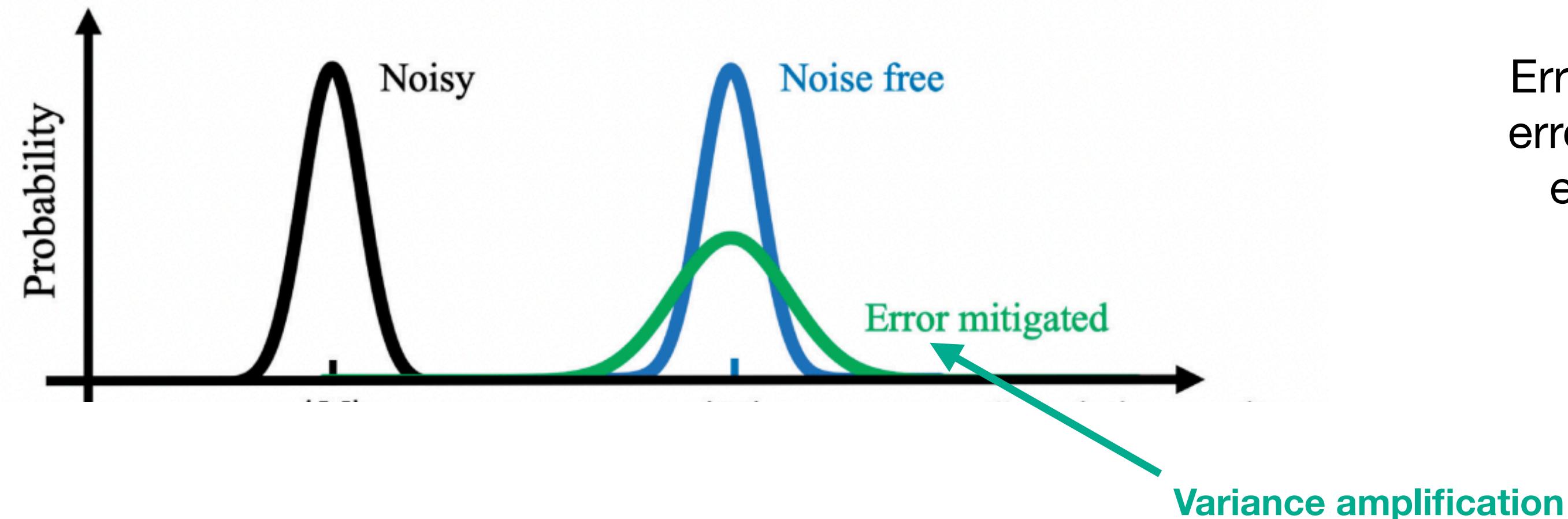
Goal: Compute expectation $\langle O \rangle = \text{Tr}(O U_D \dots U_1 \rho_0 U_1^\dagger \dots U_D^\dagger)$



Error mitigation for observables

Input: Observable O , unitary $U = U_D \dots U_1$

Goal: Compute expectation $\langle O \rangle = \text{Tr}(O U_D \dots U_1 \rho_0 U_1^\dagger \dots U_D^\dagger)$



Error mitigation aims to improve approximation error in the noisy sample mean estimator at the expense of increased sampling complexity.

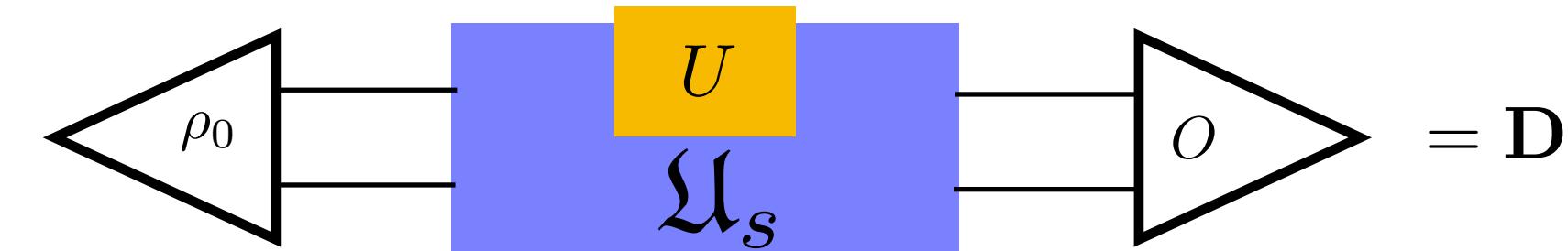
Error mitigation for observables – A modular general framework

Input: Observable O , unitary $U = U_D \dots U_1$

Output : Error-mitigated estimator $\langle \hat{O} \rangle_{EM}$

I. Noise characterisation

- Process $\mathfrak{N} = (\mathfrak{U}_0, \dots, \mathfrak{U}_S)$ produces a series of modified circuits



II. Functional model

- Motivated by noise models
- Describes relationship between measurements of modified circuit and target expectation value

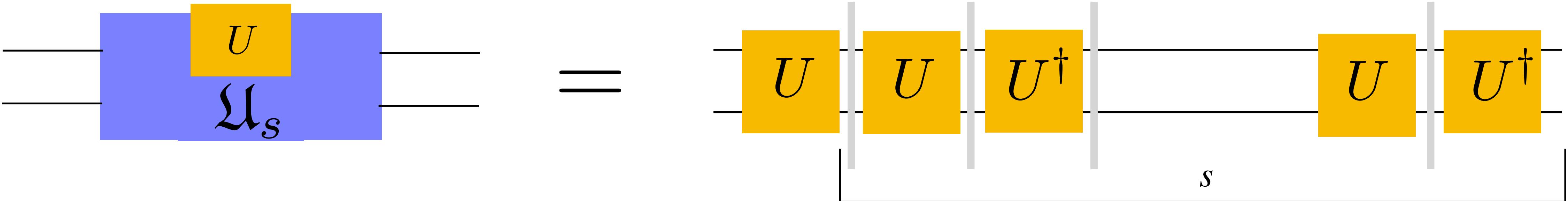
$$F(\mathbf{D}) = 0$$

III. Classical post-processing

- Determine $\langle \hat{O} \rangle_{EM}$ by fitting parameters from data collected

Zero Noise Extrapolation (ZNE)

I.



II. Functional Model

Polynomial fit $D_s = \sum_{i=0}^k F_i s^i$

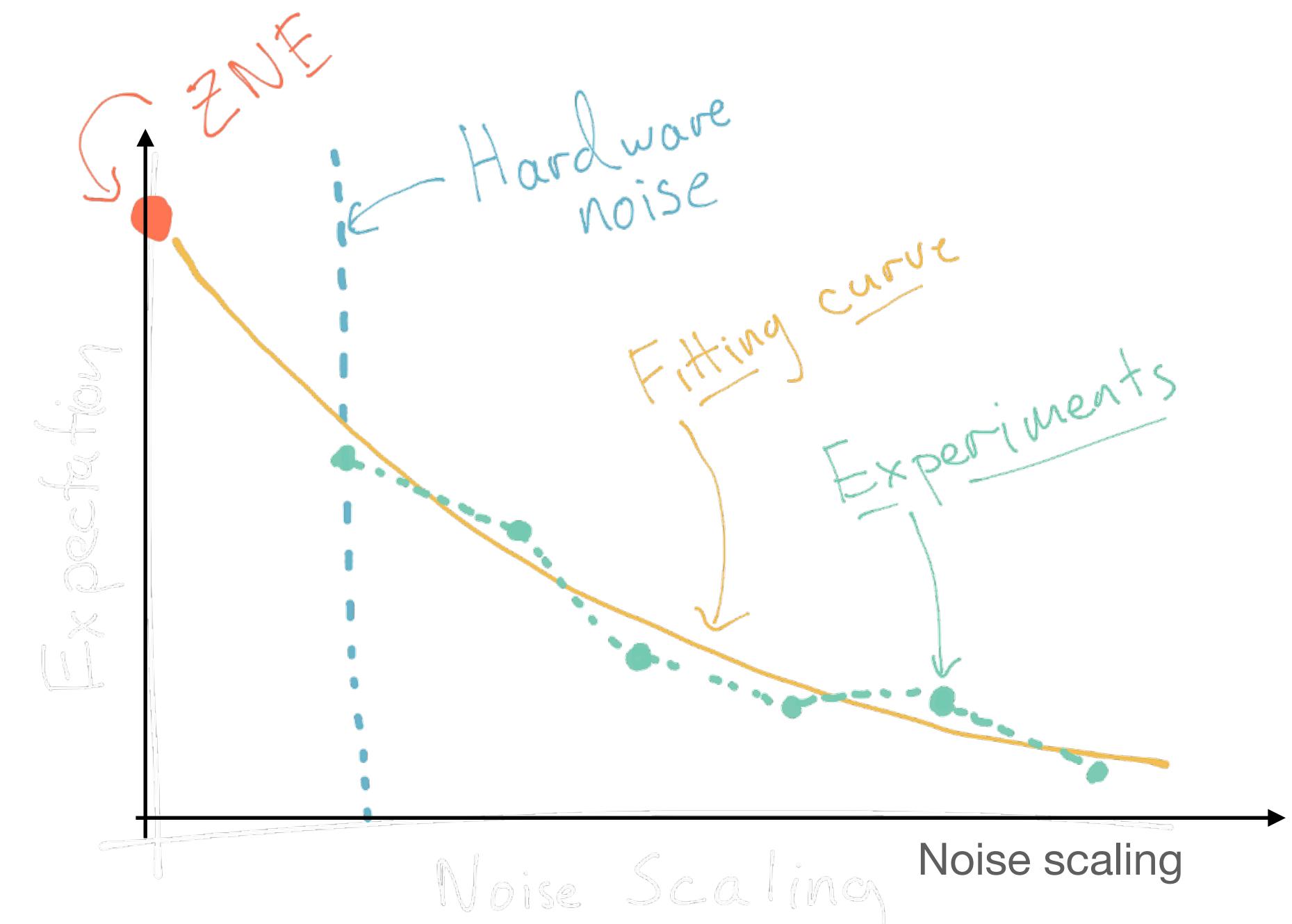
Exponential fit $D_s = F_0 e^{-sF_1}$

Global depolarising channel

$$\langle O \rangle_{\text{noisy}} = (1 - p)^s \langle O \rangle$$

III.

$$\langle \hat{O} \rangle_{EM} = \hat{F}_0$$

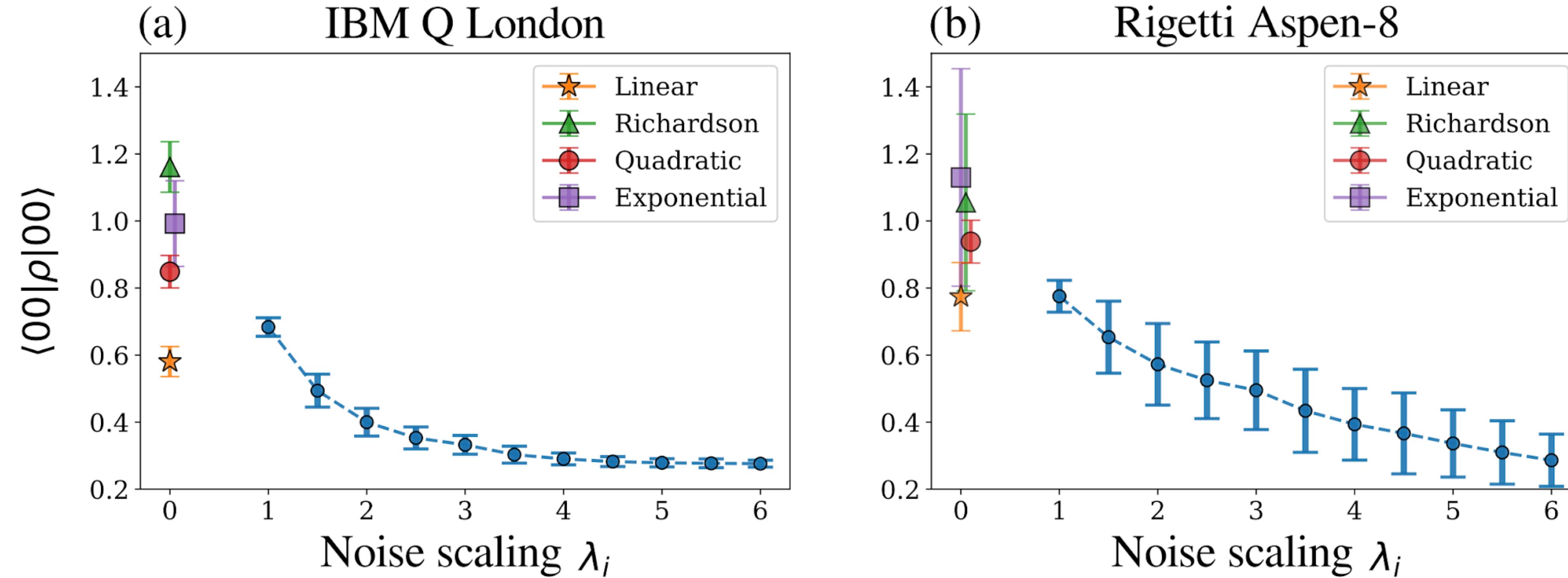


Kristan Temme, Sergey Bravyi, and Jay M Gambetta. "Error mitigation for short-depth quantum circuits". Physical review letters 119.18 (2017), p. 180509.

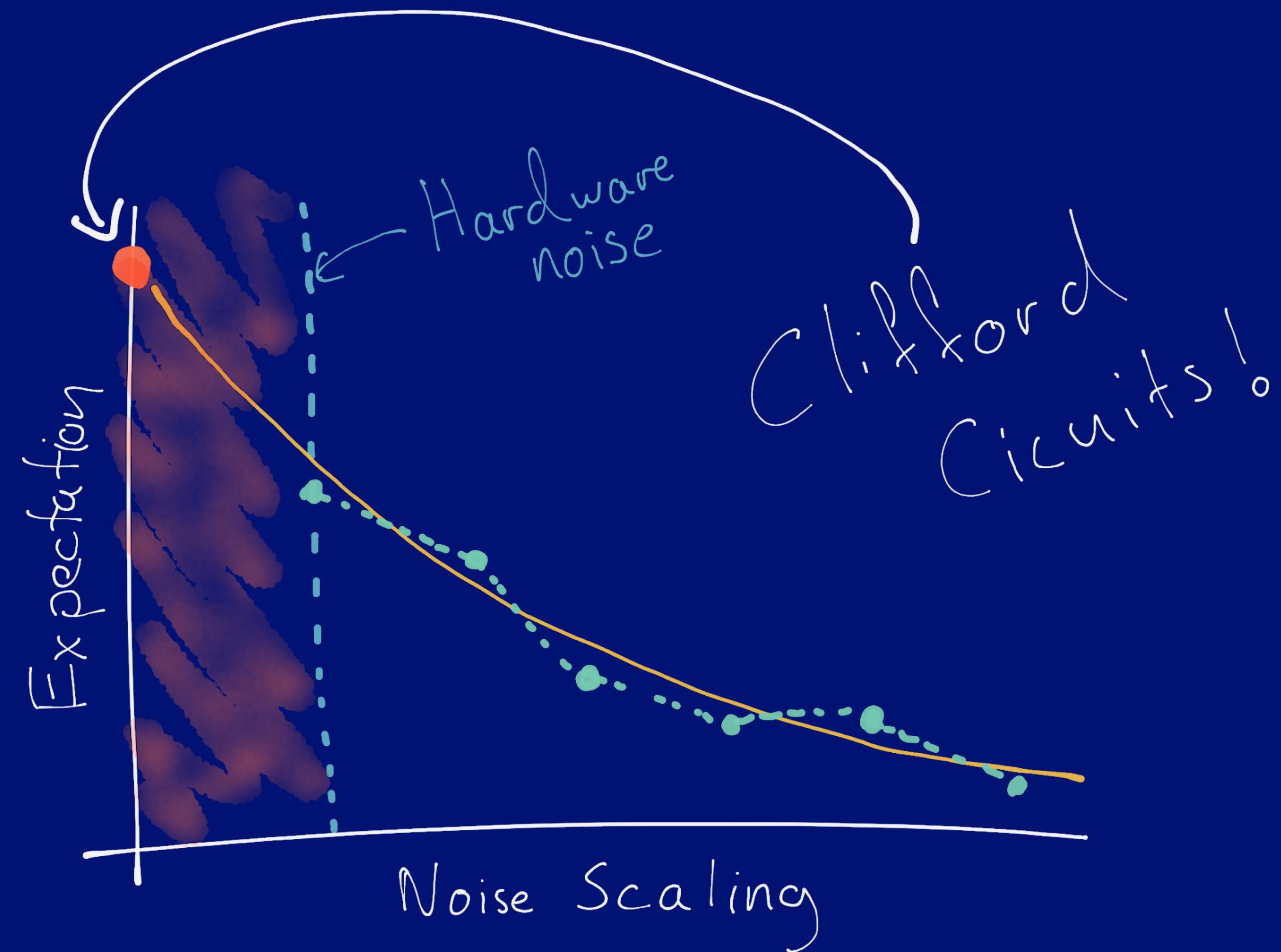
Li, Y., and S. C. Benjamin, Phys. Rev. X 7 (2), 021050. (2017)

Tudor Giurgica-Tiron et al. "Digital zero noise extrapolation for quantum error mitigation". In: arXiv preprint arXiv:2005.10921 (2020).

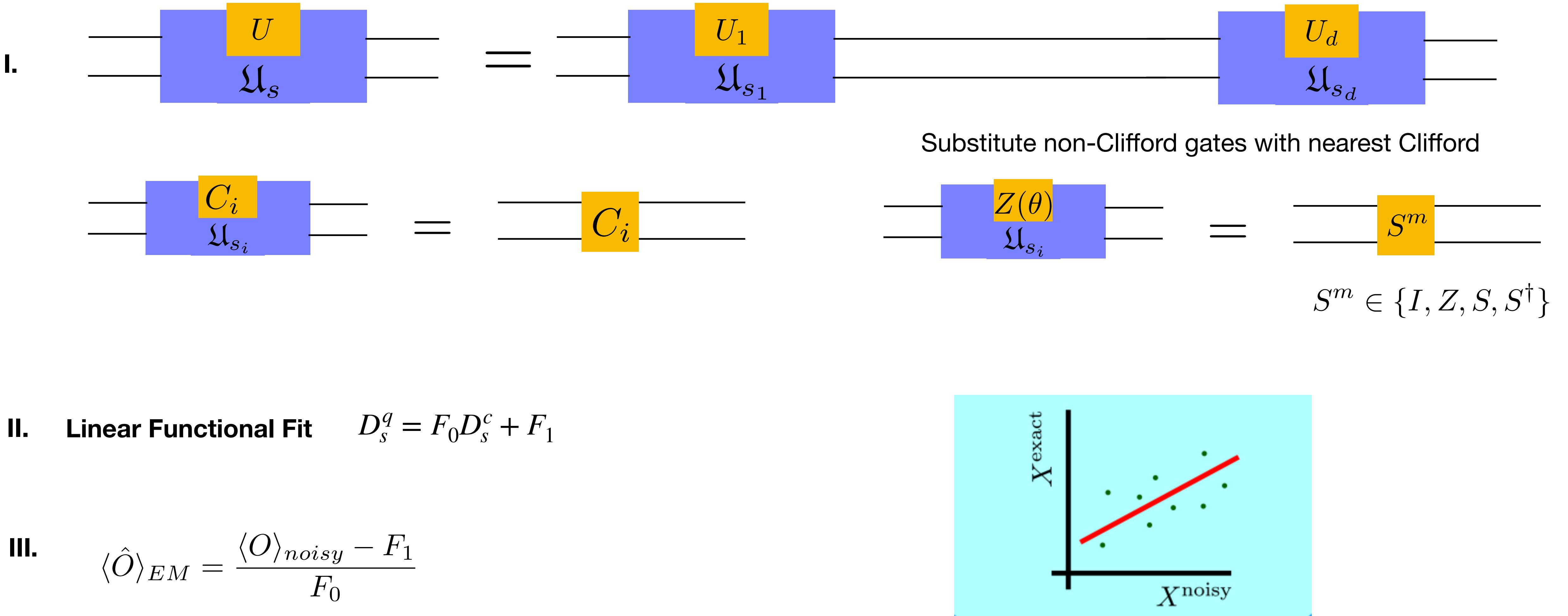
Zero Noise Extrapolation (ZNE)



Clifford Data Regression



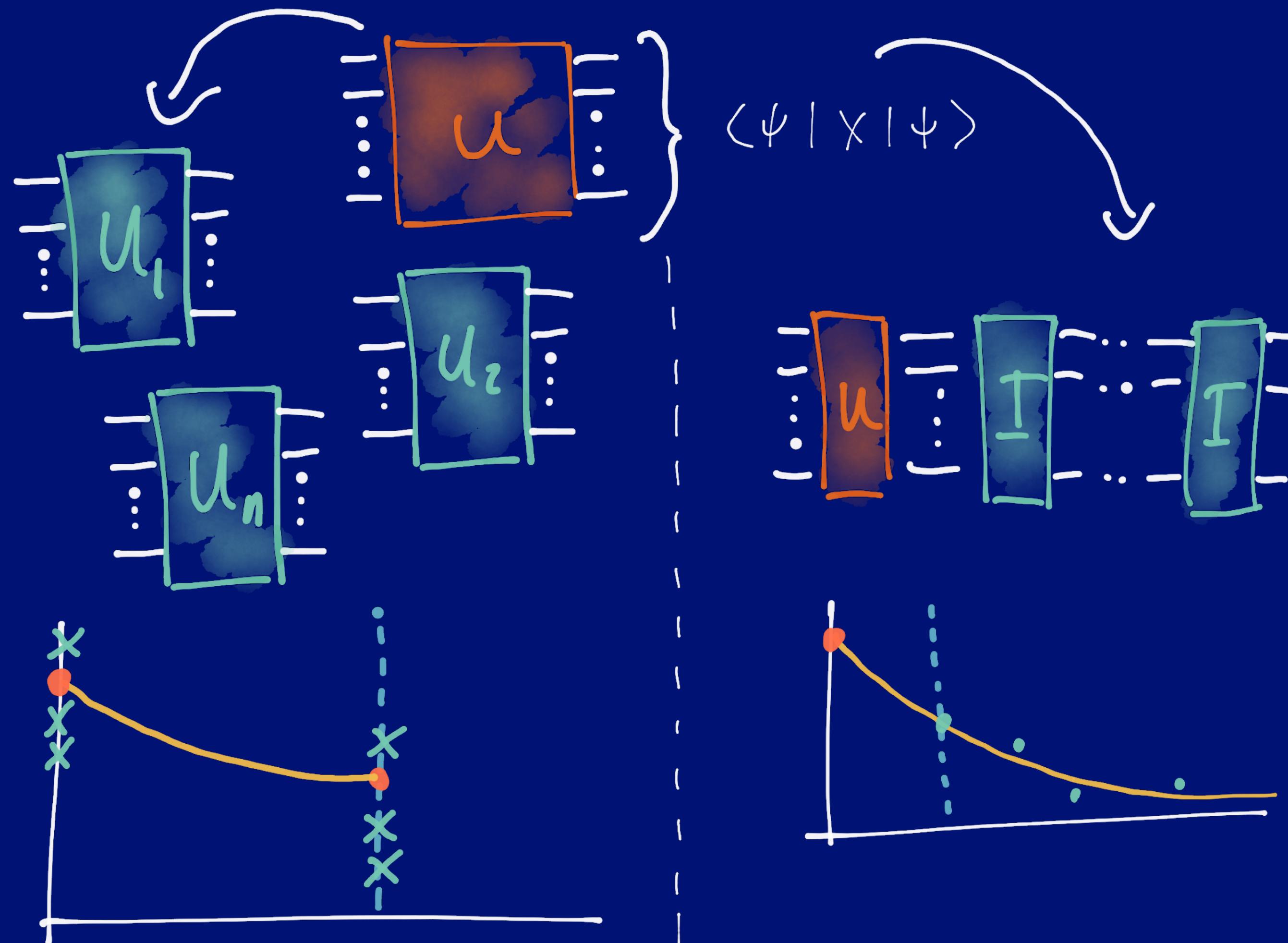
Clifford Data Regression



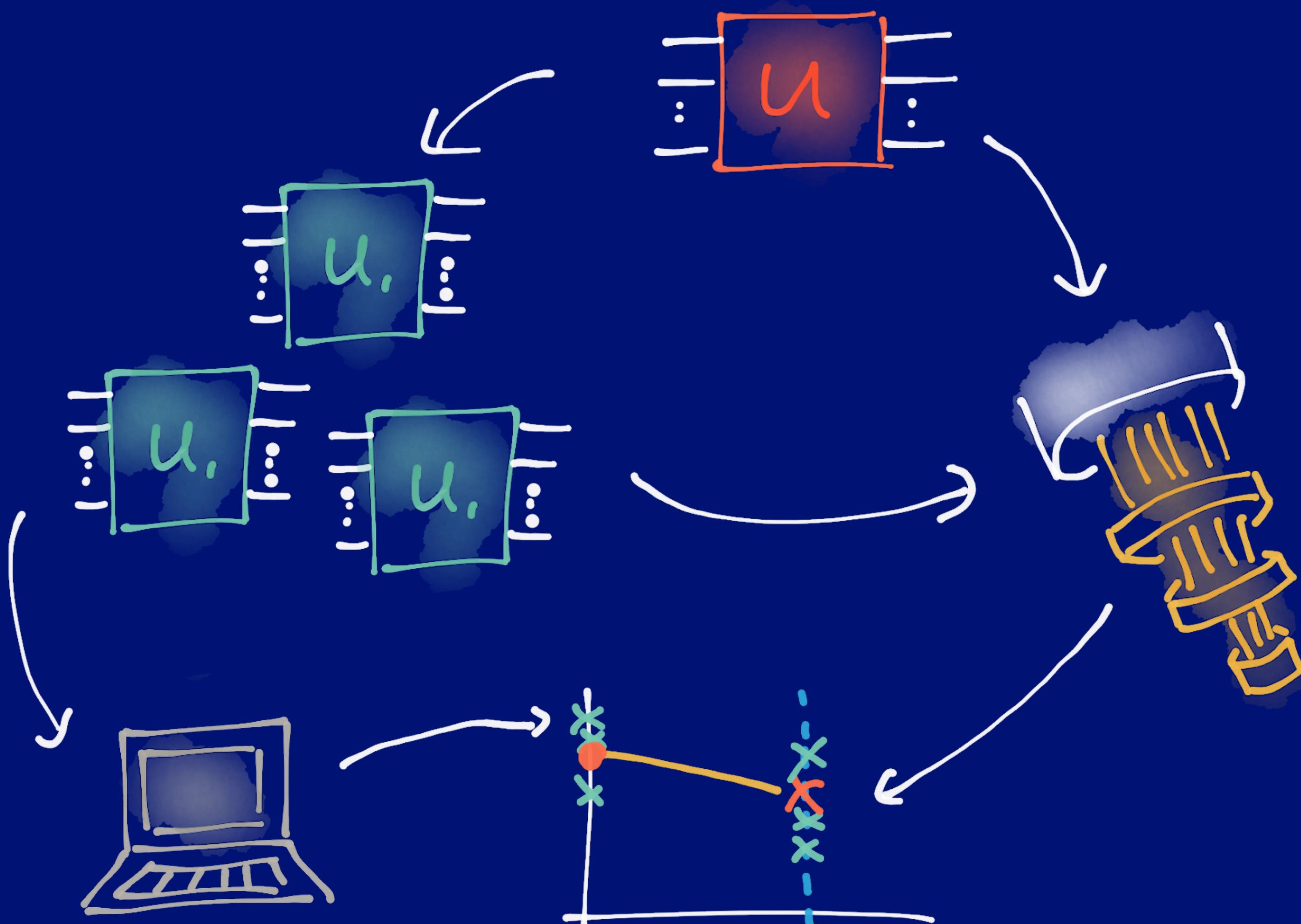
Czarnik, Piotr, et al. "Error mitigation with Clifford quantum-circuit data." *Quantum* 5 (2021): 592.

Lowe, Angus, et al. "Unified approach to data-driven quantum error mitigation." *Physical Review Research* 3.3 (2021): 033098.

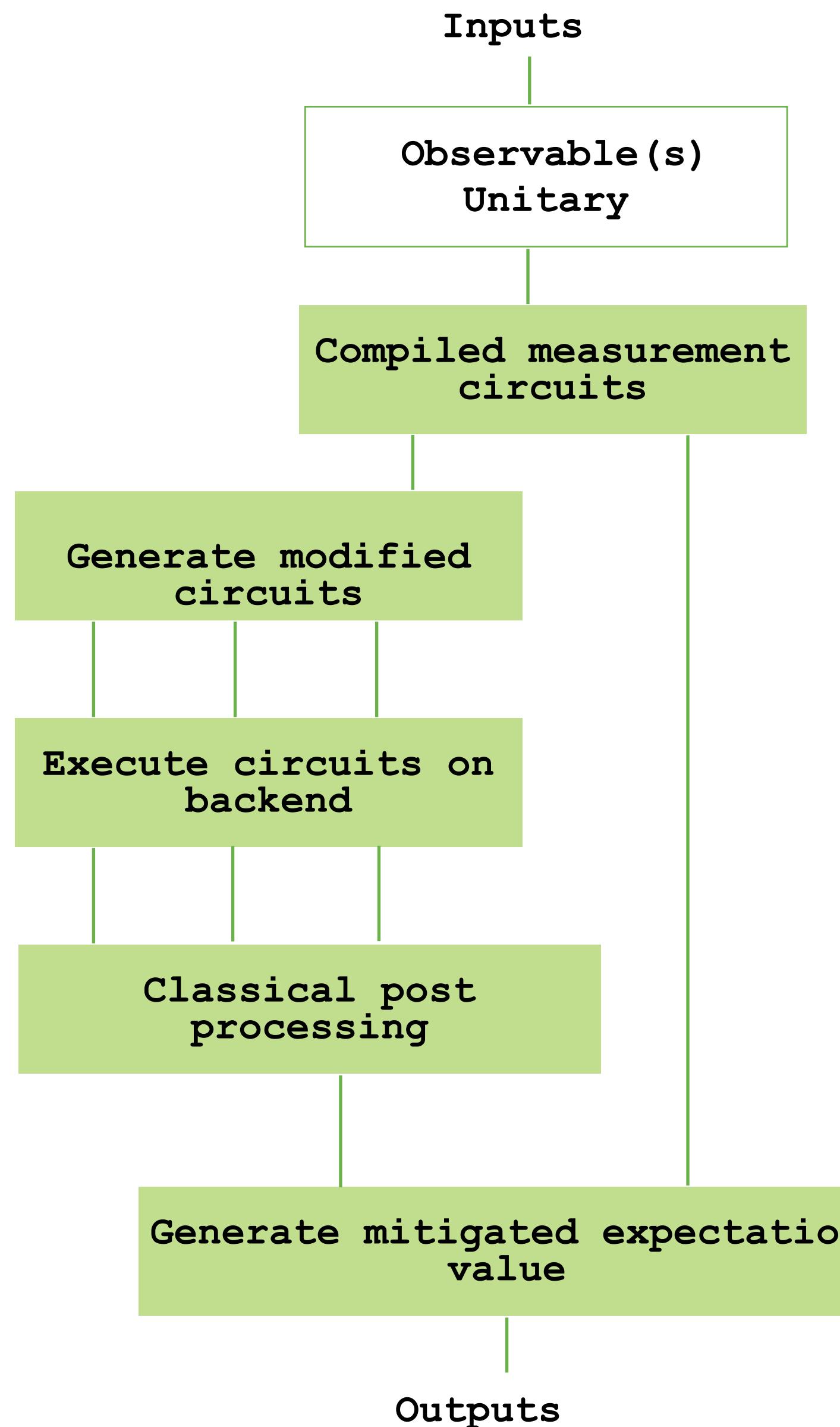
Error Mitigation Workflow



Error Mitigation Workflow



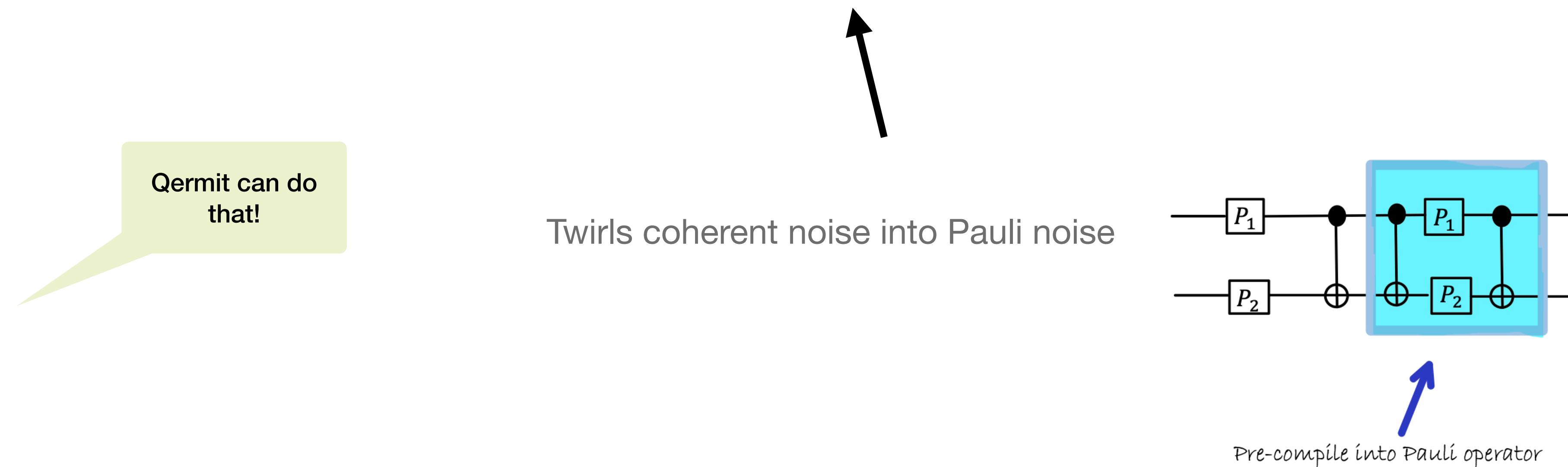
Error mitigation for observables – Experiment workflow



Effectively combining error mitigation strategies

- Noise profile should match the assumptions going into the error mitigation method
- Application-specific (post-selection) methods to enhance performance of error mitigation

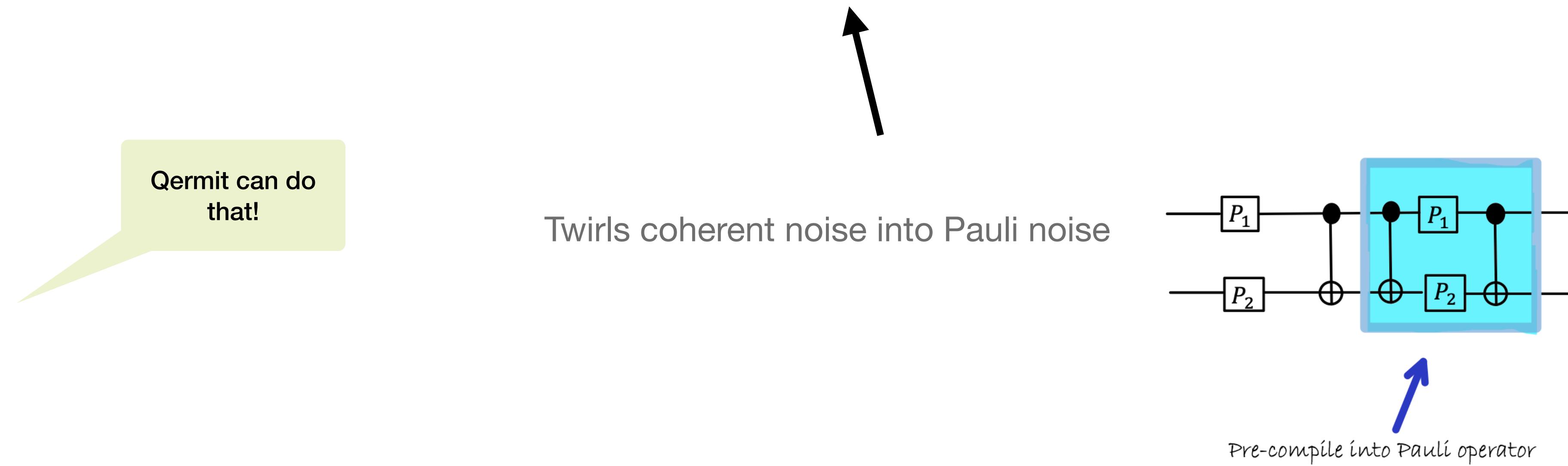
Combine Zero Noise Extrapolation + frame randomisation + spam correction



Effectively combining error mitigation strategies

- Noise profile should match the assumptions going into the error mitigation method
- Application-specific (post-selection) methods to enhance performance of error mitigation

Combine Zero Noise Extrapolation + frame randomisation + spam correction

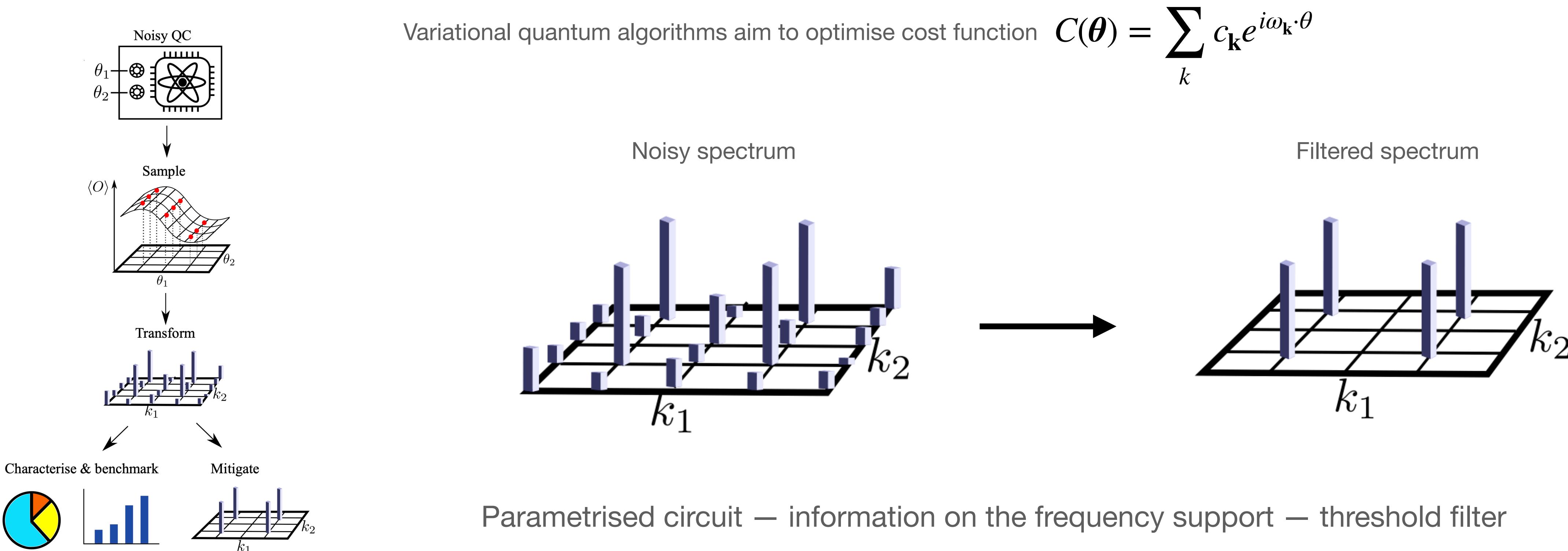


- Youngseok, K., Wood, C.J., Yoder, T.J., Merkel, S.T., Gambetta J.M , Temme K, & Kandala A. "Scalable error mitigation for noisy quantum circuits produces competitive expectation values." *arXiv:2108.09197* (2021).
- Cade, C., Mineh, L., Montanaro, A., & Stanisic, S. (2020). Strategies for solving the Fermi-Hubbard model on near-term quantum computers. *Physical Review B*, 102(23),
- Fontana, E., Rungger, I., Duncan, R., Cirstoiu, C. "Spectral analysis for noise diagnostics and filter-based digital error mitigation." *arXiv preprint arXiv:2206.08811* (2022).
- Yamamoto, K., Manrique, D. Z., Khan, I. T., Sawada, H., & Ramo, D. M. (2022). Quantum hardware calculations of periodic systems with partition-measurement symmetry verification: Simplified models of hydrogen chain and iron crystals. *Physical Review Research*, 4(3), 033110.

Effectively combining error mitigation strategies

Spectral filtering (for parametrised circuits)

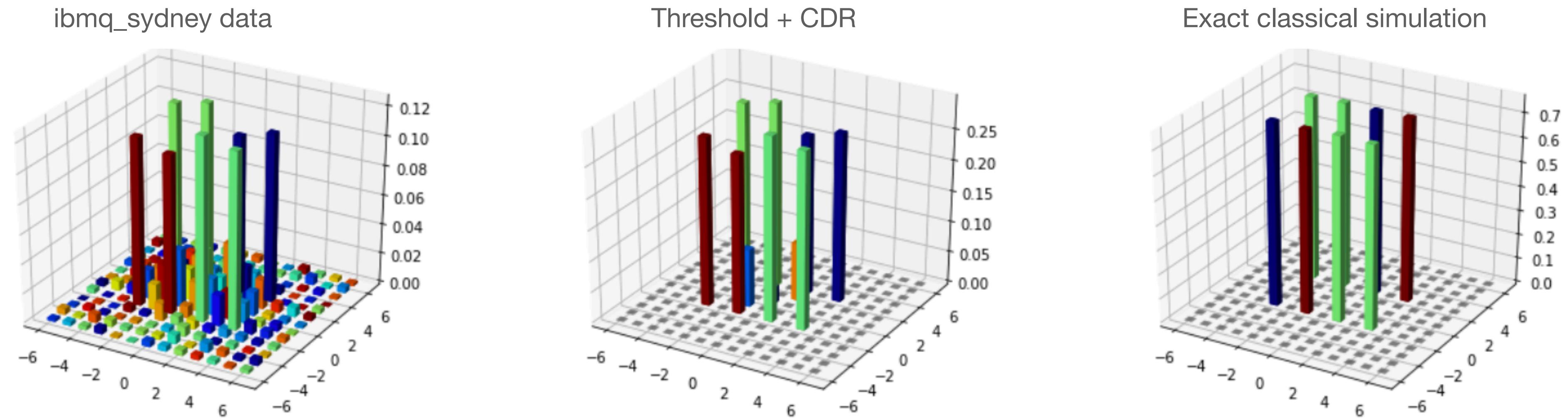
Apply signal processing to circuit families parametrised by continuous variables.



Effectively combining error mitigation strategies

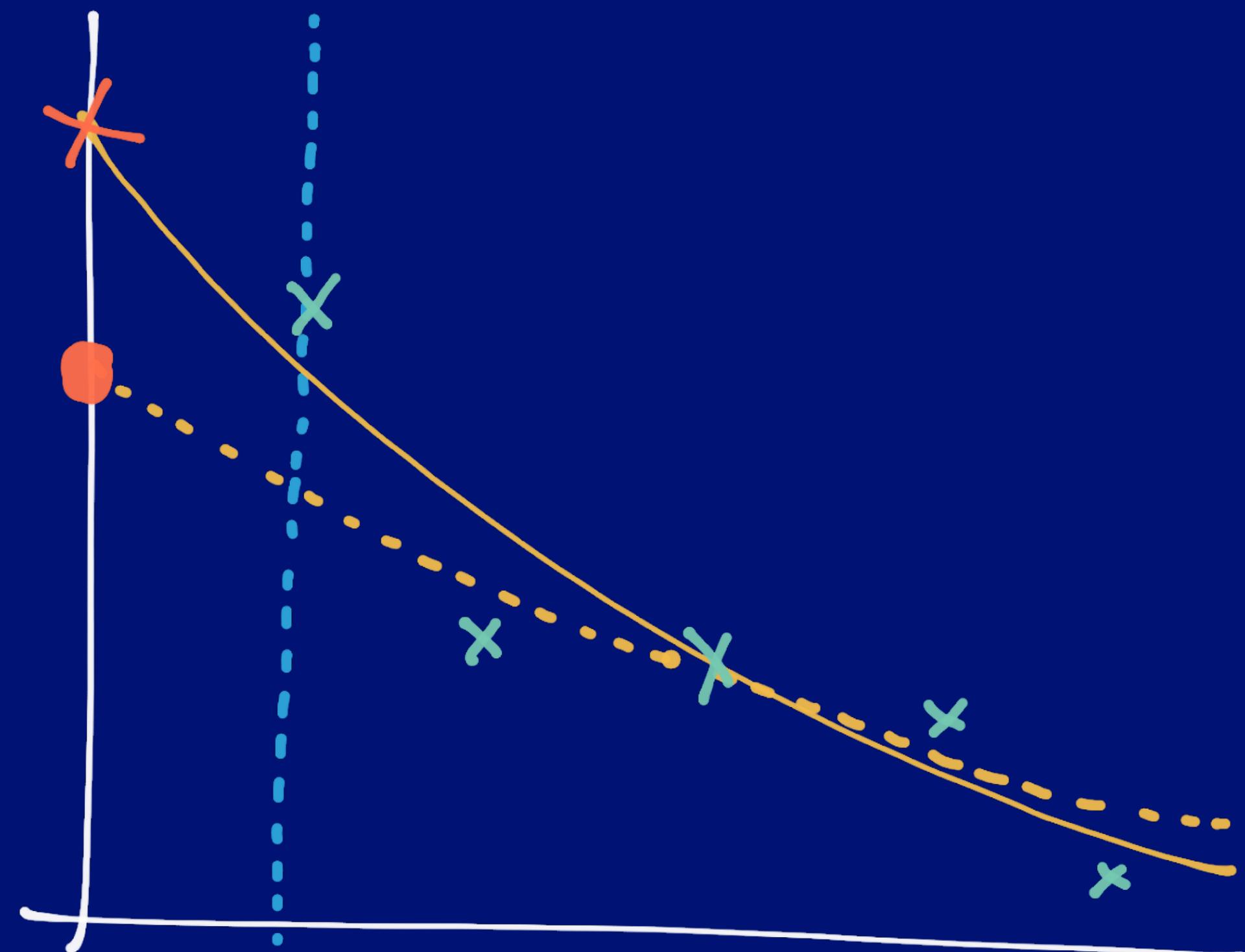
Spectral filtering (for parametrised circuits)

QAOA (1 layer) circuit with 8 qubits, 3-regular graph on IBMq_sydney



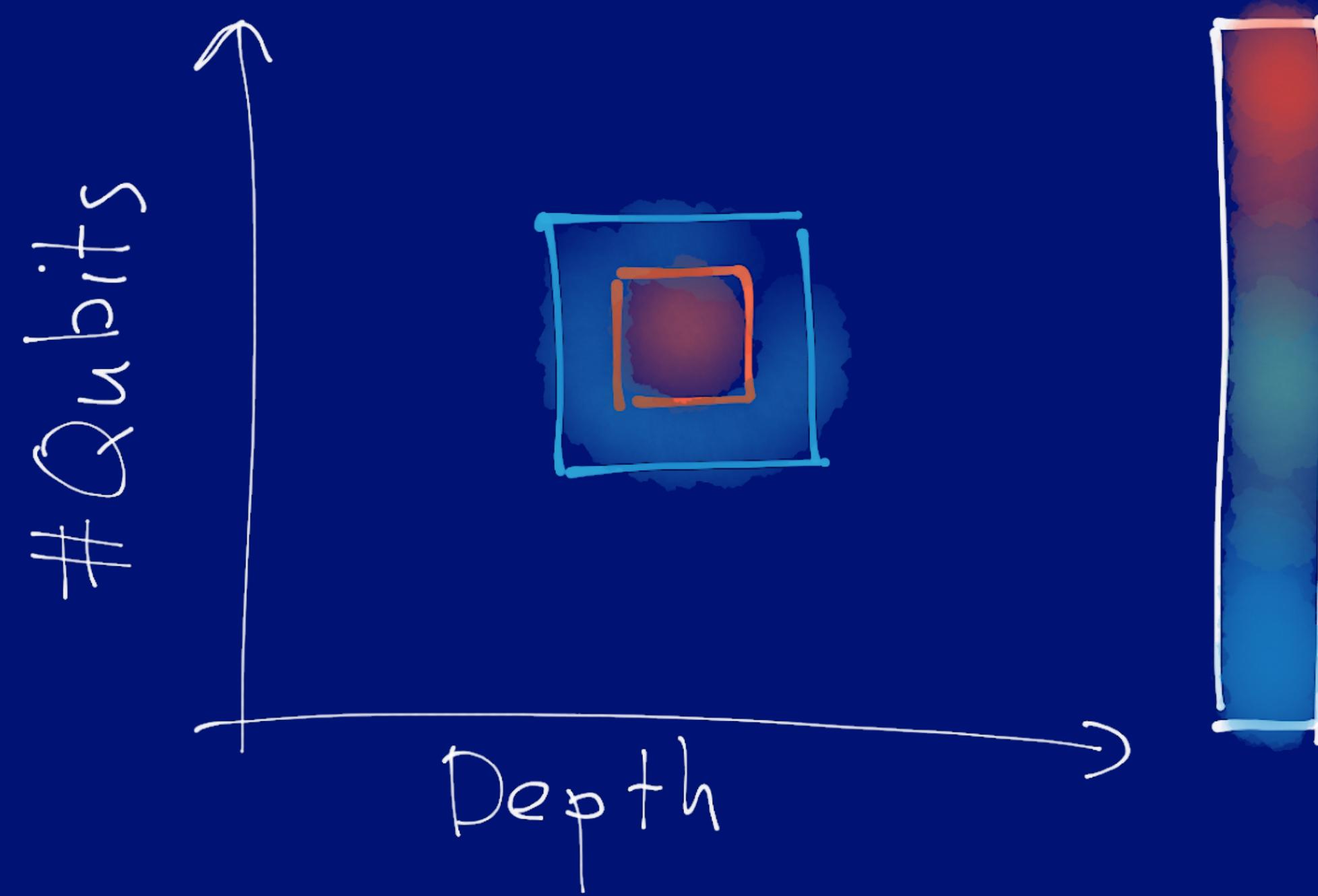
Filtering improves “shape” of the energy landscape (as measured by e.g cosine similarity)

How well do they work in practice?

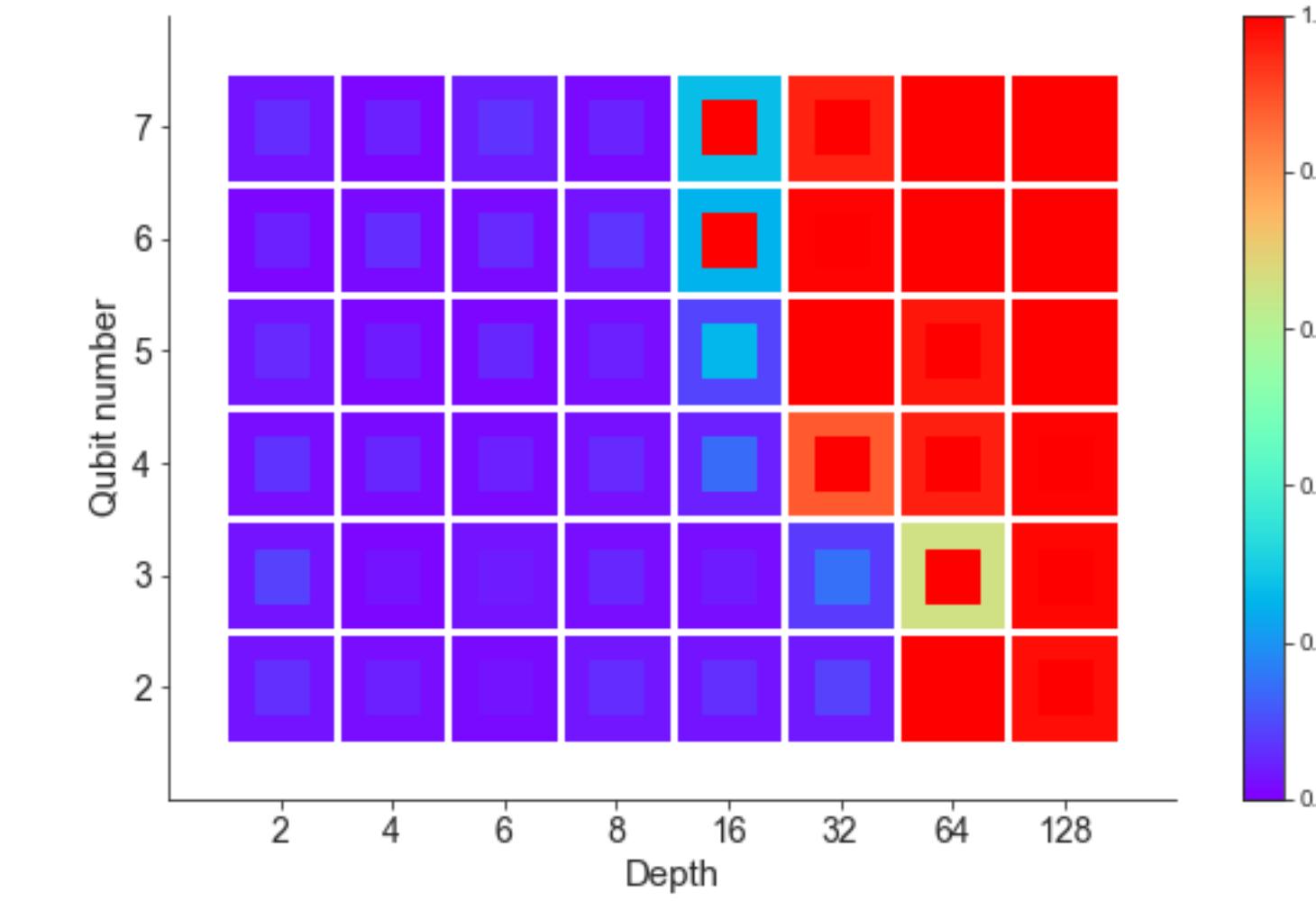
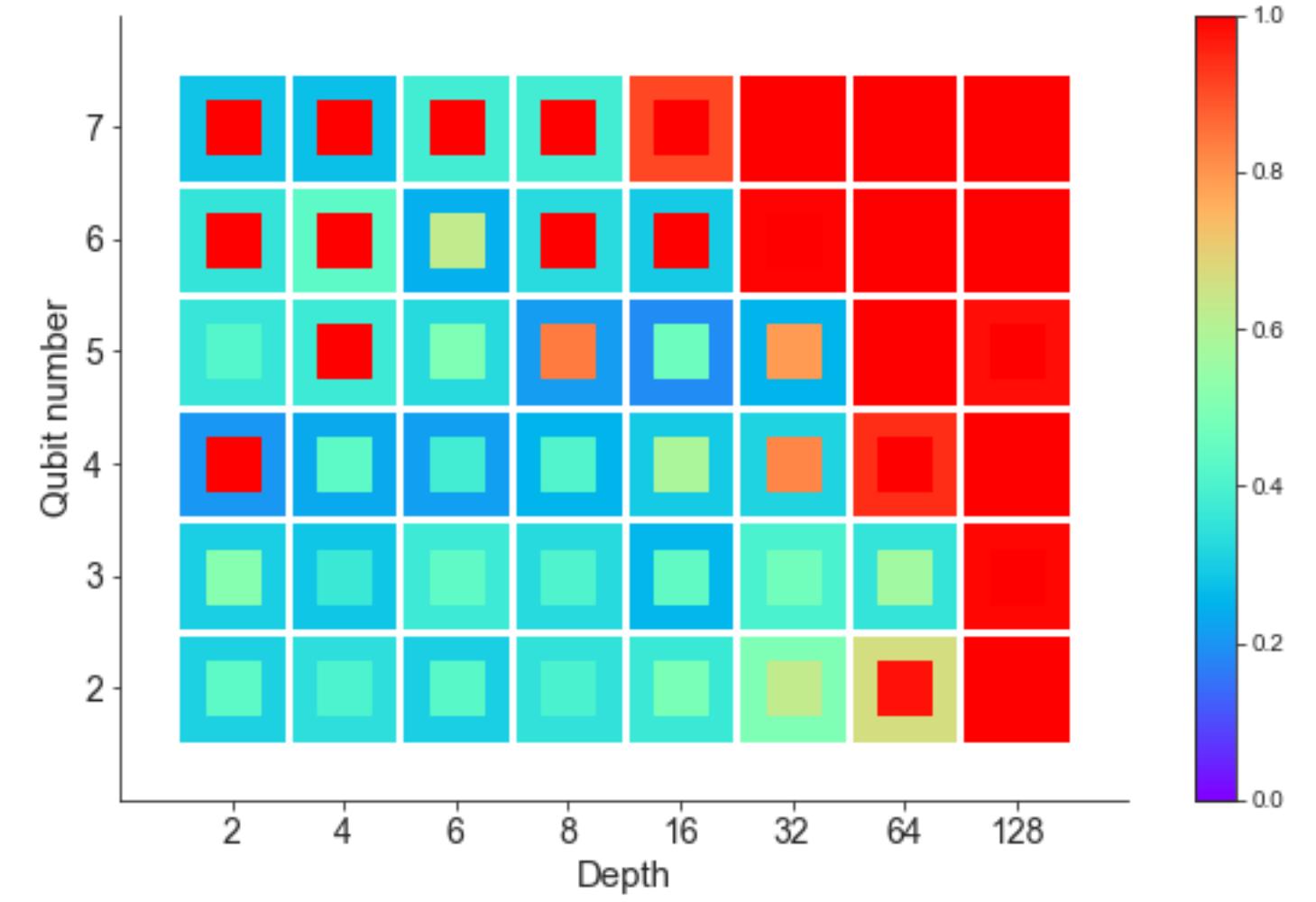


How well do they work in practice?

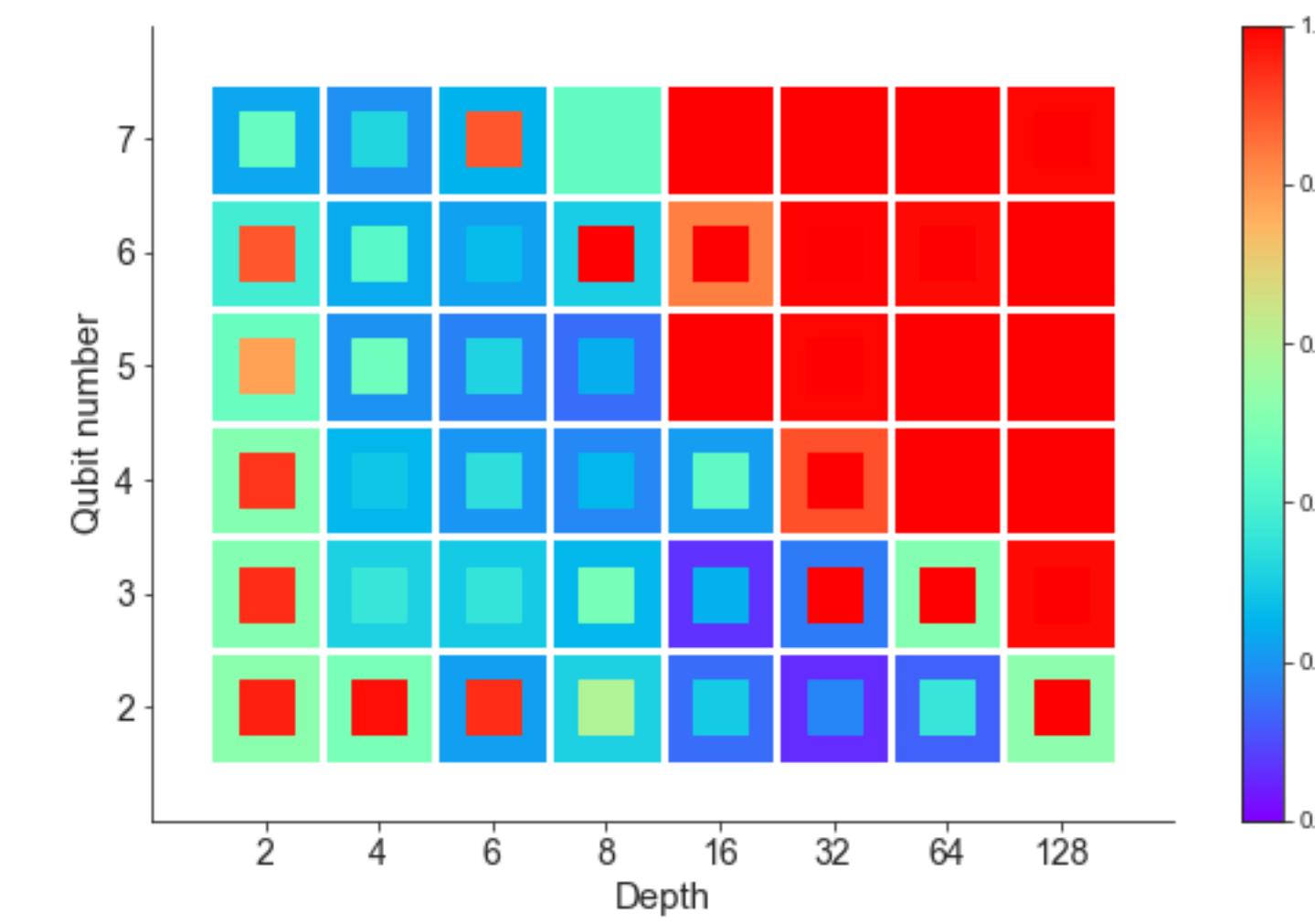
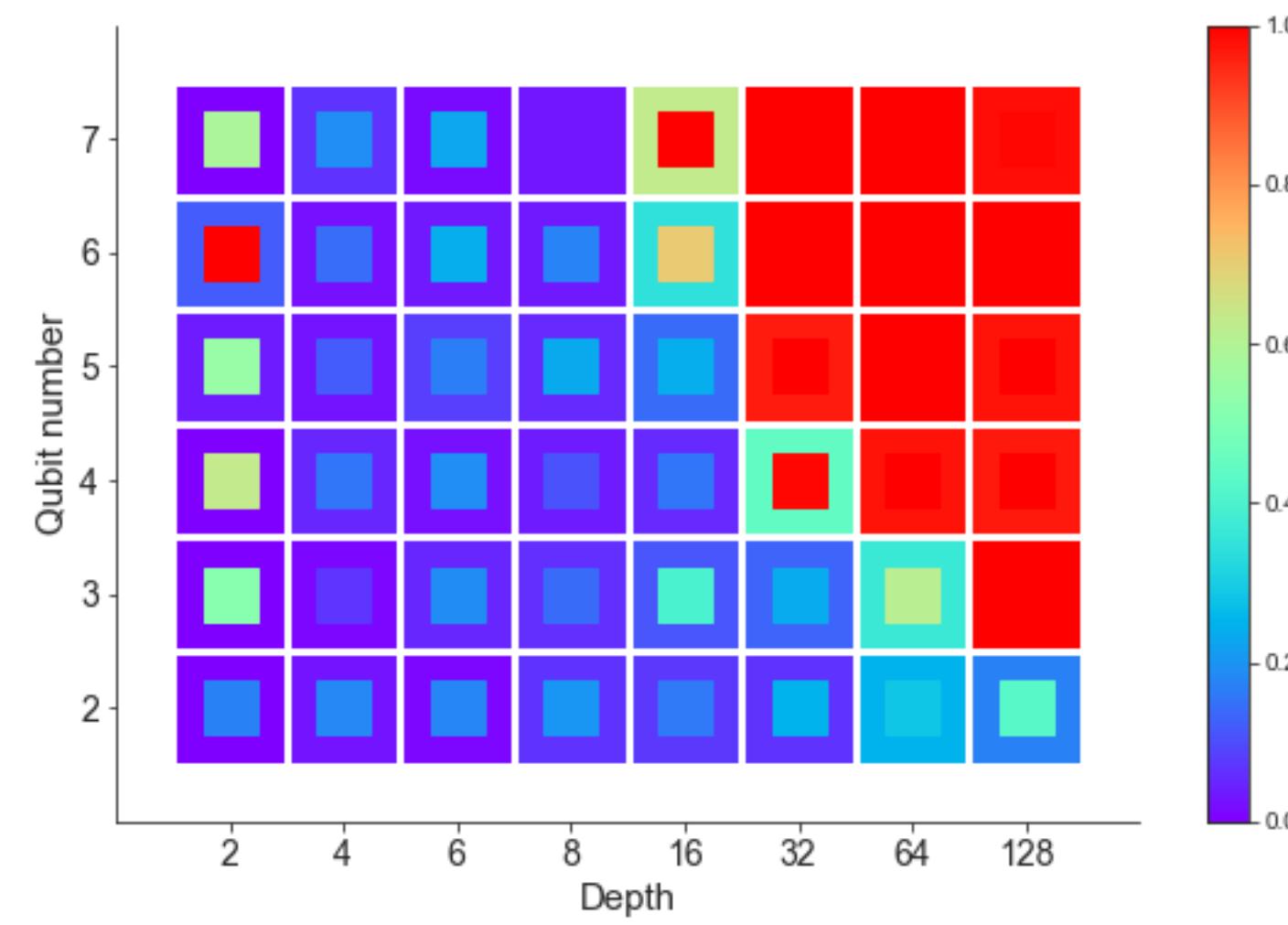
$$\frac{|\langle \hat{O} \rangle_{EM} - \langle O \rangle|}{|\langle \hat{O} \rangle_n - \langle O \rangle|}$$



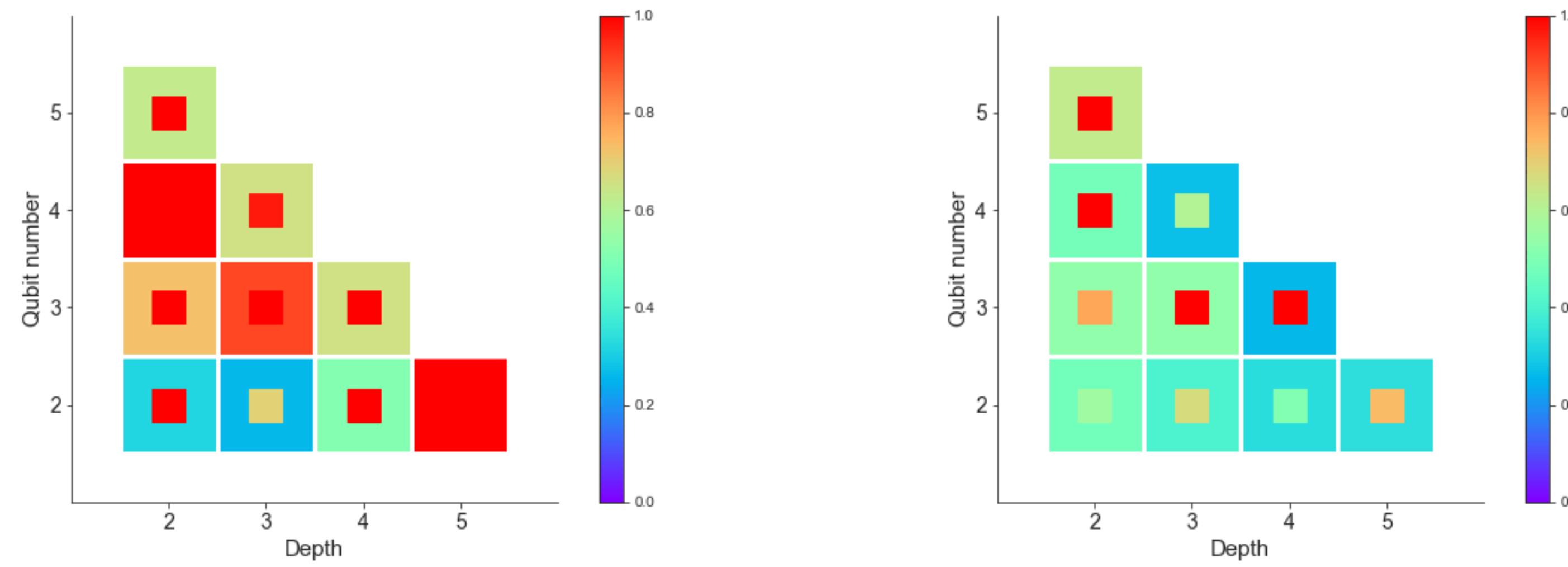
How well do they work in practice?



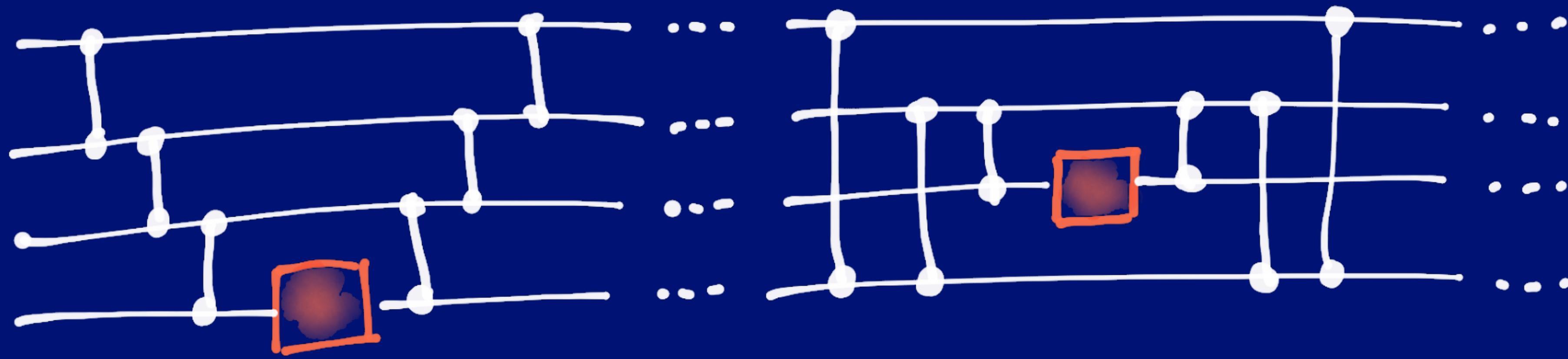
How well do they work in practice?



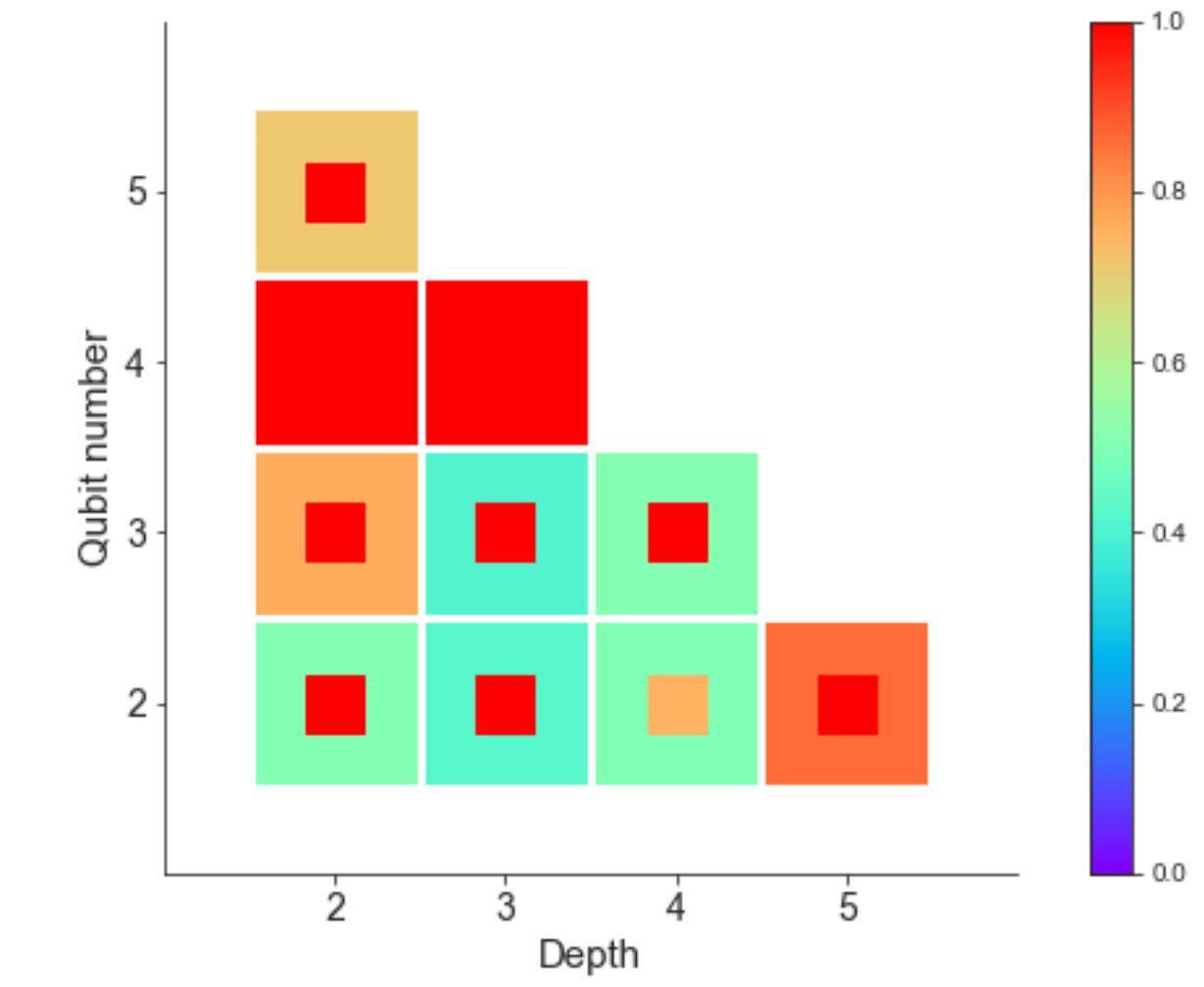
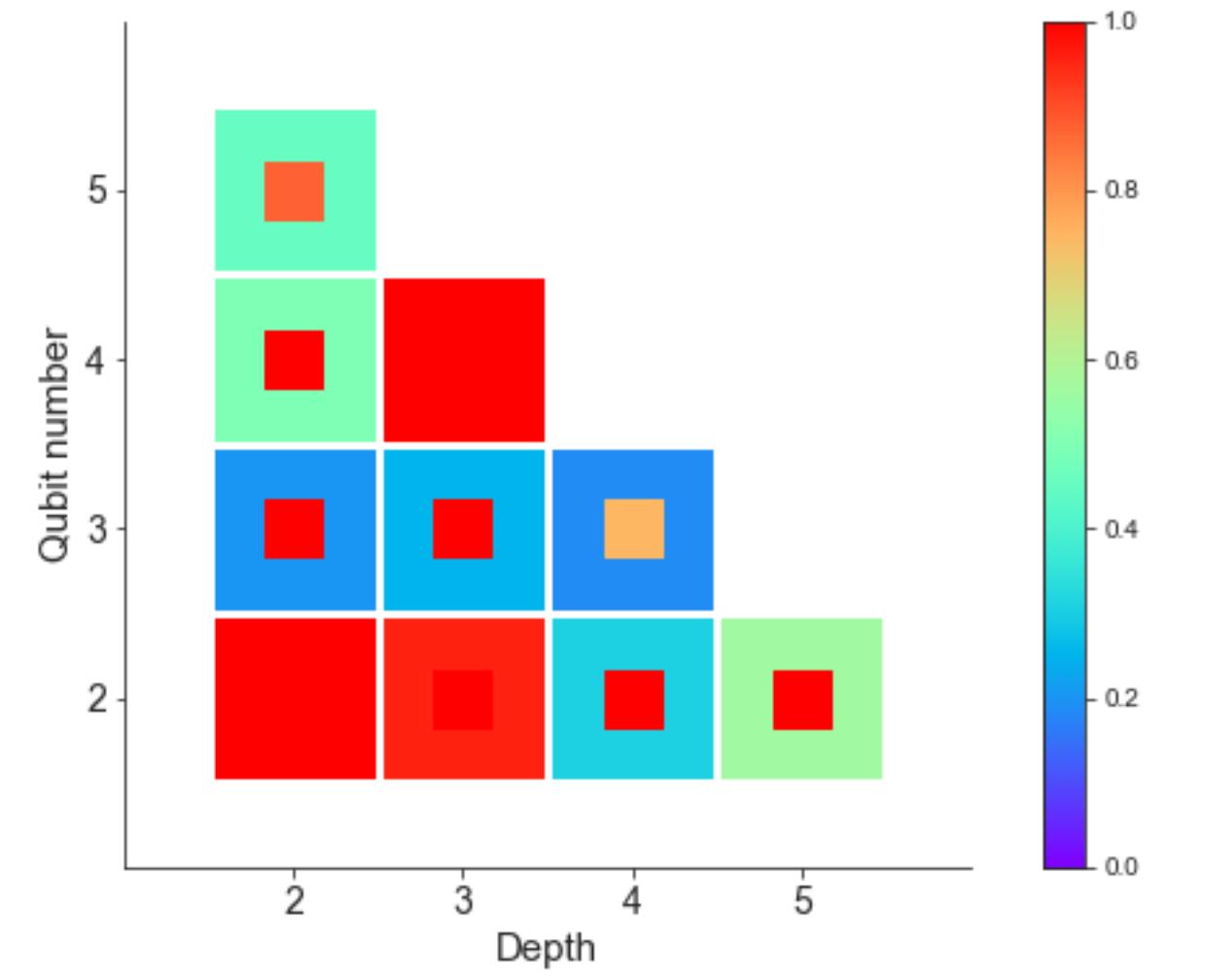
How well do they work in practice?



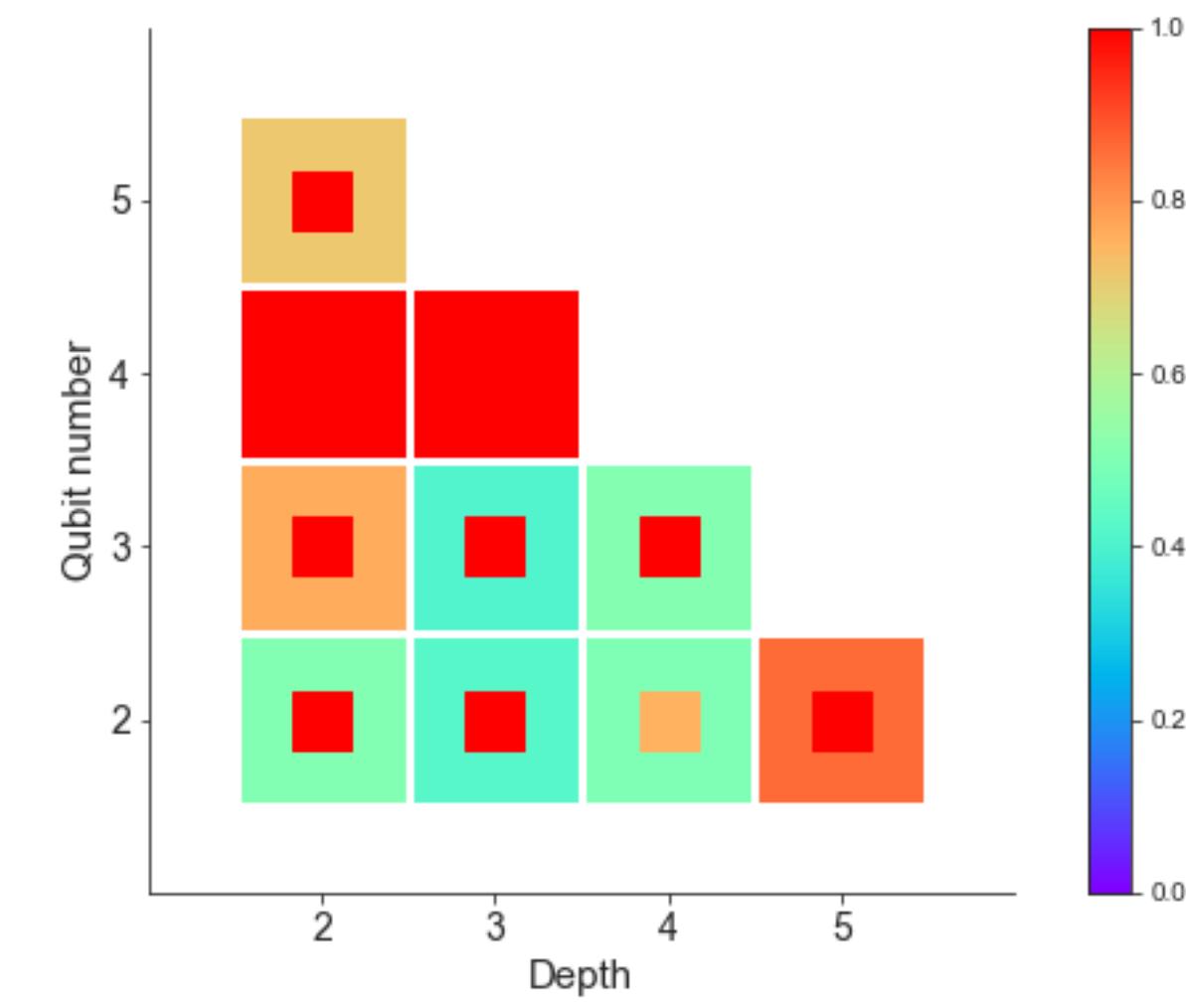
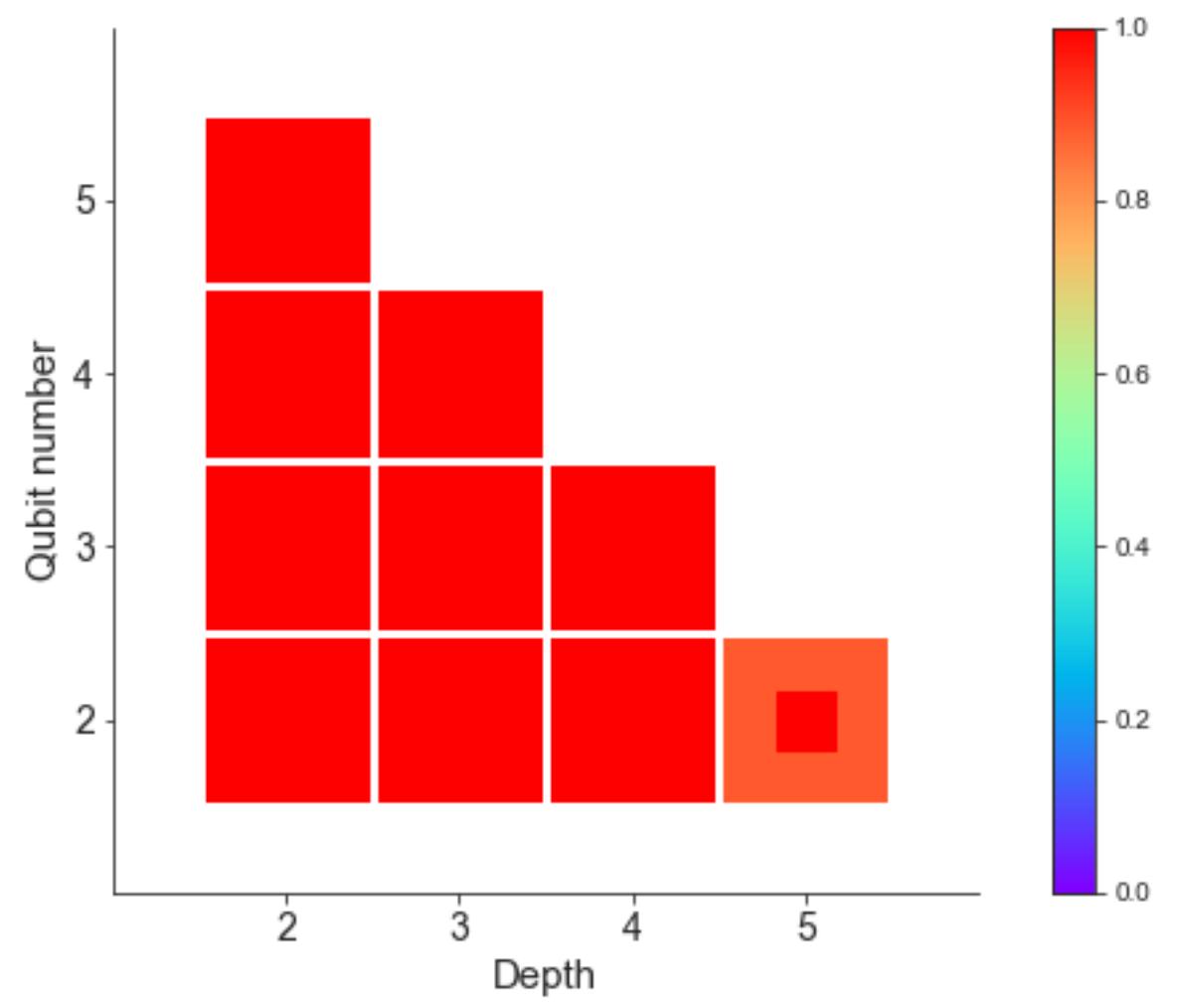
How well do they work in practice?



How well do they work in practice?



How well do they work in practice?



Summary

- ▶ QERMIT currently supports ZNE, CDR, Probabilistic error cancellation, SPAM reduction, frame randomisation and can combine existing methods, and user-defined strategies
- ▶ Seamless integration with other SDKs via Pyket backend
- ▶ Due to modular software design, methods are not hard-coded and subroutines can be re-used
- ▶ Composing different error mitigation strategies + bespoke application/hardware strategies

Summary

- ▶ QERMIT currently supports ZNE, CDR, Probabilistic error cancellation, SPAM reduction, frame randomisation and can combine existing methods, and user-defined strategies
- ▶ Seamless integration with other SDKs via Pyket backend
- ▶ Due to modular software design, methods are not hard-coded and subroutines can be re-used
- ▶ Composing different error mitigation strategies + bespoke application/hardware strategies

How about some code examples ?

References

Software for error mitigation:

Qermit — Cirstoiu, C., Dilkes, S., Mills, D., Sivarajah, S., & Duncan, R. (2022). Volumetric Benchmarking of Error Mitigation with Qermit. *arXiv:2204.09725*.

Mitiq — LaRose, R., Mari, A., Shammah, N., Karalekas, P., & Zeng, W. (2020). Mitiq: A software package for error mitigation on near-term quantum computers.

mthree — Nation, P.D, Kang, K., Sundaresan, N., and Gambetta, J, “Scalable Mitigation of Measurement Errors on Quantum Computers”, PRX Quantum 2, 040326 (2021)

SPAM mitigation

Paul D. Nation, Hwajung Kang, Neereja Sundaresan, and Jay M. Gambetta,, “Scalable Mitigation of Measurement Errors on Quantum Computers”, PRX Quantum 2, 040326 (2021)

Bravyi, S., Sheldon, S., Kandala, A., Mckay, D. C., & Gambetta, J. M. (2021). Mitigating measurement errors in multiqubit experiments. *Physical Review A*, 103(4), 042605.

Maciejewski, F. B., Zimborás, Z., & Oszmaniec, M. (2020). Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography. *Quantum*, 4, 257.

Error mitigation for observables

Kristan Temme, Sergey Bravyi, and Jay M Gambetta. “Error mitigation for short-depth quantum circuits”. In: *Physical review letters* 119.18 (2017), p. 180509.

Li, Y., and S. C. Benjamin, *Phys. Rev. X* 7 (2), 021050. (2017)

Tudor Giurgica-Tiron et al. “Digital zero noise extrapolation for quantum error mitigation”. In: *arXiv preprint arXiv:2005.10921* (2020).

Lowe, Angus, et al. "Unified approach to data-driven quantum error mitigation." *Physical Review Research* 3.3 (2021): 033098.

Czarnik, Piotr, et al. "Error mitigation with Clifford quantum-circuit data." *Quantum* 5 (2021): 592.

Youngseok, K., Wood, C.J., Yoder, T.J., Merkel, S.T, Gambetta J.M , Temme K,Kandala A. "Scalable error mitigation for noisy quantum circuits produces competitive expectation values." *arXiv:2108.09197* (2021).

Cade, C., Mineh, L., Montanaro, A., & Stanisic, S. (2020). Strategies for solving the Fermi-Hubbard model on near-term quantum computers. *Physical Review B*, 102(23),

Fontana, E, Rungger, I., Duncan, R., Cirstoiu, C. "Spectral analysis for noise diagnostics and filter-based digital error mitigation." *arXiv preprint arXiv:2206.08811* (2022).