Developing and Executing Error-Mitigated NISQ Algorithms Across Devices and Simulators

Part 2: Introduction to error mitigation

Tutorial — IEEE 2022

QUANTINUUM

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

I'll do that, find me at

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

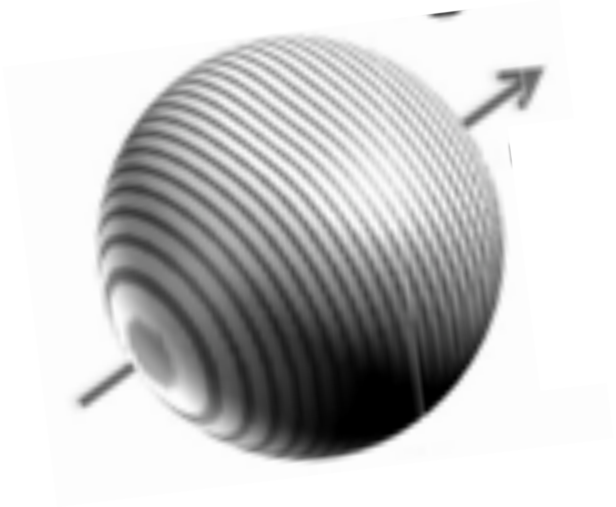
```
pip install qermit
```

Documentation and examples → www.qerm.it.

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

Noise models

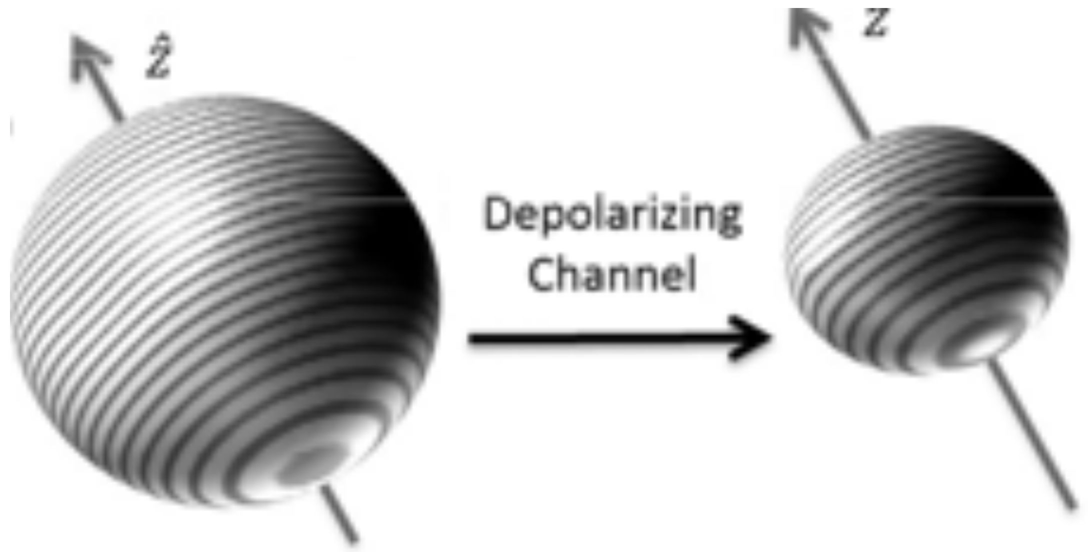
Coherent vs Incoherent noise



$$\rho \longrightarrow U\rho U^{\dagger}$$

Depolarisation

Probabilistically mixes input state with the maximally mixed state

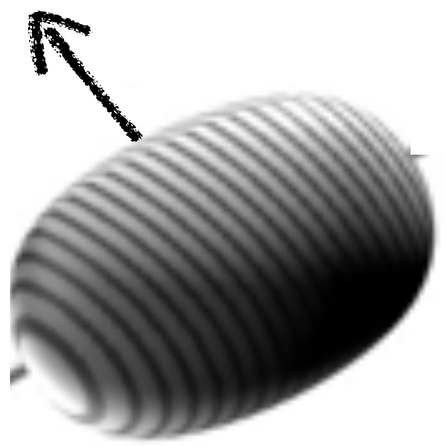


$$\rho \longrightarrow (1-p)\rho + p\,\mathbb{I}/2$$

Bit-flip errors

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

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



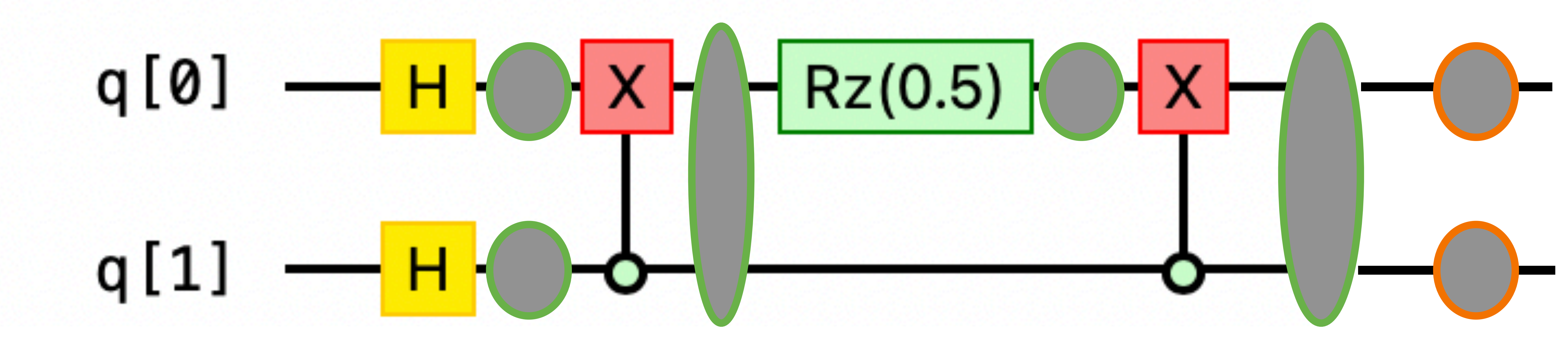
$$\rho \longrightarrow (1-p)\rho + X\rho X$$

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

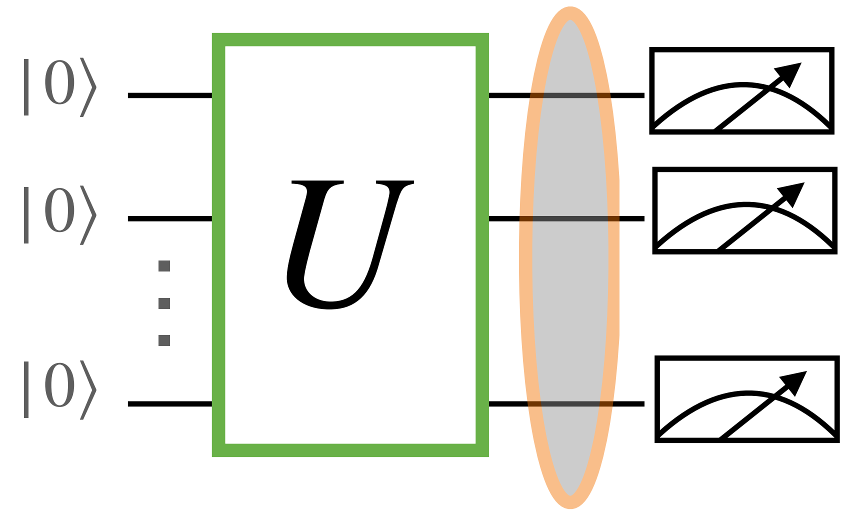
- Reduce the effect of SPAM on the outcome sample probability distribution

- Detector tomography

  Construct process matrix S of transition probabilities

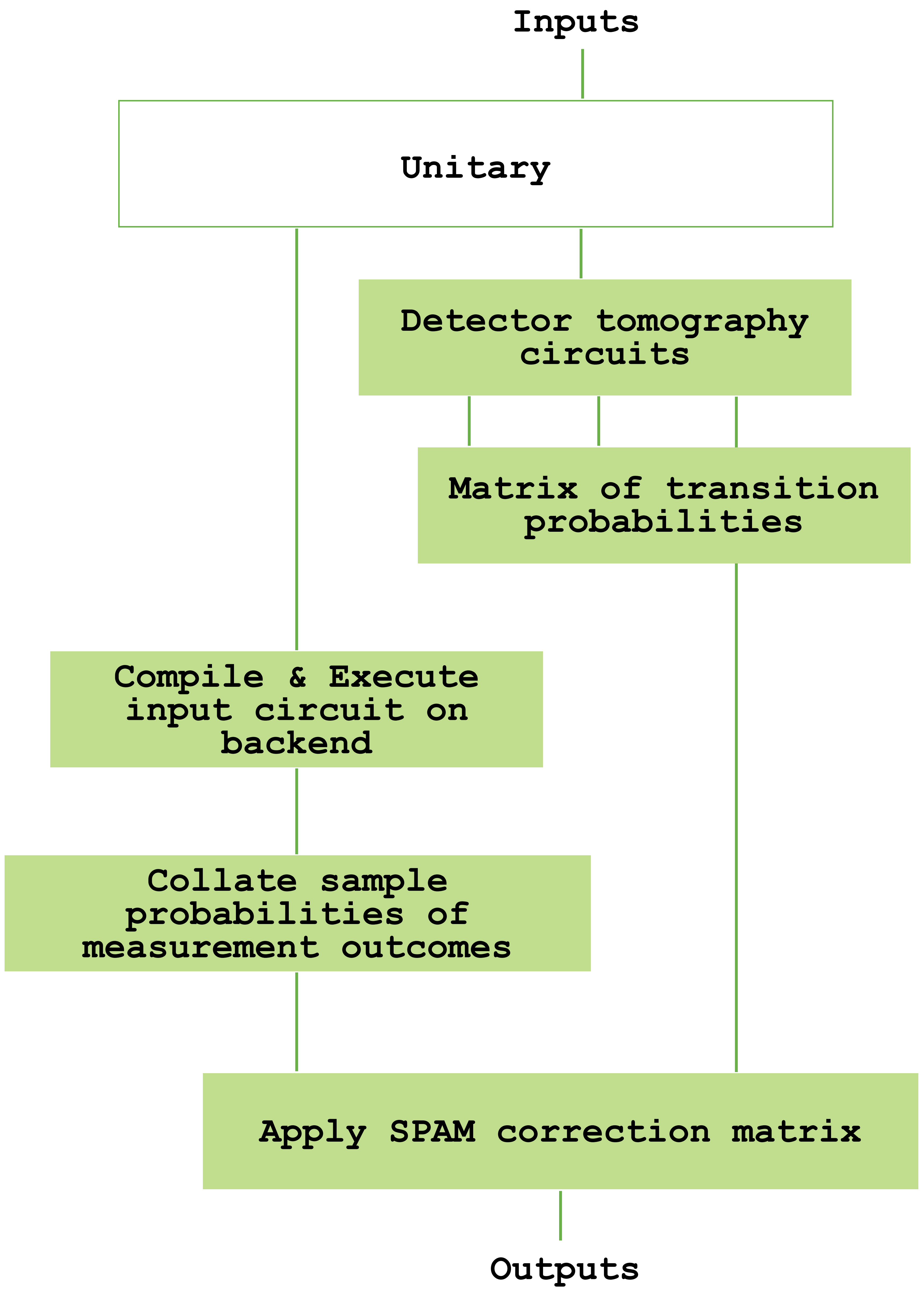
$$S_{zz'} = prob(z \mid z')$$

z - measurement outcome n bit-string



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

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

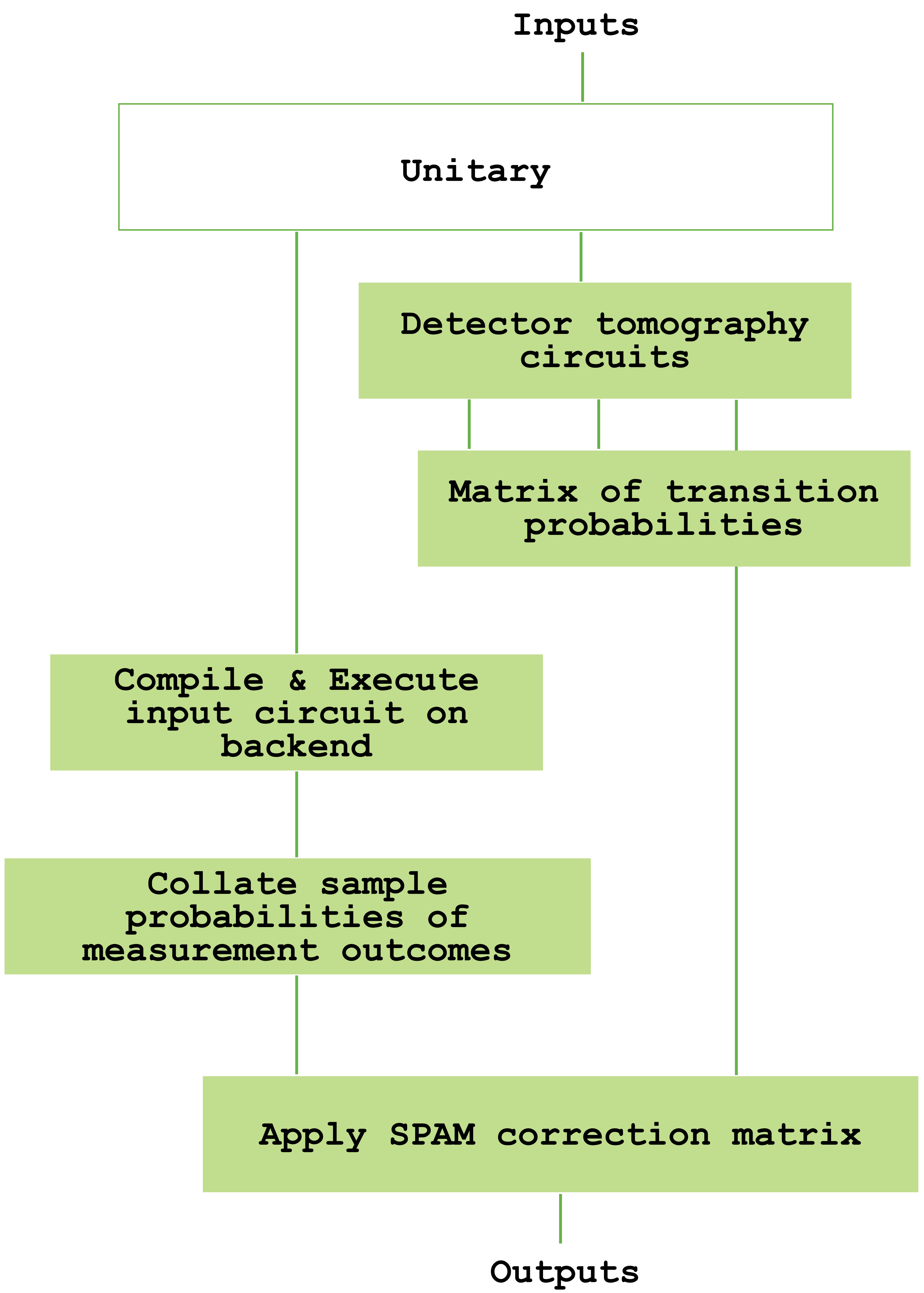


State Preparation and Measurement error mitigation

- 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

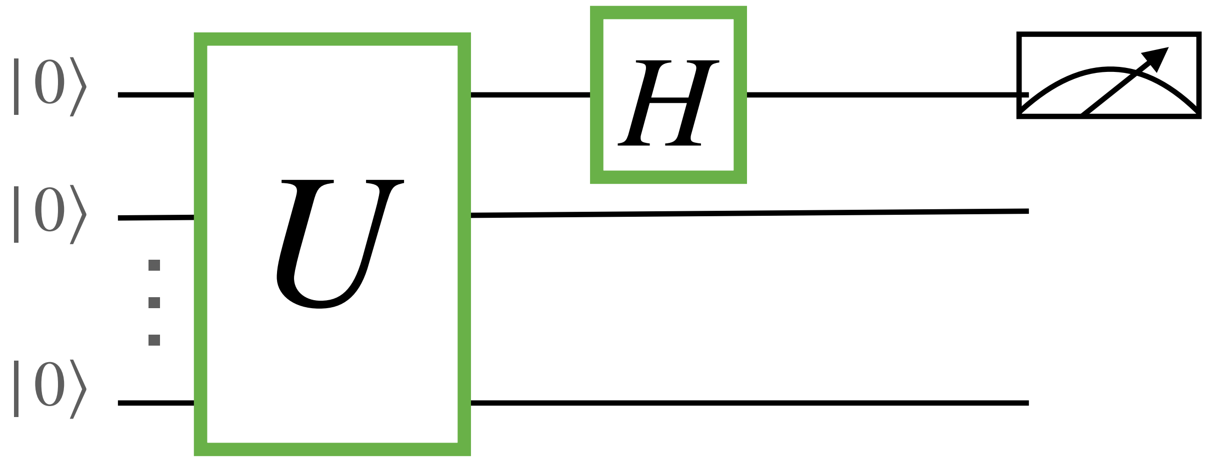
Input: Observable O, unitary $U = U_D \dots U_1$

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

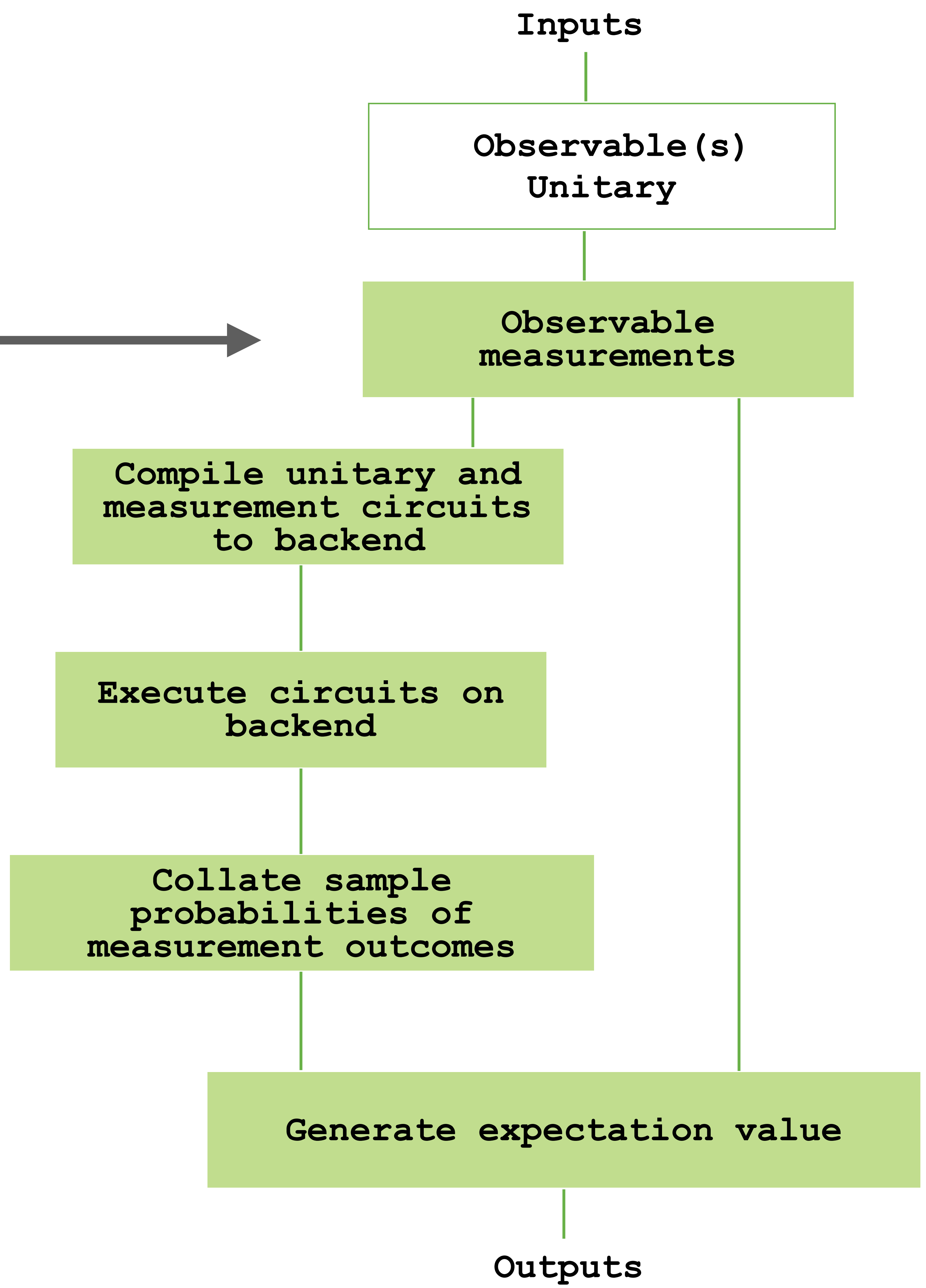
Decompose O into (projective) operator measurements

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

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



$< X > = p(0) - p(1)$

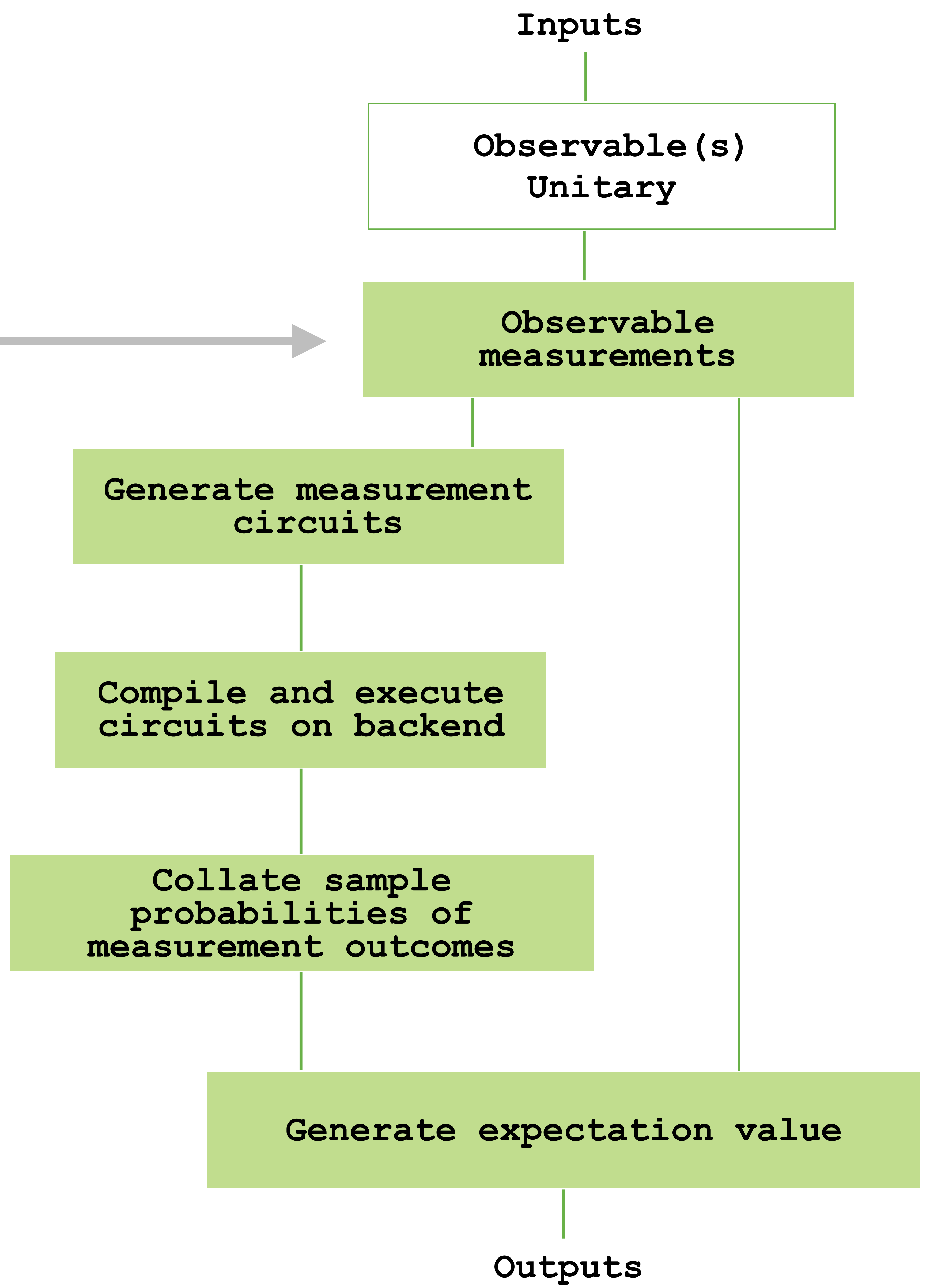


Expectation value of observables —workflow

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

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

Decompose O into (projective) operator measurements



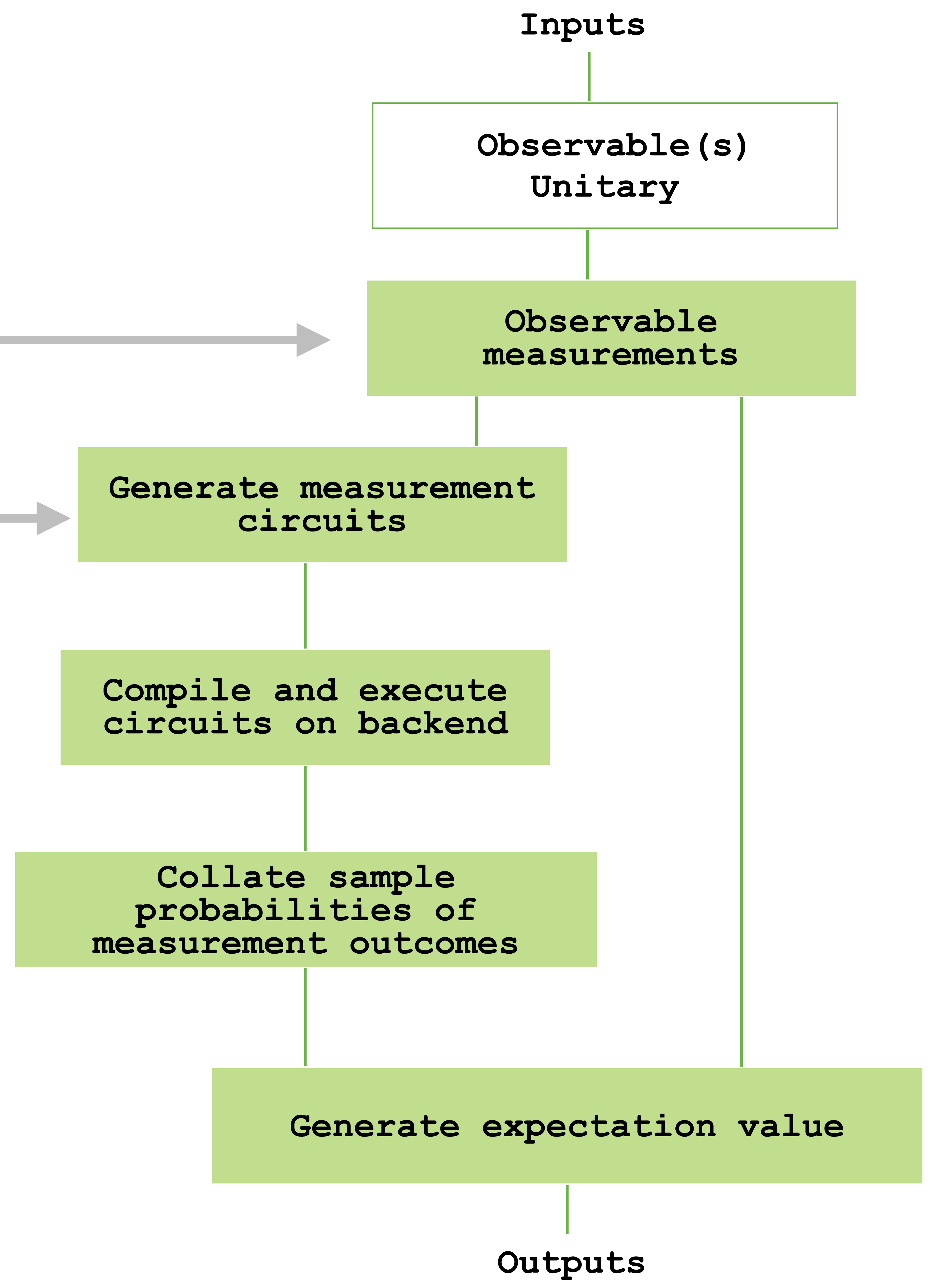
Expectation value of observables —workflow

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

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

Decompose O into (projective) operator measurements

Appends rotation gates and measurement in computational basis



Expectation value of observables —workflow

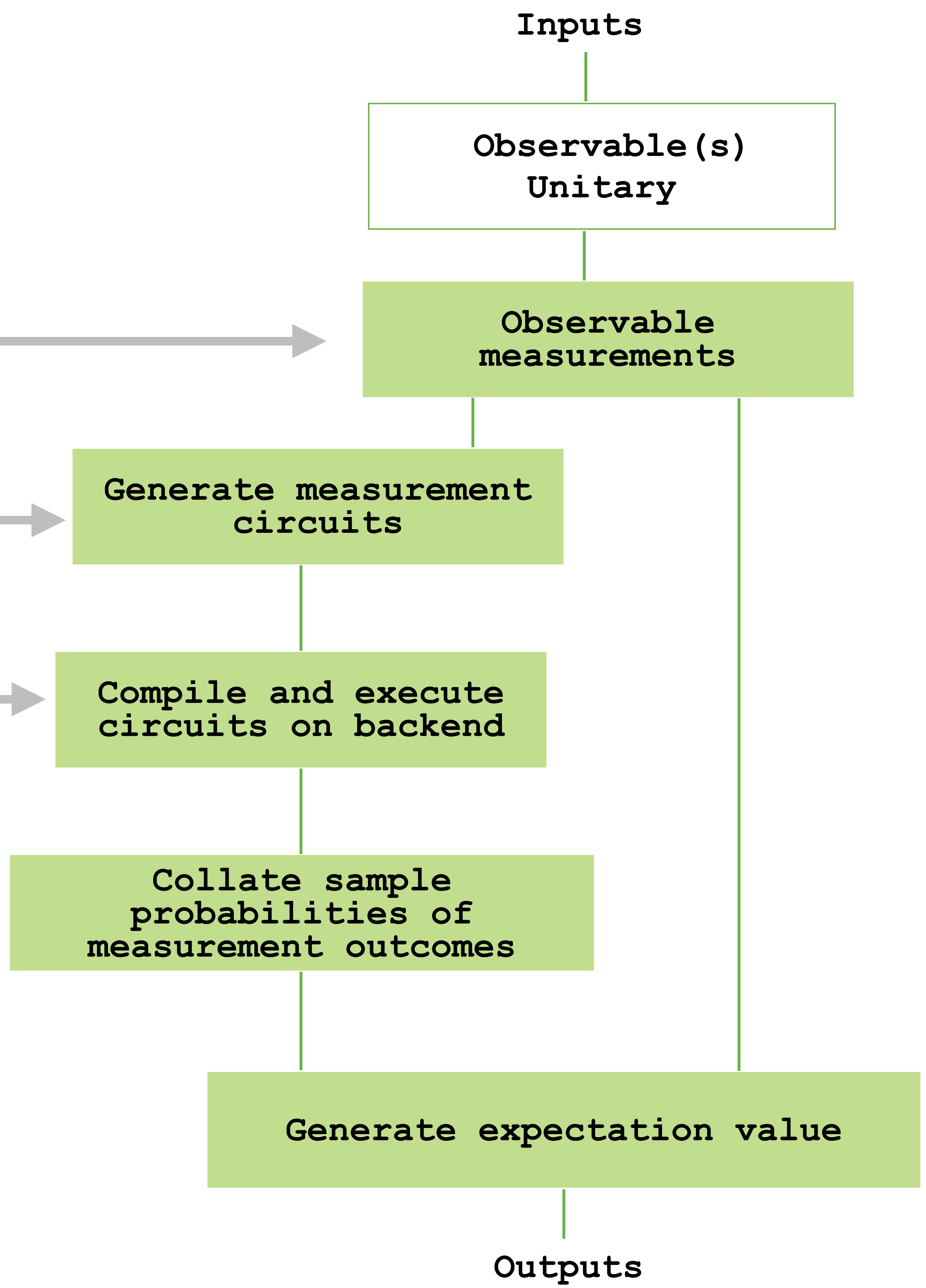
Input: Observable O, unitary $U = U_D \dots U_1$

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

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 $U = U_D \ldots U_1$

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

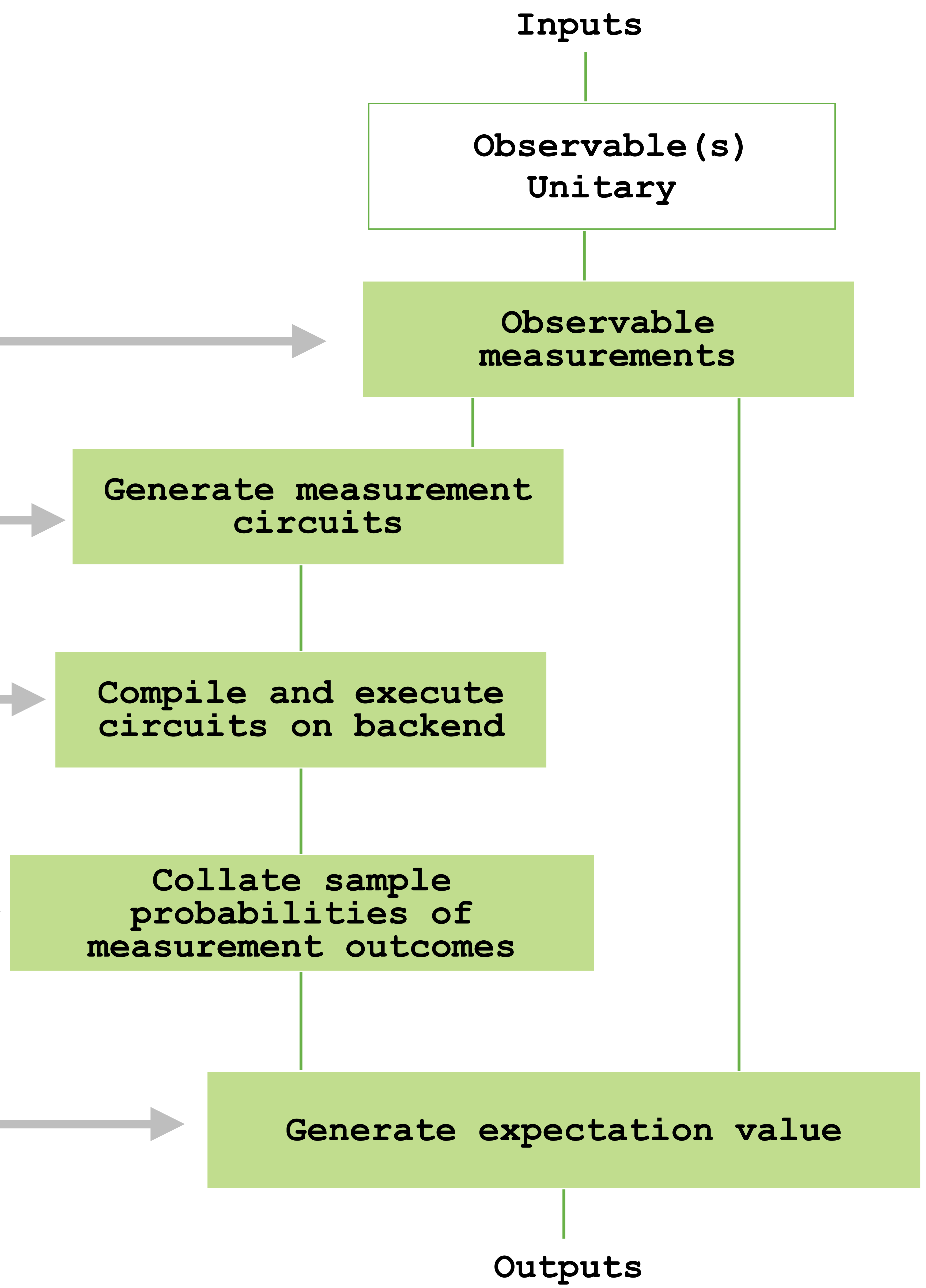
Decompose O into (projective) operator measurements

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

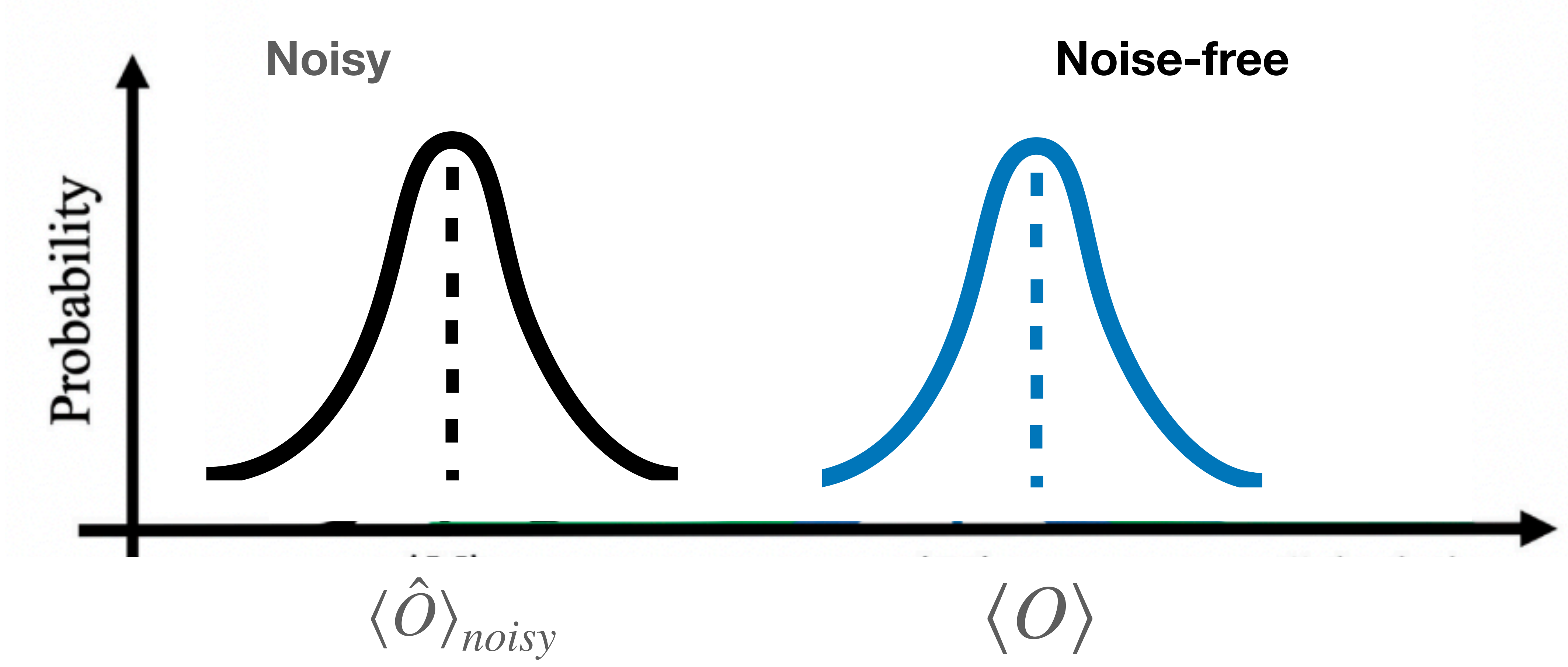


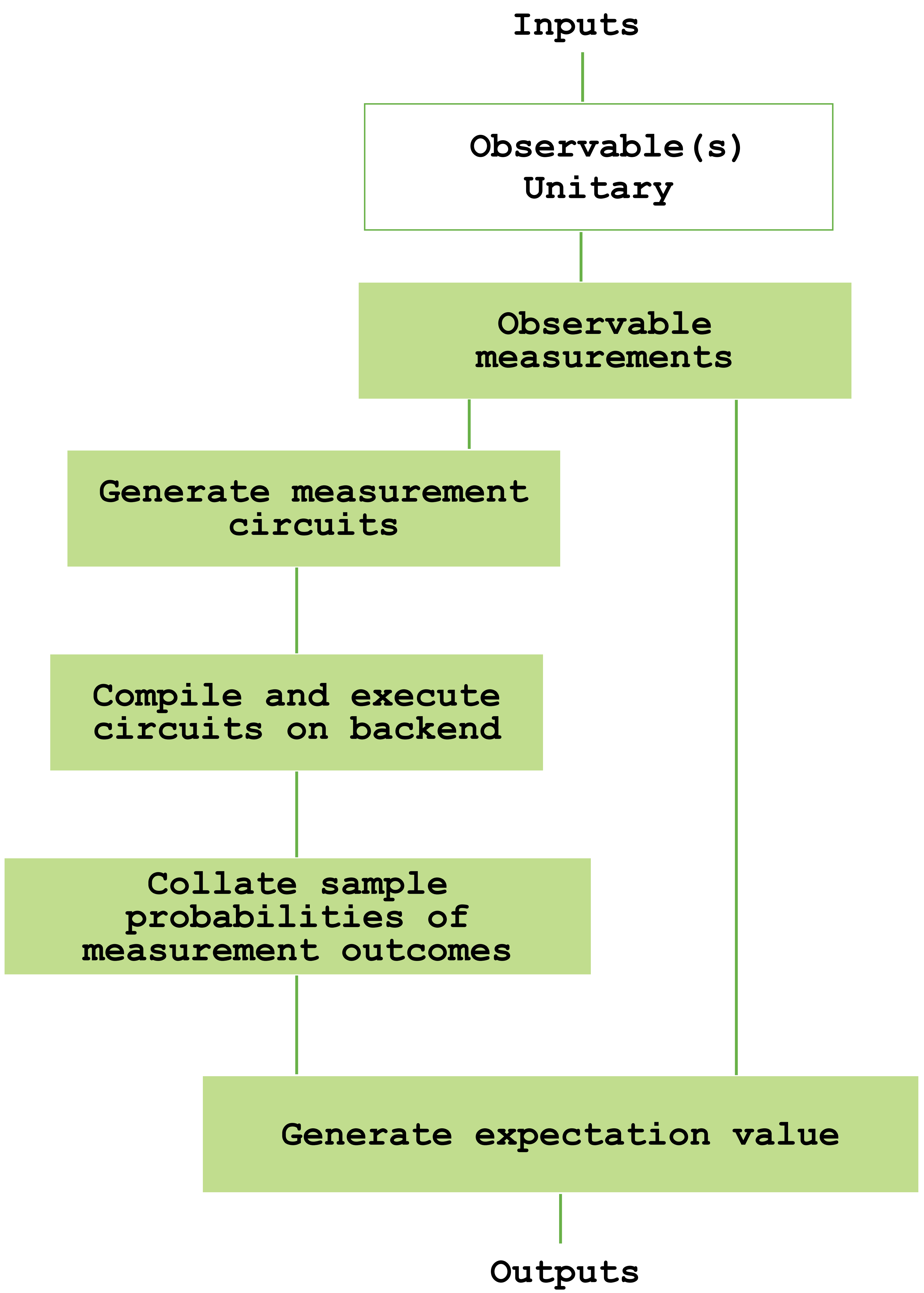
Noisy expectation value of observables — finite sampling

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

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

Variance $\sim \dfrac{\sigma_0^2}{n_s}$



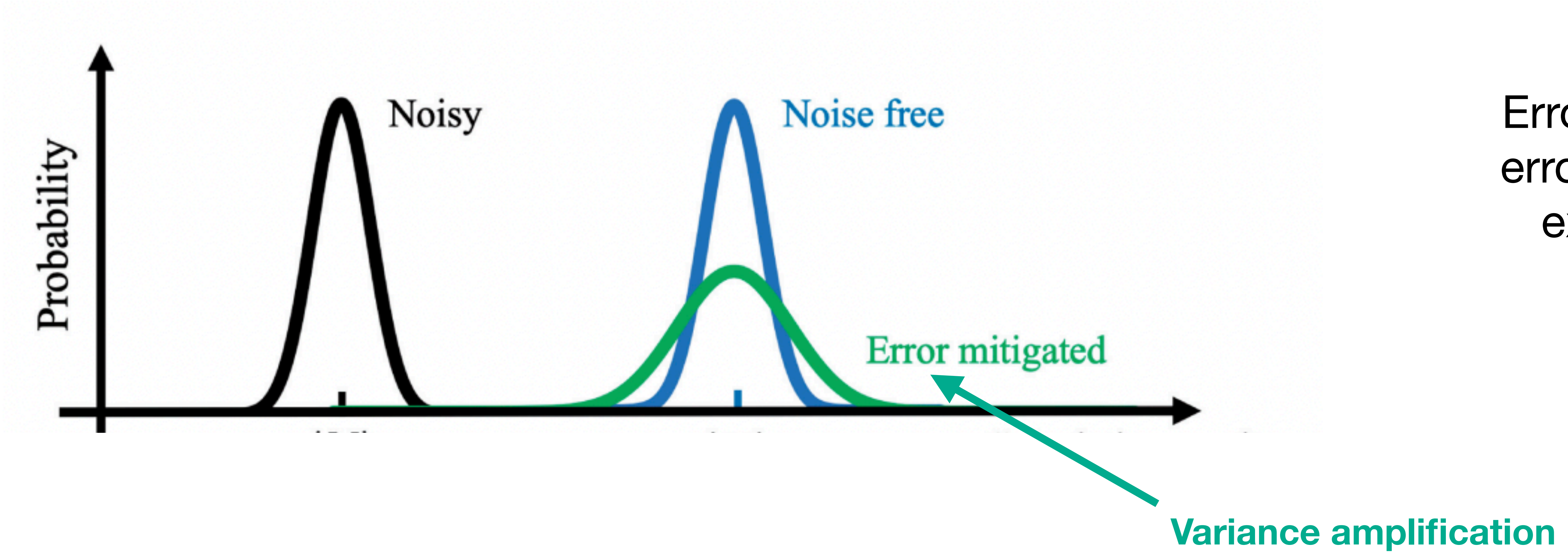


Error mitigation for observables

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

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

Variance ~ $\dfrac{\sigma_0^2}{n_s}$



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 $U = U_D \ldots U_1$ **Output :** Error-mitigated estimator $\langle \hat{O} \rangle_{EM}$

I. Noise characterisation

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

II. Functional model

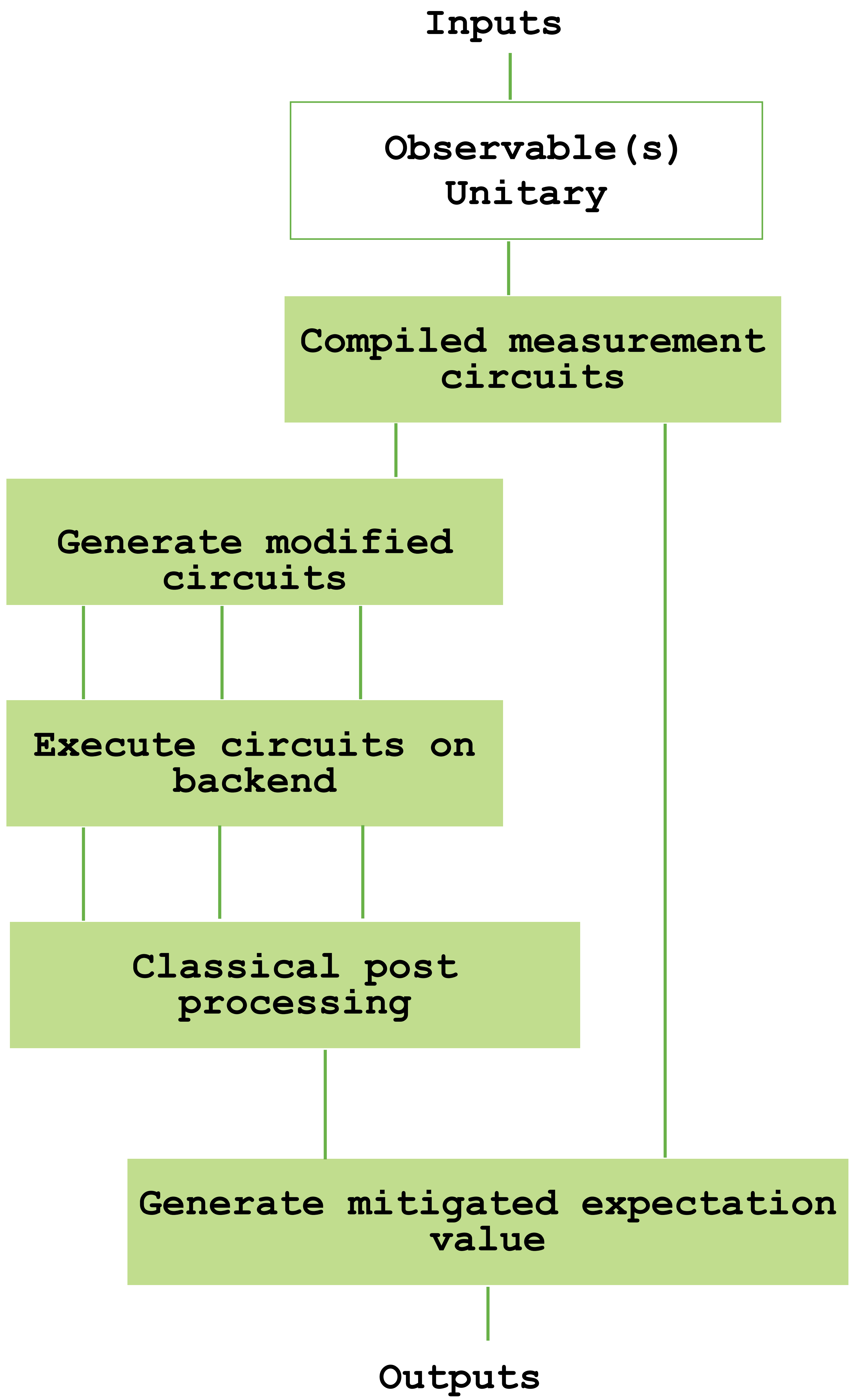
- Motivated by noise models
- Describes relationship between measurements of modified circuit and target expectation value

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

III. Classical post-processing

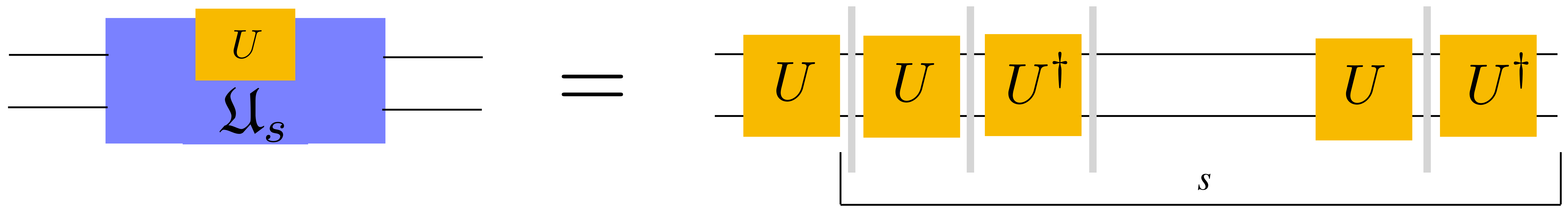
- Determine $\langle \hat{O} \rangle_{EM}$ by fitting parameters from data collected

Error mitigation for observables — Experiment workflow



Zero Noise Extrapolation (ZNE)

I.



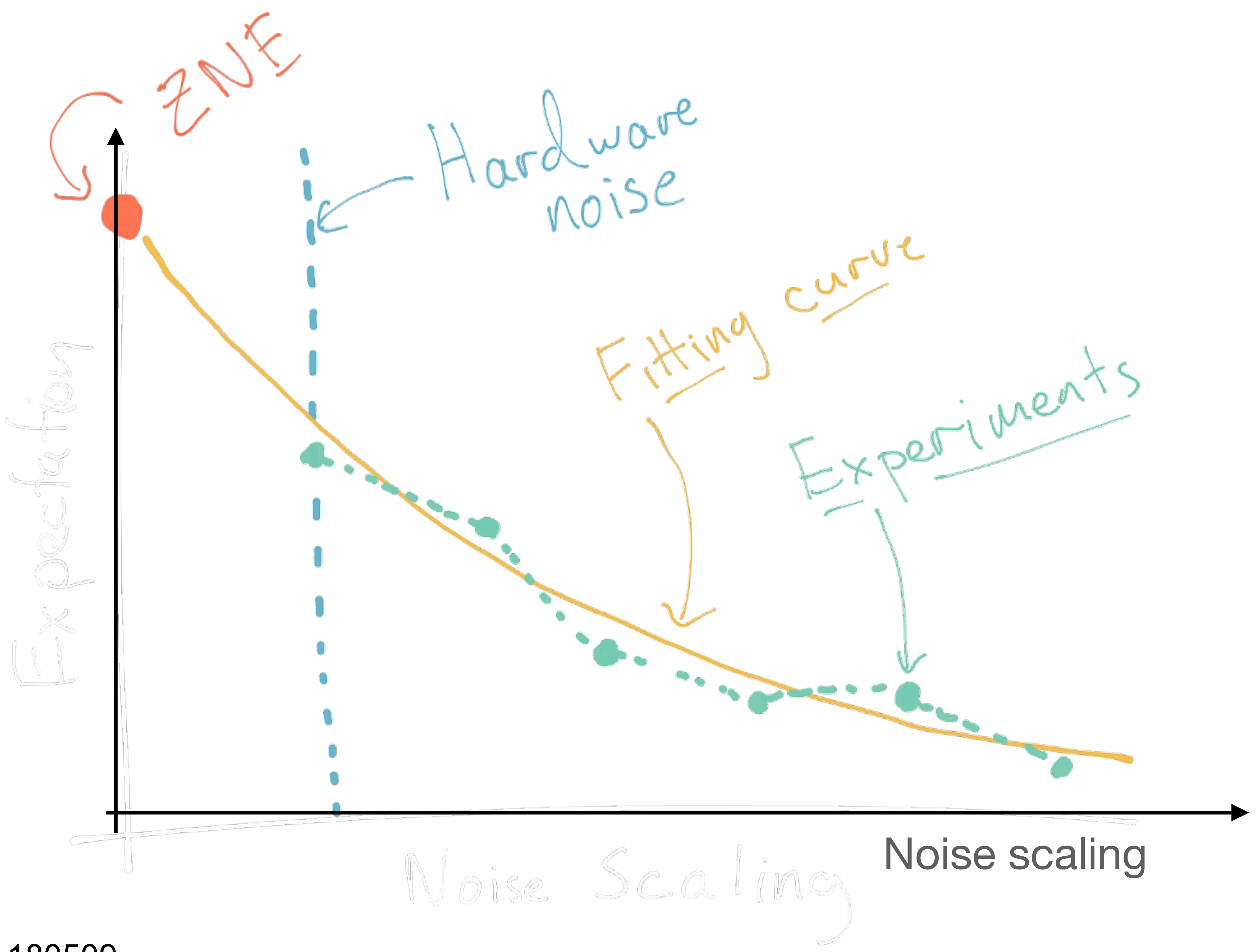
II. Functional Model

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

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

Global depolarising channel

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



III.

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

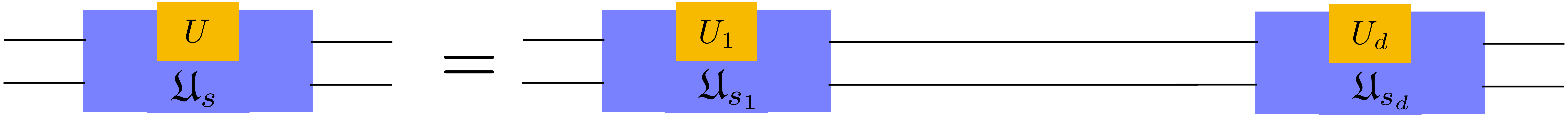
Kristan Temme, Sergey Bravyi, and Jay M Gambetta. “Error mitigation for short-depth quantum circuits”. Physical review letters 119.18 (2017), p. 180509.

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

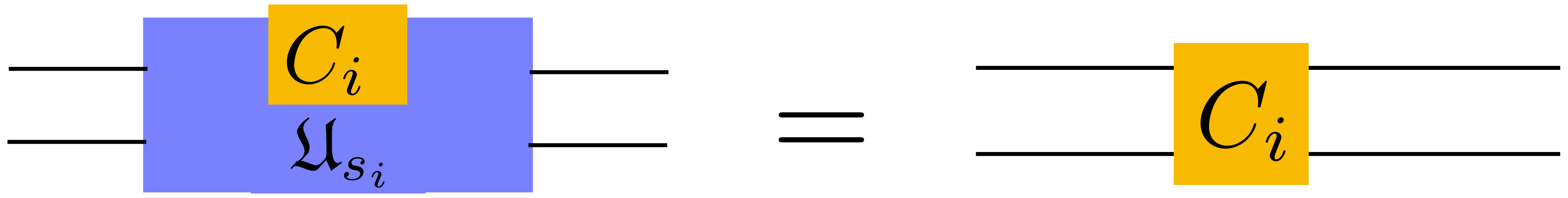
Tudor Giurgica-Tiron et al. “Digital zero noise extrapolation for quantum error mitigation”. In: arXiv preprint arXiv:2005.10921 (2020).

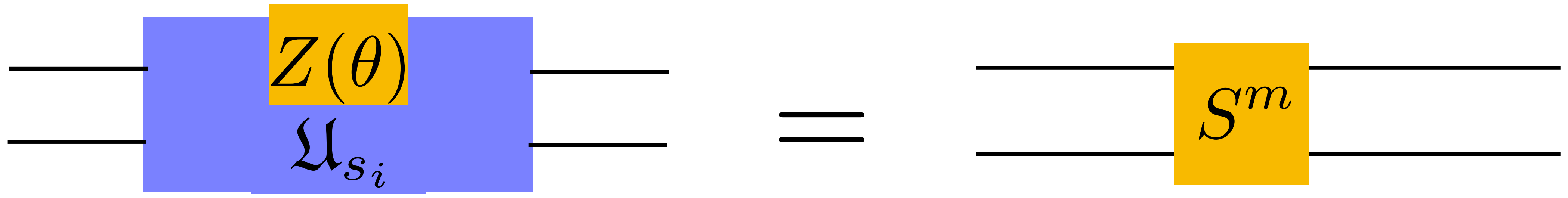
Clifford Data Regression

I.



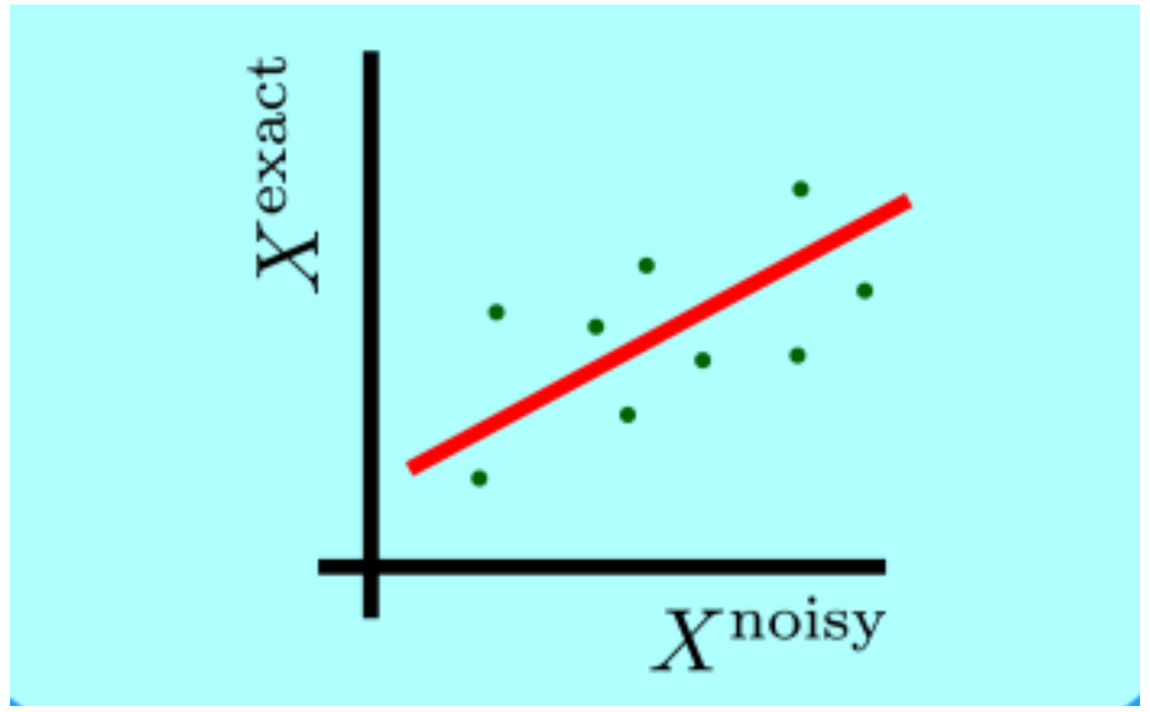
Substitute non-Clifford gates with nearest Clifford





$$S^m \in \{I, Z, S, S^\dagger\}$$

II. **Linear Functional Fit** $D_s^q = F_0 D_s^c + F_1$



III.

$$\langle \hat{O} \rangle_{EM} = \frac{\langle O \rangle_{noisy} - F_1}{F_0}$$

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.

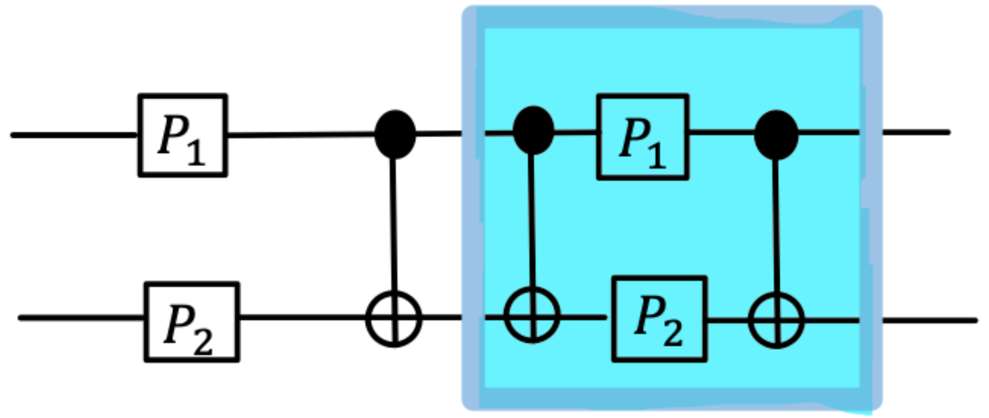
Effectively combining error mitigation strategies

- 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

Twirls coherent noise into Pauli noise

Yup, can do that too!



Pre-compile into Pauli operator

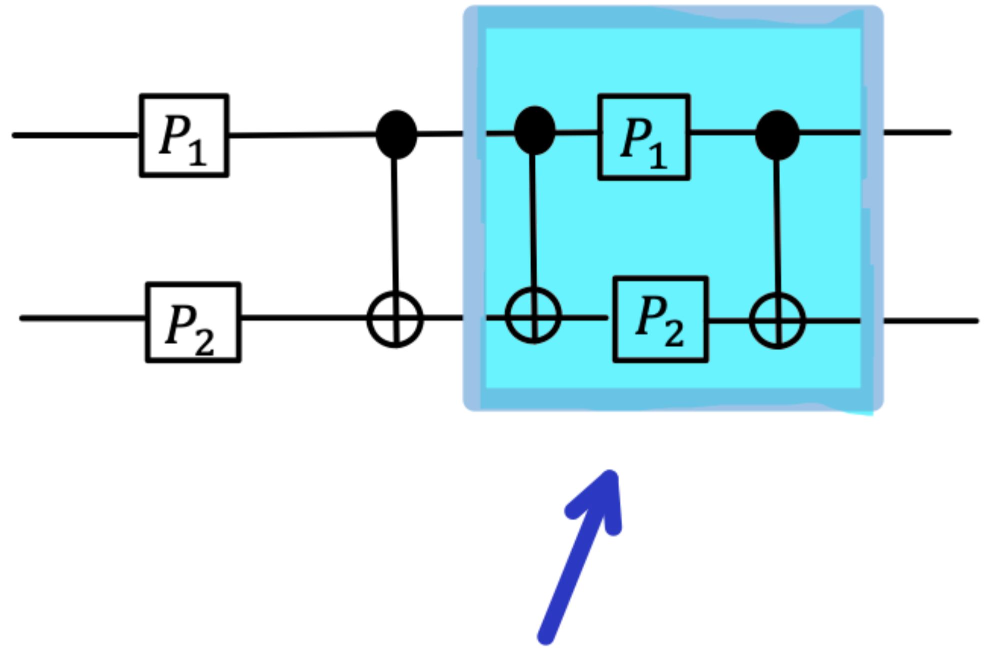
Effectively combining error mitigation strategies

- 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

Yup, can do that too!

Twirls coherent noise into Pauli noise



Pre-compile into Pauli operator

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),

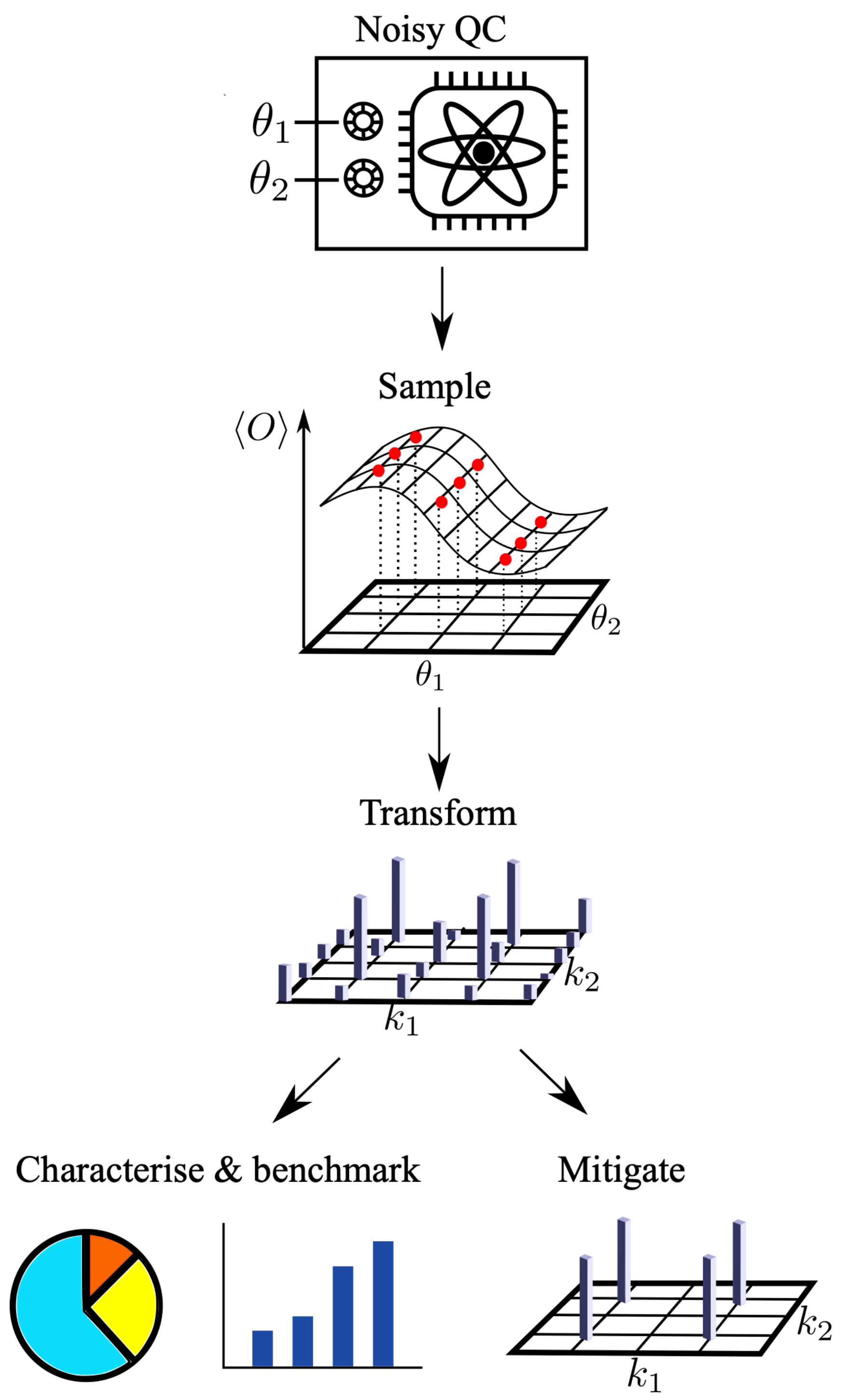
Fontana, E, Rungger, I., Duncan, R., Cirstoiu, C. "Spectral analysis for noise diagnostics and filter-based digital error mitigation." *arXiv preprint arXiv:2206.08811* (2022).

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)

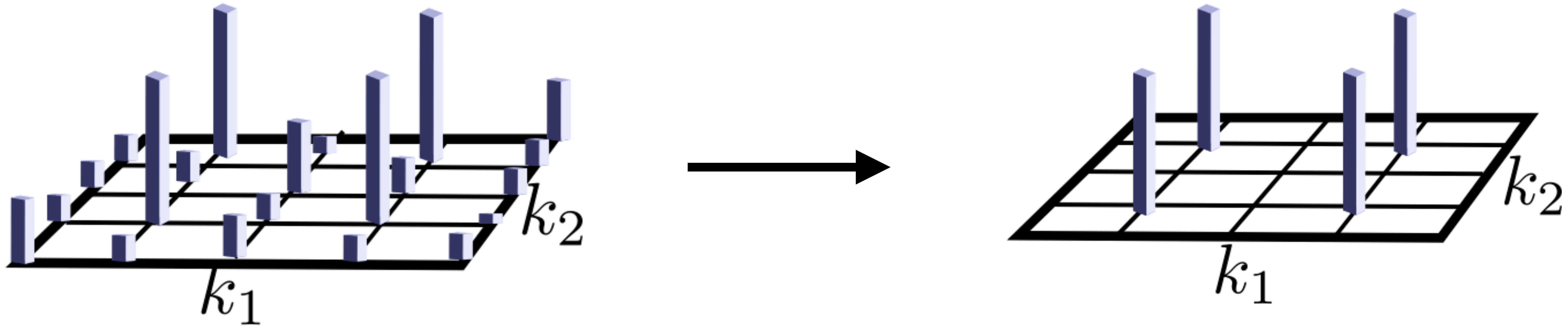
Apply signal processing to circuit families parametrised by continuous variables.



Variational quantum algorithms aim to optimise cost function $C(\boldsymbol{\theta}) = \sum_{k} c_{\mathbf{k}} e^{i\omega_{\mathbf{k}}\cdot\theta}$

Noisy spectrum

Filtered spectrum



Parametrised circuit — information on the frequency support — threshold filter

Spectral analysis for noise diagnostics and filter-based digital error mitigation arXiv:2206.08811, E. Fontana, I. Rungger, R. Duncan, CC,

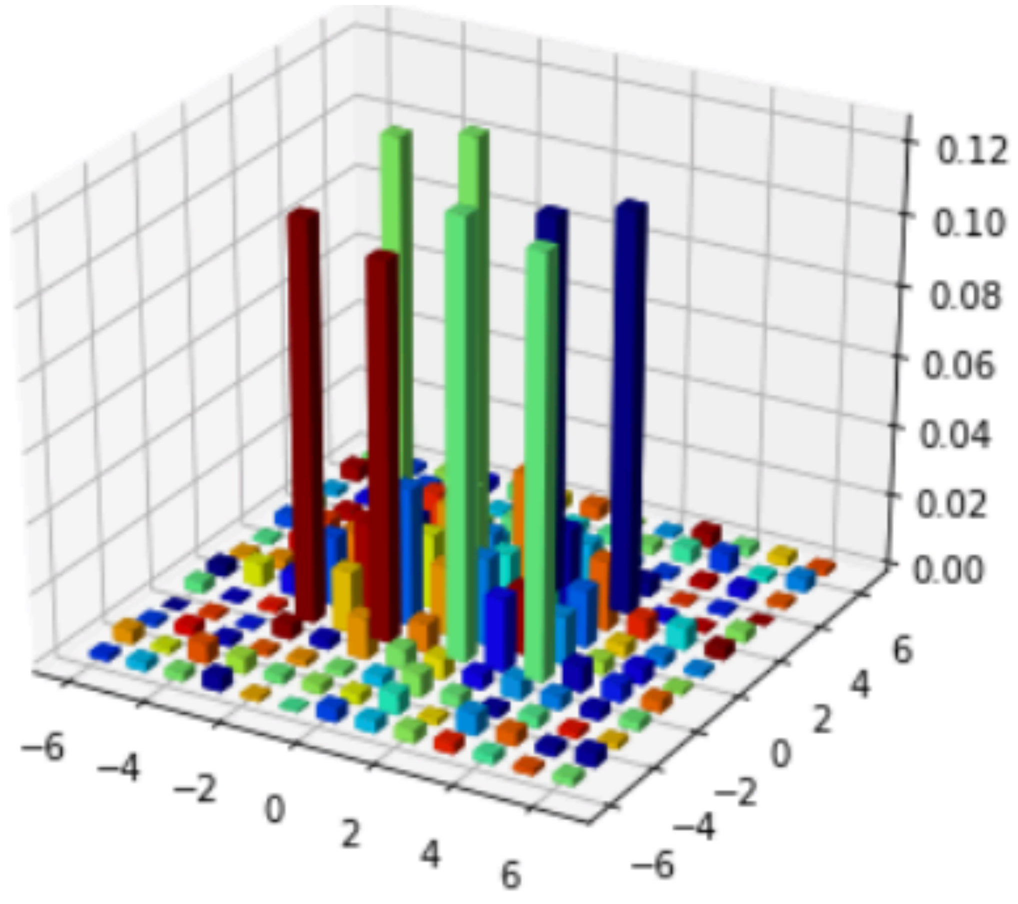
Efficient recovery of variational quantum algorithms landscapes using classical signal processing. arxiv:2208.05958, E. Fontana, I. Rungger, R. Duncan, CC,

Effectively combining error mitigation strategies

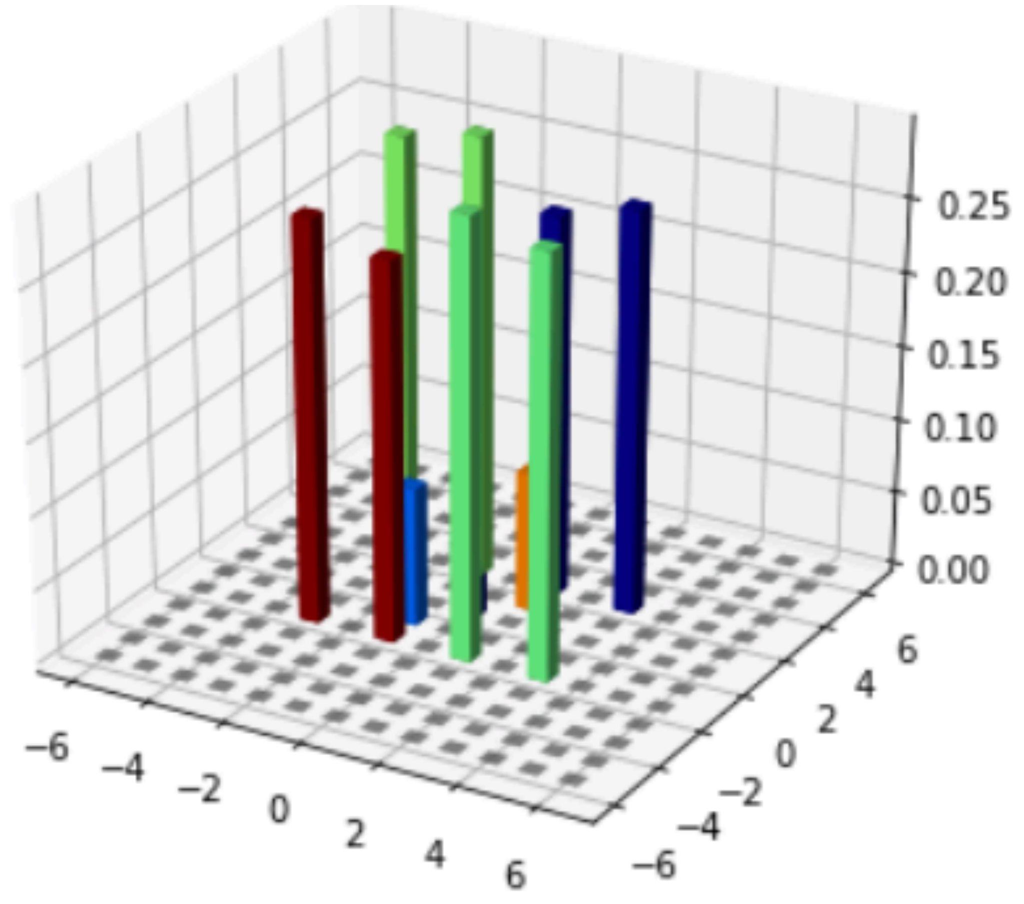
Spectral filtering (for parametrised circuits)

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

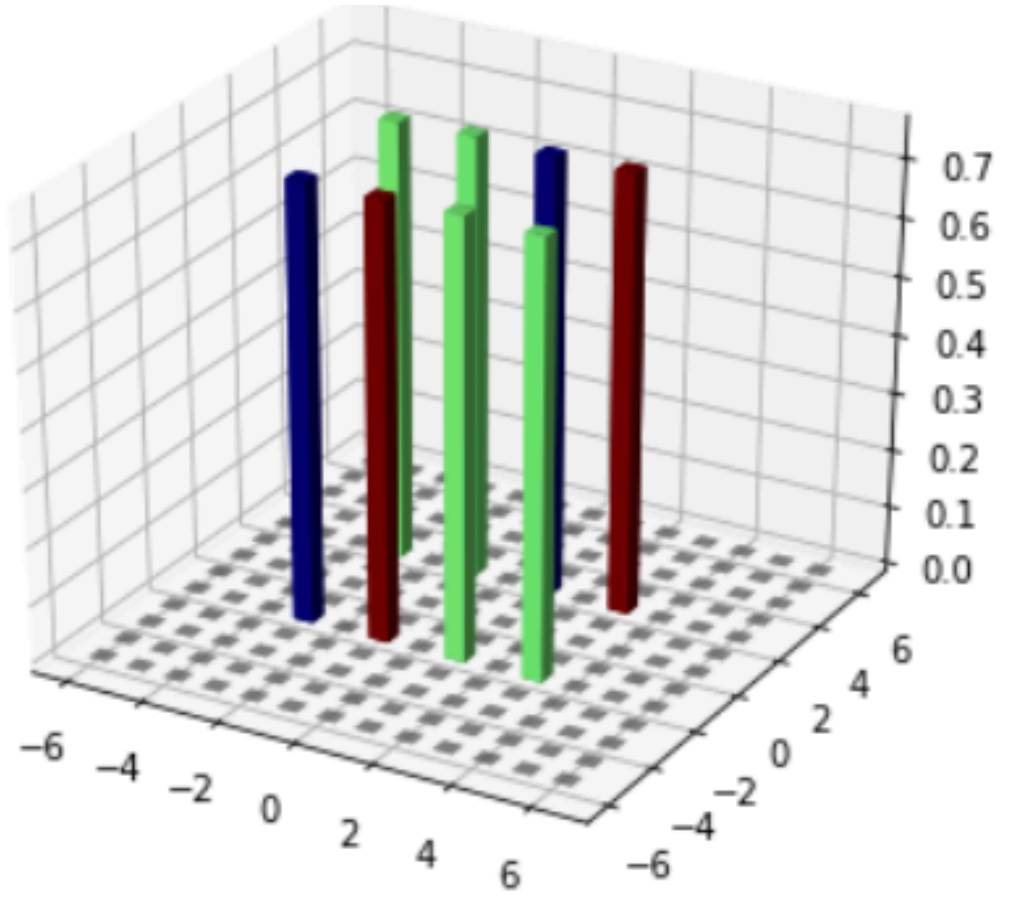
ibmq_sydney data



Threshold + CDR



Exact classical simulation



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

Spectral analysis for noise diagnostics and filter-based digital error mitigation arXiv:2206.08811, E. Fontana, I. Rungger, R. Duncan, CC,

Efficient recovery of variational quantum algorithms landscapes using classical signal processing. arxiv:2208.05958, E. Fontana, I. Rungger, R. Duncan, CC,

Summary

- QERMIT currently supports ZNE, CDR, Probabilistic error cancellation, SPAM reduction, frame randomisation; can combine existing methods and user-defined strategies
- Seamless integration with other SDKs via Pytket 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

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

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 Pytket 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 ?

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

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 computer**s.**
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).