SAND2023-06368C

Learning a quantum computer's capability

Daniel Hothem, Tommie Catanach, Kevin Young, and Timothy Proctor

5th Annual Workshop on Quantum Resource Estimation @ ISCA23

June 18, 2023

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Scenario 1

Can I solve my problem on a quantum computer?

Scenario 2

Image by DALLE-2

Commonality?

What do our users have in common?

They both need to know how well *specific* circuits will run on a *physical* quantum computer!

Capability Functions

- How do we formalize "What is my quantum computer capable of running?"
- Capability functions:

$Let\ \mathbb{Q}\ be\ a\ quantum\ processing\ unit\ (QPU), \mathcal{C}\ a\ collection\ of\ circuits\ executable\ on\ \mathbb{Q},\ and\ \mathcal{A}\ a\ collection\ of\ context\ variables. A\ \boldsymbol{capability\ function}\ is\ a\ real\ valued\ function\ s: \mathcal{C} \times \mathcal{A} \rightarrow \mathbb{R}\ defined\ by:$

$$s(c, a) = \epsilon[\gamma(c), \tilde{\gamma}(c, a)],$$

$where\ \gamma(c)\ is\ the\ ideal\ unitary\ implementing\ c, \tilde{\gamma}(c, a)\ is\ the\ noisy\ quantum\ channel\ implementing\ c\ on\ \mathbb{Q}\ in\ context\ a, and\ \epsilon\ is\ some\ success\ metric.$

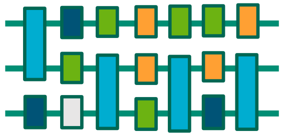
Both of scientists need to learn a ***capability function.***

Learning capability functions

- Incredibly challenging!
- Quantum computers are highly error-prone
- Noise is unique and complex
- Left with learning an approximation or surrogate function

Approximating Capability Functions

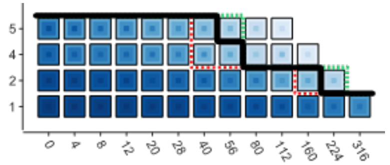
- Current quantum computers are noisy and error prone
- Phenomenological models[1]
  - Built on benchmarking tools
  - Rely on human extracted features
  - Poor performance
- Quantum process models[2]
  - Informed by “tomography”
  - Depend on circuit structure
  - Specious assumptions
  - Hard to scale
- Neural network models
  - Extract their own features
  - Few assumptions
  - Potentially scalable



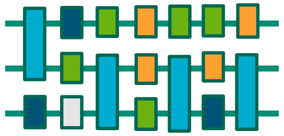
Map to 1-2 features

(w,d)

Look up performance on table



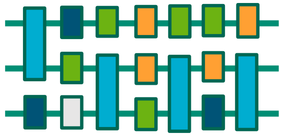
$p_{success}$

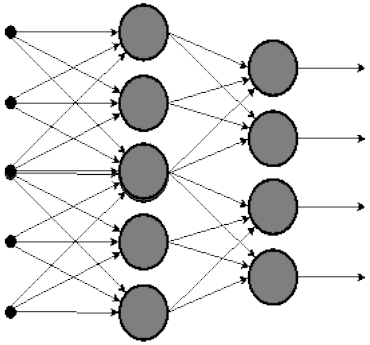


Multiple elements from a set of (very large) matrices

$$\{M, \mathcal{G}_1, \cdots, \mathcal{G}_n, \rho\}$$

$p_{success}$





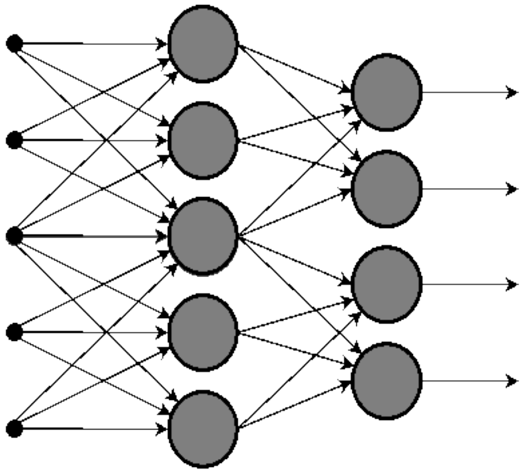
$p_{success}$

[1]*Characterizing Quantum Gates via Randomized Benchmarking*, Magesan et al, Phys. Rev. A **85,** 042311, 11 April 2012

[2]*Gate Set Tomography*, Nielsen et al, Quantum **5**, 2021

Background – Neural Networks

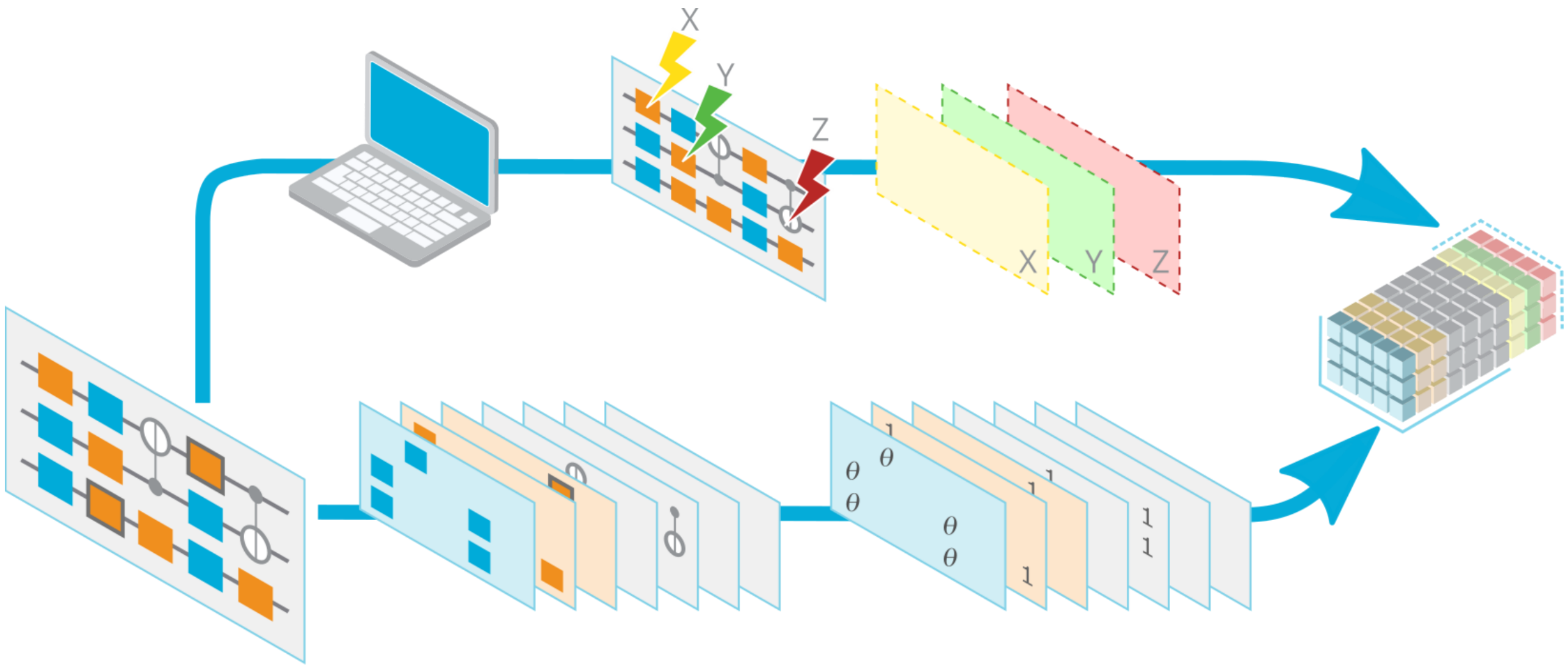
- Generic structure
  - Sequential layers of “neurons”
  - “Neurons” perform different operations
  - Previous layer feeds into the next layer
- Convolutional Neural Networks
  - Process images
  - Two components
    - Convolutional
    - Multi-layer Perceptron
  - Convolutional component extracts features
  - MLP processes features and makes predictions



The Big Picture

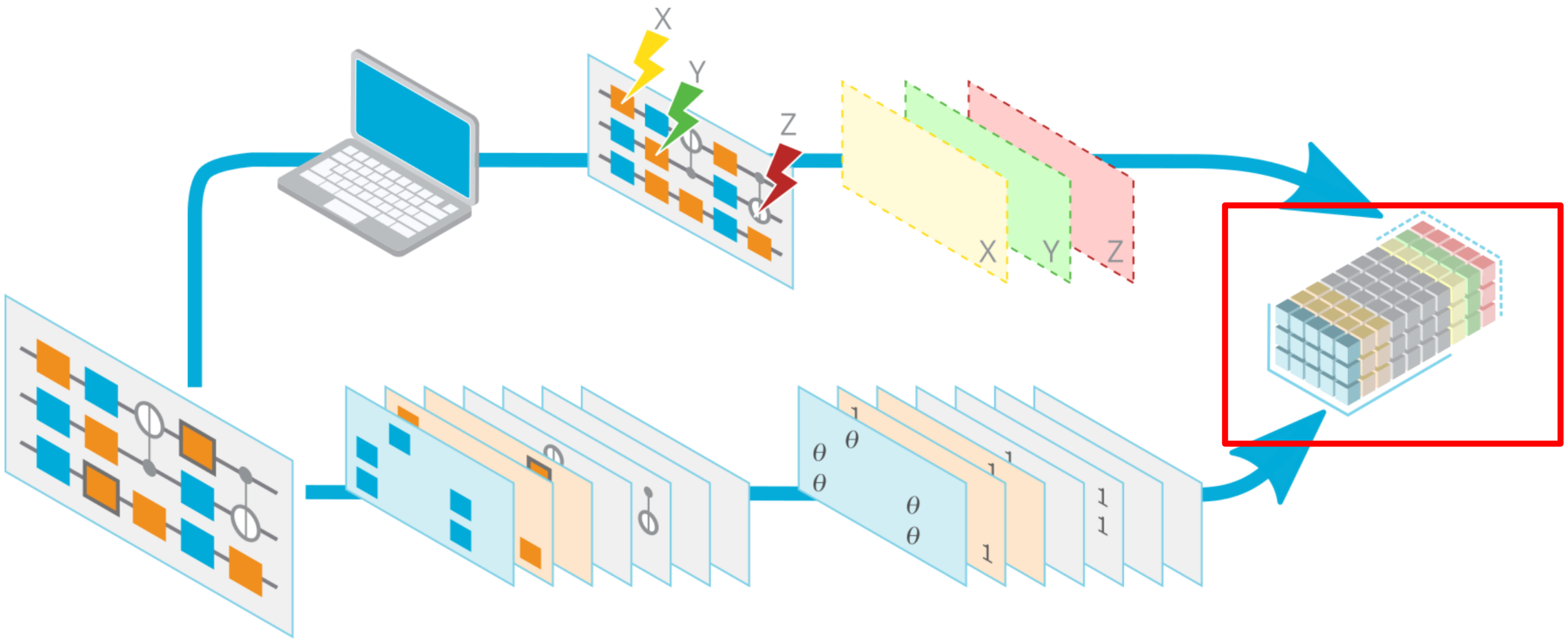
- The idea
  - Run lots of circuits on a device
  - Train neural networks to predict which circuits run successfully
  - Use the neural network as a proxy for the capability function!
- What circuits?
  - Focus on random (and periodic) mirror Clifford circuits
- What is "success?"
  - Focus on probability of successful trial
- Encoding the circuits?
  - Image encoding
- Which networks?
  - Convolutional neural networks

Circuit Encoding



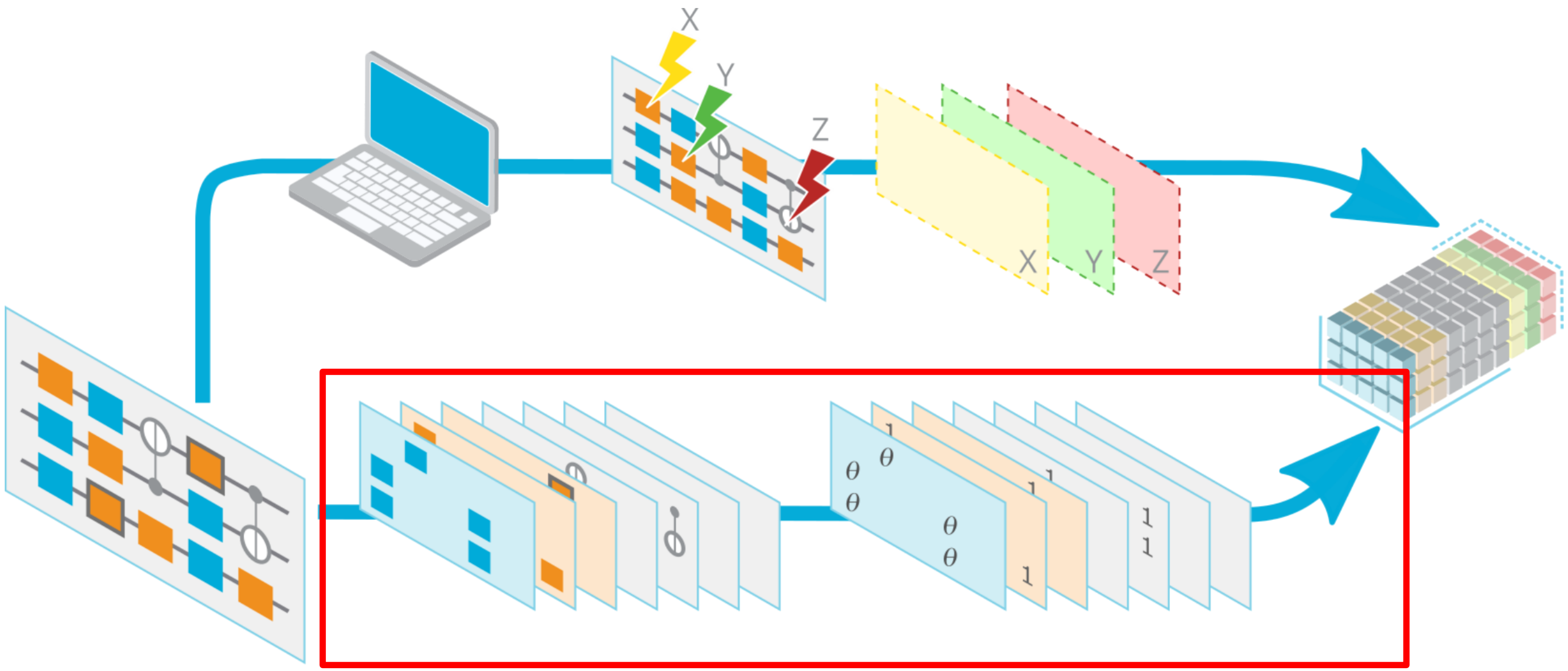
We need a way to input a quantum circuit into a convolutional neural network.

Circuit Encoding



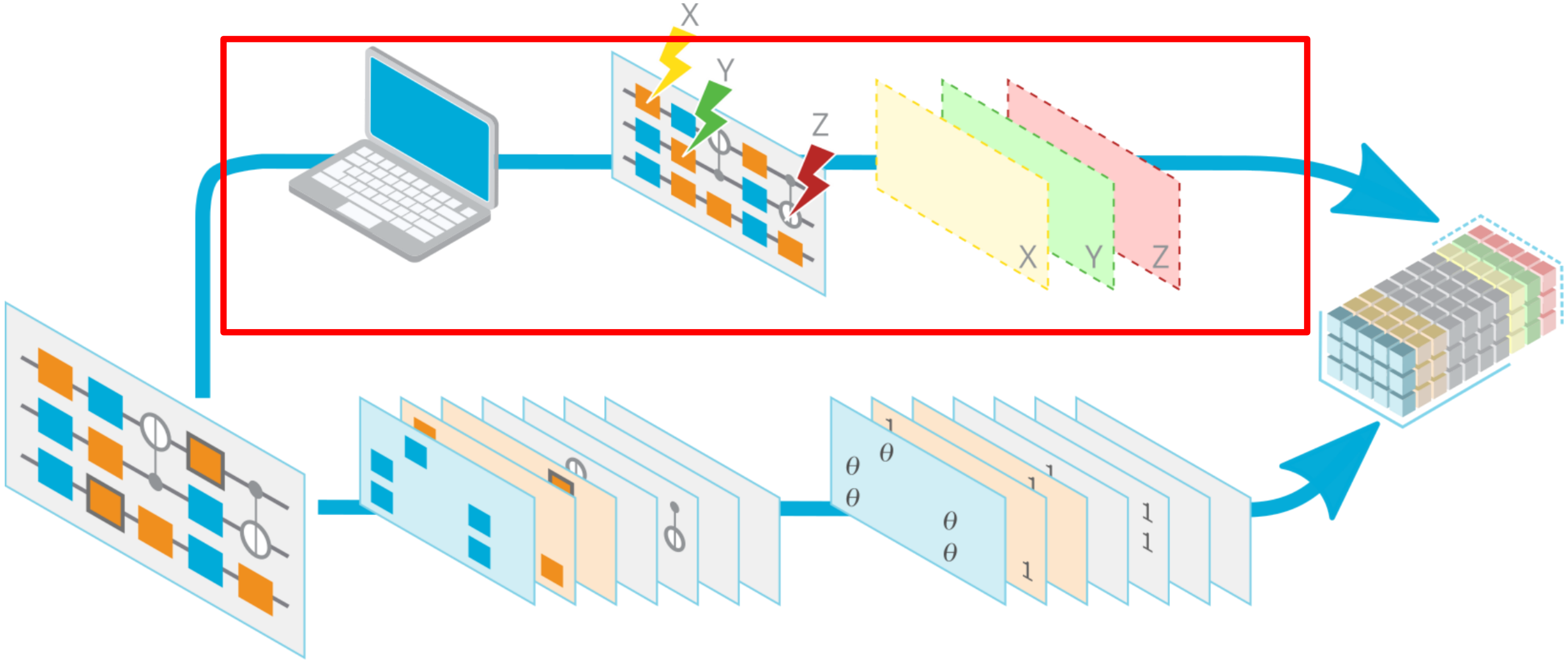
We will encode a quantum circuit as a 3D tensor (i.e., a color image).

Circuit Encoding



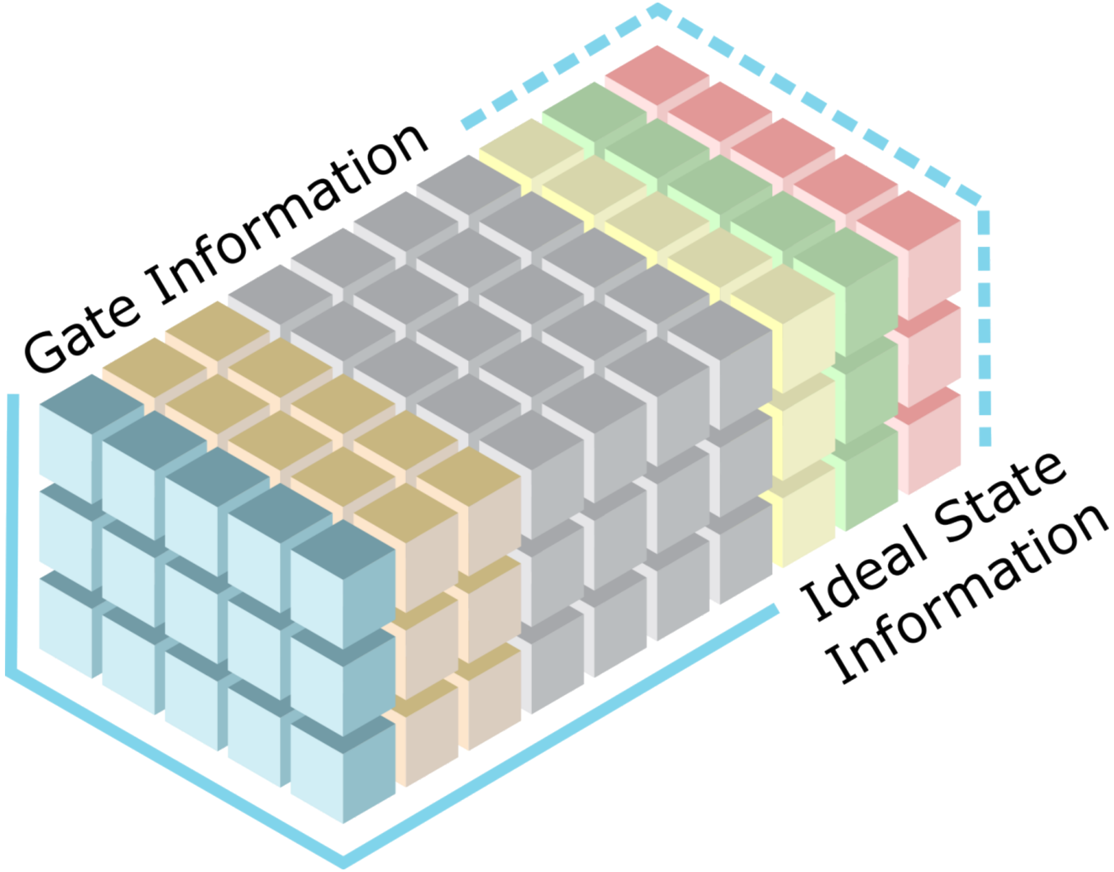
We will provide gate information…

Circuit Encoding



...and error-sensitivity information.

Circuit encoding

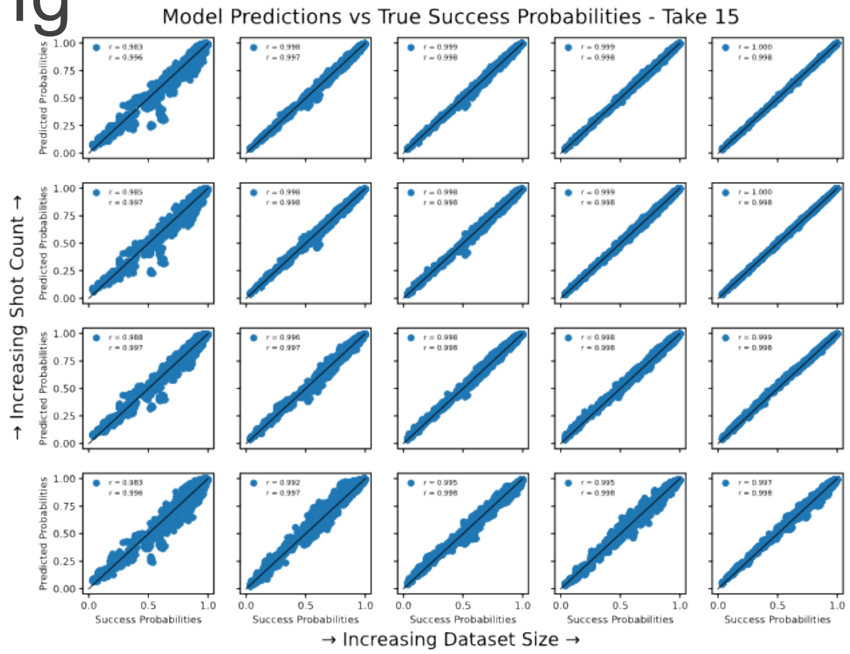


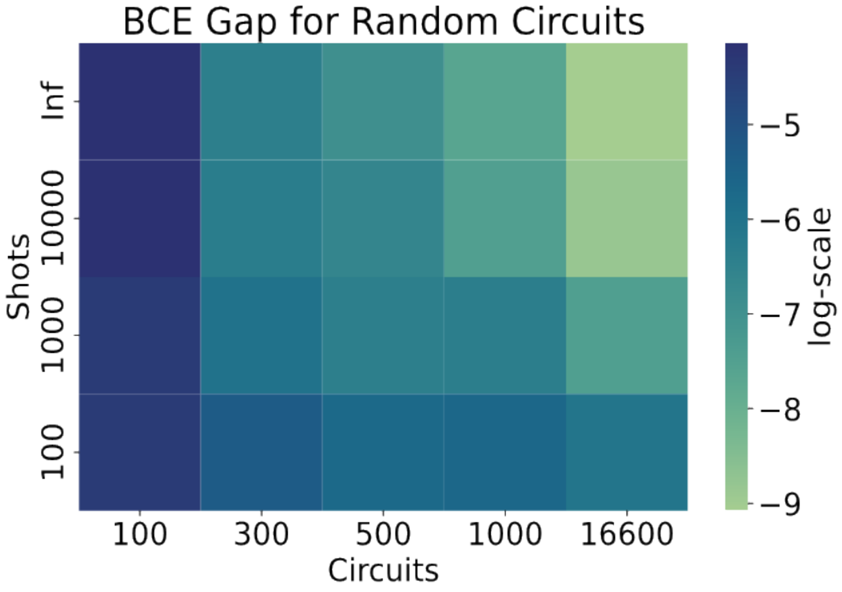
Our research

- Explore how dataset size and quality affects predictions
  - Multiple datasets with 100, 300, 500, 1000, 16600 circuits
    - 11 different 100 circuit datasets
    - 5 different datasets per circuit count for the rest
  - Each dataset was simulated at four levels of precision (shot count)
    - Same Markovian error model
- Predict under non-Markovian noise*
  - Very difficult for other techniques
- Experimental demonstration
- Impact of coherent noise

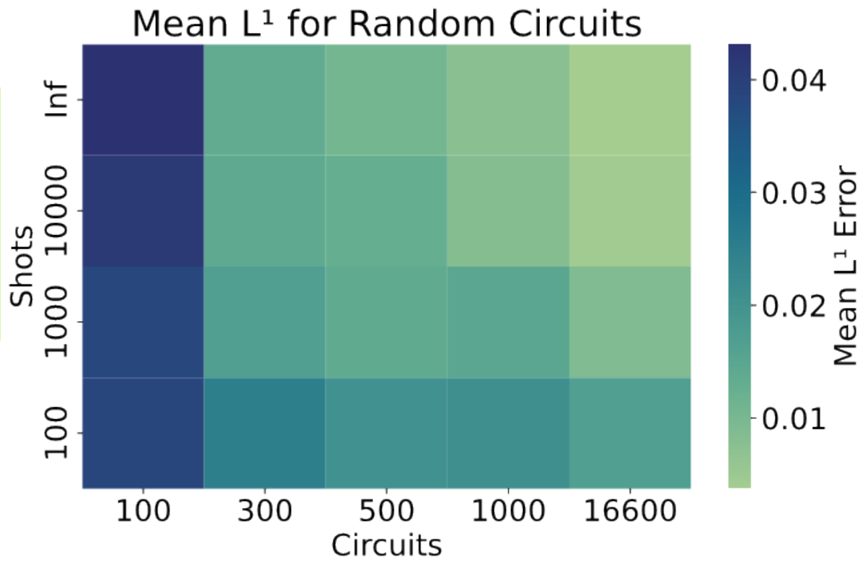
Dataset	Circuits	Shots	Take	Trials
Random	1000	100	11	35
Random	1000	100	12	N/A
Random	1000	100	13	N/A
Random	1000	100	14	N/A
Random	1000	100	15	N/A

Performance Scaling



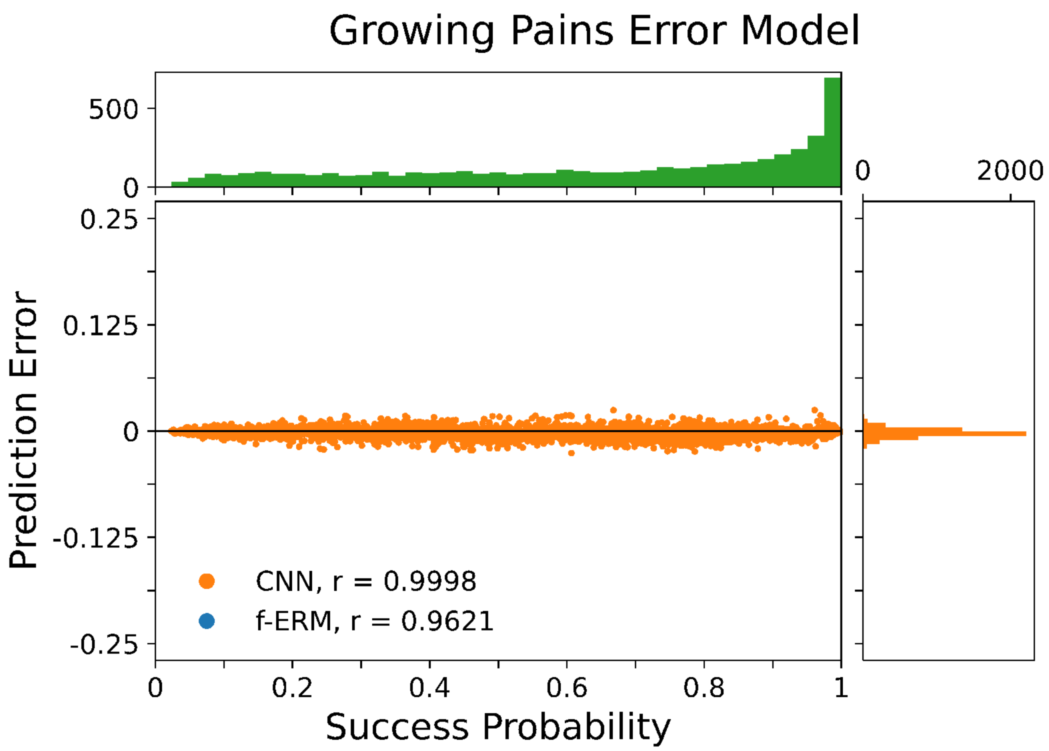


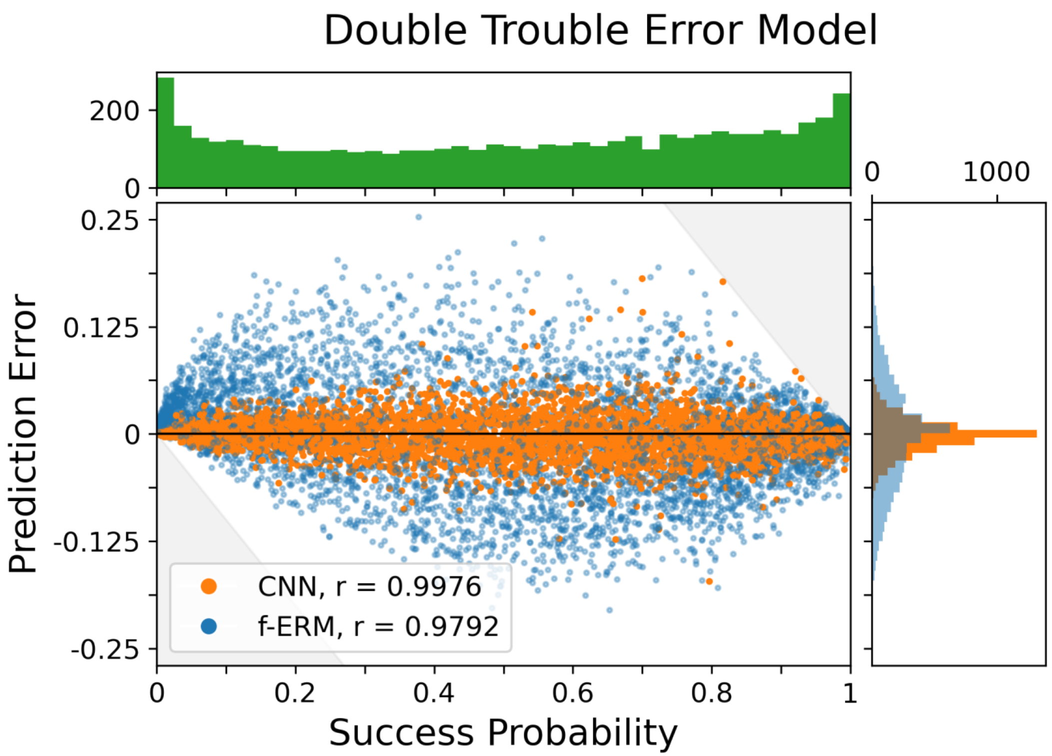
More circuits/Better data =
More useful features =
Tractable problem



Non-Markovian Noise

- 49-qubit wide circuits
  - First-order approximate simulator

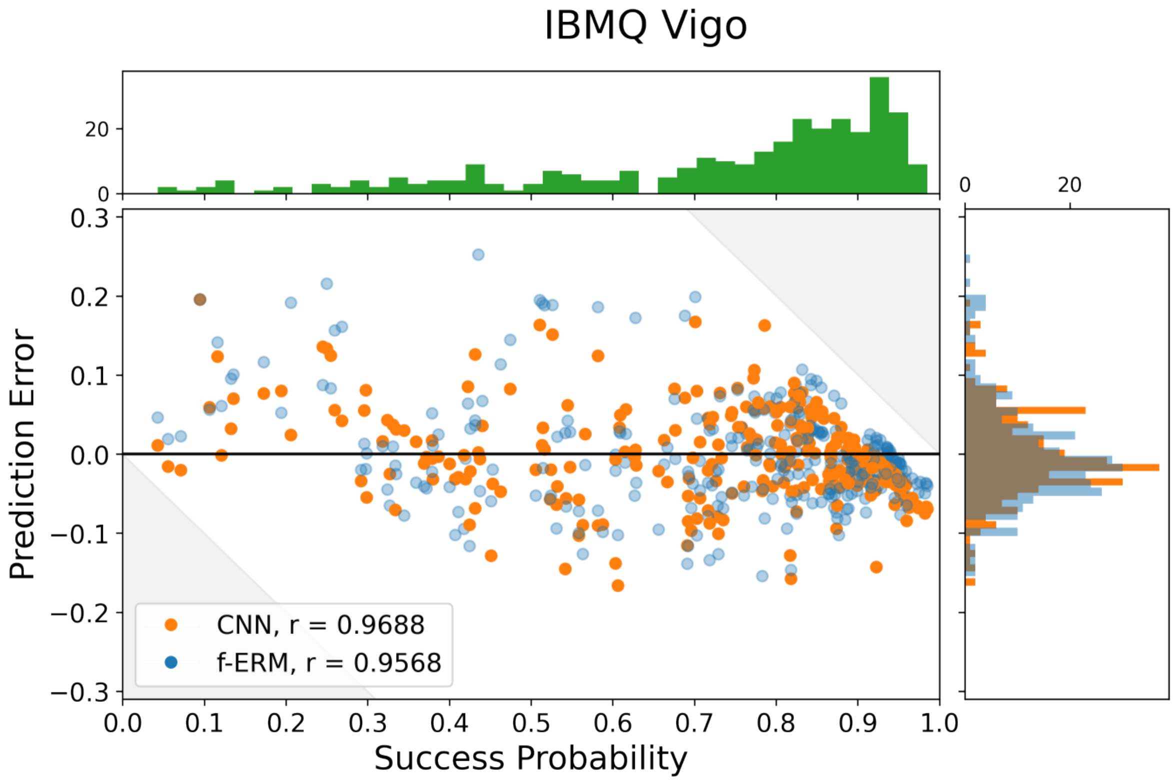


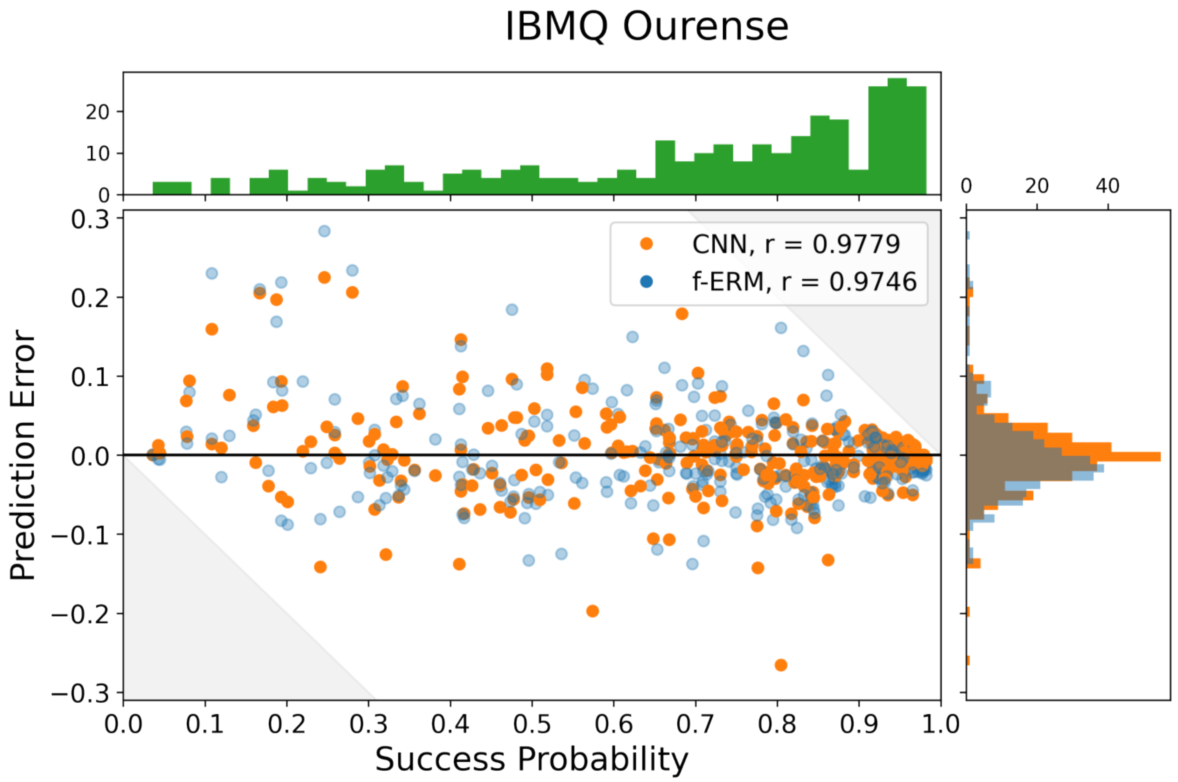


Outperform sophisticated phenomenological models[1].

[1] D. Hothem, J. Hines, K. Nataraj, R. Blume-Kohout, T. Proctor, *Predictive models from quantum computer benchmarks* (2023). arXiv:2305.08796v1

Experimental results



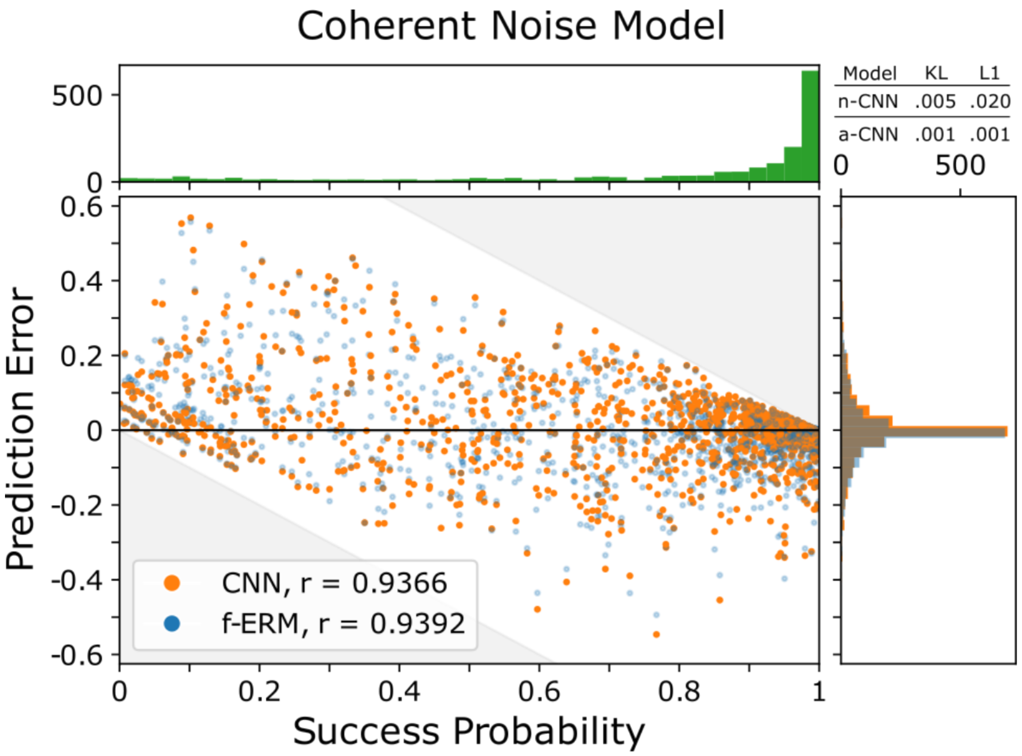


Decreased performance on experimental data. Why?

Possible reasons

- Model-free approach is a boon and a bane
  - Lack physics-intuition
- No notion of time
  - Context-dependent errors
- Limited error model information
- Pernicious coherent errors

Coherent noise



Coherent noise is the *major* cause of poor performance on experimental data.

Conclusions and next steps

- A QPUs performance is captured by a capability function
- Neural networks are promising surrogates for a particular capability function
- Coherent noise limits practical utility
- Modified approaches are needed
  - Additional error sensitivity information
  - New architectures (e.g., GNNs)