Quantum AI

Benchmarking NISQ and QEC experiments with tensor networks

Benjamin Villalonga

S. Boixo, S. Mandrà, M. Newman, N. Shutty, K. Kechedzhi, S. Isakov, X. Mi, V. Smelyanskiy, and many others

Workshop on Tensor Networks

IPAM @ UCLA

Los Angeles 2024

Google

Outline

1. Motivation
    - The *status quo* of quantum computing experiments.
    - Two use cases for tensor networks
2. Benchmarking large NISQ experiments
    - A simulation problem
    - Making brute force TN contractions less brute: the RCS case study
    - Some results
    - The future of NISQ applications: noise vs. computational volume
3. Decoding early QEC demonstrations
    - The setup: 3 performance contributing factors
    - The hyper-graph error model.
    - The (maximum-likelihood) decoding problem
    - A TN maximum-likelihood decoder for all hyper-graph error models
    - Results
4. Conclusion

Outline

1. Motivation
   - The *status quo* of quantum computing experiments.
   - Two use cases for tensor networks
2. Benchmarking large NISQ experiments
   - A simulation problem
   - Making brute force TN contractions less brute: the RCS case study
   - Some results
   - The future of NISQ applications: noise vs. computational volume
3. Decoding early QEC demonstrations
   - The setup: 3 performance contributing factors
   - The hyper-graph error model.
   - The (maximum-likelihood) decoding problem
   - A TN maximum-likelihood decoder for all hyper-graph error models
   - Results
4. Conclusion

The *status quo* of quantum computing experiments

Useful or physically motivated **applications** (error mitigation):

- Topological phases of matter, majorana edge modes, non-abelian statistics (Satzinger et al. 2021, Mi et al. 2022, Andersen et al. 2022, ...)
- Time crystals (Mi et al. 2022, ...)
- Information scrambling quantum systems (Mi et al. 2021)
- Floquet evolution of transverse field Ising model (Y. Kim et al., 2023)
- MERA implementation (Haghshenas et al. 2023)
- Dissipative cooling (Mi et al. 2023)
- Graph problems (Deng et al. 2023)
- Other experiments from Harvard/QuEra, IBM, Quantinuum, USTC, ...
- . . .

NISQ

Beyond-classical demonstration attempts (usually no error mitigation involved):

- Random circuit sampling (Arute et al. 2019, Wu et al., 2021, Zhu et al. 2022, Morvan et al. 2023, Bluvstein et al. 2024)
- Gaussian BosonSampling (Zhong et al. 2020, Zhong et al. 2021, Madsen et al. 2022, Deng et al. 2023)

Early demonstrations of **quantum error correction**:

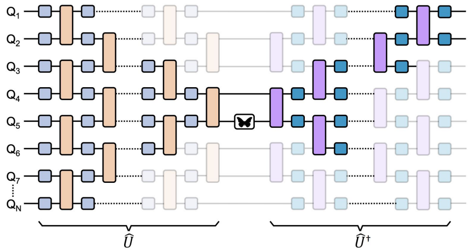
- Surface code implementations (Krinner et al. 2022, Zhao et al. 2022)
- Surface code error suppression (Google 2022)
- Other codes (Ofek et al. 2016, Fluhmann et al. 2019, Champagne-Ibarcq et al. 2020, Grimm et al. 2020, Chen et al. 2021, Egan et al. 2021, Ryan-Anderson et al. 2021, Sundaresan et al. 2022)

QEC

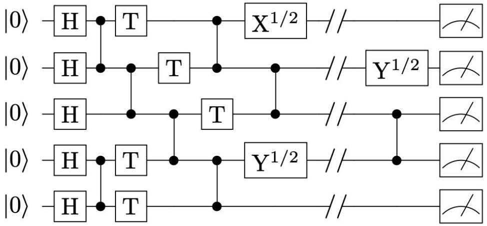
Two use cases for tensor networks

NISQ

Benchmarking experiments with tensor networks



OTOC
(Mi, et al. 2021)



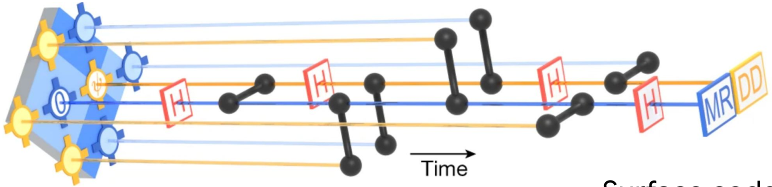
RCS
(Boixo, et al. 2017)

Exploiting structure: compressibility, low entanglement, ...

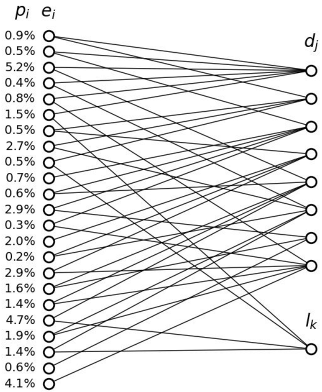
Worst case: brute force contraction

QEC

Decoding (and benchmarking) with tensor networks



Surface code
(Google. 2023)



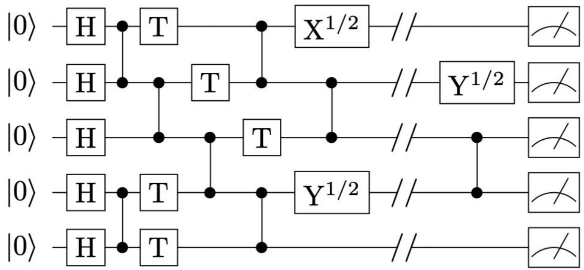
- **Mapping** decoding to TN contraction
- **Contract** efficiently

Outline

1. Motivation
    - The *status quo* of quantum computing experiments.
    - Two use cases for tensor networks
2. Benchmarking large NISQ experiments
    - A simulation problem
    - Making brute force TN contractions less brute: the RCS case study
    - Some results
    - The future of NISQ applications: noise vs. computational volume
3. Decoding early QEC demonstrations
    - The setup: 3 performance contributing factors
    - The hyper-graph error model.
    - The (maximum-likelihood) decoding problem
    - A TN maximum-likelihood decoder for all hyper-graph error models
    - Results
4. Conclusion

A simulation problem (1 / 2)

Experiment



- Observable
- Probability amplitude
- Samples
- ...

Characterization:

- Fidelity?
- Is the experiment giving right results?
- What kind of results do we expect?
- ...

Challenging beyond-classical claims:

- What's the classical computational cost?
- What are the hardness guarantees?
- Is the experiment beyond-classical?
- ...

(classical) **Simulation**

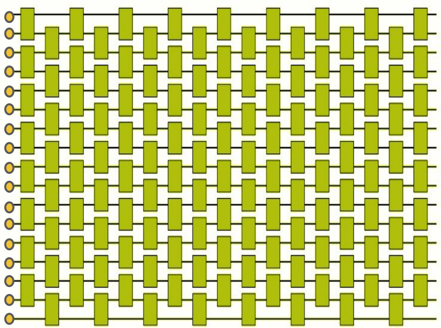
A simulation problem (2 / 2)

Under special circumstances there are specialized techniques:

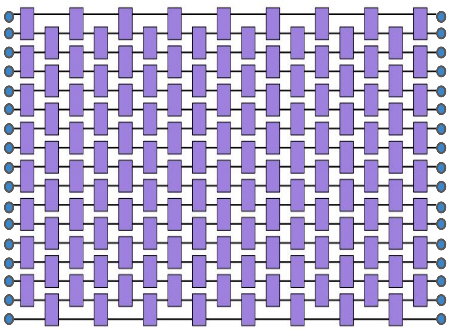
- Clifford circuits
- Clifford + T circuits
- Matchgate circuits
- Localized dynamics
- Large noise rates (hinder entanglement formation)
- ...

In the generic case we need brute force

Circuit



Tensor network

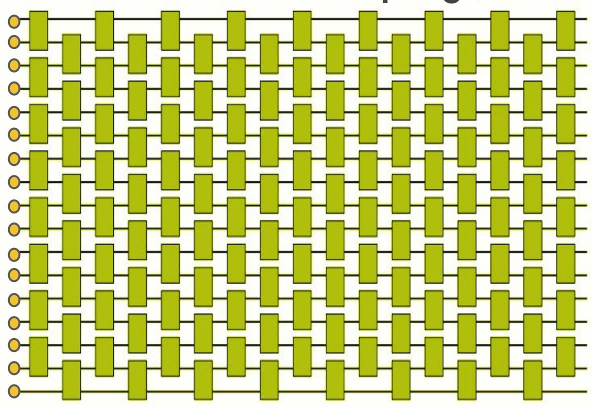


- Observable
- Probability amplitude
- Samples
- ...

As **quantity** or **primitive** for it

Making brute force less brute: the RCS case study (1/3)

Random Circuit Sampling (RCS)



```
11101101...
01111011...
10110011...
11011110...
01001100...
01101101...
10001100...
10110011...
11010101...
01001101...
11110110...
01001111...
00110000...
11010111...
11100110...
00101101...
10110001...
10111010...
```

N ($\sim 10^6$)

Estimate fidelity **f** (from samples)

f > 0 within a few σ's?

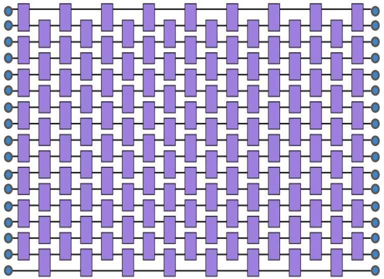
Can a classical computer perform this task (*in a reasonable amount of time*)?

Fairly strong complexity theory guaranties for the hardness of this task.

(Boixo et al. 2016, Aaronson et al. 2016, Bouland et al. 2019 & 2021, Movassagh et al. 2020, ...)

Tensor network

Classical adversary



Sampling algorithm (Markov et al. 2018):

1. Compute **p(x)** for bit strings **x** chosen uniformly at random
2. Accept **x** with probability **p(x)N/M**

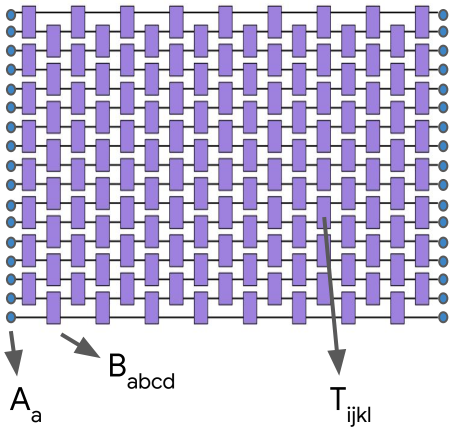
Acceptance ratio **1/M ~ 1/10**

Modified rejection sampling: **frugal sampling**


```
11101101...
01111011...
10110011...
11011110...
01001100...
01101101...
10001100...
10110011...
11010101...
01001101...
11110110...
01001111...
00110000...
11010111...
11100110...
00101101...
10110001...
10111010...
```

Making brute force less brute: the RCS case study (2/3)

p(x) =



$$\underbrace{\sum_c \sum_a \sum_f \sum_b}\ldots A_a B_{abcd}\ldots T_{ijkl}\ldots$$

Order of contraction dramatically affects computational cost.
Time and memory requirements lower bounded by **treewidth of line graph** (Markov & Shi 2008)

Goal: optimize tensor network contraction ***ordering*** **(O)**. (Gray & Kourtis 2020)

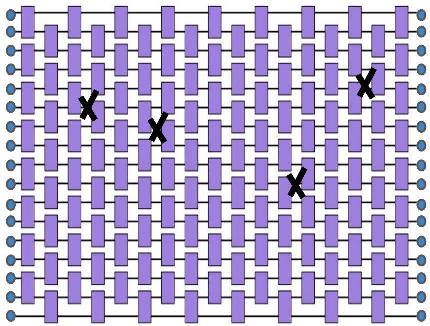
Memory?

For current experiments this leads memory requirements ~10^4× the total memory of the largest supercomputer on Earth

Solution: project (slice) a set of variables **(S)** and perform sum *a posteriori* (Alibaba, 2018).

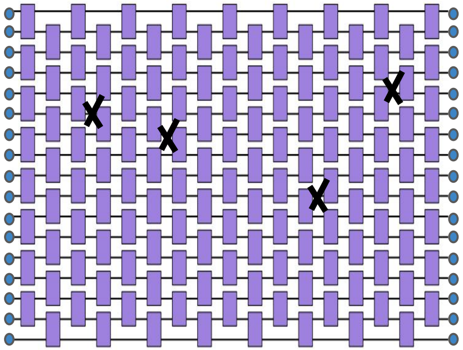
✅Alleviates memory requirements *and* parallelizes computation

❎Need to contract an exponential number TNs



Quantum AI

Making brute force less brute: the RCS case study (3/3)



Optimization problem:

Contraction cost is function of ordering and slices **C(O, S)**.
Memory usage **M(O, S)<M*** (total memory available).

Pedantic detail of our encoding:

We take slices from an ordered set of candidates **P** until **M<M*** is satisfied, so our cost function is really **C(O, P)**.
Contraction cost is function of ordering and slices **C(O, S)**.

Plethora of "tricks" from the literature can be beneficial in practice:

- **Sparse output:** so millions of amplitudes can be computed with a single contraction (**F. Pan** et al. 2021)
- **Details of hardware gates:** `fSim` gate can be exploited for faster contractions (Google 2019 & **F. Pan** et al. 2021)
- **Memoization:** reuse of intermediate computations across branches of the computation (Kalachev et al. 2021)
- **Experimental fidelity:** low target fidelity speeds up simulation (Markov et al. 2018, Villalonga et al. 2019)

All these accounted for in **C(O, P)**.
Highly optimized evaluation of **C(O, P)**. Current experiments are close to ~1000 two-qubit gates.

Example results (1/2)

Optimized runtimes for RCS experiments:

Exp.	1 amp.	1 million noisy samples		
	FLOPs	FLOPs	XEB fid.	Time
SYC-53 [4]	6×10^{17}	2×10^{17}	2×10^{-3}	6 s
ZCZ-56 [5]	6×10^{19}	6×10^{19}	6×10^{-4}	20 min
ZCZ-60 [6]	1×10^{21}	1×10^{23}	3×10^{-4}	40 days
SYC-70	5×10^{23}	6×10^{25}	2×10^{-3}	50 yr
SYC-67	2×10^{23}	2×10^{37} 2×10^{28} 2×10^{25}	1×10^{-3}	1×10^{13} yr 1×10^{4} yr* 12 yr**

Google, 2023

Parallelizing over independent GPUs on Frontier

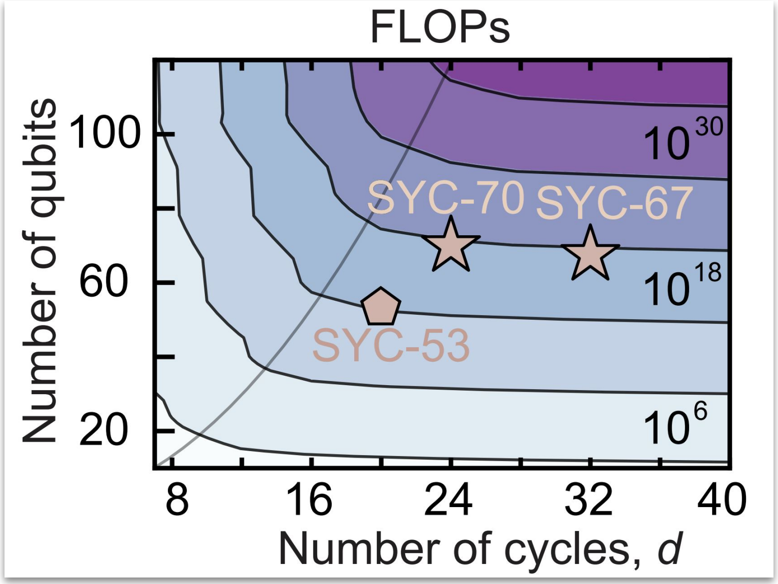
*Assuming distributed contractions over all RAM.

**Assuing distributed contractions using secondary storage.

*&** without inter-node communication times (for stronger adversary against BC claim)

Example results (1/2)

Complexity vs. circuit size:



Google, 2023

2D architecture (**L × L**) similar to experiment

- At low depth, cost **exp(d × L)**
- At large depth, cost **exp(L × L) = exp(#qubits)**

The future of NISQ applications: noise vs. computational volume

Signal (fidelity) decreases exponentially with volume of computation (for generic circuits, ~#two-qubit gates).

Computations are limited to finite sizes, which limits their classical computational cost.

RCS experiments beyond classical?	**Useful / physical** experiments beyond classical?
Strongly established	**Not yet**

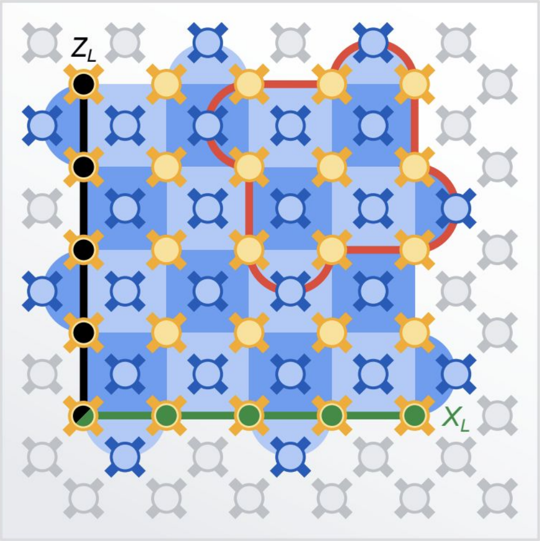
Strongly supported by highly optimized TN contraction results

Will there be a useful NISQ application before QEC is achievable?

Outline

1. Motivation
    - The *status quo* of quantum computing experiments.
    - Two use cases for tensor networks
2. Benchmarking large NISQ experiments
    - A simulation problem
    - Making brute force TN contractions less brute: the RCS case study
    - Some results
    - The future of NISQ applications: noise vs. computational volume
3. Decoding early QEC demonstrations
    - The setup: 3 performance contributing factors
    - The hyper-graph error model.
    - The (maximum-likelihood) decoding problem
    - A TN maximum-likelihood decoder for all hyper-graph error models
    - Results
4. Conclusion

The setup: 3 performance contributing factors (1/2)



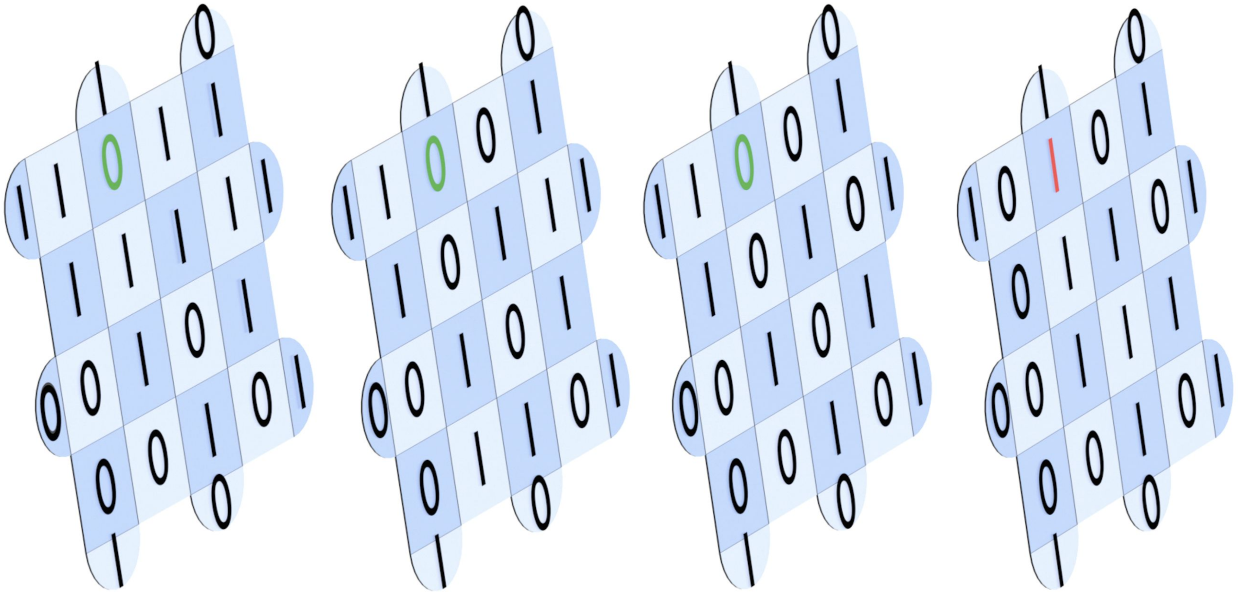
For low enough error rates:

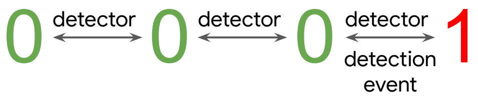
- Encode logical operators (qubit) over many physical qubits
- Early demonstration with **memory experiment:**
  - Initialize system in eigenstate of **X** or **Z**
  - Run several rounds of surface code, each one measuring parity checks (operators)
  - Decode: infer from parity checks whether logical operator has changed value
- Decoding has as input an understanding of physical errors: **error model**

What determines the quality of the experiment (of the logical qubit)?

- Hardware: roughly physical error rates
- Error model
- **Decoder**

The setup: 3 performance contributing factors (2/2)



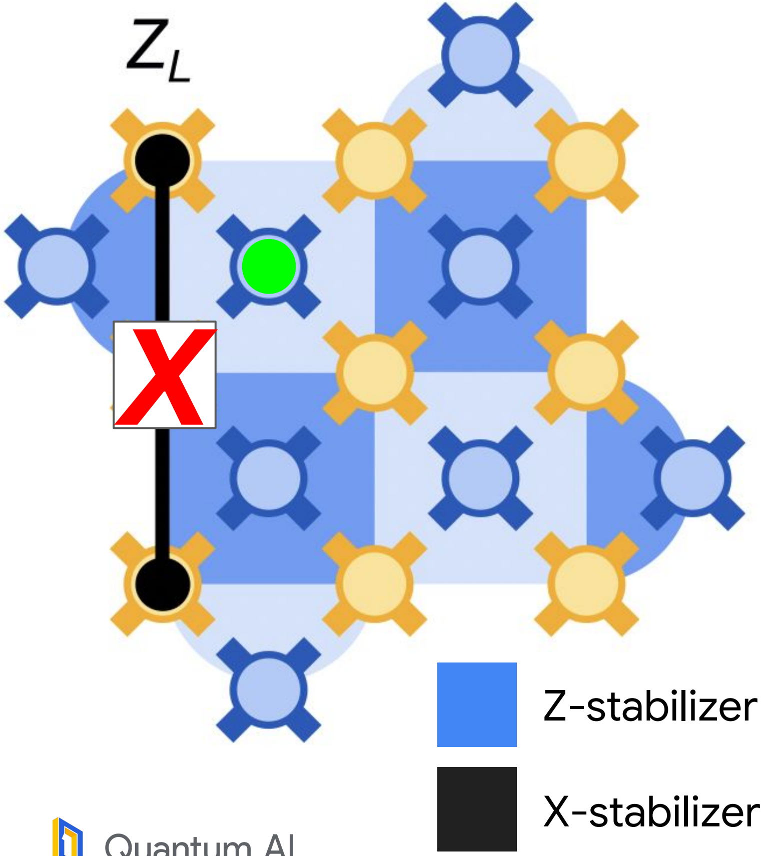


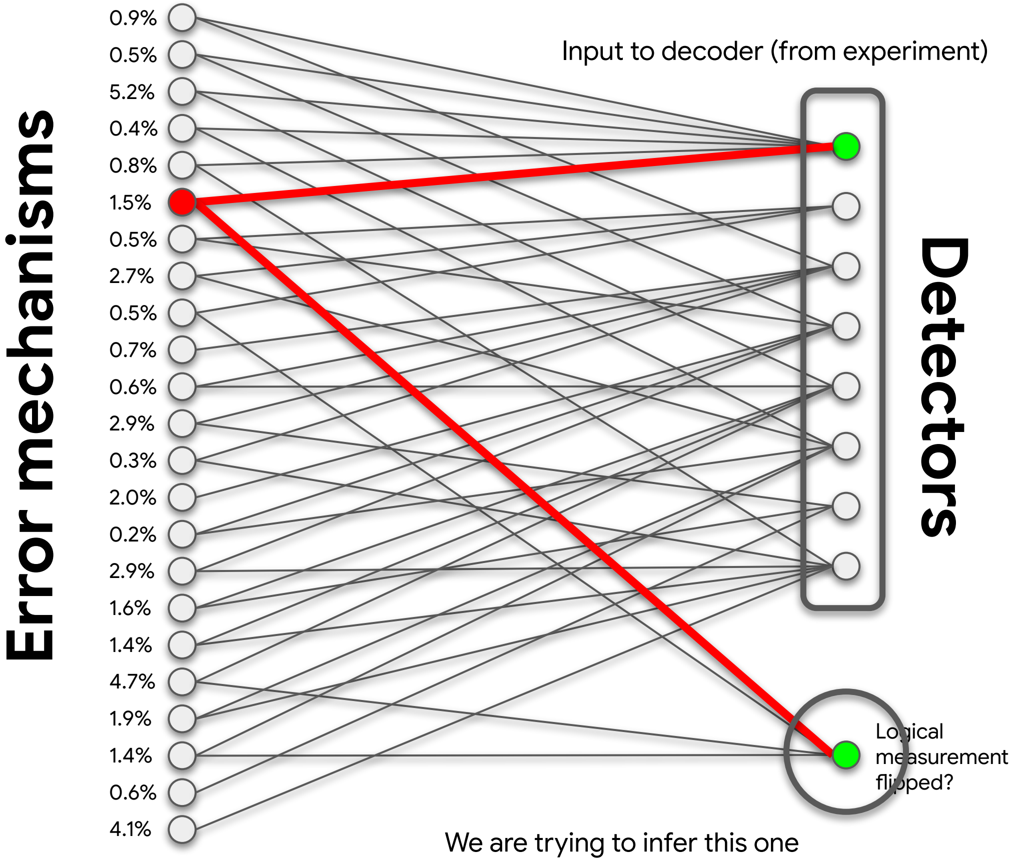
Detectors — comparisons of measurements that should agree

Detection events — when they don't

From detection events, can we make a guess on the the logical operator flipped/not flipped?

The hyper-graph error model



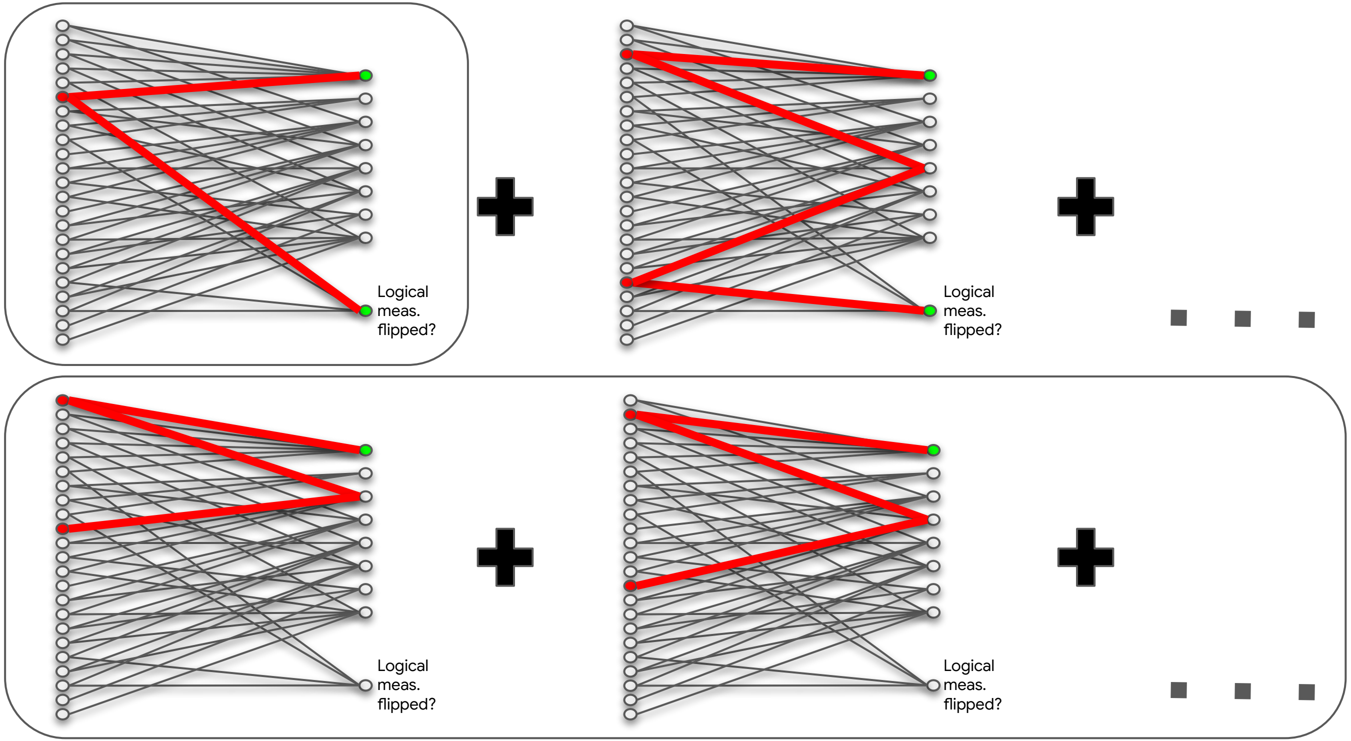


Quantum AI

The (maximum-likelihood) decoding problem

Most likely error...

... might not belong
to the most likely
set of errors.



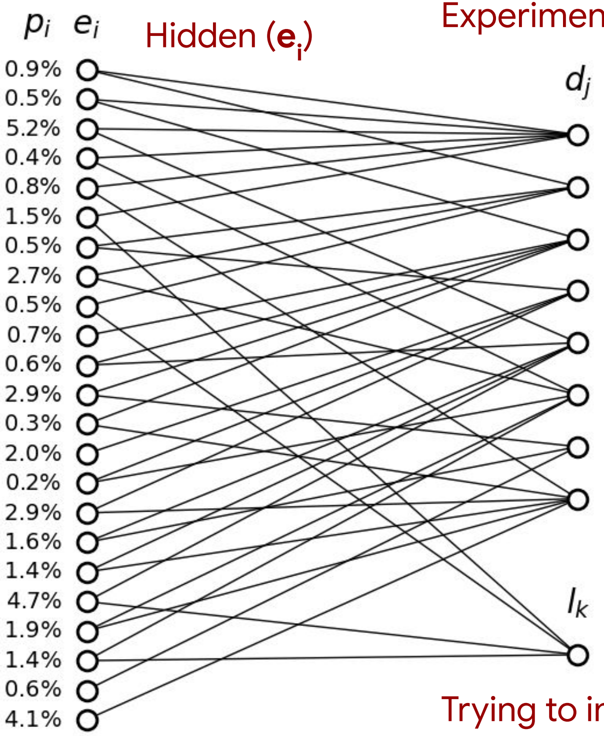
Maximum likelihood decoding = ***optimal*** decoding

A tensor network ML decoder for all hyper-graph error models (1/5)

Error model (**p$_i$**+ graph)

Hidden (**e$_i$**)

Experiment (**d$_j$**)

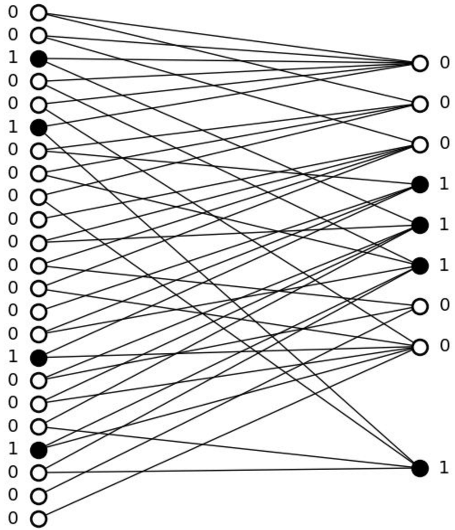


Trying to infer (**l$_k$**)

$$\Pr(\vec{e}) = \prod_i p_i^{e_i} \cdot (1-p_i)^{1-e_i}$$

$$\vec{d} = \vec{f}(\vec{e}) \text{ and } \vec{l} = \vec{g}(\vec{e})$$

$$L\left(\vec{l}\middle|\vec{d}\right) \propto \sum_{\vec{e}:\left[\vec{f}(\vec{e})=\vec{d}\right]\wedge\left[\vec{g}(\vec{e})=\vec{l}\right]} \Pr(\vec{e})$$



A tensor network ML decoder for all hyper-graph error models (2/5)

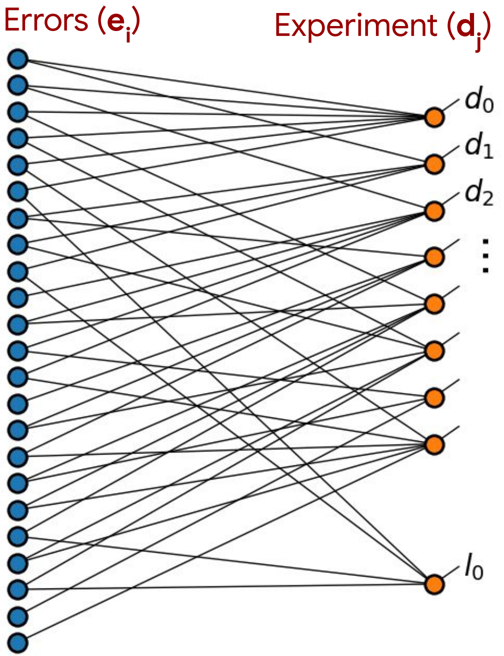
(Initial proponent: Bravyi et al. 2014)

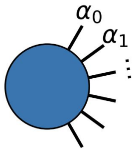
$$\Pr(\vec{e}) = \prod_i p_i^{e_i} \cdot (1-p_i)^{1-e_i}$$

$$\vec{d} = \vec{f}(\vec{e}) \text{ and } \vec{l} = \vec{g}(\vec{e})$$

$$L\left(\vec{l}|\vec{d}\right) \propto \sum_{\vec{e}:[\vec{f}(\vec{e})=\vec{d}]\wedge[\vec{g}(\vec{e})=\vec{l}]} \Pr(\vec{e})$$

$$L\left(l_0|\vec{d}\right) =$$

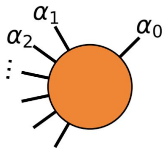




$$= \begin{cases} p_i & \text{if } \alpha_0 = \alpha_1 = \ldots = 1 \\ 1-p_i & \text{if } \alpha_0 = \alpha_1 = \ldots = 0 \\ 0 & \text{otherwise} \end{cases}$$

Propagates error to:

- Detectors
- Logical operator(s)



$$= \begin{cases} 1 & \text{if } \alpha_0 + \alpha_1 + \ldots \text{ even} \\ 0 & \text{if } \alpha_0 + \alpha_1 + \ldots \text{ odd} \end{cases}$$

Two ways of seeing it:

- Enforces right parity with d_j and l_k
- Kills error configurations that violate constraints

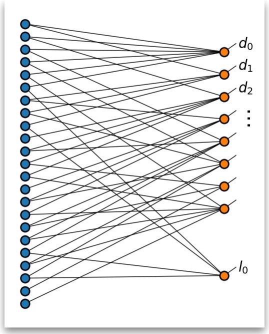
(Piveteau et al. 2023 also uses error hyper-graph as starting point)

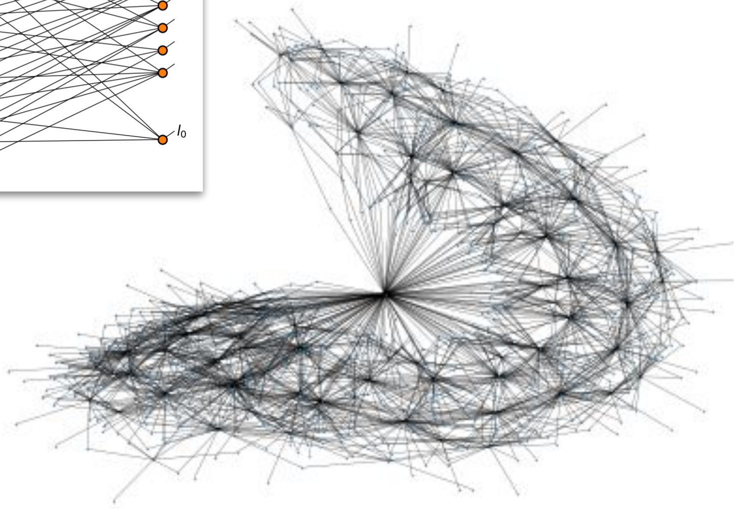
Decoding:

$$L\left(l_0 = 0|\vec{d}\right) \geq L\left(l_0 = 1|\vec{d}\right)$$

Quantum AI

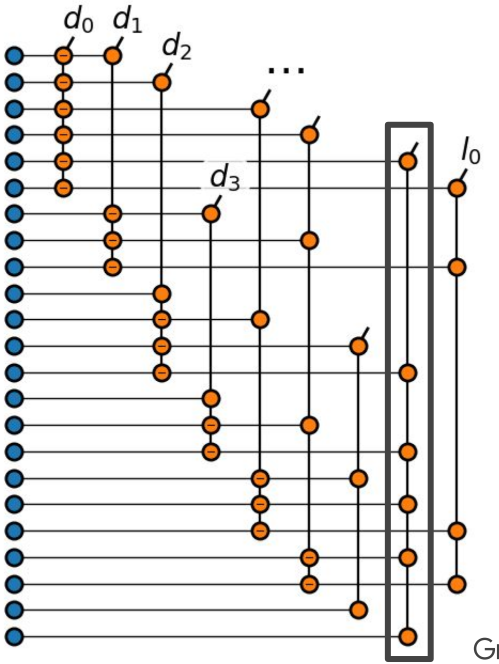
A tensor network ML decoder for all hyper-graph error models (3/5)

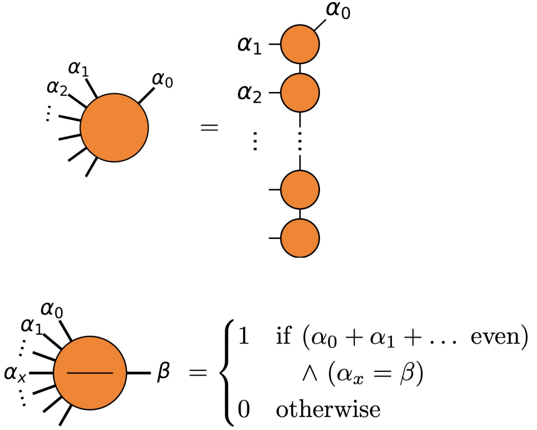




Exact contraction not scalable

Approximate contraction?

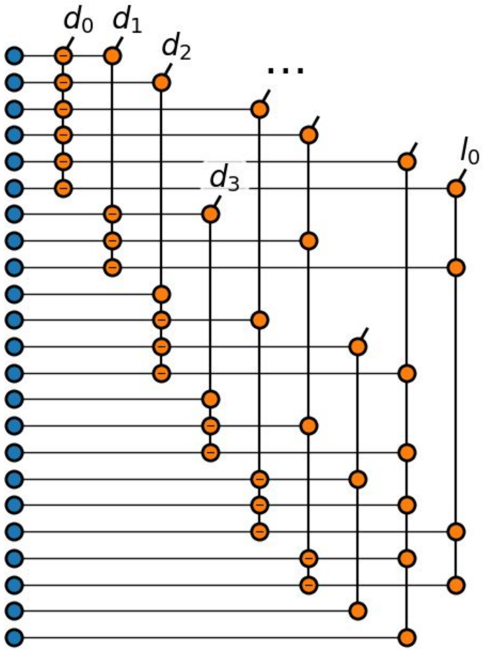




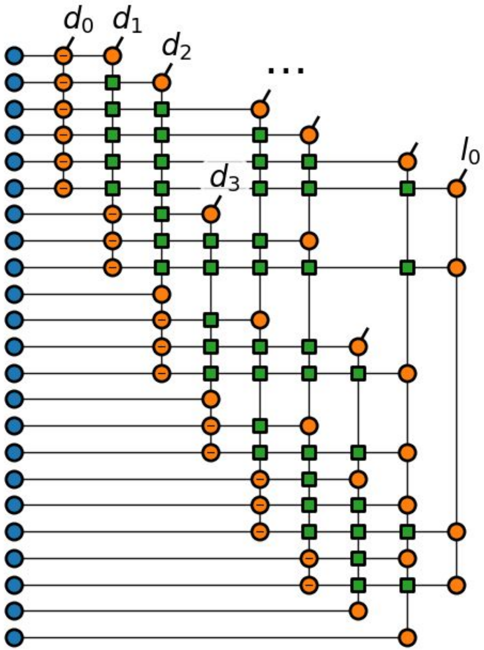
Graph structure is **adjacency matrix** of error hyper-graph

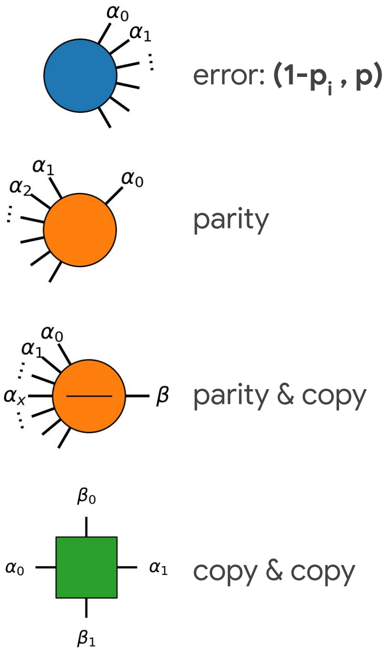
A tensor network ML decoder for all hyper-graph error models (4/5)

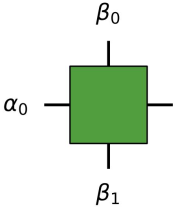
$$L\left(l_0|\vec{d}\right) =$$



=

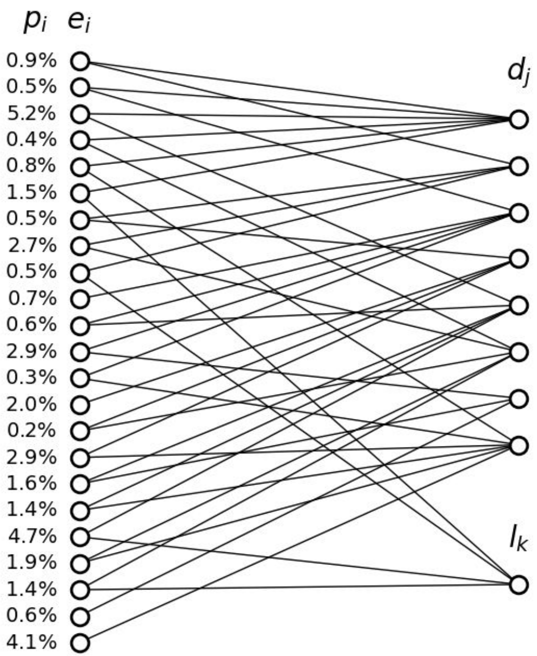


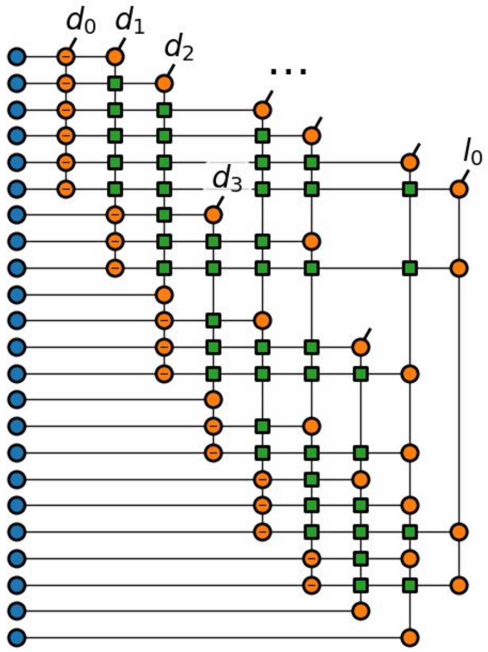




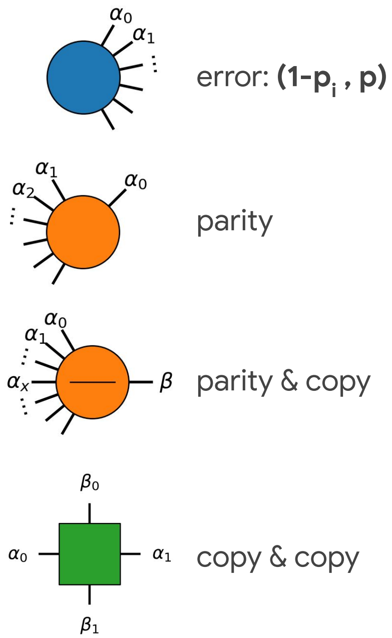
$$= \begin{cases} 1 & \text{if } (\alpha_0 = \alpha_1) \ \wedge \ (\beta_0 = \beta_1) \\ 0 & \text{otherwise} \end{cases}$$

A tensor network ML decoder for all hyper-graph error models (5/5)





Approximate contraction:
MPS evolution with finite **χ**
(left to right)



Decoding:

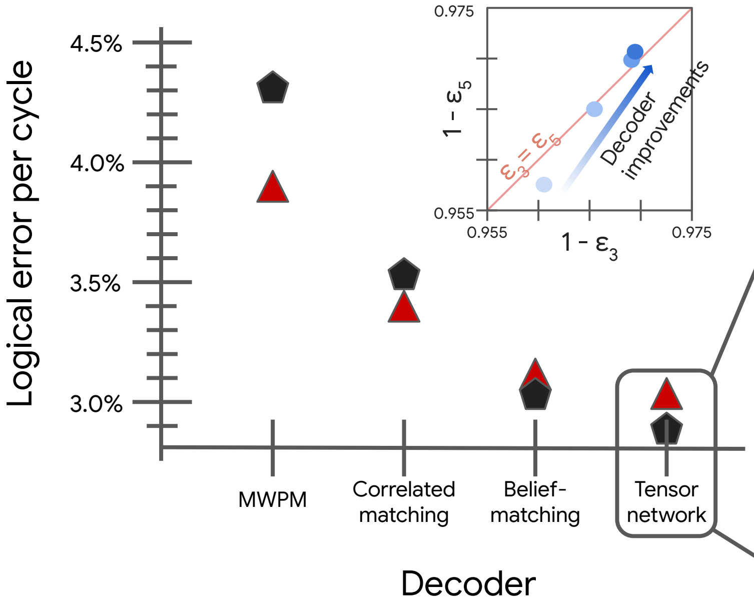
$$L\left(l_0 = 0|\vec{d}\right) \geq L\left(l_0 = 1|\vec{d}\right)$$

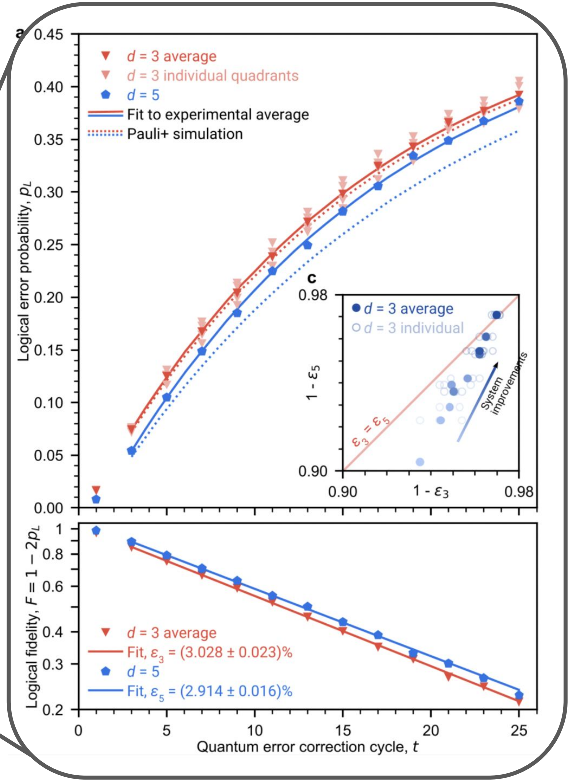
?

Results (1/2)

Milestone experiment on error suppression using the surface code

Google, 2023 - *Nature* 614, no. 7949 (2023): 676-681



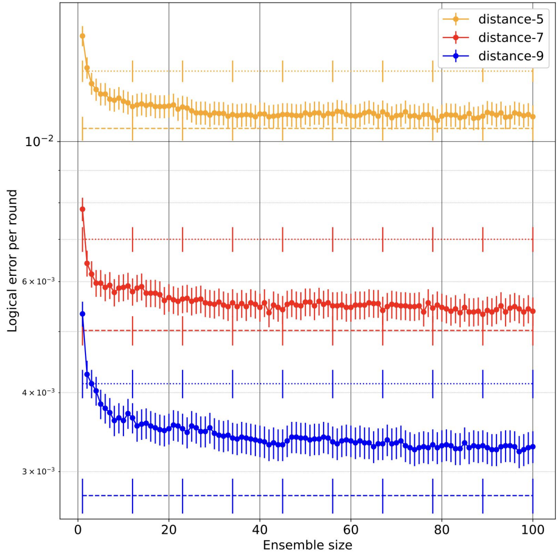


Results (2/2)

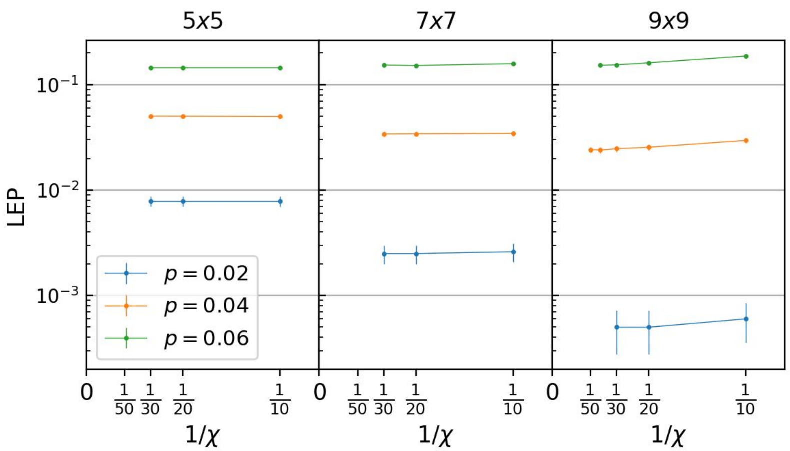
Benchmarking performance of faster / scalable decoders

N. Shutty, M. Newman, **BV**, 2024 - *arXiv:2401.12434* (2024)

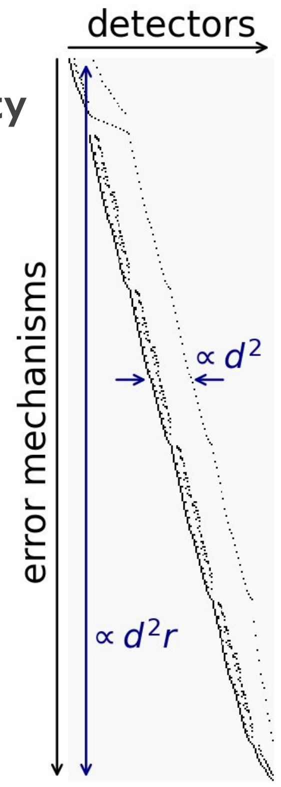
Benchmark of Harmony



Convergence



Complexity



$O(d^4 r \chi^3)$

Quantum AI

Outline

1. Motivation
    - The *status quo* of quantum computing experiments.
    - Two use cases for tensor networks
2. Benchmarking large NISQ experiments
    - A simulation problem
    - Making brute force TN contractions less brute: the RCS case study
    - Some results
    - The future of NISQ applications: noise vs. computational volume
3. Decoding early QEC demonstrations
    - The setup: 3 performance contributing factors
    - The hyper-graph error model.
    - The (maximum-likelihood) decoding problem
    - A TN maximum-likelihood decoder for all hyper-graph error models
    - Results
4. Conclusion

Conclusion

Two applications of tensor networks to experimental quantum computing:

- Highly-optimized tensor network contraction for benchmarking NISQ experiments:
  - Strong evidence for RCS being beyond classical
  - Insightful method to challenge useful beyond-classical claims
- Decoding for QEC:
  - Decode *arbitrary* error hyper-graph codes
  - Benchmark experimental hardware and error model quality
  - Benchmark performance of fast, scalable decoders

References

Latest RCS paper: Google, *arXiv:2304.11119* (2023)

Surface code error suppression: Google, *Nature* 614, no. 7949 (2023): 676-681

Harmony decoding: Noah Shutty, Michael Newman, and Benjamin Villalonga, *arXiv:2401.12434* (2024)

Acknowledgements

Salvatore Mandrà, Guifre Vidal, Sergio Boixo, Noah Shutty, Michael Newman, Hartmut Neven, K. Kechedzhi, S. Isakov, X. Mi, V. Smelyanskiy, ... and lots of people at Google Quantum AI