nature reviews physics

https://doi.org/10.1038/s42254-024-00796-z

Perspective

Benchmarking quantum computers

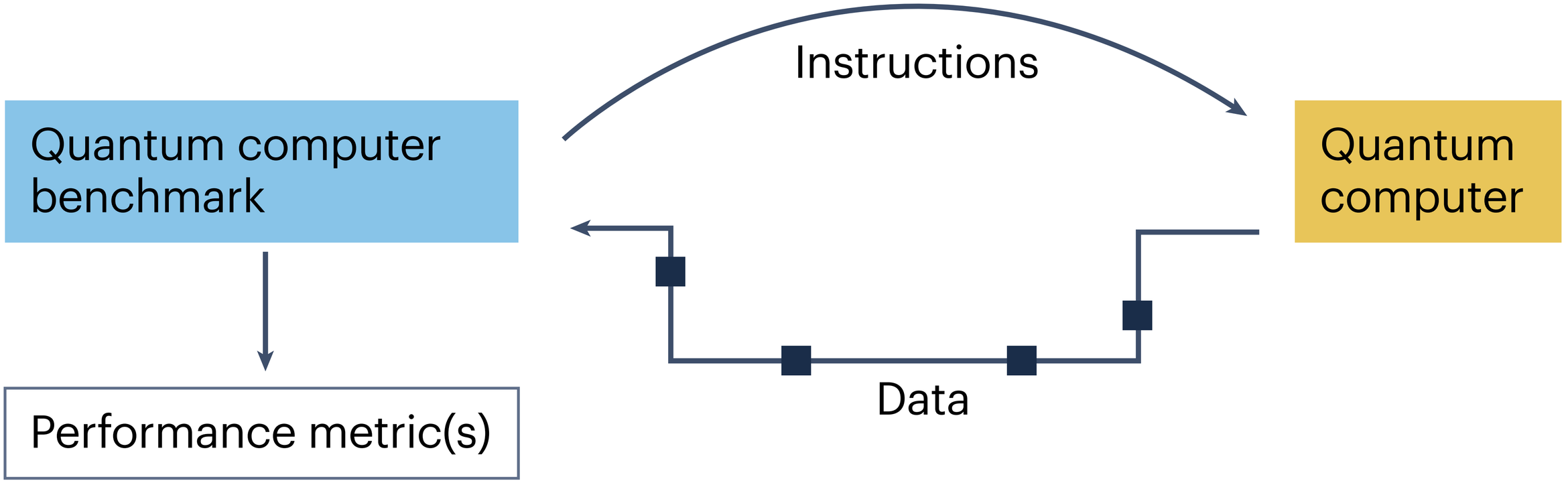
Timothy Proctor [1] ✉, Kevin Young[1], Andrew D. Baczewski [2] & Robin Blume-Kohout

U.S. DEPARTMENT OF ENERGY

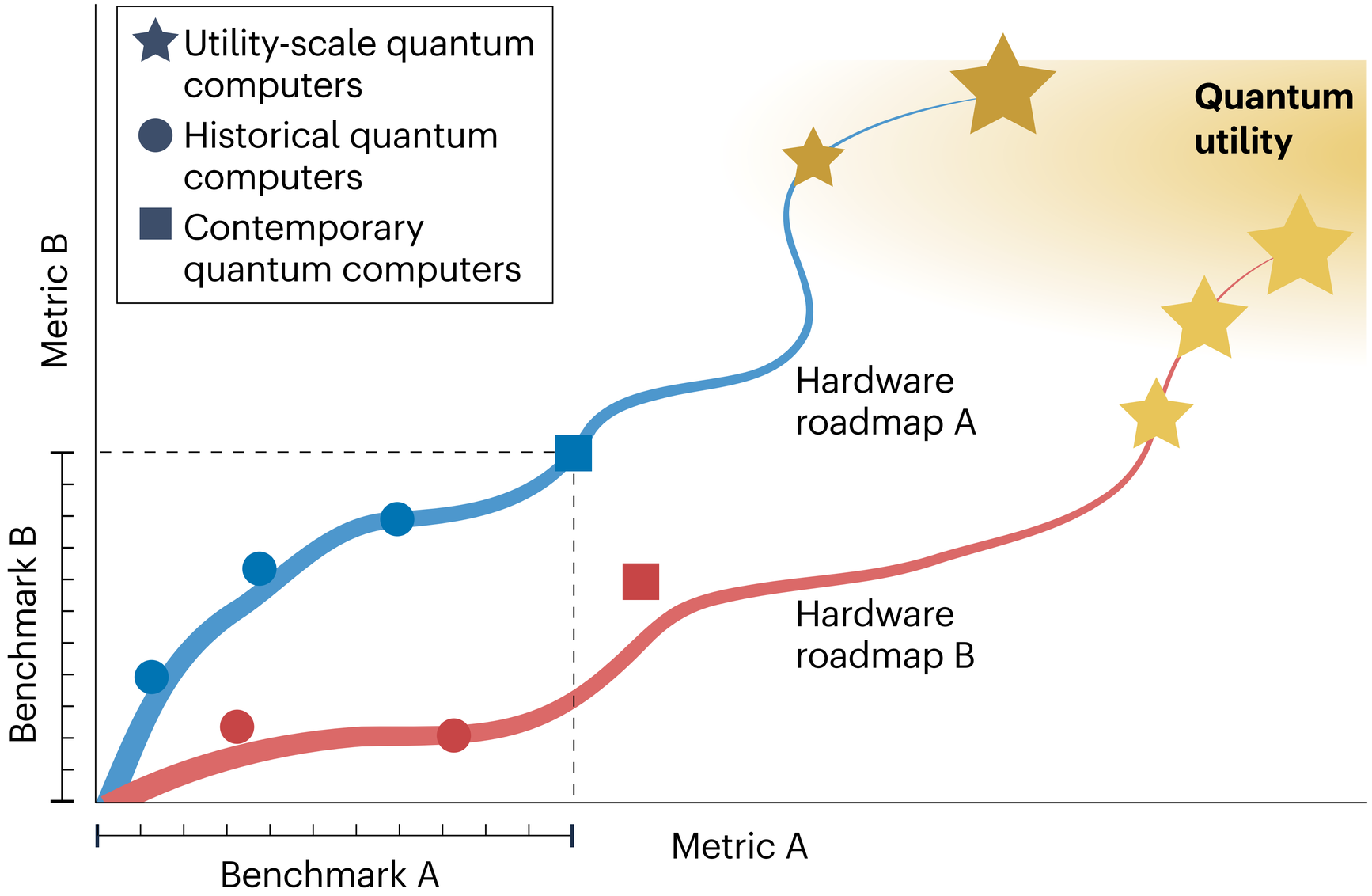
Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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-NA-0003525

Benchmarks are methods used to measure performance



Benchmarking should track progress toward utility



Desirable properties for tracking progress to utility

- **Well-motivated**: A benchmark should measure well-motivated metrics of performance.
- **Well-defined**: A benchmark should have an unambiguous procedure —any unspecified steps should be intentional configurable parameters (e.g. compilation).
- **Implementation-robust**: It should not be possible to exploit the configurable parameters of a benchmark to obtain misleading results.
- **System-robust**: The results of a benchmark should not be corrupted by *a priori* unknown (small) errors.
- **Efficient**: A benchmark should use a reasonable amount of all resources (e.g. quantum and classical computing time).
- **Technology independent**: A benchmark shouldn't be limited to particular technologies or architectures unless its metrics are only relevant in those contexts.

How to track progress toward utility

(1) Challenge Problems

A specific instance of a computational problem that is:
(1) useful,
(2) feasible to solve with a quantum computer but infeasible or more costly with any other computer.

(2) Resource Estimates

"How big, fast, and reliable would a quantum computer need to be to solve a particularly computational problem?"

A description of a minimal quantum computer that could solve that problem

(3) Roadmaps

A plausible, detailed engineering route to a device that can solve a challenge problem.

A sequence of increasingly capable prototype devices (may not necessarily increase computational power right away!)

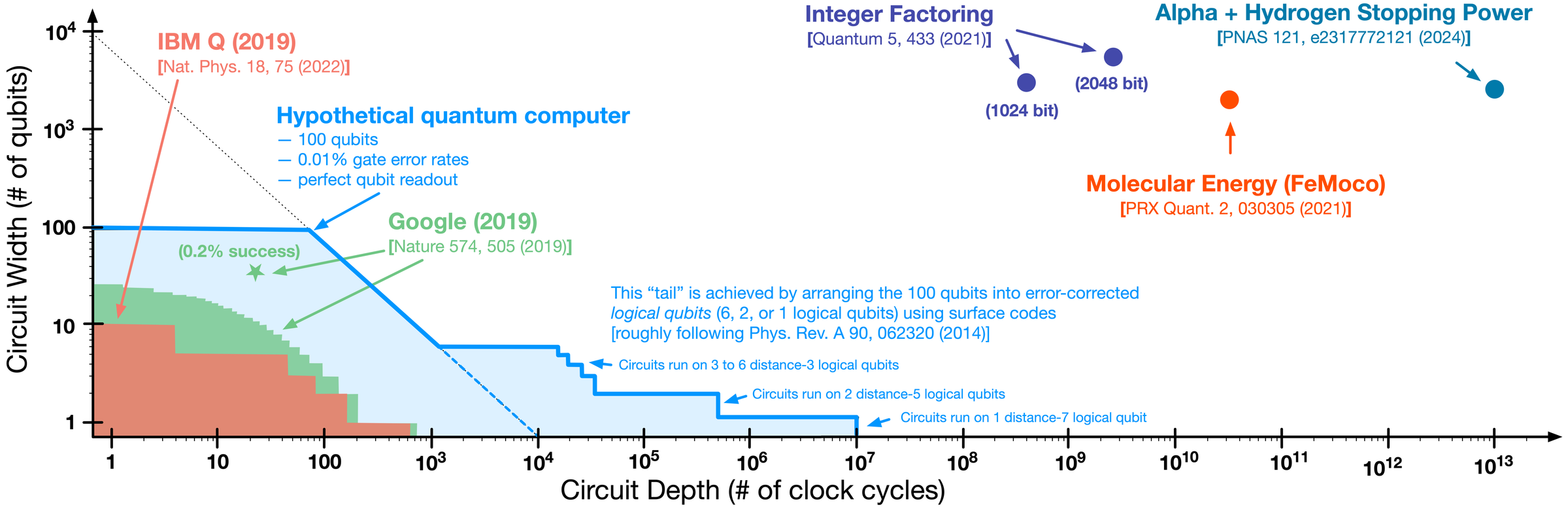
(4) Benchmarks

…that measure performance metrics specific to a particular roadmap.

…that quantify progress towards utility with high-level, intuitive, technology-agnostic performance metrics.

No single number can track progress toward utility

The most promising “metric” we’ve come up with is **capability** — what set of circuits (parameterized by **features**) can the QC run?



Part II

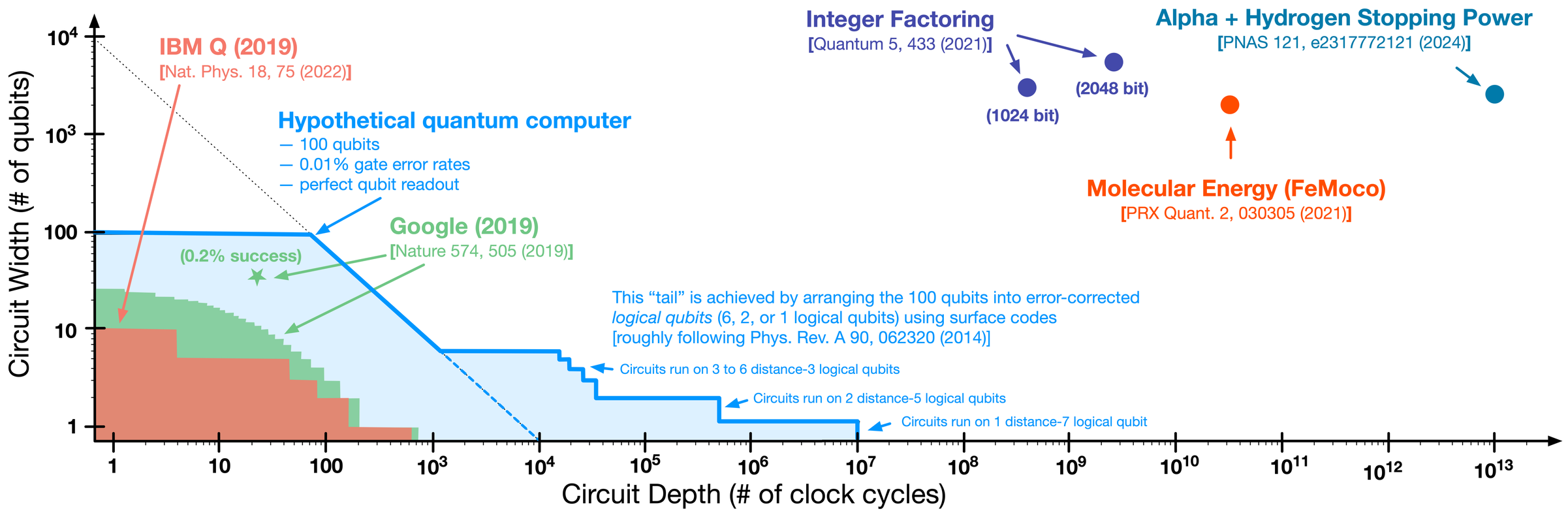
In our view, the largest open question in benchmarking is “How can our field create and deploy benchmarks that reliably quantify progress towards quantum utility?”

Here’s how *we* (at the QPL) are stumbling toward that goal.

- **Bridging the gap from NISQ to FTQC benchmarks**
- **How to benchmark tiny quantum computers with big circuits**
- **Quantifying the impact of error mitigation**

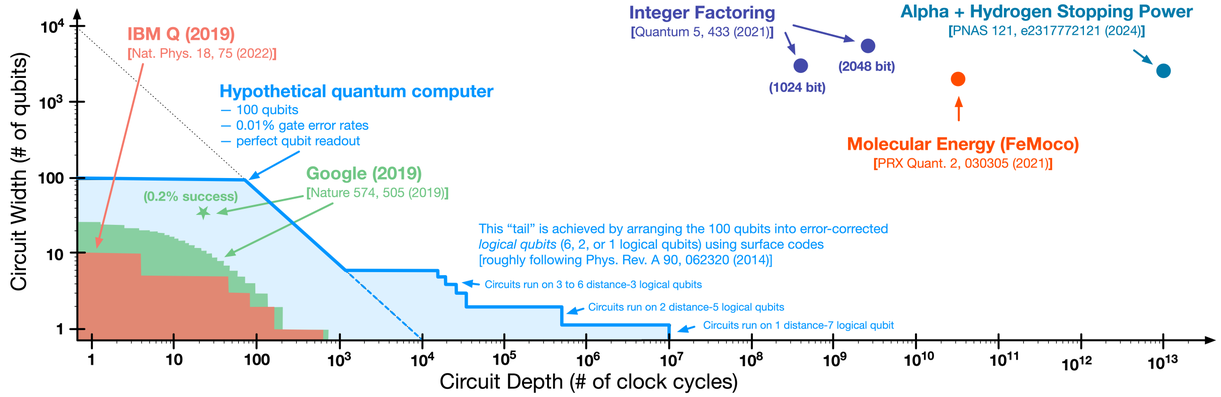
From NISQ to FTQC

This is a *very* oversimplified cartoon of how FTQEC will impact us.



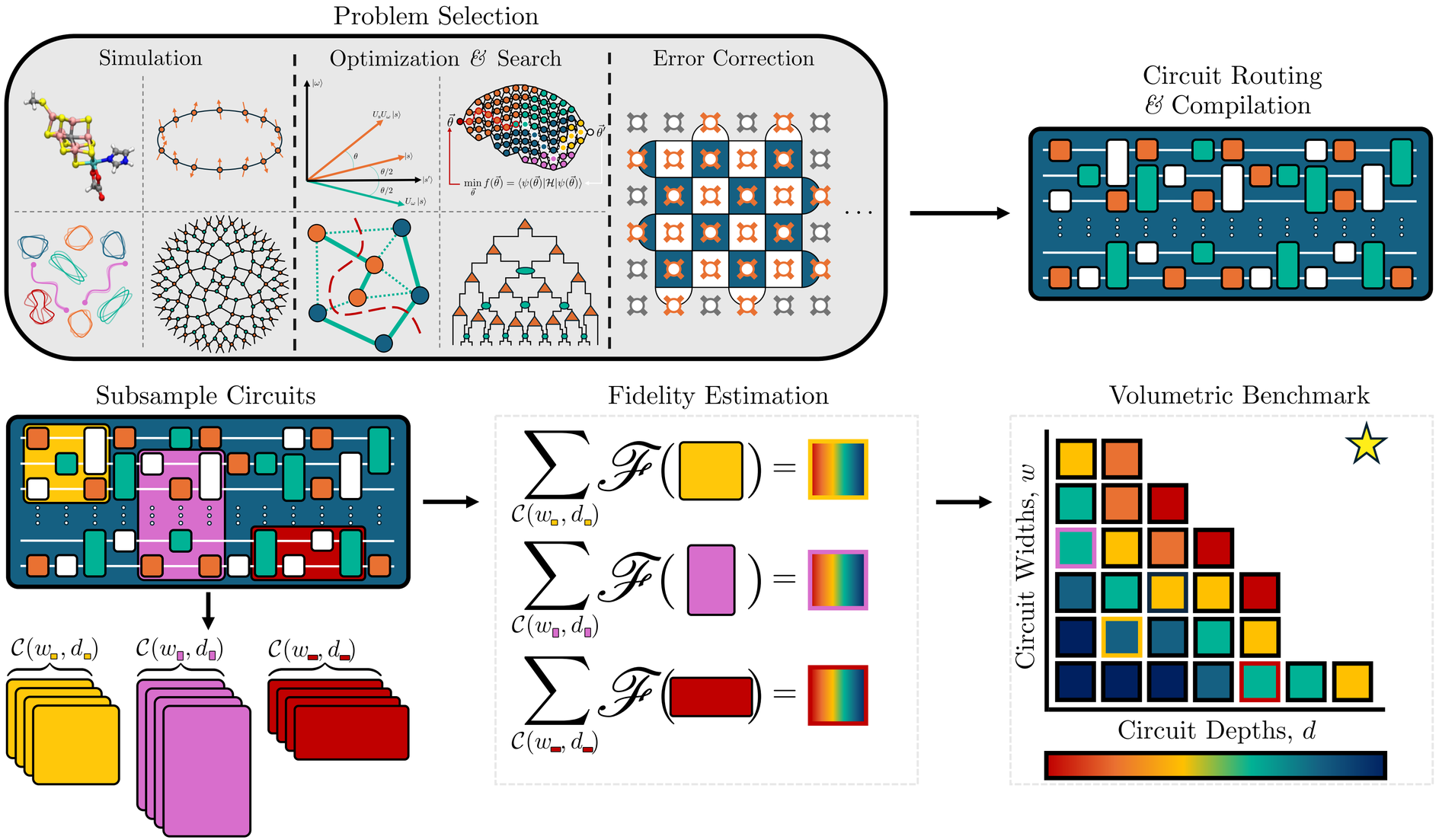
Challenges to benchmarking FTQC

- **Eastin-Knill Theorem ==> logical qubit control is very different!**
- **Factories ==> scheduling may be very weird at utility scale!**
- **Solovay-Kitaev ==> cost function of arbitrary unitaries is jagged.**
- **Utility-scale algorithms may be *serialized* to avoid parallel T gates**
- **Some gates may not even be *possible* until larger scales (e.g. lattice surgery).**
- **There may be *multiple* phase transitions from NISQ to utility.**



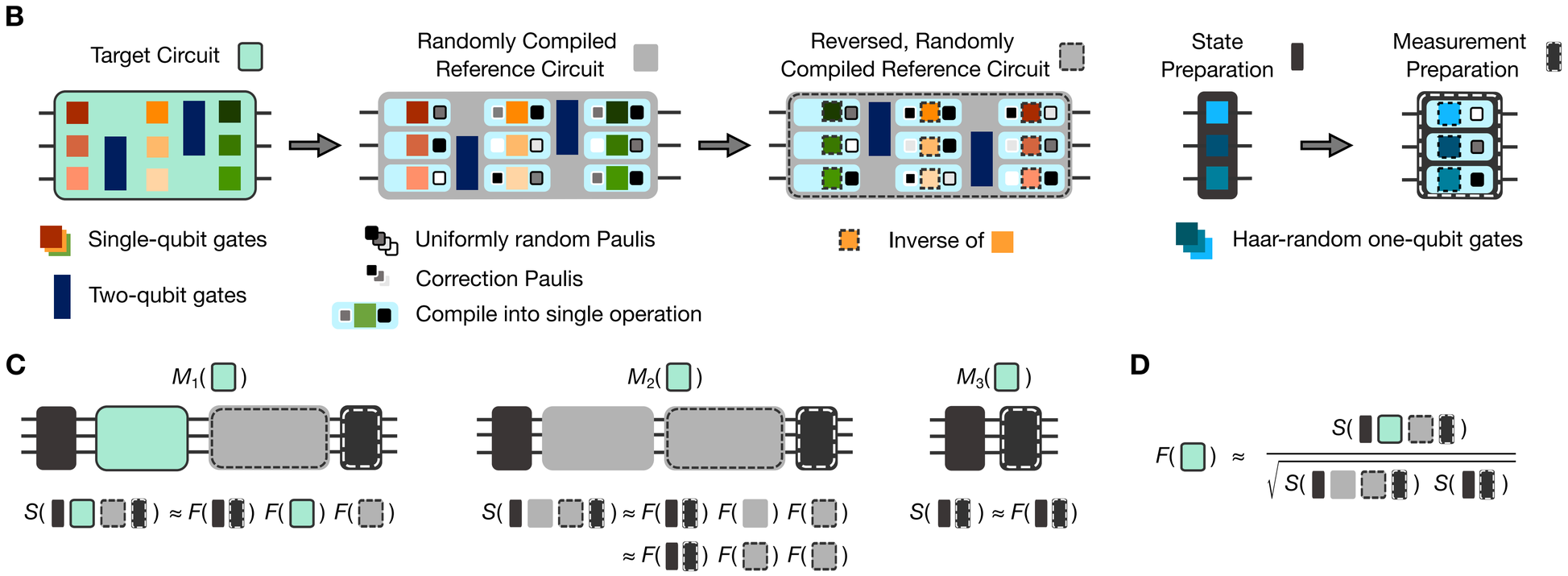
Benchmarking tiny quantum computers with big circuits

Subcircuit Volumetric Benchmarking



Technical Bits

- **Once you “snip out” a sub circuit… how do you actually test whether you ran it right???**
- ***Circuit mirroring* to the rescue!**

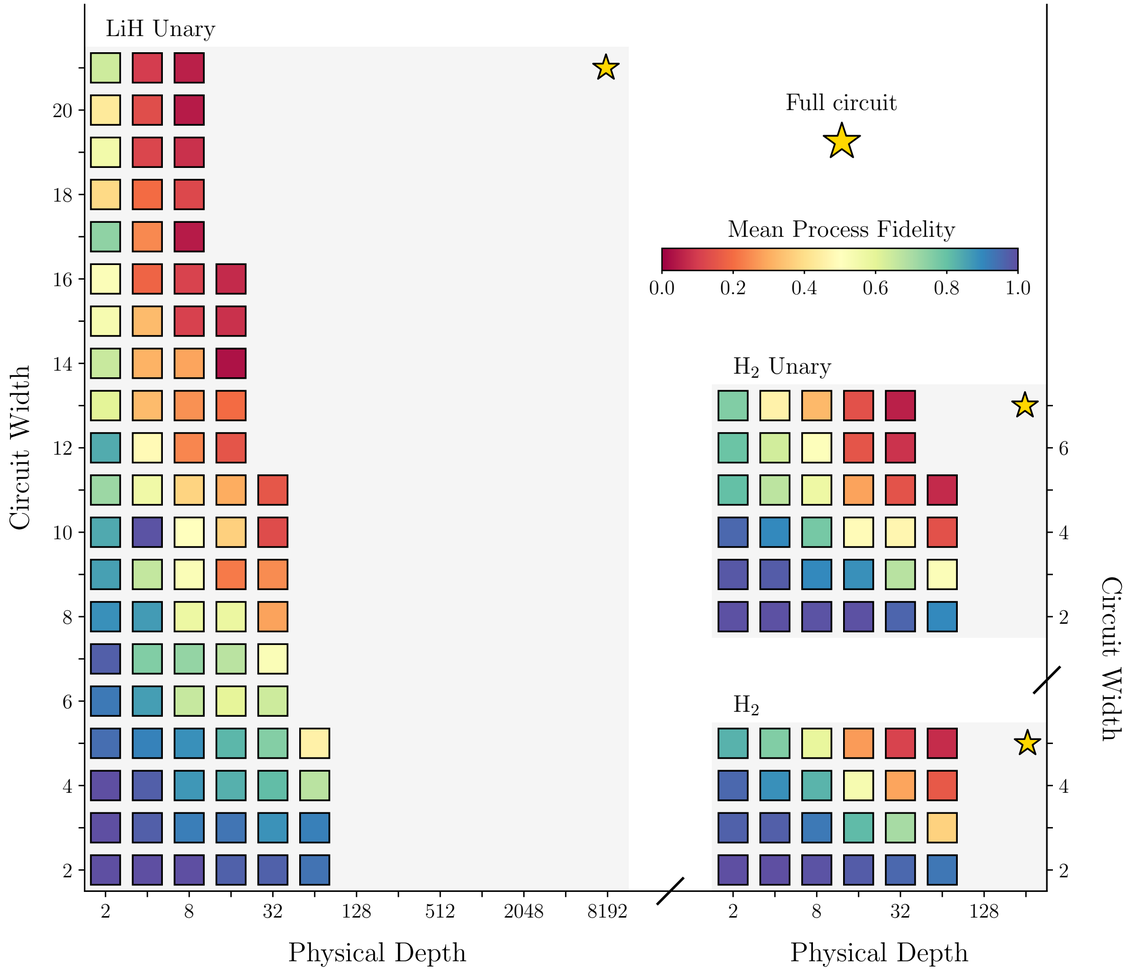


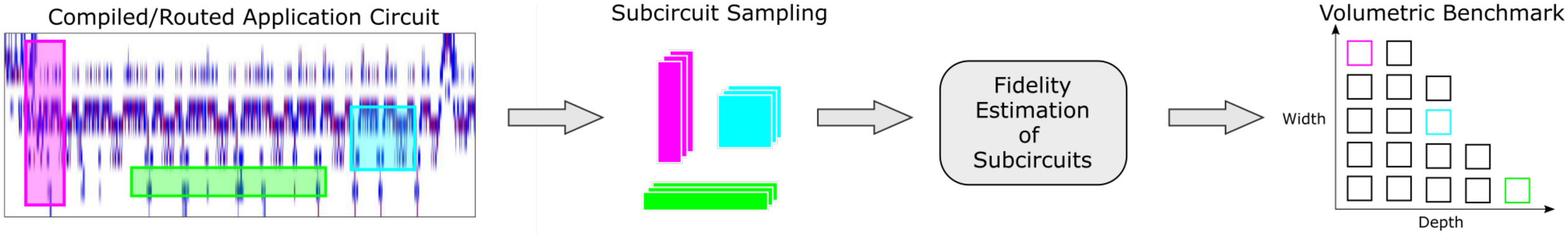
- T. Proctor *et al*, “Establishing trust in quantum computations”. arXiv preprint:2204.07568 (2022)

Results

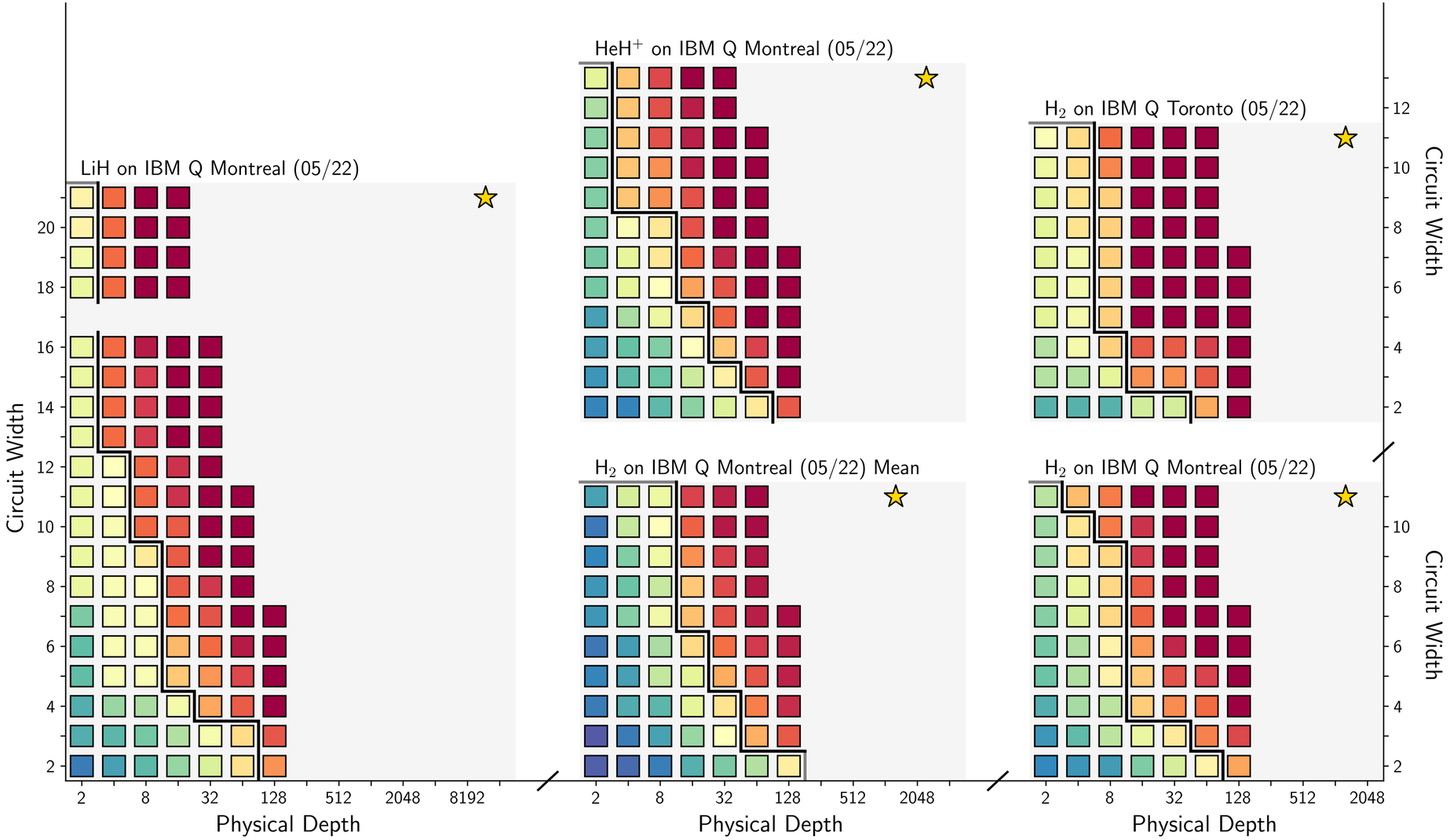
- We applied this process to a quantum chemistry simulation algorithm with a 21 x 8192 (!) circuit, and ran the resulting benchmarks on ibm_sherbrooke.

- Doing this on successive generations of device allows tracking the frontier's movement toward the full algorithm.





Delineating capability with Pareto frontiers



Oliver Maupin

Most *NISQ* algorithms rely heavily on error mitigation.
This allows algorithmic success even when circuits run with very low success probability!

Can we extend “capability regions” to capture error-mitigated performance?

Incorporating error mitigation

We want to account for the use of error mitigation

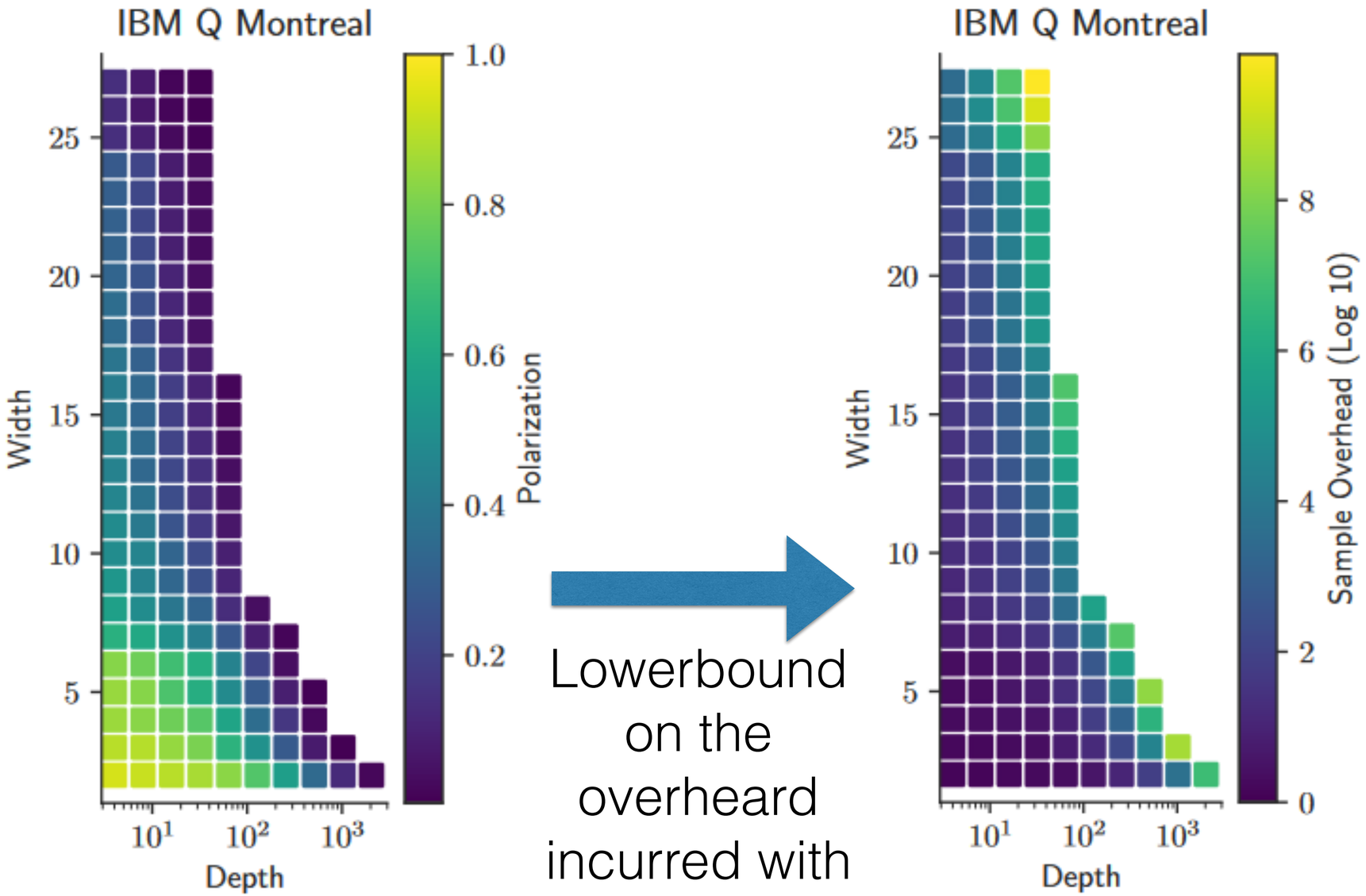
↓

We want to report a single simple metric of success for benchmarks

↓

Extending capability-region benchmarks to show the overhead of various error mitigation techniques

WIP experimental capability region



Lowerbound on the overheard incurred with PEC

Punchlines

- Benchmarking can (should?) track *progress toward utility*
- Challenge problems, roadmaps, and resource estimates
- We need >1 number. Can capability benchmarking suffice?
- If “utility scale” means *huge circuits* on *logical qubits*…
  … then benchmarking will get very different. Are we ready?
- Measuring *overhead* can unify error mitigation with capability.