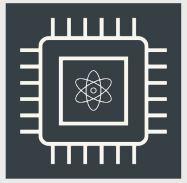
BQC | Reduce resources from holistic approach

Trustless computation on remote quantum computers

Let's take a holistic approach to BQC and break down its components:

Goal: Offload as much as possible to the server!

1

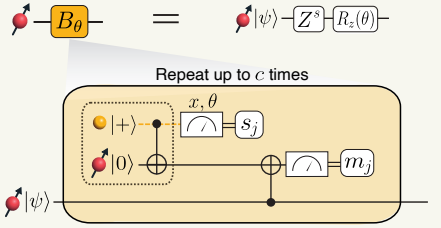


- The server is a powerful quantum computer
- It has low gate errors and fast computation speeds
- It can operate as a circuit-based quantum computer

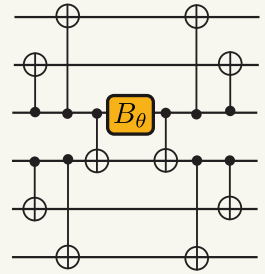
2

- The network connection will need photonic connections
- Subject to more loss and errors than the local computation
- Will be slower than local operations

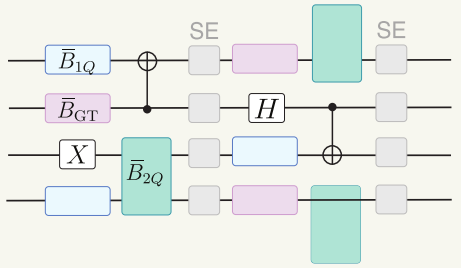
- **Element 1:** Blind gates in a hybrid architecture



- **Element 2:** Circuit designs for practical blindness



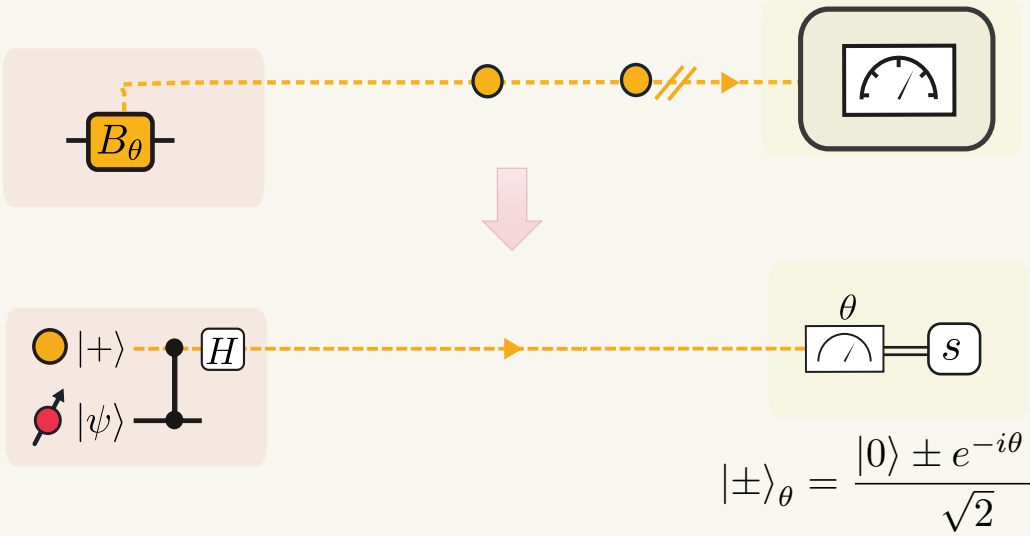
- **Element 3:** Efficient FT operation



Element 1: Blind gates in a hybrid architecture

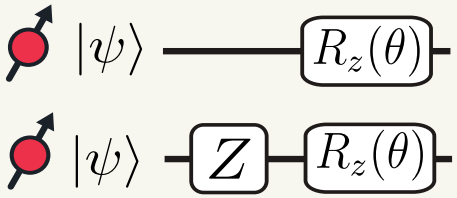
Element1 | Blind gates

Blind single qubit gate through teleportation



$$|\pm\rangle_\theta = \frac{|0\rangle \pm e^{-i\theta}|1\rangle}{\sqrt{2}}$$

After successful measurement



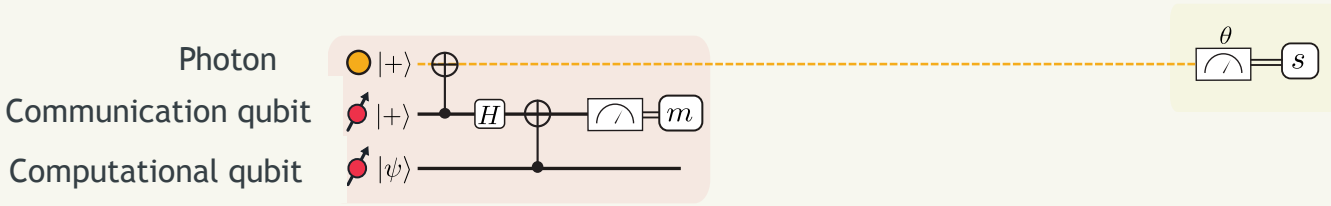
if $s = 0$

if $s = 1$

$$\mathrm{U_s} = \mathrm{Z}^s R_z(\theta)$$

Server observes a θ-independent density matrix:
$\mathbb{E}_s(U_s|\psi\rangle\langle\psi|U_s^\dagger)$

Making it loss-tolerant



- Repeat until successful measurement
- Perform local teleportation of gate onto computational qubit

$$\mathrm{U_{s,m}} = \mathrm{Z}^s R_z[\,(-1)^m\theta]$$

With $\mathrm{p} = \frac{1}{2}$ get $m = 0$ ⟶ **Done!**

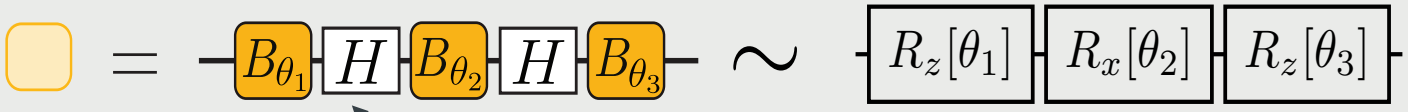
With $\mathrm{p} = \frac{1}{2}$ get $m = 1$ ⟶ **Repeat with $\boldsymbol{\theta \to 2\theta}$** $[-\boldsymbol{\theta} + \boldsymbol{2\theta} = \boldsymbol{\theta}]$

- On average 2 photons needed but can in require more repetitions.
- **If $\boldsymbol{\theta} = \{\frac{2\pi j}{2^c}\}$ for $\boldsymbol{j = 1, \ldots, 2^c}$, then we repeat this circuit at most $\boldsymbol{c - 1}$ times.**

Element1 | Blind gates

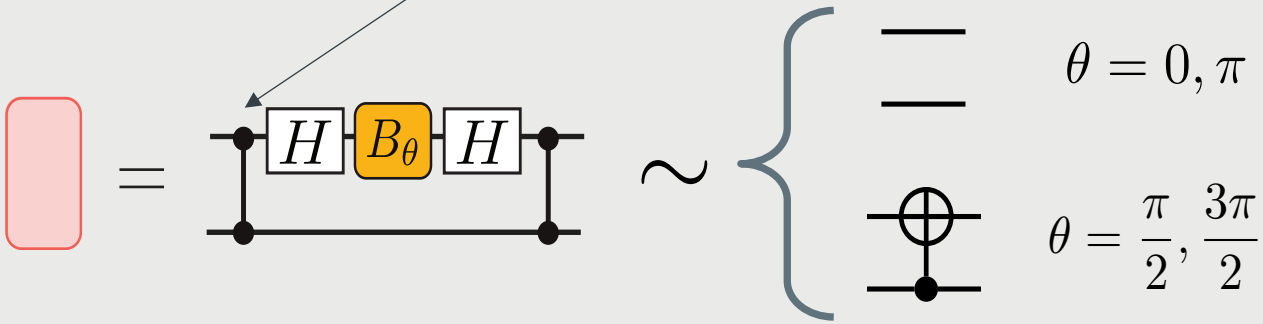
Universal blind gate set

Arbitrary **blind** 1Q rotation



Clifford gates which preserves structure of the Pauli frame (only known to to the client)

Blind entangling 2Q gate



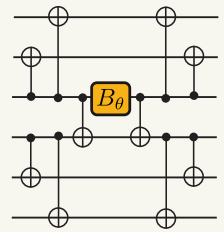
We now have

- Loss tolerant implementation of blind gates in a hybrid framework
- A universal blind gate set on physical qubits

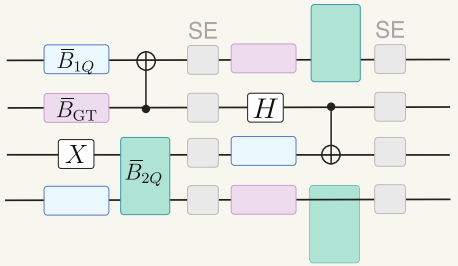
We still need

- To lower the cost of BQC!
- Extend this to a fault tolerant BQC

Element 2: Circuit designs for practical blindness



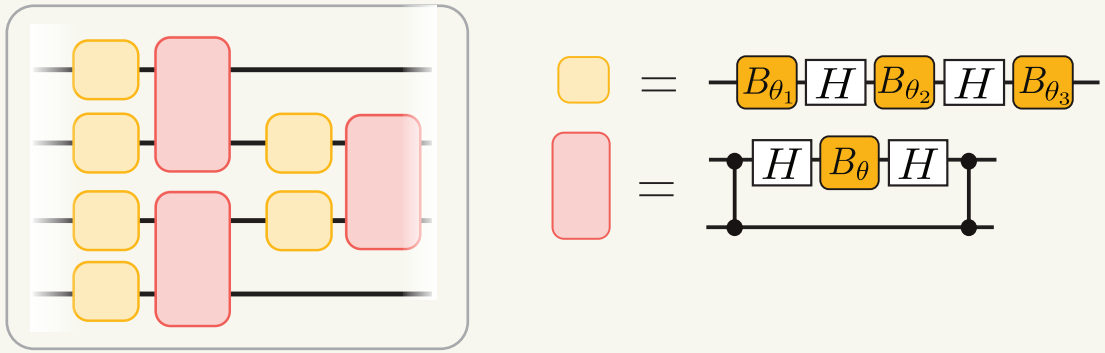
Element 3: Efficient FT operation



Element 2: Circuit designs for practical blindness

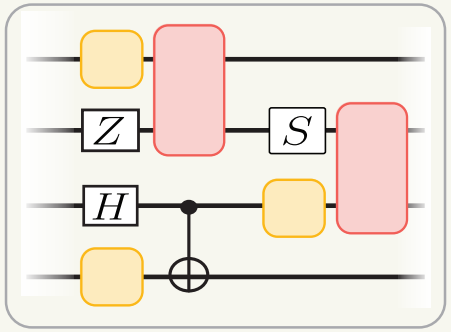
Element 2| Practical blindness

Fully-Blind Bricklayer

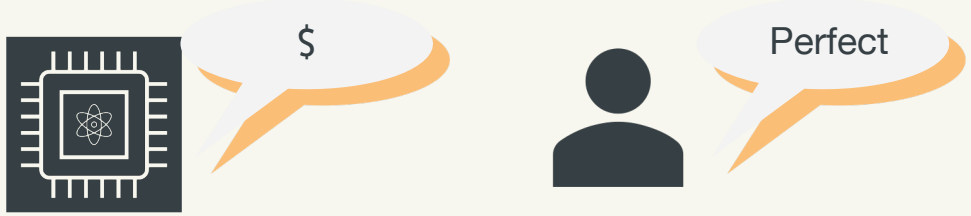


- Requires costly networking connection for every gate
- Requires a lot of non-Clifford gates to hide circuit completely
- More demanding error correction due to higher networking error

Make some of parts of the circuit known



- Reveals more information about the computation to the server
- Allows the execution of low-cost and high-fidelity gates at the server
- Reduces the error correction overhead due to fewer noisy blind gates.



Element 2| Practical blindness

Hamiltonian simulation

Key application of QC

$$H = \sum_i h_i P_i$$

Need to implement:

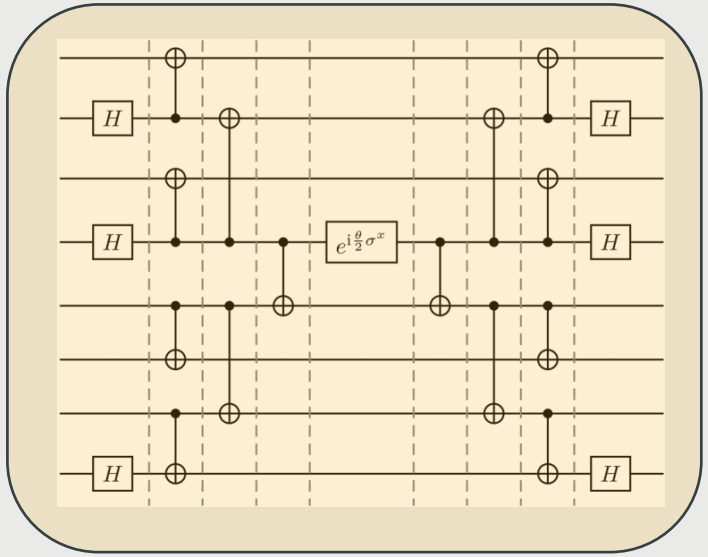
$$e^{-iHt} \approx \left[\prod_i e^{-i(h_i t/K) P_i} \right]^K$$

Key Primitive: $e^{i\frac{\theta_i}{2}P_i}$

Fully blind implementation

Hides information that you are performing Hamiltonian simulation all together

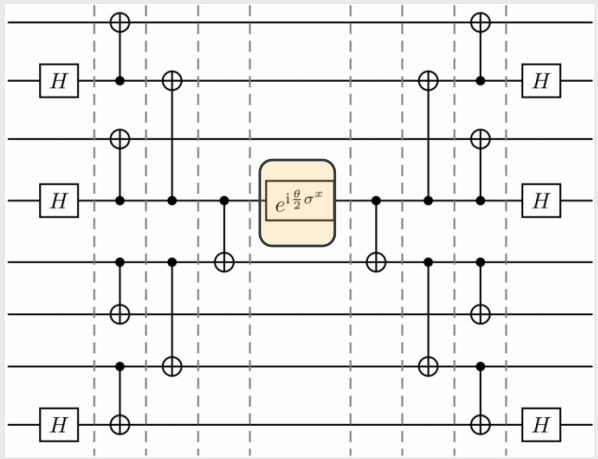
$e^{i\frac{\theta}{2}P} =$



Hiding both P and θ:
$O(poly(n))$ blind gates

Partially blind implementation

Client is okay with leaking information that they are performing Hamiltonian simulation but wants to hide which Hamiltonian they are simulating.



Hiding only θ:
$O(1)$ blind gates

Element 2| Practical blindness

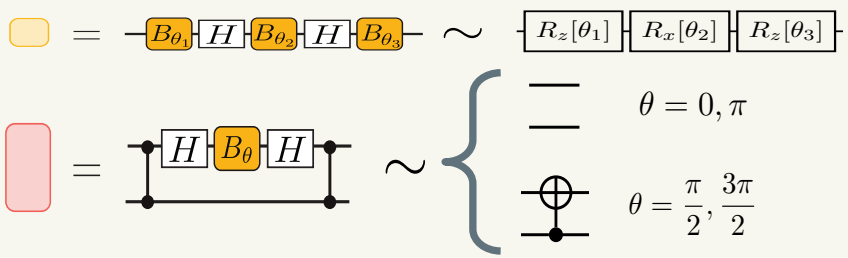
Limited access to blind gates

The server charges a high cost per blind gate in the circuit and the client wants to maximize their value per blind gate

Expressibility: A measure for the ability of a circuit to generate a uniformly distributed ensemble in the space of all possible unitaries

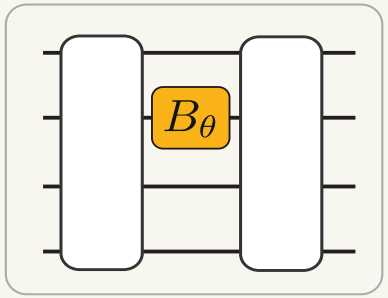
Low expressibility: Blind circuit only has access to a space of very similar unitaries

High expressibility: Blind circuit has access to a space of wide variety of unitaries

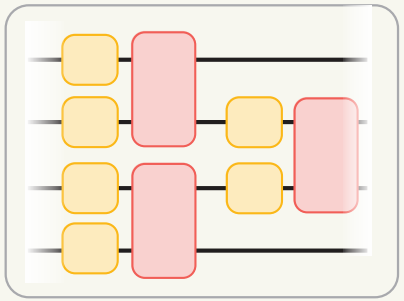


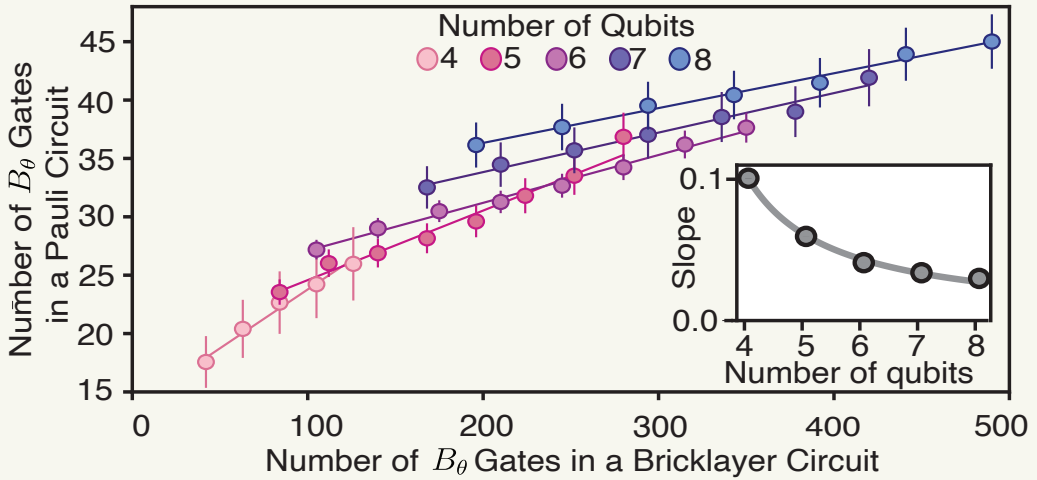
Expressibility of set with all possible (secret) input parameters

Blind Pauli rotation



Brick layer

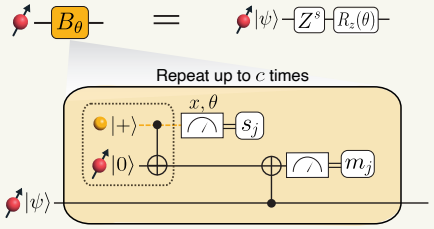




Intermezzo| Current status

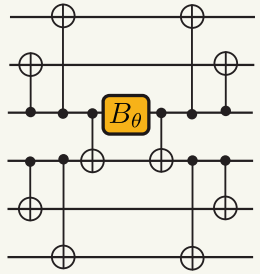
Goal: Offload as much as possible to the server!

- **Element 1:** Blind gates in a hybrid architecture



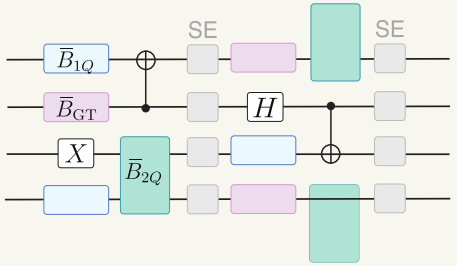
Done

- **Element 2:** Circuit designs for practical blindness



Done

- **Element 3:** Efficient FT operation



Next

What have we achieved?

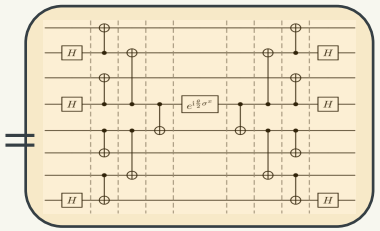
Loss tolerant framework for BQC in a hybrid framework

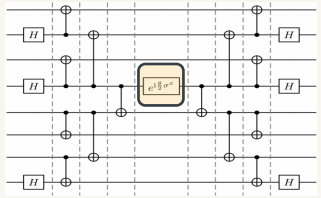
- Implement a universal blind gate set in a circuit based QC through gate teleportation allowing for local non-blind gates

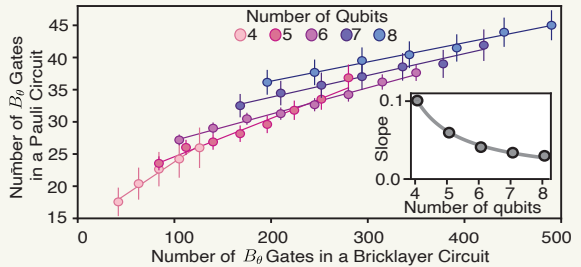
Flexibility in circuit design and selective blindness for more efficient BQC

- Reduce number of blind gates but keep valuable information secret
- Expressibility as a measure for “Get most value for your buck” circuit compilation

$e^{i\frac{\theta}{2}P} =$







Element 3: Efficient FT operation

Element 3| Efficient QC operation

Observation 1:
Error correction is not an interesting thing to hide

→ Do not mind offloading error correction overhead to server

Observation 2:
Error correction overhead cannot reveal information about the logical operations

→ You **can** offload error correction overhead to server

Observation 3:
Matter qubits can form effective logical qubits, while photons apply delegated physical gates

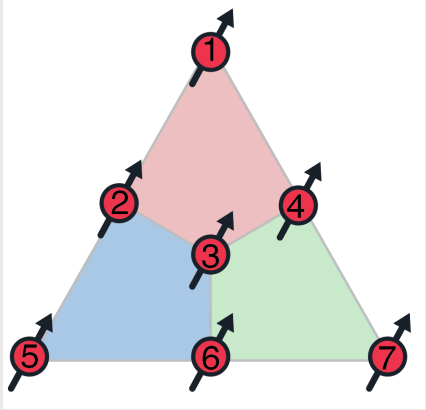
→ Can sustain and correct errors induced by the photonic link

Element 3| Efficient QC operation

Step 1: QEC code

Make choice of QEC code:

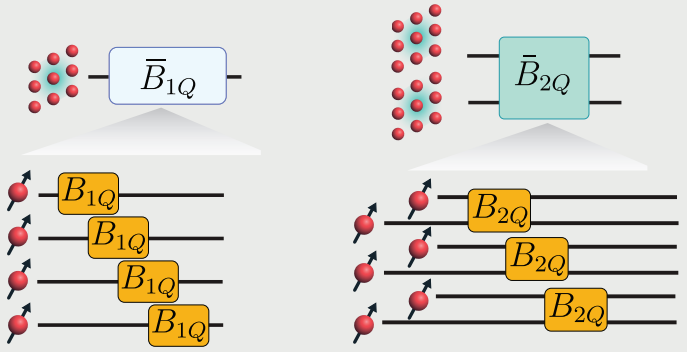
- Resources available
- Computational goal
- Compatibility of code with hardware



Step 2: Logical gate set

Construct the implementation of logical gates:

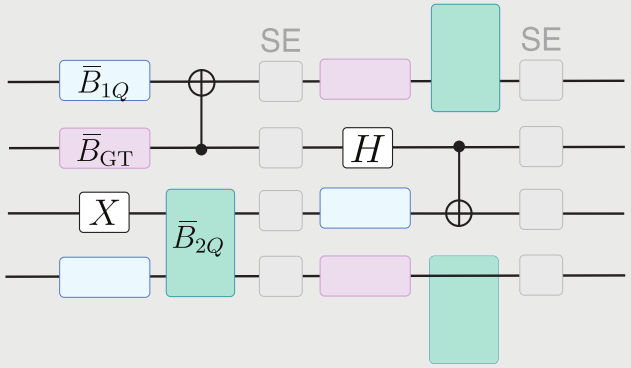
- Set of transversal gates
- Teleported gates e.g. for magic states
- E.g. folded surface code has transversal Clifford gates



Step 3: Compilation

Compile the algorithm into the logical gate set:

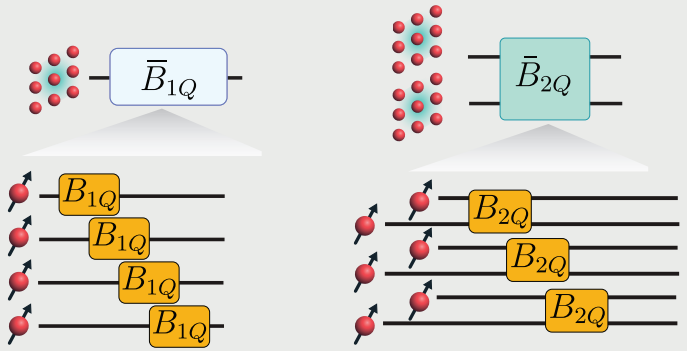
- Choose blind elements
- Interleave operations with syndrome extraction performed locally



Element 3| Efficient QC operation

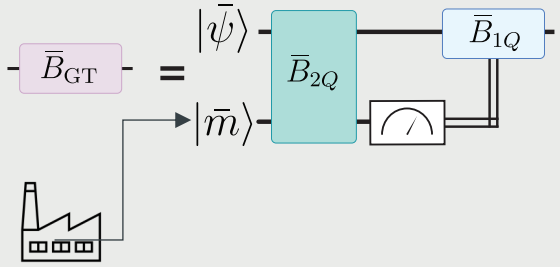
Efficient transversal gates

- [[*n*,*k*,*d*]] – code requires *n* blind gates for implementation of 1 logical transversal gate
- Errors from the blind gates result in local errors on matter qubits



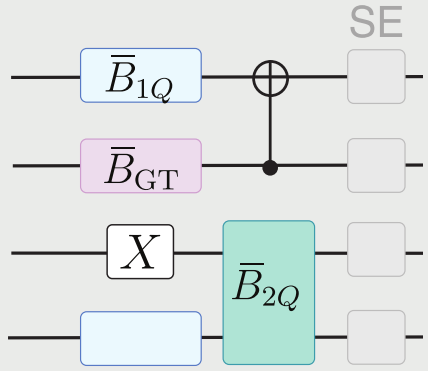
Magic state teleportation

- Requires same resources as transversal 1Q and 2Q gates
- Magic state is generated and distilled entirely at the server
- Steane code example: Teleport T gate and singel qubit gate correction is transversal S gate



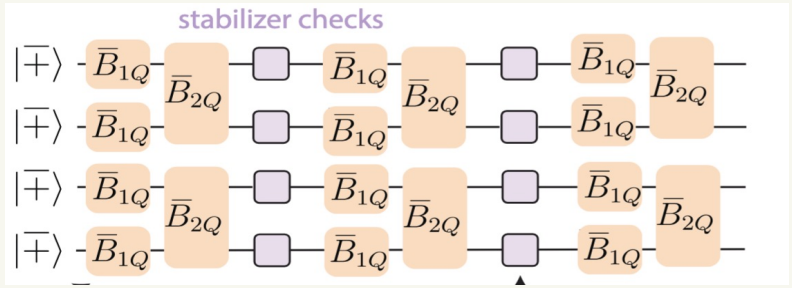
Local stabilizer checks

- QEC stabilizers are measured locally at the server
- Only classical information sent to the client for decoding.
- Lower local errors -> higher error thresholds for the non-local (blind) operations



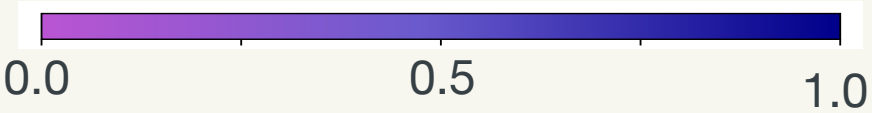
Element 3| Efficient QC operation

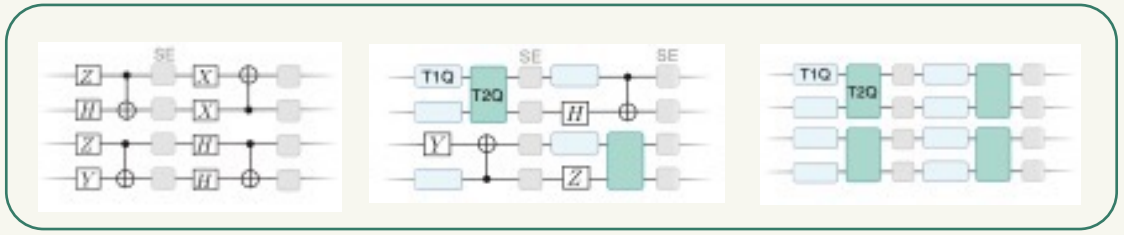
Simulating random, deep quantum circuits



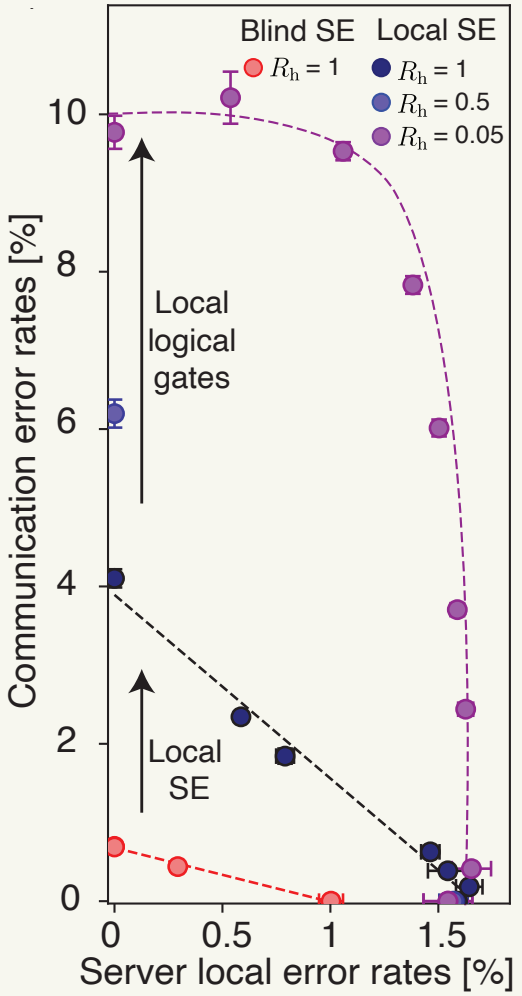
- Simulation of random quantum (Clifford) circuits with the surface code.
- Decoding conducted with a most-likely-error correlated decoder
- Includes simulations with different blind gate fraction

Blind gate fraction, R_{h}



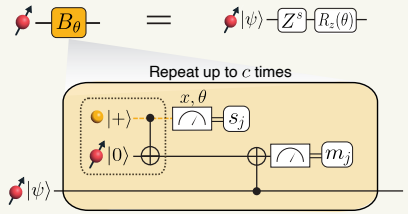


Higher error thresholds!

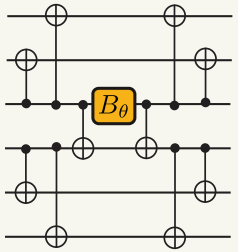


Summary

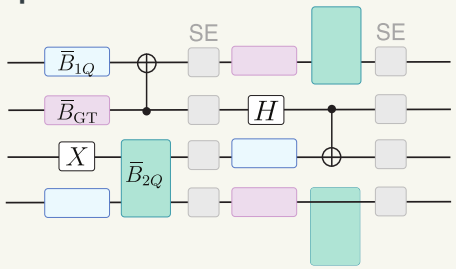
Element 1: Blind gates in a hybrid architecture



Element 2: Circuit designs for practical blindness



Element 3: Efficient FT operation



Loss tolerant framework for BQC in a hybrid framework

- Implement a universal blind gate set in a circuit based QC through gate teleportation allowing for local non-blind gates

Flexibility in circuit design and selective blindness for more efficient BQC

- Reduce number of blind gates but keep valuable information secret
- Expressibility as a measure for “Get most value for your buck” circuit compilation

Off load FT operation entirely to server

- Transversal logical gates for efficient QEC
- Magic state generation and distillation entirely at the server
- Logical stabilizer measurements -> Higher networking error threshold

References and people

References for this work:

Theoretical framework: Gefen Baranes *et al.* *"Designing Fault-Tolerant Blind Quantum Computing"*, arXiv: 2505.21621

Experimental implementation: Y.-C. Wei *et al.* *"Universal distributed blind quantum computing with solid-state qubits"*, Science **388**, 509-513 (2025)