// Xanadu Presents

Weight Reduced Stabilizer Codes with Lower Overhead

CAP Congress 2024, Western University, London Ontario

Michael Vasmer

Motivation

- Noise severely limits the performance of today's quantum computers
- Quantum error correction enables scalable quantum computation for applications
  - Factoring [Sho96], chemical simulation [BBM+20], optimisation [OML19], ...
  - But space & time overheads are currently prohibitive
- Big focus for academia, industry, and government
- Surprising connections to other fields in physics
  - Topological codes ↔ topological phases of matter [Kit03, Kit06]
  - Tensor network codes ↔ AdS/CFT correspondence [PYH+15]

Contents

XANADU

01

A bit on Xanadu

A bit on Xanadu

- A photonic quantum computing company with eyes set on fault-tolerance
- Full-stack:
    - Hardware, including Architecture
    - Software and Algorithms
- Located in downtown Toronto, Canada:

Premise of Xanadu's architecture

need low-depth physical circuits

photonics → MBQC

building blocks

nonlocal connectivity

path to scaling up

benefit of existing R&D

Probabilistic sources (with cryostats)

Deterministic entanglement generation (offline) and gates (via adaptive local measurements), at room temperature

Silicon chips, waveguides, fibers
Phase shifters, squeezers
Beamsplitters
Homodyne detectors
Photon **N**umber **R**esolving detectors

want fast, room-temperature, deterministic, accessible gates

GKP qubits

Premise of Xanadu's architecture

Gaussian ~ "easy"
Non-Gaussian ~ hard

photonics

need low-depth physical circuits

MBQC

building blocks

nonlocal connectivity

path to scaling up

benefit of existing R&D

Probabilistic sources (with cryostats)

Deterministic entanglement generation (offline) and gates (via adaptive local measurements), at room temperature

Silicon chips, waveguides, fibers
Phase shifters, squeezers
Beamsplitters
Homodyne detectors
Photon **N**umber **R**esolving detectors

want fast, room-temperature, deterministic, accessible gates

GKP qubits

Premise of Xanadu's architecture

Gaussian ~ "easy"
Non-Gaussian ~ hard

photonics

need low-depth physical circuits

MBQC

building blocks

nonlocal connectivity

path to scaling up

benefit of existing R&D

Probabilistic sources (with cryostats)

Deterministic entanglement generation (offline) and gates (via adaptive local measurements), at room temperature

Silicon chips, waveguides, fibers
Phase shifters, squeezers
Beamsplitters
Homodyne detectors
Photon **N**umber **R**esolving detectors

want fast, room-temperature, deterministic, accessible gates

GKP qubits

Gaussian ~ “easy”
Non-Gaussian ~ hard

A bit on Xanadu

Premise of Xanadu’s architecture

photonics

need low-depth physical circuits

MBQC

building
blocks

nonlocal
connectivity

path to scaling up

Probabilistic sources (partly cold)

Deterministic entanglement generation (offline) and gates (via adaptive local measurements), at room temperature

benefit of
existing R&D

Silicon chips, waveguides, fibers
Phase shifters, squeezers
Beamsplitters
Homodyne detectors
Photon **N**umber **R**esolving detectors

want fast, room-temperature,
deterministic, accessible gates

GKP qubits

Premise of Xanadu's architecture

Gaussian ~ "easy"

Non-Gaussian ~ hard

photonics → MBQC

Probabilistic sources

Deterministic entanglement generation and gates

GKP qubits

02

Quantum error correction

For photonics

Quantum error correction 101

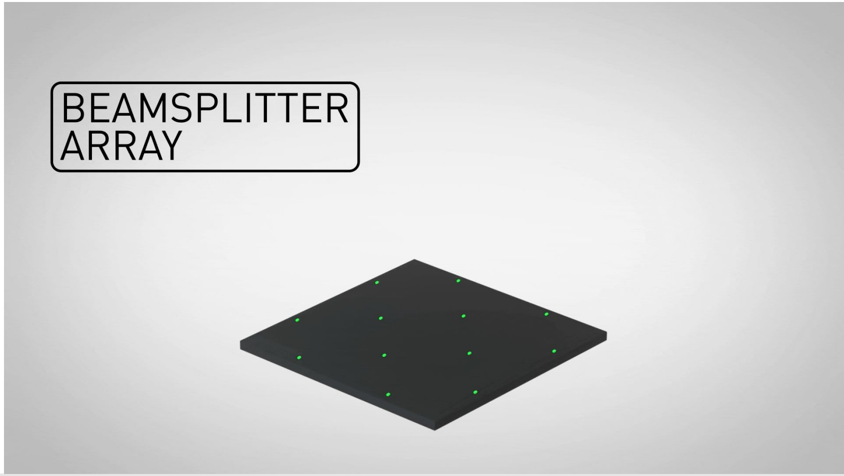
- n-qubit Hilbert space $(\mathbb{C}^2)^{\otimes n}$
- Quantum error correcting code: subspace of $(\mathbb{C}^2)^{\otimes n}$
- Code parameters $[\![n,k,d]\!]$
  - Number of physical qubits n
  - Number of encoded qubits k
  - Code distance d

Quantum error correction 101

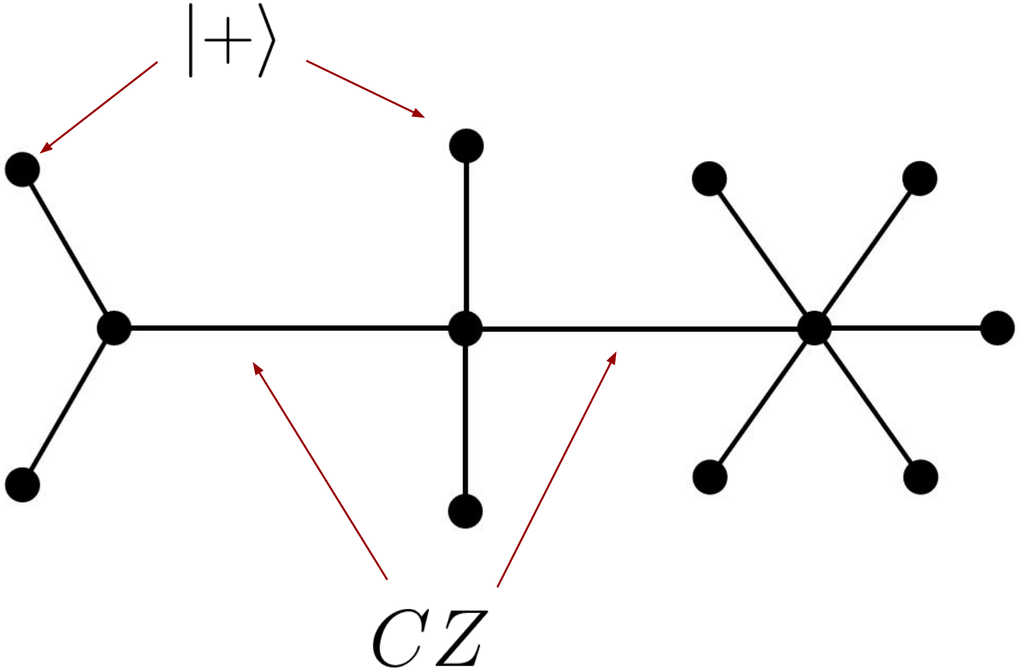
- Pauli matrices X, Y, and Z
- n-qubit Pauli group generated by tensor products of Paulis and I
- Stabilizer codes [Got97]
  - Abelian subgroup S of Pauli group
  - Code (sub)space $\{ |\Psi\rangle \text{ such that } g|\Psi\rangle = |\Psi\rangle \text{ for all } g \in S \}$
- Measure stabilizers to diagnose errors

Quantum error correction for photonics

- Measurement-based quantum computing [RBB03] natural for photonics
  - Start with entangled resource state (cluster state)
  - Computation proceeds via single-qubit measurements
- Foliation [BD-CP+16]
  - Input: quantum error-correcting code
  - Output: fault-tolerant cluster state



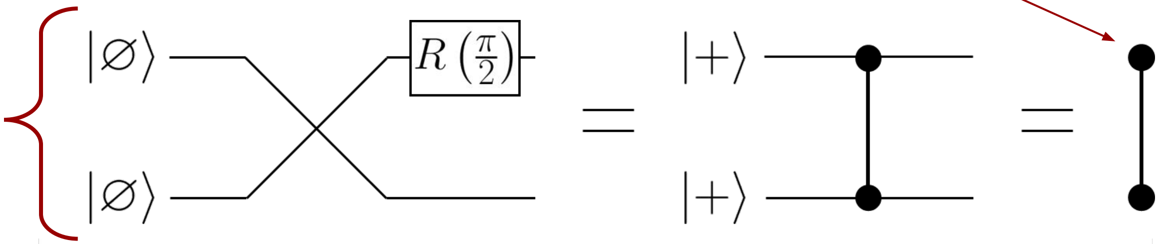
The target: a cluster state—a resource for FT MBQC [RHG06]

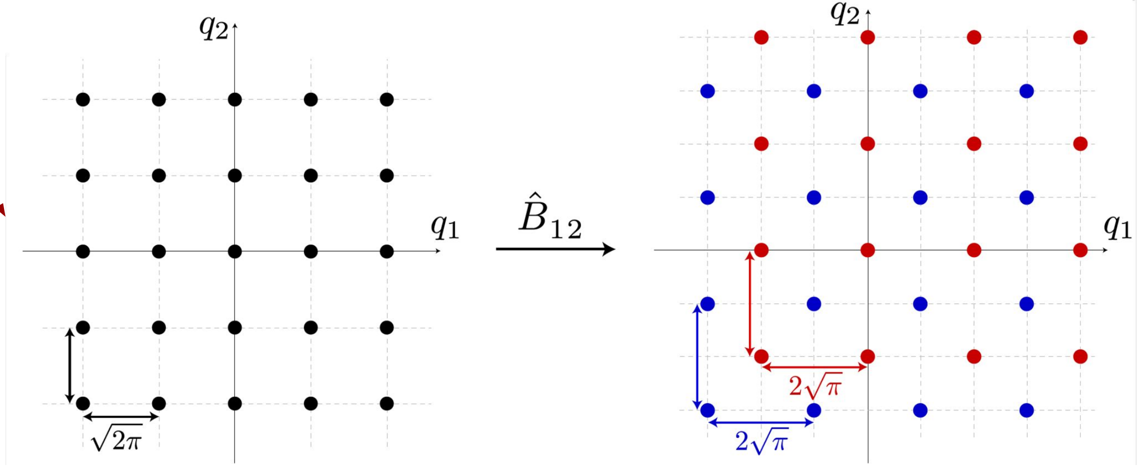


Entanglement generation with arbitrary connectivity [WBA+20,TMA+21]

Step 1: Create entangled GKP pairs ("dumbbells")

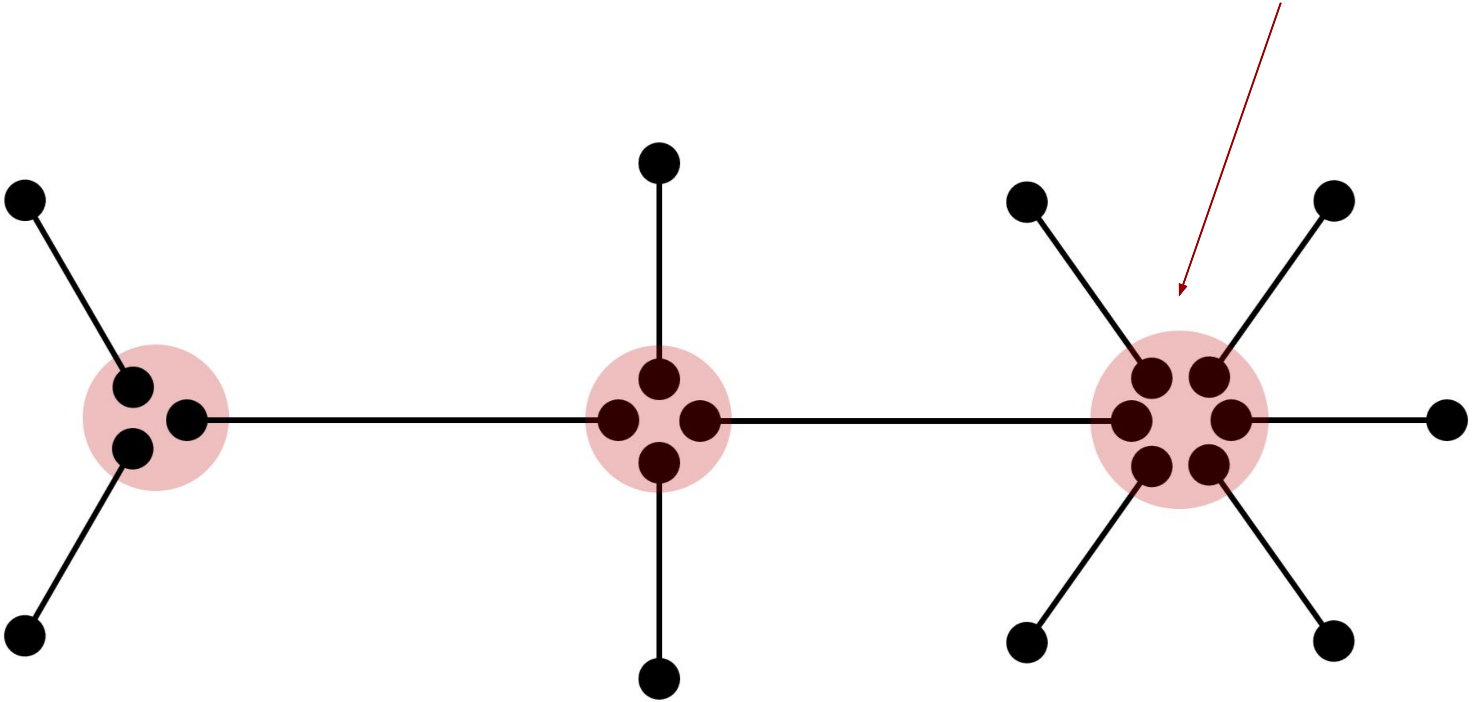
GKP sensor states
(q-naught states)





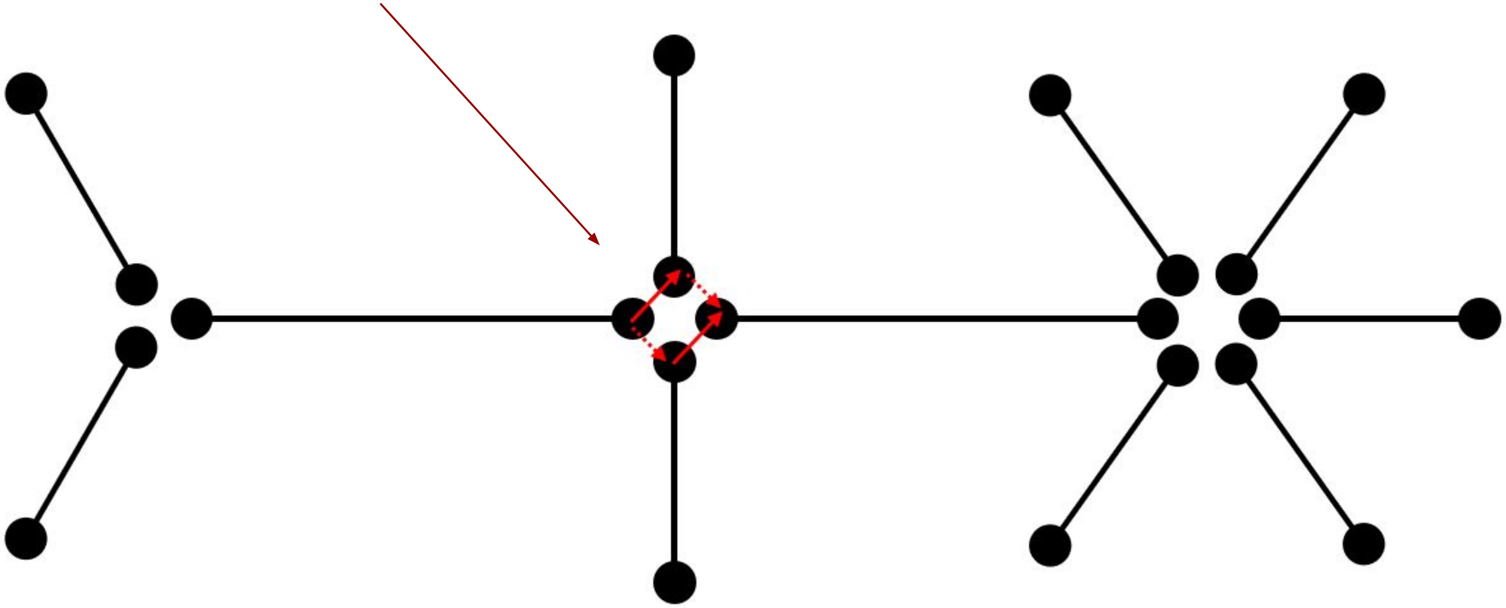
Entanglement generation with arbitrary connectivity

Step 2: Route modes from dumbbells to meet at macronodes



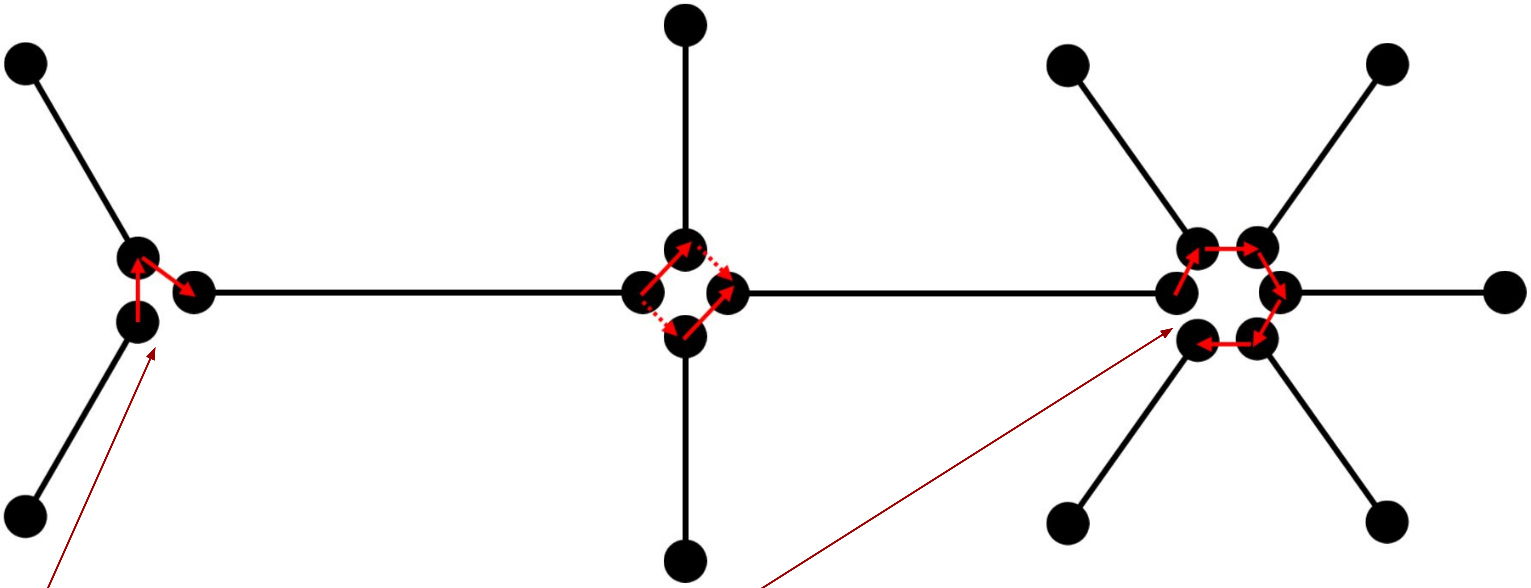
Step 3: Entangle modes within macronodes

balanced foursplitter: static array of 50:50 beamsplitters



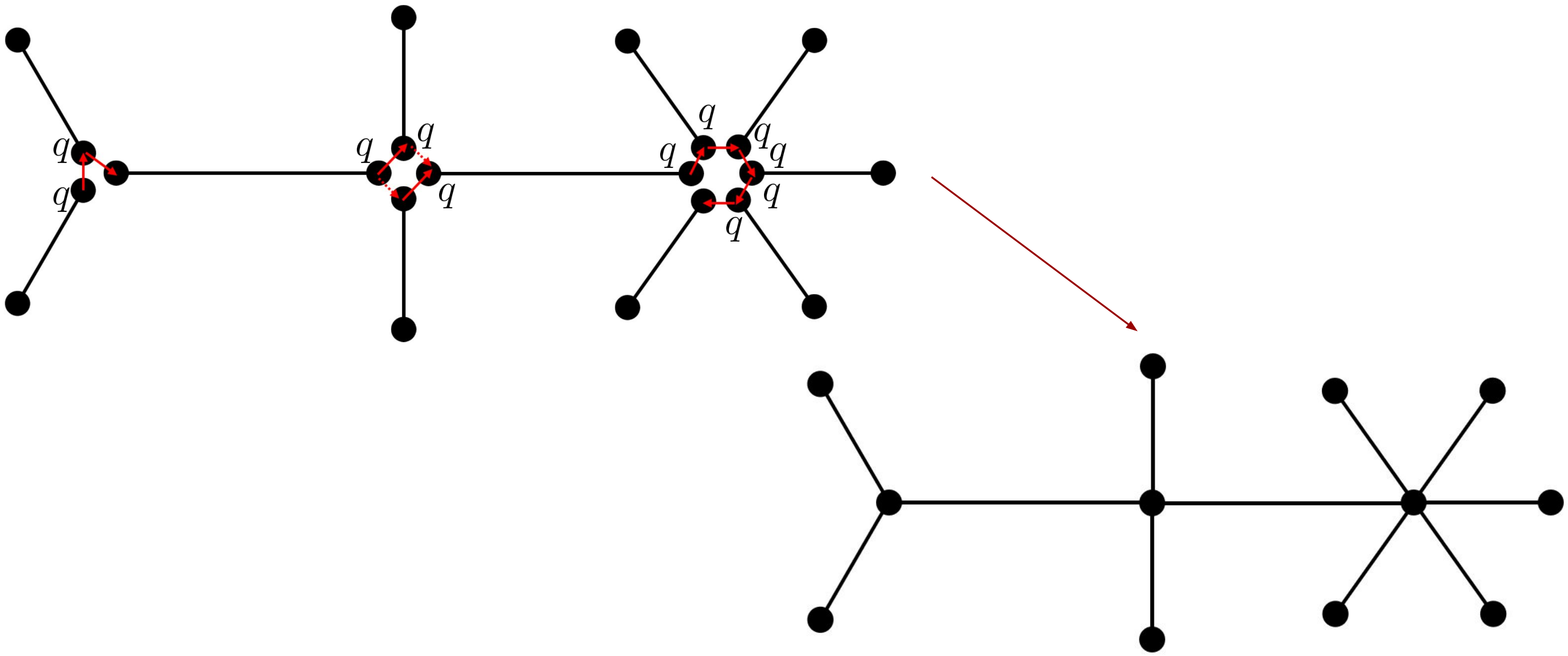
Entanglement generation with arbitrary connectivity

Step 3: Entangle modes within macronodes

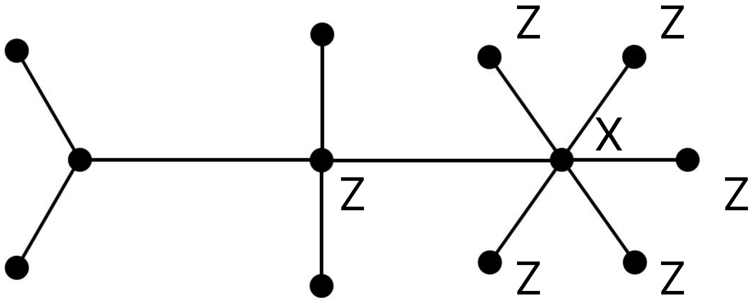


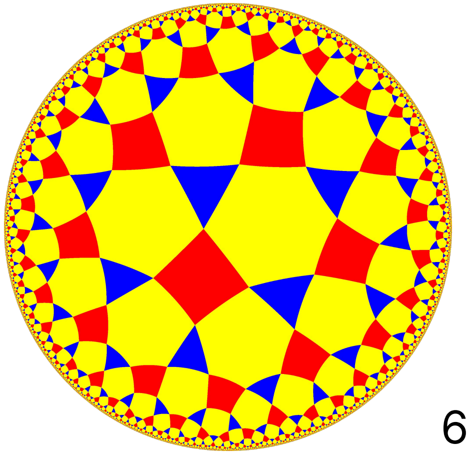
generalized splitters, e.g. linear cascade

Step 4: Finish reduction via homodyne measurements



Step 5: Implement QEC code cluster state of any valence





6.6.4.3 tiling

Implement checks corresponding to, e.g., a quantum code with arbitrary, potentially non-local, connectivity—at almost no cost to hardware

03

Weight reduction

For stabilizer codes

// Problem

The effective error in a stabilizer measurement usually scales with the weight of the stabilizer.

// Problem

The effective error in a stabilizer measurement usually scales with the weight of the stabilizer.

// Question

Given a code, can we reduce the weights of its stabilizers while retaining the desirable properties of the code?

Notation

- $[\![n,k,d]\!]$ stabilizer codes
  - Maximum stabilizer weight w
  - Maximum qubit degree q
- CSS codes
  - Parity-check matrices H_X and H_Z
  - Maximum stabilizer weights w_X and w_Z
  - Maximum qubit degrees q_X and q_Z

Weight reduction review

- Stabilizer code → subsystem / Floquet code [BD-CP+13, BFH+15, HH21, ...]
- Layer codes [WB23]
- Hastings's weight reduction [H16]
  - Works for any stabilizer code
  - Output code has stabilizer weights ≤ 5
  - Constant factor increase in n, constant factor decrease in d
- Hastings's procedure is complex
  - Essentially because of the requirement of preserving stabilizer commutation

Weight reduction review

- Weight reduction for classical codes
  - No need to worry about commutation
  - Can achieve check weights ≤ 3 [HHO21]
  - Checks can also be made geometrically local in two dimensions [B23]
- Many quantum code constructions use classical codes as input [TZ14, PK21, BE21, ...]

Classical weight reduction

- Let H be the parity-check matrix of an $[\,n,k,d\,]$ (classical) linear code
- Let h be a row of H with weight w
- Replace $h \mapsto [\,I_w\; 0\; H_w^{\mathsf{T}}\,]$ (H_w is the parity-check matrix of the w-bit repetition code)

$$
\begin{array}{cccccc}
v_1 & \cdots & v_w & & & \\
\end{array}
\!
\begin{array}{c}
\\
\begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix}
\end{array}
\mapsto
\begin{array}{c}
\begin{array}{ccccccccccc}
v_1 & v_2 & \cdots & v_{w-1} & v_w & & v'_1 & v'_2 & \cdots & v'_{w-2} & v'_{w-1}
\end{array} \\
\begin{pmatrix}
1 & & & & & \cdots & 1 & & & & \\
 & 1 & & & & \cdots & 1 & 1 & & & \\
 & & \ddots & & & \cdots & & & \ddots & & \\
 & & & 1 & & \cdots & & & & 1 & 1 \\
 & & & & 1 & \cdots & & & & & 1
\end{pmatrix}
\end{array}.
$$

Classical weight reduction: Example

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Classical weight reduction

- Algorithm
  - Input: parity-check matrix H
    1. Apply weight reduction to each row with weight ≥ 4
    2. Transpose the output and repeat 1
    3. Undo the transpose
  - Output: new parity-check matrix H' with row and column weights ≤ 3

Examples: Hypergraph product codes

- For an input linear code with parity-check matrix H and parameters $[\ n,k,d\ ]$, the hypergraph product code HGP(H) has
  - parameters $[\![\ \Theta(n^2), \Theta(k^2), \Theta(d)\ ]\!]$
  - parity-check matrices $H_X = (\ H \otimes I \ I \otimes H^T\ )$ and $H_Z = (\ I \otimes H^T \ H \otimes I\ )$
- Let r and c be the row and column weights of H
- HGP(H) has $w = r + c$ and $q = \max(\ r,c\ )$
- Classical weight reduction: HGP(H) $\mapsto$ HGP(H') with $w' = 6$ and $q' = 3$

Examples: Hypergraph product codes

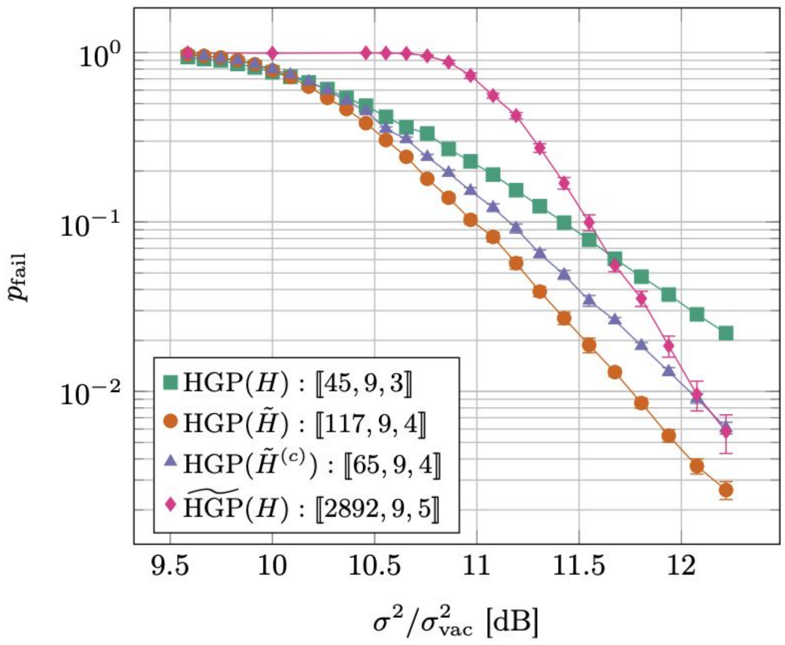
- *H* is the parity-check matrix of a [6,3,3] code, where *r* = 4 and *c* = 3
- HGP(*H*) has *w* = 7 and *q* = 4

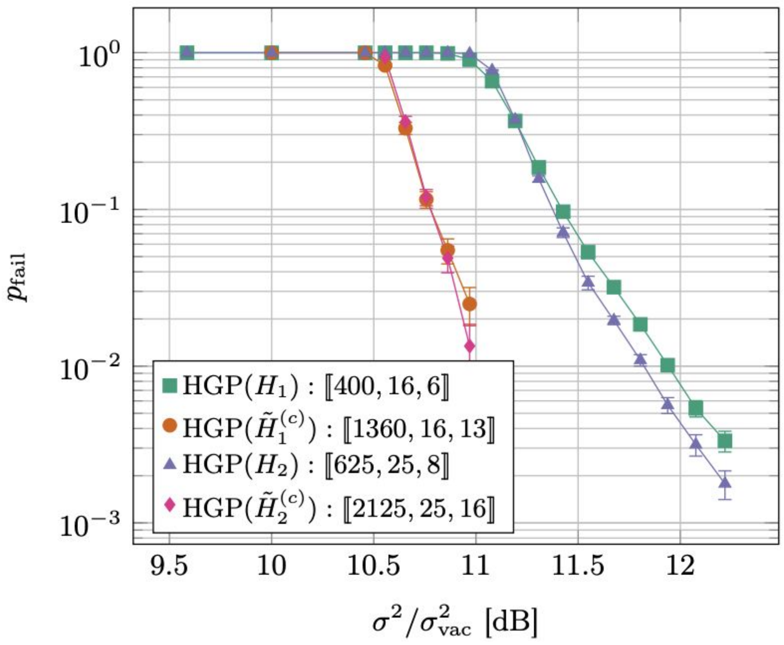
$\mathcal{C}(H)$	$\mathrm{HGP}(H)$	R	$\mathrm{HGP}(\tilde{H})$	R	$\mathrm{HGP}(\tilde{H}^{(c)})$	R
$[6,3,3]$	$[\![45,9,3]\!]$	0.200	$[\![117,9,4]\!]$	0.077	$[\![65,9,4]\!]$	0.138

- Compare with Hastings's weight reduction

$\mathcal{C}(H)$	$\mathrm{HGP}(H)$	R	(w_X, q_X, w_Z, q_Z)	$\widetilde{\mathrm{HGP}}(H)$	R	$(\tilde{w}_X, \tilde{q}_X, \tilde{w}_Z, \tilde{q}_Z)$
$[6,3,3]$	$[\![45,9,3]\!]$	0.200	(7, 4, 7, 4)	$[\![2892,9,5]\!]$	0.003	(6, 6, 6, 3)

Simulations: Xanadu architecture





Conclusion

Summary

- Classical weight reduction can reduce the stabilizer weights of quantum product codes while retaining competitive parameters
- The weight reduced codes have superior performance in Xanadu’s architecture for a fault-tolerant photonic quantum computer
- Paper on the arXiv [SGI+24]
  - Also contains a self-contained explanation of Hastings’s weight reduction algorithm with lots of examples (and some optimizations)
  - Both algorithms at https://github.com/esabo/CodingTheory

Open questions

- Can Hastings's algorithm be improved further?
- How does classical weight reduction compare with the layer codes approach?
- How to compare with the stabilizer → subsystem code approach?
- What is the lowest stabilizer weight compatible with e.g. constant encoding rate?

References

- [Sho96] Peter W. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”, *SIAM Journal on Computing*. https://doi.org/10.1137/S0097539795293172
- [BBM+20] Bela Bauer, Sergey Bravyi, Mario Motta & Garnet Kin-Lic Chan “Quantum Algorithms for Quantum Chemistry and Quantum Materials Science”, *Chemical Reviews*. https://doi.org/10.1021/acs.chemrev.9b00829
- [OML19] Román Orús, Samuel Mugel & Enrique Lizaso, “Quantum computing for finance: Overview and prospects”, *Reviews in Physics*. https://doi.org/10.1016/j.revip.2019.100028

References

- [Kit03] A. Yu. Kitaev, “Fault-tolerant quantum computation by anyons”, *Annals of Physics*. https://doi.org/10.1016/S0003-4916(02)00018-0
- [Kit06] Alexei Kitaev, “Anyons in an exactly solved model and beyond”, *Annals of Physics*. https://doi.org/10.1016/j.aop.2005.10.005
- [PYH+15] Fernando Pastawski, Beni Yoshida, Daniel Harlow & John Preskill, “Holographic quantum error-correcting codes: toy models for the bulk/boundary correspondence”, *Journal of High Energy Physics*. https://doi.org/10.1007/JHEP06(2015)149

References

- [Got97] Daniel Gottesman, “Stabilizer Codes and Quantum Error Correction”, *PhD thesis (Caltech)*. https://doi.org/10.48550/arXiv.quant-ph/9705052
- [RBB03] Robert Raussendorf, Daniel E. Browne & Hans J. Briegel, “Measurement-based quantum computation on cluster states”, *Physical Review A*. https://doi.org/10.1103/PhysRevA.68.022312
- [BD-CP+16] A. Bolt, G. Duclos-Cianci, D. Poulin & T. M. Stace, “Foliated Quantum Error-Correcting Codes”, *Physical Review Letters*. https://doi.org/10.1103/PhysRevLett.117.070501

References

- [RHG06] R. Raussendorf, J. Harrington & K. Goyal, “A fault-tolerant one-way quantum computer”, *Annals of Physics*. https://doi.org/10.1016/j.aop.2006.01.012
- [WBA+20] Blayney W. Walshe, Ben Q. Baragiola, Rafael N. Alexander & Nicolas C. Menicucci, “Continuous-variable gate teleportation and bosonic-code error correction”, *Physical Review A*. https://doi.org/10.1103/PhysRevA.102.062411
- [TMA+21] Ilan Tzitrin, Takaya Matsuura, Rafael N. Alexander, et al., “Fault-Tolerant Quantum Computation with Static Linear Optics”, *PRX Quantum*. https://doi.org/10.1103/PRXQuantum.2.040353

References

- [BD-CP+13] Sergey Bravyi, Guillaume Duclos-Cianci, David Poulin & Martin Suchara, “Subsystem surface codes with three-qubit check operators”, *Quantum Information & Computation*. https://doi.org/10.48550/arXiv.1207.1443
- [BFH+15] Dave Bacon, Steven T. Flammia, Aram W. Harrow & Jonathan Shi, “Sparse Quantum Codes from Quantum Circuits”, *IEEE Transactions on Information Theory*. https://doi.org/10.1145/2746539.2746608
- [HH21] Matthew B. Hastings & Jeongwan Haah, “Dynamically Generated Logical Qubits”, *Quantum*. https://doi.org/10.22331/q-2021-10-19-564
- [WB23] Dominic J. Williamson, Nouédyn Baspin, “Layer Codes”, *arXiv preprint*. https://doi.org/10.48550/arXiv.2309.16503

References

- [H16] M. B. Hastings, “On Quantum Weight Reduction”, *arXiv preprint*. https://doi.org/10.48550/arXiv.2102.10030
- [HHO21] Matthew B. Hastings, Jeongwan Haah & Ryan O’Donnell, “Fiber Bundle Codes: Breaking the $N^{1/2}$polylog(N) Barrier for Quantum LDPC Codes”, *STOC ‘21*. https://doi.org/10.1145/3406325.3451005
- [B23] Nouédyn Baspin, “On combinatorial structures in linear codes”, *arXiv preprint*. https://doi.org/10.48550/arXiv.2309.16411
- [TZ14] Jean-Pierre Tillich & Gilles Zemor, “Quantum LDPC codes with positive rate and minimum distance proportional to $n^{1/2}$”, *IEEE Transactions on Information Theory*. https://doi.org/10.1109/TIT.2013.2292061

References

- [PK21] Pavel Panteleev & Gleb Kalachev, "Quantum LDPC Codes with Almost Linear Minimum Distance", *IEEE Transactions on Information Theory*. https://doi.org/10.1109/TIT.2021.3119384
- [BE21] Nikolas P. Breuckmann & Jens N. Eberhardt, "Balanced Product Quantum Codes", *IEEE Transactions on Information Theory*. https://doi.org/10.1109/TIT.2021.3097347
- [SGI+24] Eric Sabo, Lane G. Gunderman, Benjamin Ide, Michael Vasmer, Guillaume Dauphinais, "Weight Reduced Stabilizer Codes with Lower Overhead", *arXiv preprint*. https://doi.org/10.48550/arXiv.2402.05228

Thank you

xanadu.ai
Twitter → @xanadu.ai

Michael Vasmer
michael.vasmer@xanadu.ai

Classical weight reduction

Theorem

Let H be the parity-check matrix of an $[\, n,k,d \,]$ binary, linear code, w be the maximum row weight of H, and q be the maximum column weight of H. Then Algorithm 1 outputs a parity-check matrix H' with $w' = q' = 3$ whose code has parameters $[\, N,k,D \,]$, where N = O(max(w, q) n) and $D \geq d$.

Proof

In the worst case, all checks have weight ≥ 4, which gives the bound on N.

Weight reducing the rows does not change the column weights and vice versa, so $w' = q' = 3$.

Classical weight reduction

$$f\big|_{\mathrm{supp}\,h_i} = I_{w_i}$$

$$\left(\begin{array}{c|c} h_1 & \\ \vdots & 0 \\ h_{i-1} & \\ \hline f & H^{\mathrm{T}}_{w_i\text{-}1} \\ \hline h_{i+1} & \\ \vdots & 0 \\ h_{n-k} & \end{array}\right),$$

Bits to the left = old, bits to the right = new.

Suppose c is a codeword of this parity-check matrix.

$$0 = \begin{pmatrix} f & H^{\mathrm{T}}_{w_i-1} \end{pmatrix} c = \begin{pmatrix} f & H^{\mathrm{T}}_{w_i-1} \end{pmatrix} \begin{pmatrix} c\big|_{\mathrm{old}} \\ c\big|_{\mathrm{new}} \end{pmatrix} = fc\big|_{\mathrm{old}} + H^{\mathrm{T}}_{w_i-1} c\big|_{\mathrm{new}}$$

$$H^{\mathrm{T}}_{w_i-1} c\big|_{\mathrm{new}} = fc\big|_{\mathrm{old}} = c\big|_{\mathrm{supp}\,h_i}$$

The image of $H_w{}^T$ contains only even weight strings.

Therefore $c|_{\mathrm{old}}$ is a codeword of the original code.

If $c|_{\mathrm{old}} = 0$ then $c|_{\mathrm{new}} = 0$ as $\ker H_w{}^T = 0$.

Hence $K = k$ and $D \geq d$. ∎

Examples: Lifted product codes

- The lifted product is a generalization of the hypergraph product to (amongst other things) quasi-cyclic code inputs.
- Quasi-cyclic codes are defined using matrices whose entries are elements of a polynomial quotient ring. Classical weight reduction generalizes straightforwardly to this case.
- Lifted product codes often have superior parameters to hypergraph product codes of similar size.

Examples: Lifted product codes

- Each weight reduction step can be randomized: $h \mapsto [\ \Pi\ I_w\ 0\ H_w^{\top}\ ]$.
- We find empirically that this can give substantial increases in the distance.

$\mathcal{C}(A)$	$\mathrm{LP}(A)$	R	$\mathcal{C}(\tilde{A})$	$\mathrm{LP}(\tilde{A})$	R	$\mathcal{C}(\tilde{A}^{(c)})$	$\mathrm{LP}(\tilde{A}^{(c)})$	R
$[52, 27, 6]$	$[[260, 58, \leq 6]]$	0.223	$[130, 27, 12 \to 14]$	$[[2132, 58, \leq 14]]$	0.027	$[78, 27, 6 \to 8]$	$[[676, 58, \leq 8]]$	0.086
$[28, 9, 10]$	$[[175, 19, \leq 10]]$	0.109	$[91, 9, 28 \to 33]$	$[[2191, 19, \leq 39]]$	0.009	$[49, 9, 14 \to 18]$	$[[595, 19, \leq 18]]$	0.032
$[36, 11, 12]$	$[[225, 21, \leq 12]]$	0.093	$[117, 11, 36 \to 40]$	$[[2817, 21, \leq 48]]$	0.007	$[63, 11, 18 \to 22]$	$[[765, 21, \leq 22]]$	0.027
$[68, 19, 18]$	$[[425, 29, \leq 18]]$	0.068	$[221, 19, 54 \to 62]$	$[[5321, 29, \leq 74]]$	0.005	$[119, 19, 32 \to 34]$	$[[1445, 29, \leq 34]]$	0.020
$[76, 21, 20]$	$[[475, 31, \leq 20]]$	0.065	$[247, 21, 60 \to 68]$	$[[5947, 31, \leq 93]]$	0.005	$[133, 21, 37 \to 38]$	$[[1615, 31, \leq 38]]$	0.019
$[124, 33, 24]$	$[[775, 43, \leq 24]]$	0.055	$[403, 33, 71 \to 84]$	$[[9703, 43, \leq 115]]$	0.004	$[217, 33, 44 \to 48]$	$[[2635, 43, \leq 48]]$	0.016