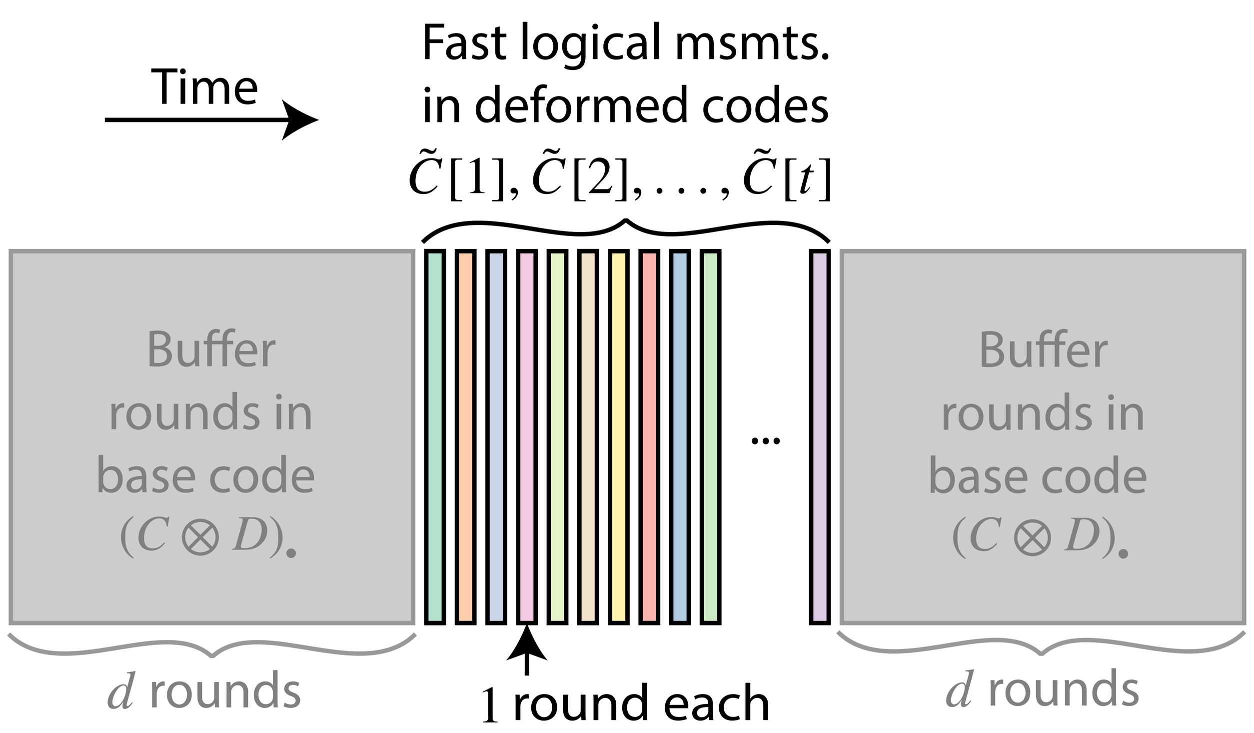
Constant Time Surgery on 2D Hypergraph Product Codes with Near-Constant Space Overhead

Kathleen (Katie) Chang*[1,2] · Zhiyang He*[3] · Theodor J. Yoder[1] · Guanyu Zhu[1] · Tomas Jochym-O'Connor[1]

[1]IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
[2]Department of Physics, Yale University, New Haven, Connecticut 06511, USA
[3]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Generalized code surgery is a versatile and low-overhead technique for performing fault-tolerant computation on quantum low-density parity-check (qLDPC) codes. In many settings, surgery exhibits practical space overheads, while its time overhead remains a bottleneck at O(d) syndrome rounds per operation. In this work, we construct surgery gadgets that perform **parallel logical measurements** on **2D hypergraph product codes** in **constant time overhead ($\mathcal{O}(1)$)** and **near-constant space overhead ($\tilde{\mathcal{O}}(1)$).** The reduced time overhead is a result of amortization, as we show, following the formulation by Cowtan et al. (arXiv:2510.14895), that performing d surgery operations in $\mathcal{O}(d)$ time is fault tolerant. Our gadgets combine the strengths of different approaches to fault-tolerant logical operations: they partially retain the **flexibility of surgery** while **achieving overheads comparable to transversal gates**.

Main results



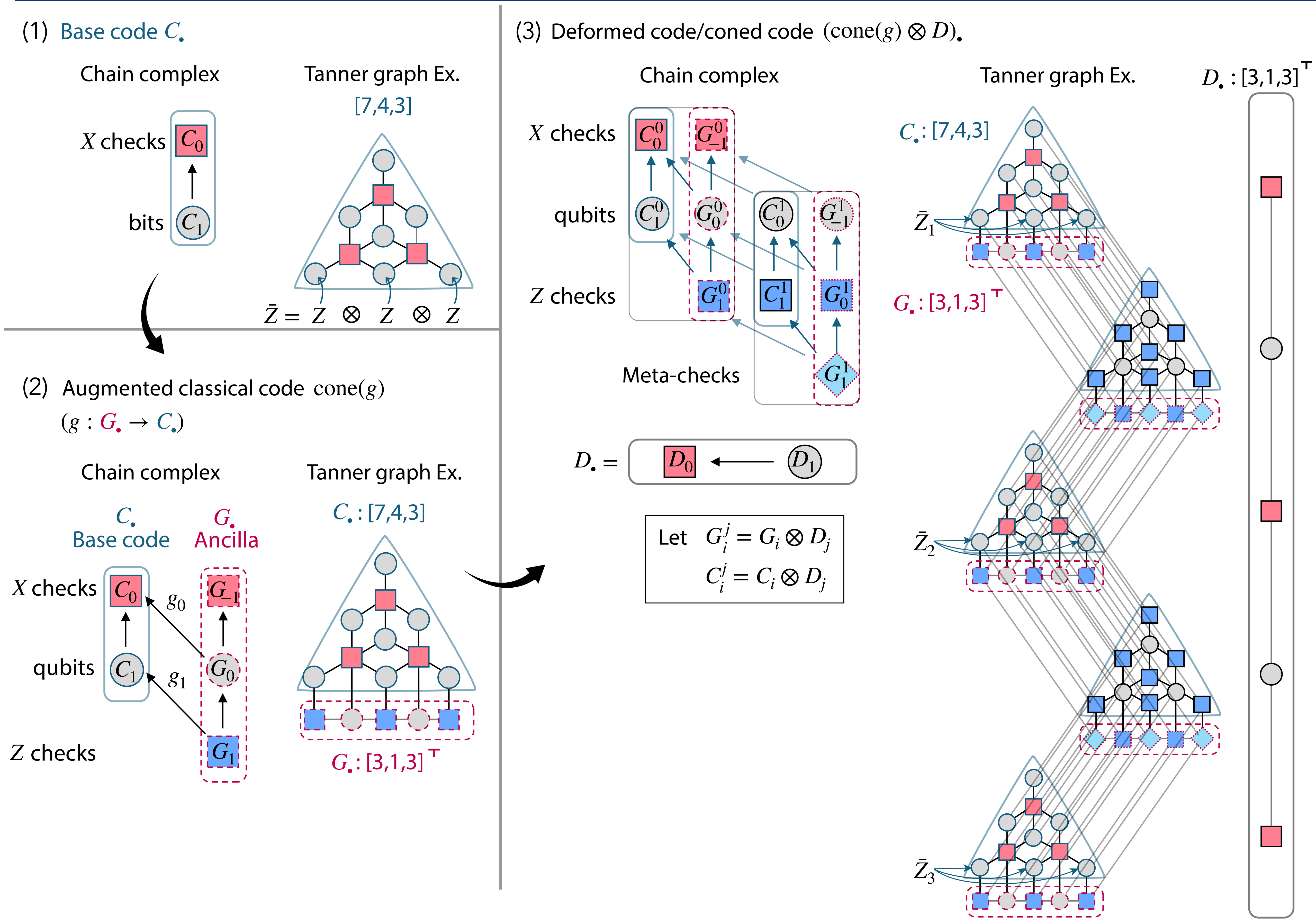
Constant time surgery in amortization

1. Measure d rounds in the base hypergraph product code, $(C \otimes D)_\bullet$
2. Measure 1 round of stabilizers of the deformed code $\tilde{C}[1]_\bullet$, which consists of the ancilla complex $A[1]_\bullet$ attached to $(C \otimes D)_\bullet$. This performs a logical measurement of $(C \otimes D)_\bullet$.
3. Repeat Step 2 for a sequence of deformed codes, $\tilde{C}[2]_\bullet, \ldots, \tilde{C}[t]_\bullet$, each performing a different logical measurement.
4. Measure d rounds in the base hypergraph product code, $(C \otimes D)_\bullet$

Each logical measurement takes $\mathcal{O}(1)$ rounds (for $t \geq d$).

Theorem 1. Given a $[[n,k,d]]$ hypergraph product code $(C \otimes D)_\bullet$, we can construct a sequence of ancillary complexes $A[1]_\bullet, \ldots, A[t]_\bullet$ which **satisfy fault-tolerant fast surgery conditions in [1]**. Performing fast surgery with these complexes, lets us measure **rows (and similarly columns) of Z logical operators (and similarly X operators)** in **parallel**, with **each surgery operation taking $\mathcal{O}(1)$ times in amortization**. Furthermore, each complex utilizes $\tilde{\mathcal{O}}(n)$ **physical ancilla qubits**. Therefore the overall spacetime overhead of these logical gates is $\tilde{\mathcal{O}}(1)$.

Constant time surgery complexes



Ex: Constructing deformed code $\tilde{C}[i]$

(Left) chain complex. (Right) tanner graph.

(1) **Take a classical code, $C_\bullet$**, used in the base hypergraph product code, $(C \otimes D)_\bullet$ Here, we show the [7,4,3] Hamming code.

(2) **Augment the classical code** by attaching an ancilla complex, $G_\bullet$, to $C_\bullet$ with a chain map, $g_i : G_i \to C_i$. This ancilla complex measures a logical of $C_\bullet$, and can be constructed with known techniques [2,3,7]. Here, we choose $G_\bullet$ to be the dual of the 3-bit repetition code. It measures the $\bar{Z}$ logical highlighed in (1).

(3) The final deformed code is the **hypergraph product of the augmented classical code in (2) with $D_\bullet$**, the other classical code in $(C \otimes D)_\bullet$. For this example, we take $D_\bullet$ to also be the dual of the 3-bit repetition code.

Perspective & Outlook

- We construct constant time surgery gadgets for **codes without single-shot state preparation.** (See previous work [4-6]).
- These gadgets maintain a very low space overhead of $\mathcal{O}(n \log(n))$, **avoiding trading space for time**. (Using techniques in [7]).
- Future work includes finding constant time with low space overhead gadgets for **other product codes**, such as lifted product and balanced product codes

References

[1] CHWY'25 (arXiv:2510.14895)
[2] WY'24 (arXiv:2410.02213)
[3] IGND'25 (PRX **15,** 10.1103)
[4] BBC'25 (arXiv:2510.04521)
[5] HDYV'24 (arXiv:2410.12963)
[6] THLGH'25 (arXiv:2510.08552)
[7] YCHLW'26 (arXiv:2603.05082)