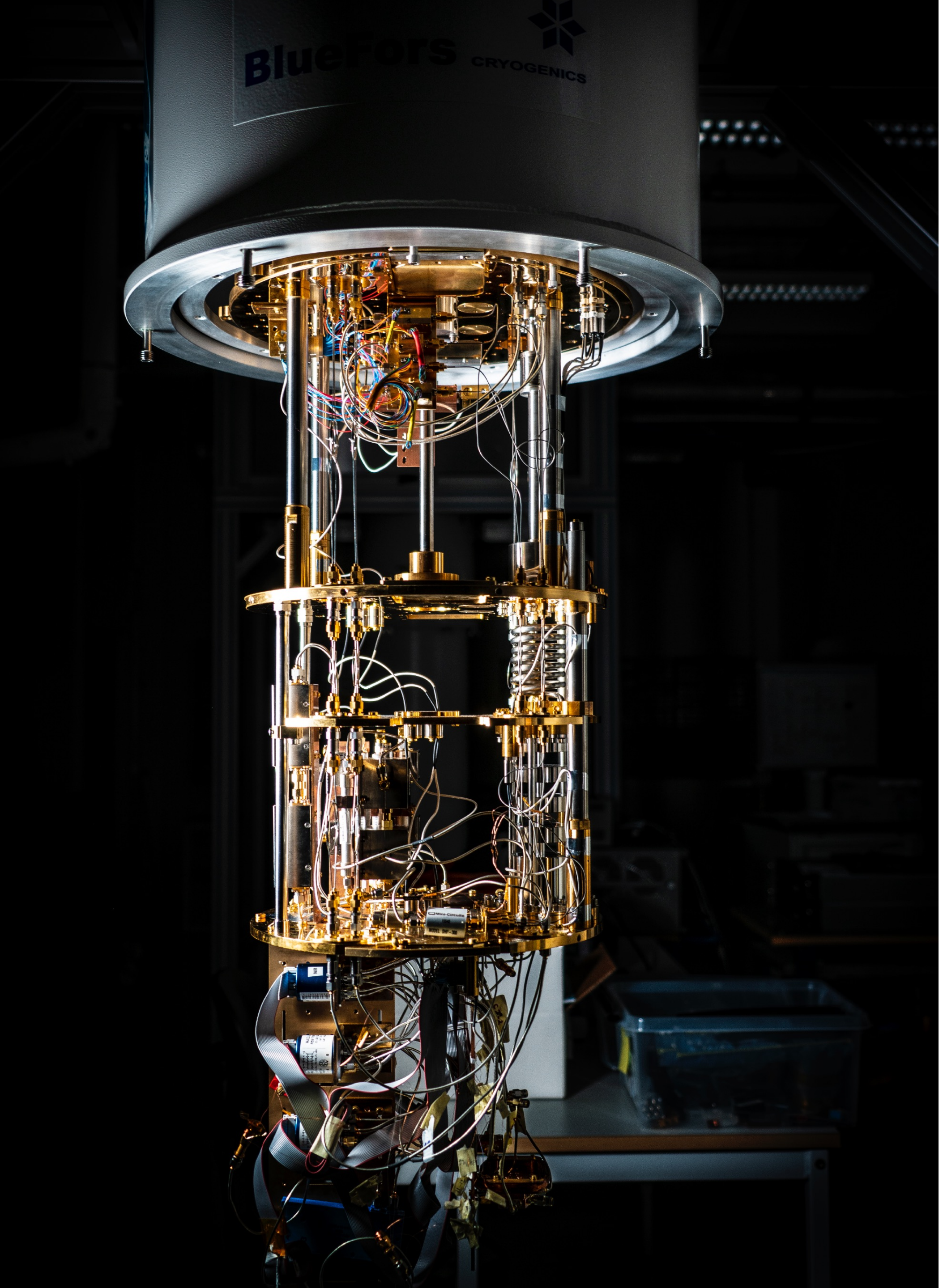


Graph neural network based decoders for quantum error correcting codes

Mats Granath
Department of Physics
University of Gothenburg

2nd Workshop of Machine Learning for Quantum Technology
Max Planck Institute for the Science of Light
Erlangen
Nov 6, 2024

WACQT Wallenberg Centre for Quantum Technology

UNIVERSITY OF GOTHENBURG

Quantum computing limited by decoherence

Superconducting qubits:

- lifetime - 100 microsecond
- two qubit gate times – few 100 nanoseconds

Maximum few 100 gates deep circuits (error rates ~ 10^{-3})

Photograph by Thomas Prior for TIME

To factor $N=2^{2048}$ size integer using Shor's algorithm takes > $(\log N)^2 = 10^7$ deep circuit. (error rates < 10^{-7})

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney[1] and Martin Ekerå[2,3]

[1]Google Inc., Santa Barbara, California 93117, USA
[2]KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
[3]Swedish NCSA, Swedish Armed Forces, SE-107 85 Stockholm, Sweden

Published: 2021-04-15, volume 5, page 433
Eprint: arXiv:1905.09749v3
Doi: https://doi.org/10.22331/q-2021-04-15-433
Citation: Quantum 5, 433 (2021).

Assuming all gates readily available!

Longer qubit lifetimes needed to get ``quantum advantage''?

Quantum error correction

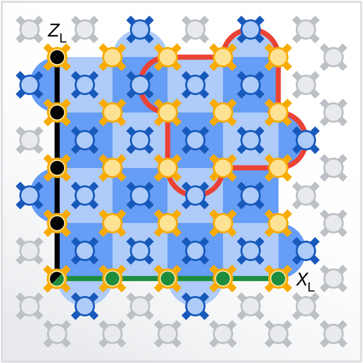
Distribute information over many physical qubits --> Lower error rate logical qubit

Peter W. Shor, "Scheme for reducing decoherence in quantum computer memory," Physical Review A **52**, R2493–R2496 (1995).
A. M. Steane, "Error Correcting Codes in Quantum Theory," Physical Review Letters **77**, 793–797 (1996).
Daniel Gottesman, "Stabilizer Codes and Quantum Error Correction," (1997), arXiv:quant-ph/9705052.

Article **676** | Nature | Vol 614 | 23 February 2023

Suppressing quantum errors by scaling a surface code logical qubit

https://doi.org/10.1038/s41586-022-05434-1
Google Quantum AI*
Received: 13 July 2022



25 qubit "surface code"

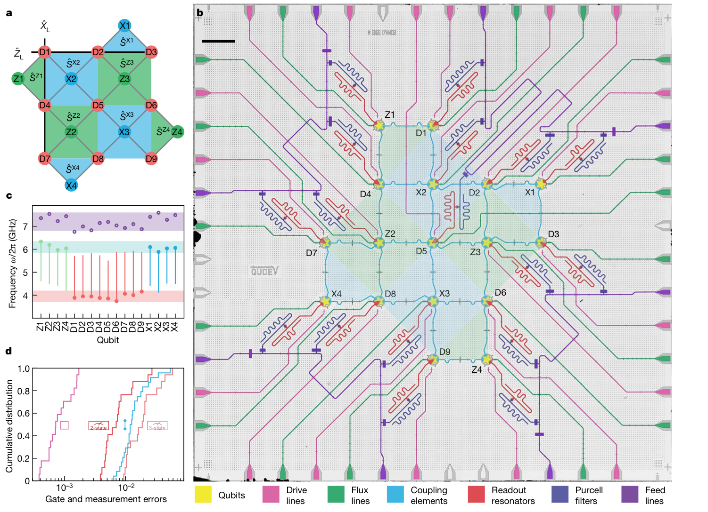
Article

Realizing repeated quantum error correction in a distance-three surface code

https://doi.org/10.1038/s41586-022-04566-8
Received: 15 November 2021
Accepted: 9 February 2022
Published online: 25 May 2022

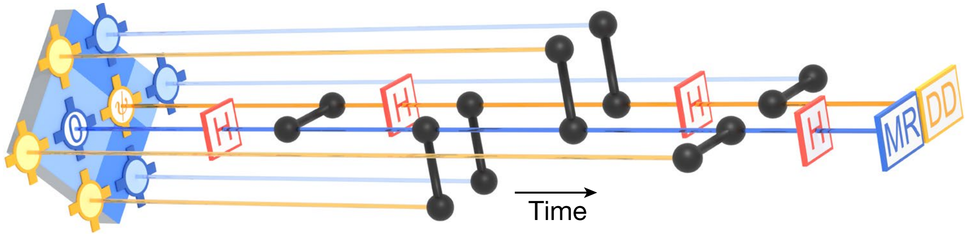
Sebastian Krinner[1,9], Nathan Lacroix[1,9], Ants Remm[1], Agustin Di Paolo[2,3], Elie Genois[2,3], Catherine Leroux[2,3], Christoph Hellings[1], Stefania Lazar[1], Francois Swiadek[1], Johannes Herrmann[1], Graham J. Norris[1], Christian Kraglund Andersen[1,8], Markus Müller[4,5], Alexandre Blais[2,3,6], Christopher Eichler[1] & Andreas Wallraff[1,7]

Article



9 qubit "surface code"

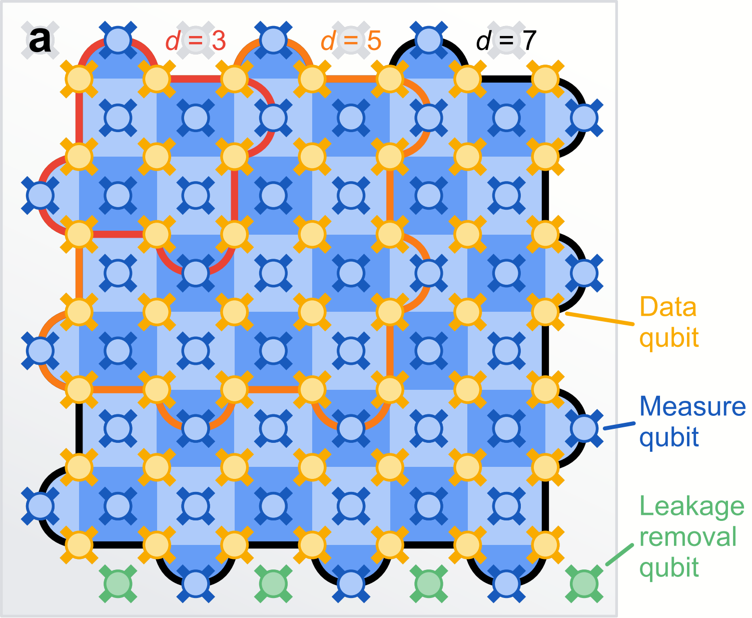
Repeated projective parity measurements allows for error correction



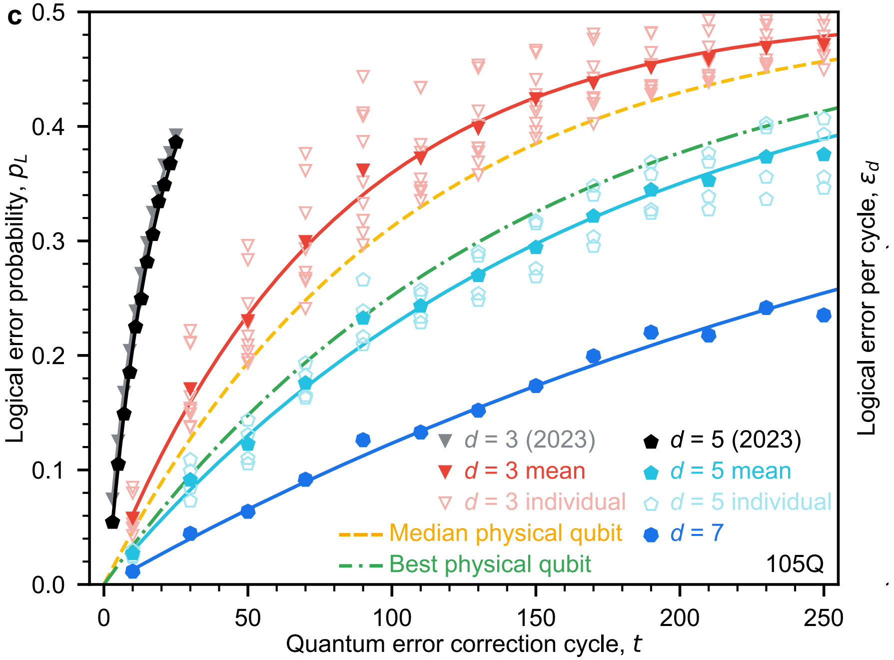
Scalable error suppression

Quantum error correction below the surface code threshold

Google Quantum AI and Collaborators
(Dated: August 27, 2024)



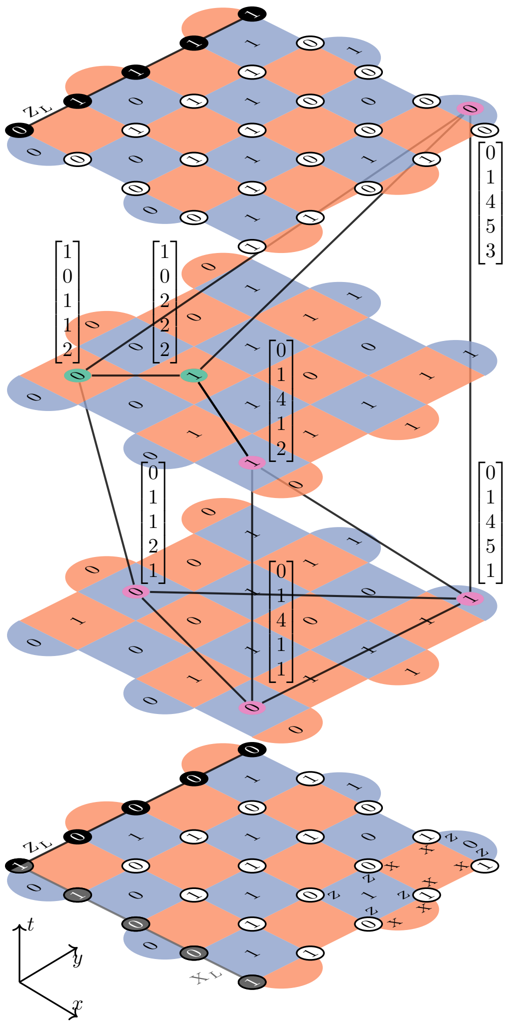
49 qubit surface code



Best results rely on machine learning for decoding! Bausch et al. 2023

Outline

- The surface code and the decoding problem
- Matching Decoders
- Why machine learning decoders?
- Graph neural networks
- Results and work in progress



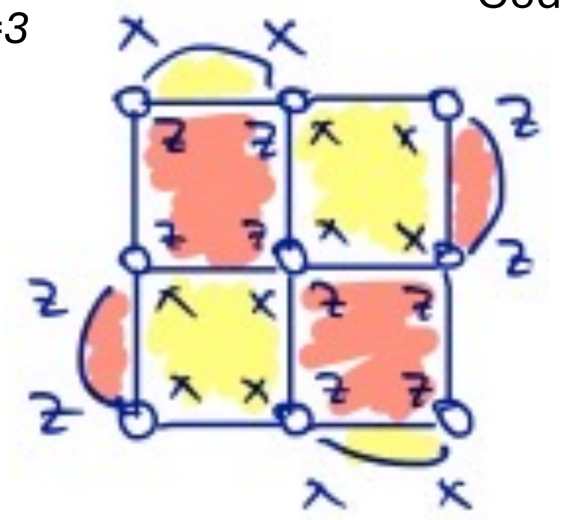
Surface code recap

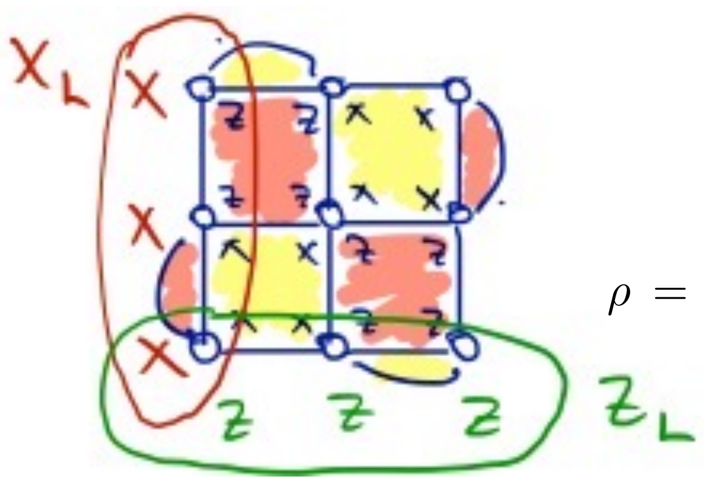
S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary (1998), arXiv:quant-ph/9811052.
E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, Journal of Mathematical Physics **43**, 4452 (2002).
A. Kitaev, Fault-tolerant quantum computation by anyons, Annals of Physics **303**, 2 (2003).

Planar version of Kitaev's toric code

- $n=d^2$ (data) qubits
- d^2 -1, 4 and 2 qubit stabilizer (generators)
- Commuting and independent
- $k=1$ logical qubit
- Code-distance d

Example $d=3$





$$Z_L|0\rangle_L = |0\rangle_L$$

$$|1\rangle_L = X_L|0\rangle_L$$

$$\rho = \sum_{i,j\in\{0,1\}} \rho_{ij}|i\rangle_L\langle j|_L$$

- Hilbert space partitioned by the ±1 eigenvalues of the stabilizers into d^2 -1, 2-dimensional sectors
- Any of these can serve as the logical qubit

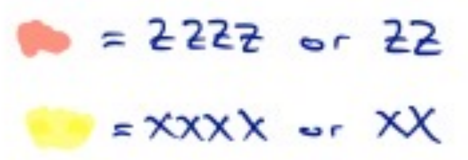
Logical Pauli operators:

commute with stabilizers

outside stabilizer group

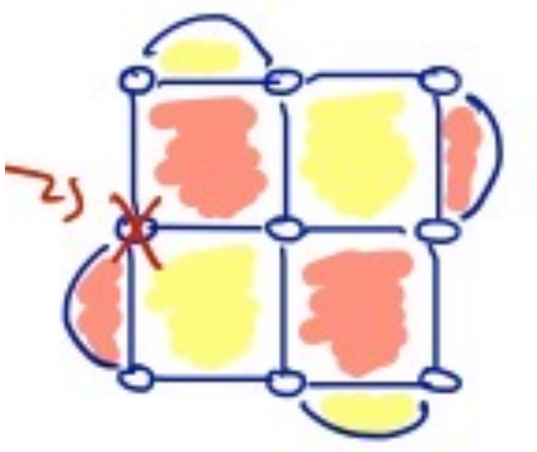
minimal undetectable error=code-distance

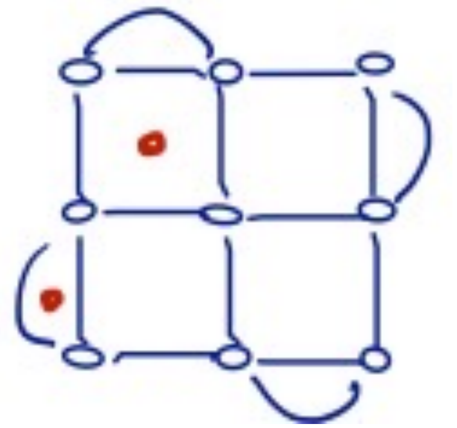
Decoding basics



Syndrome is measured
Not the error

Bit-flip error

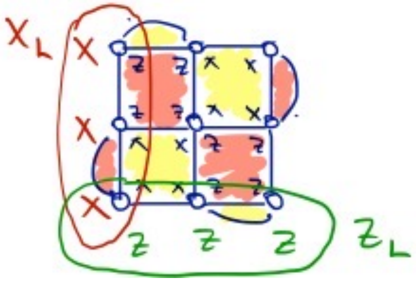




This is my example sydrome

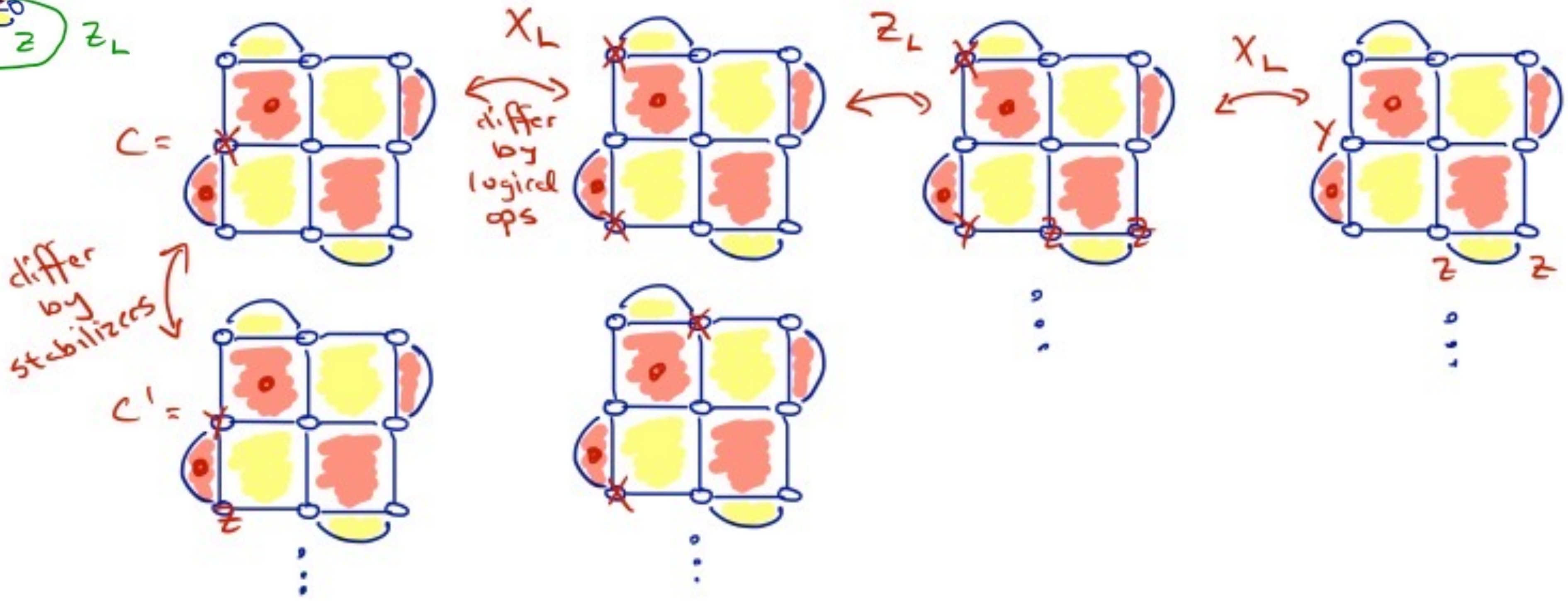
Decoder: Syndrome => Correction

Challenge: 2^{d^2+1} errors (Pauli strings) consistent with any 1 syndrome!



Equivalence classes of errors

Class I Class X Class Y Class Z

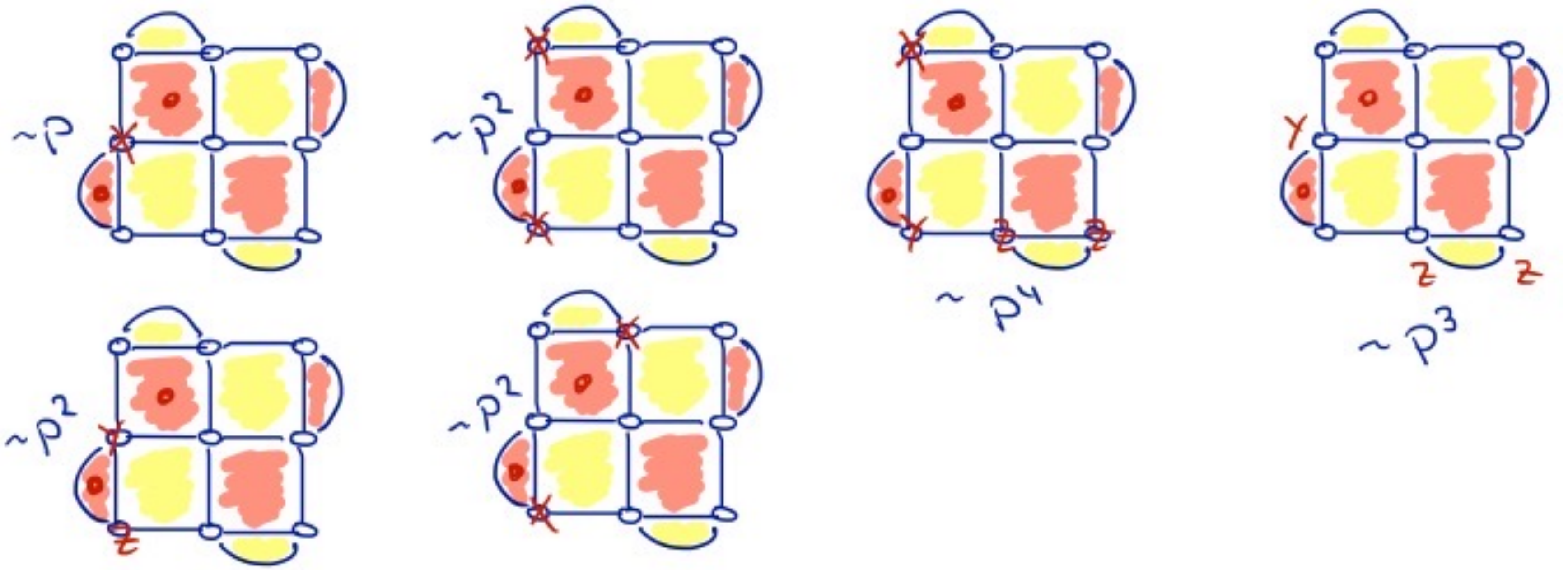


- Errors within a class are equivalent, since the logical qubit is an eigenstate of any stablizer
- Optimal decoder: suggest a correction from the most likely class

Estimating class probabilities

Consider simple depolarizing iid error channel $p_x = p_y = p_z = p/3$

Class *I* Class *X* Class *Y* Class *Z*



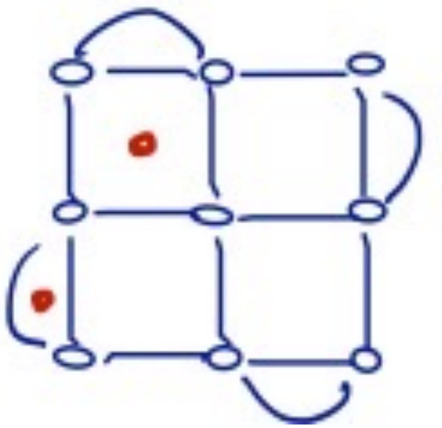
Probability of an error chain C, with n_c errors: $\pi_C = (p/3)^n(1-p)^{N-n} = (1-p)^N(\frac{p/3}{1-p})^n = (1-p)^N e^{-n/T}$ $1/T = -\ln(\frac{p/3}{1-p})$

Probability of an equivalence class *E*: $P_E \sim Z_E = \sum_{C \in E} e^{-n_C/T}$

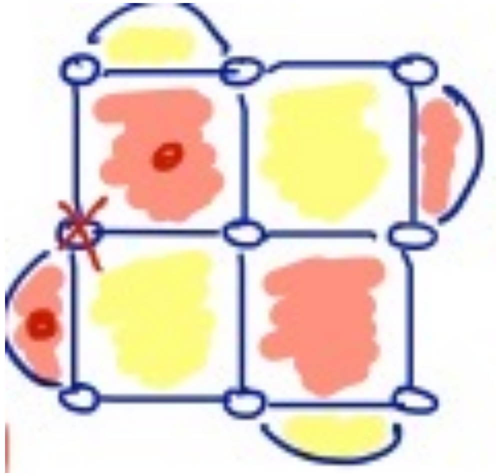
Optimal, Maximum-likelihood decoder: calculate the partition functions

Asymptotic logical failure rate

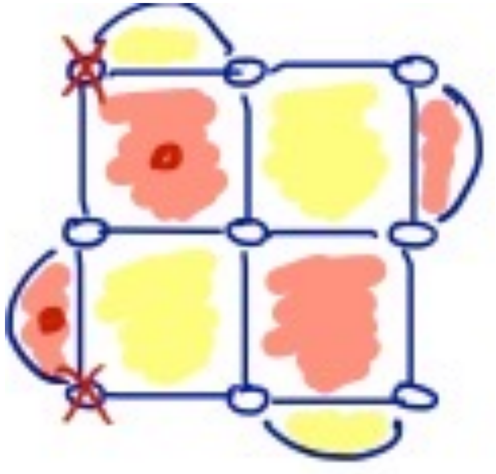
Assume error rate $p \ll 1$



Most likely error



Most likely error in other class



When this error occurs
=> logical (bit-flip) failure

Asymptotic logical failure: $P_f \sim p^{(d+1)/2}$

Logical errors are exponentially supressed with code distance *d*

Decoders, MLD versus MLE

MLD: Maximum-likelihood decoder.

MLE: Most likely error decoder

Metropolis - Monte Carlo

PRL **109,** 160503 (2012) PHYSICAL REVIEW LETTERS week ending 19 OCTOBER 2012

High Threshold Error Correction for the Surface Code

James R. Wootton and Daniel Loss
Department of Physics, University of Basel, Klingelbergstrasse 82, CH-4056 Basel, Switzerland
(Received 1 March 2012; published 18 October 2012)

PHYSICAL REVIEW A **105**, 042616 (2022)

Error-rate-agnostic decoding of topological stabilizer codes

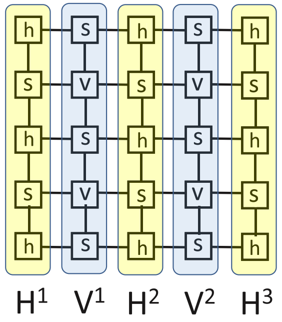
Karl Hammar,[1] Alexei Orekhov,[2] Patrik Wallin Hybelius,[1] Anna Katariina Wisakanto,[1]
Basudha Srivastava,[1] Anton Frisk Kockum,[2] and Mats Granath[1,*]
[1]*Department of Physics, University of Gothenburg, 41296 Gothenburg, Sweden*
[2]*Department of Microtechnology and Nanoscience, Chalmers University of Technology, 41296 Gothenburg, Sweden*

Tensor network based

PHYSICAL REVIEW A **90**, 032326 (2014)

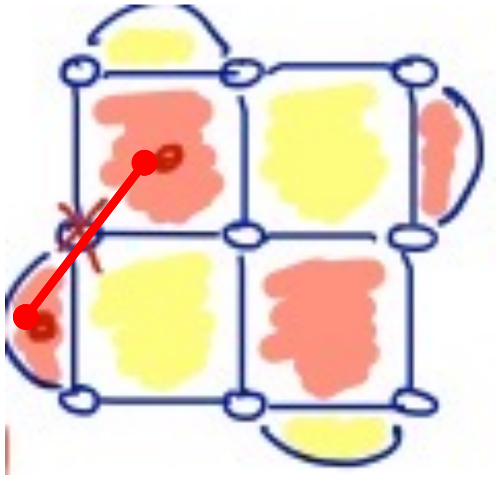
Efficient algorithms for maximum likelihood decoding in the surface code

Sergey Bravyi, Martin Suchara, and Alexander Vargo
IBM Watson Research Center, Yorktown Heights, New York 10598, USA
(Received 23 June 2014; published 25 September 2014)



Accurate but slow.

Matching decoders (next slide)

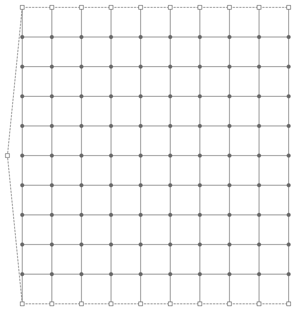


D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, Threshold error rates for the toric and planar codes, Quantum Inf. Comput. **10**, 456 (2010).
D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Surface code quantum computing with error rates over 1%, Phys. Rev. A **83**, 020302(R) (2011).

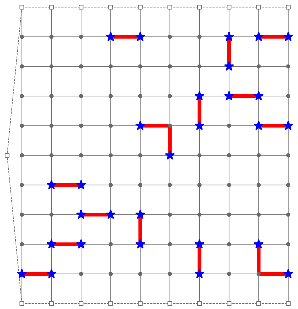
Suboptimal, but fast

Matching decoders (Dijkstra + Blossom)

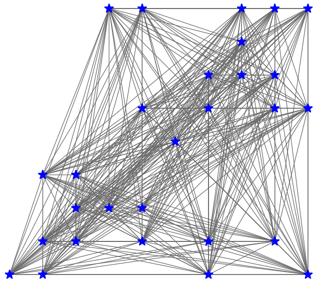
Encodes propabilites of errors as edge-weights



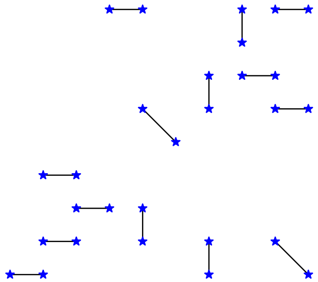
(a) Matching graph



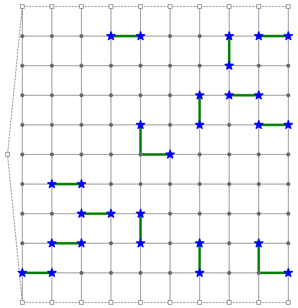
(b) Error



(c) Syndrome graph



(d) Minimum-weight perfect matching



(e) Correction

- Fast
- Accurate

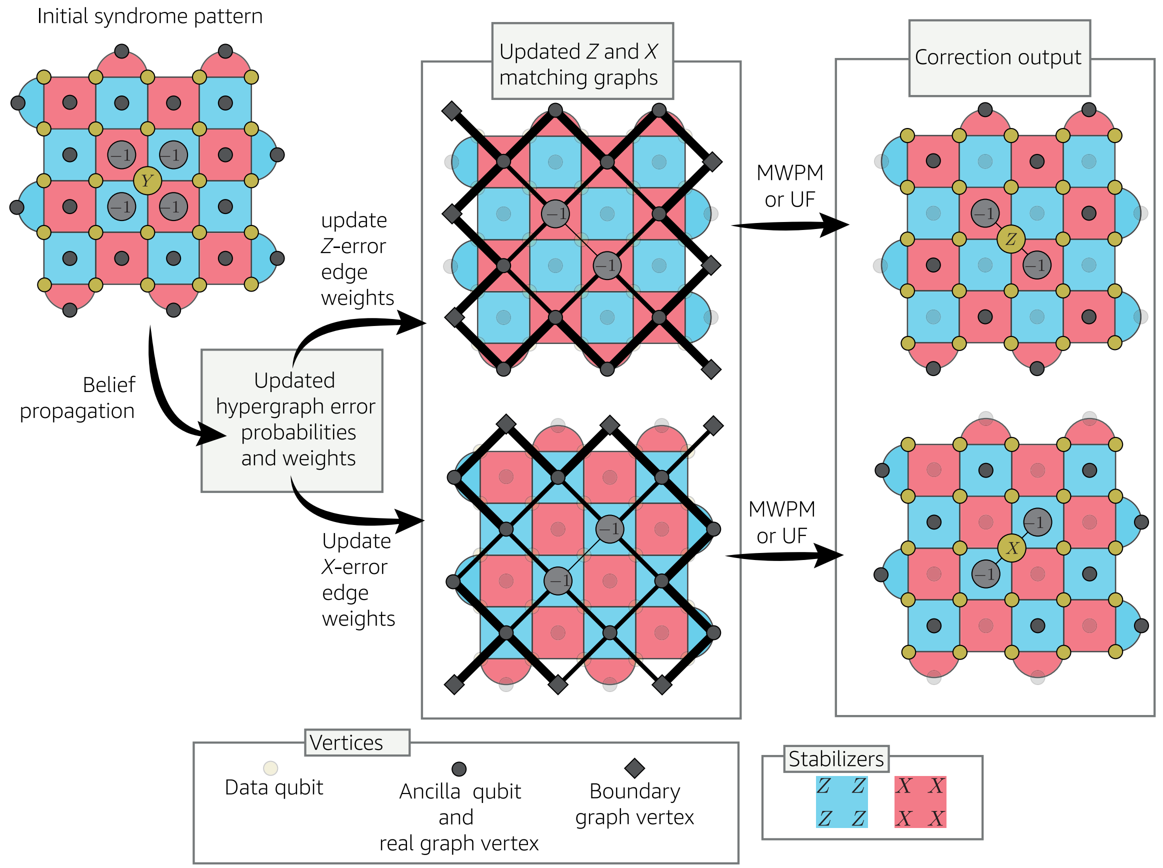
- Suboptimal, X and Z stabilizers decoded separately, and only identifies most likely error

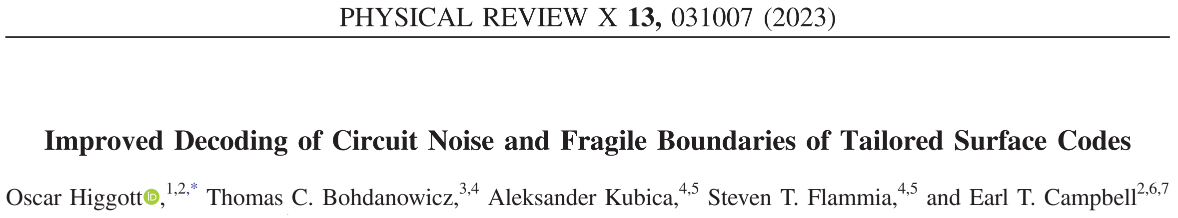
PyMatching: A Python package for decoding quantum codes with minimum-weight perfect matching

Oscar Higgott*[1]

Belief-matching

Accounts for correlations between X and Z stablizers, due to Y errors



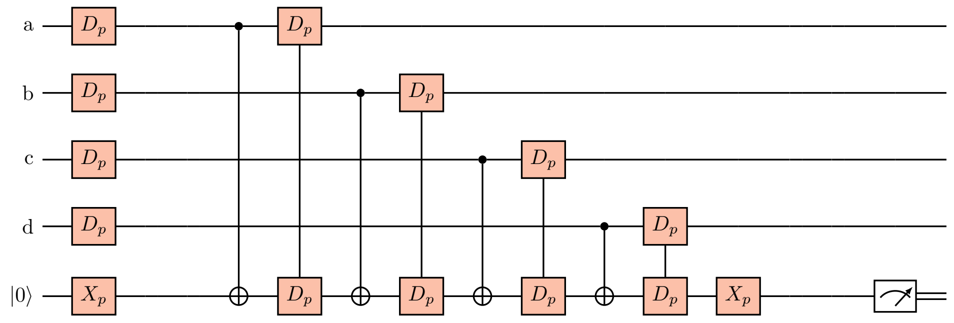


Improved accuracy
Slower

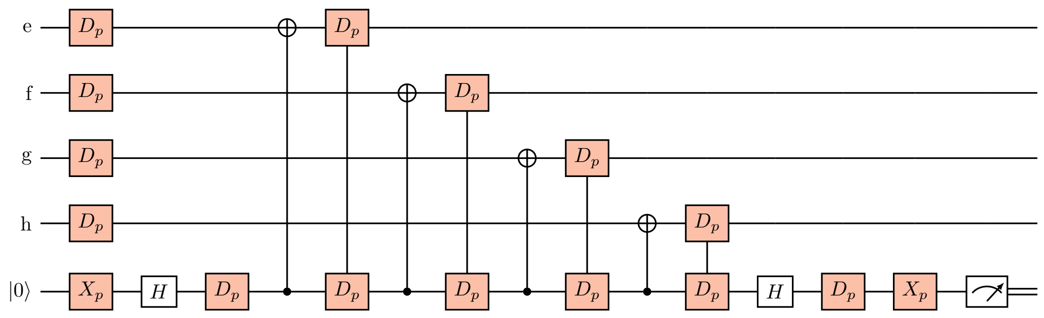
Stabilizer measurement circuits

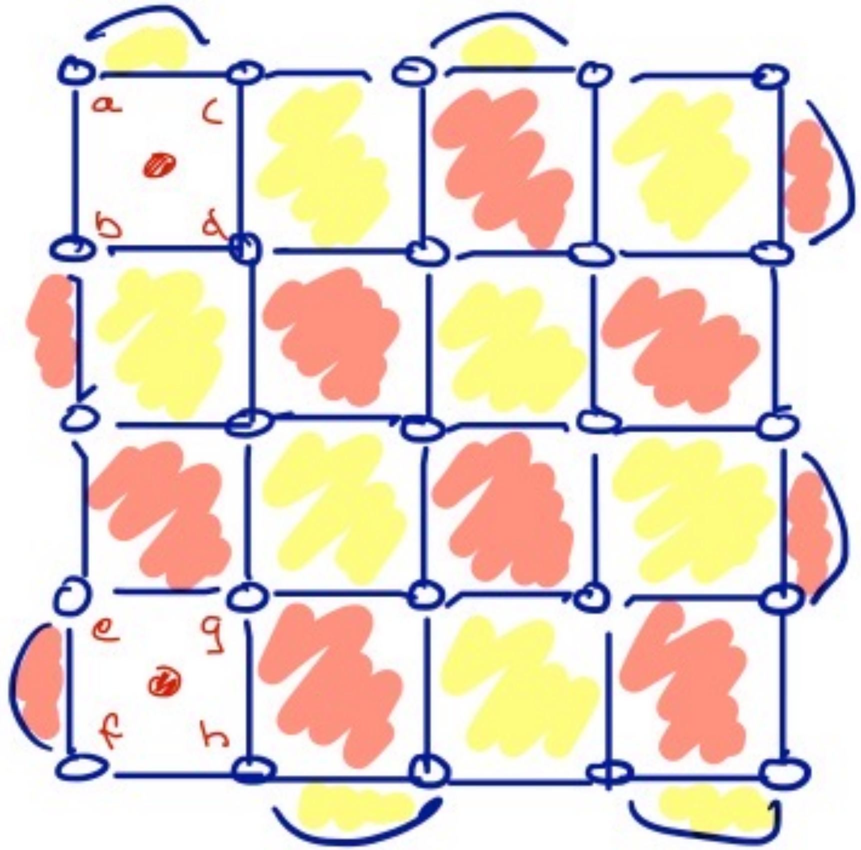
To measure stabilizers we use ancilla (measure) qubits

ZZZZ stabilizer



XXXX stabilizer





Simulating experiments with circuit-level noise

[Stim: a fast stabilizer circuit simulator](#)

Craig Gidney

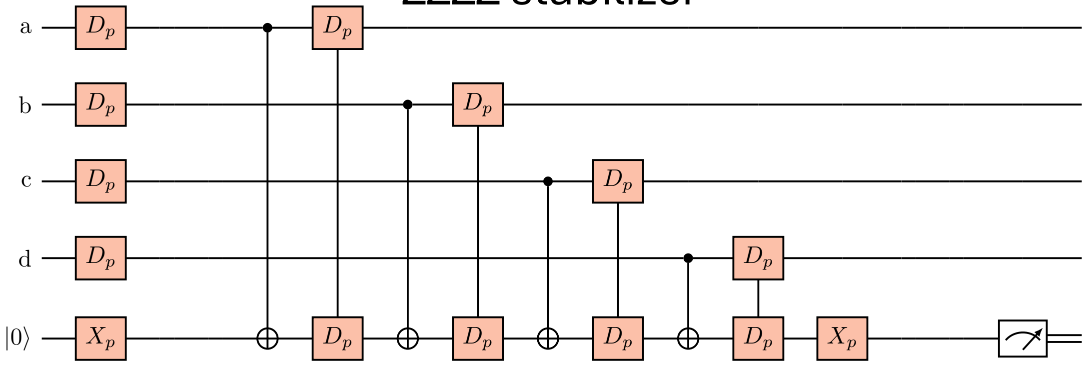
Google Inc., Santa Barbara, California 93117, USA

Published: 2021-07-06, volume 5, page 497
Eprint: arXiv:2103.02202v3
Doi: https://doi.org/10.22331/q-2021-07-06-497
Citation: Quantum 5, 497 (2021).

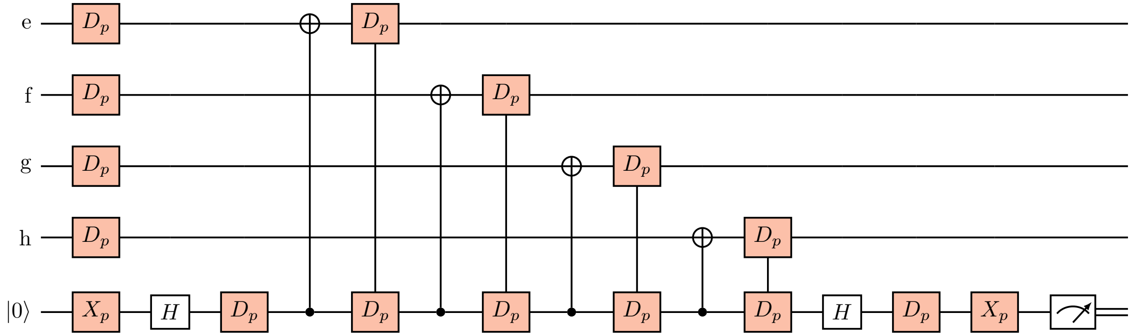
We use Stim to generate simulated "experiments"

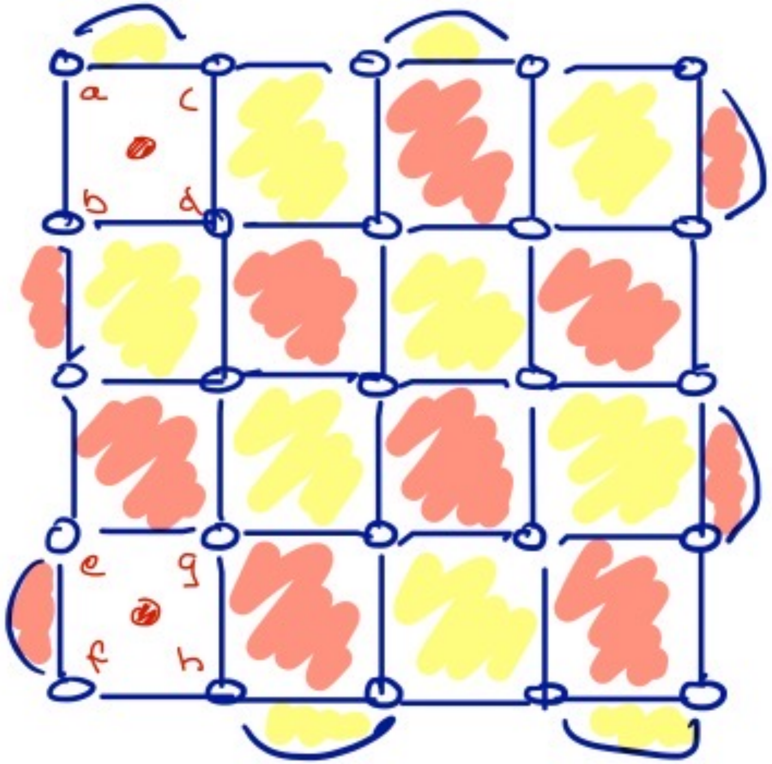
Standard circuit-level noise model: Depolarizing single qubit idle and two-qubit gate errors, and reset and readout errors (Excludes leakage, amplitude damping, crosstalk, …)

ZZZZ stabilizer



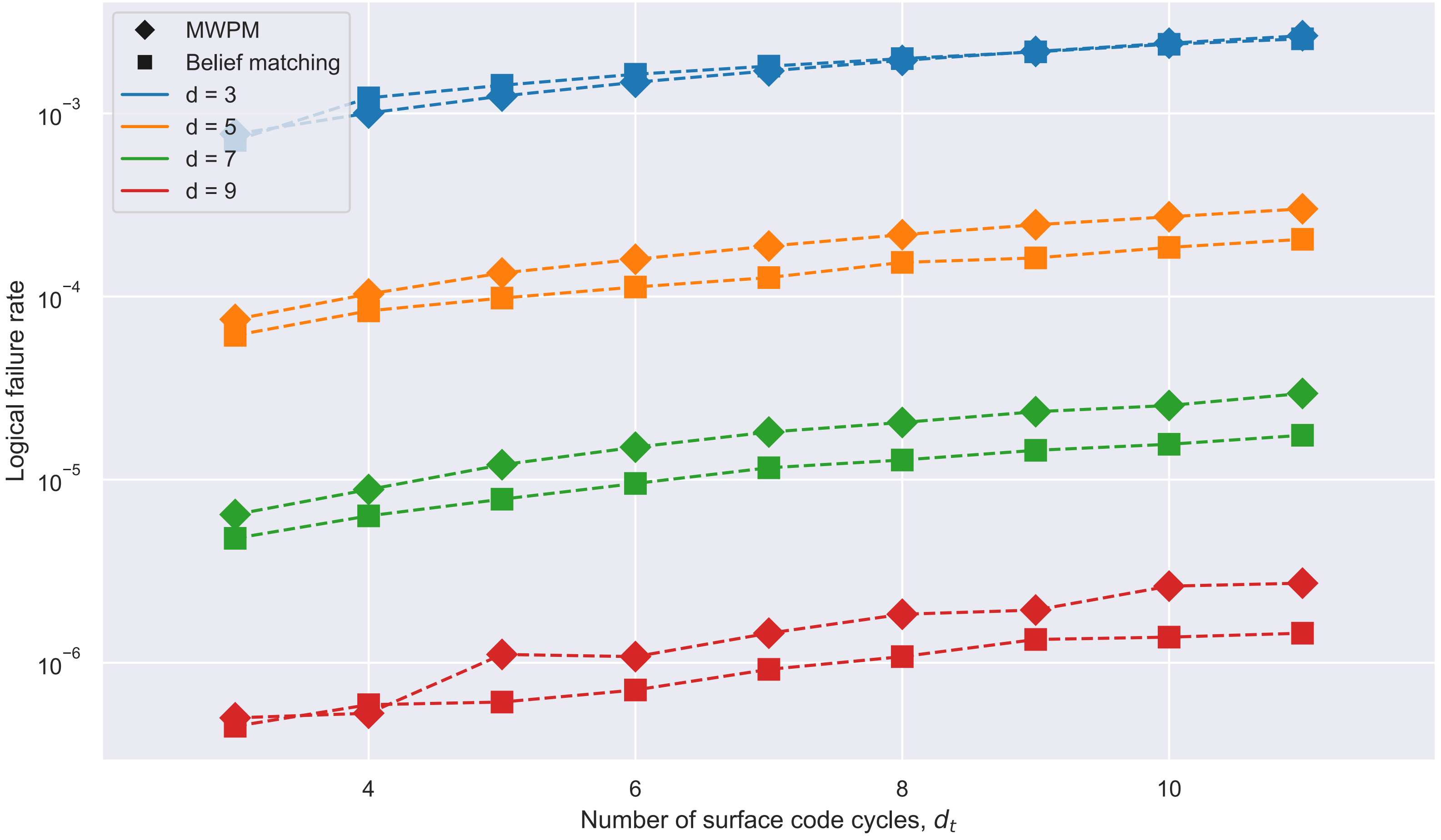
XXXX stabilizer





Logical failure rates for matching

At overall error rate p=1.0 x 10^{-3}



Very high accuracies!

Requires detailed knowledge of error channel

Q. Can we use a data driven machine learning approach to decoding?

Motivated by e.g. natural language processing where large deep learning models made strutured (gramatical) approaches obsolete.

- Model free! (non-Pauli error channel)
- Potentially fast and scalable

Extreme requirements on:

- Accuracy – 0.999999 (or even higher)
- Inference time – μs

Previous work on deep learning based decoders

TABLE I. A comprehensive literature survey and the comparison of the machine learning based syndrome decoders.

Paper	Error correction code	d_{max}	Threshold	ML Technique	Noise model
A scalable and fast artificial neural network syndrome decoder for surface codes **[This Work]**	Surface code with boundaries, braiding and lattice surgery structures	1025	0.138	Supervised learning convolution neural network	Depolarizing, Inhomogeneous and Biased noise models
Scalable Neural Decoder for Topological Surface Codes [22]	Toric code	255	0.162(5)	Supervised learning dense neural network	Depolarizing noise
Reinforcement learning for optimal error correction of toric codes [45]	Toric code	9	0.103	Reinforcement learning, Deep convolutional net	Bit-flip
Neural Network Decoders for Large-Distance 2D Toric Codes [35]	Toric code	64	0.095	Supervised learning, Renormalization group based neural network	Bit flip
Neural ensemble decoding for topological quantum error-correcting codes [46]	Surface code	11	Not reported	Supervised learning, Neural network ensemble learning	Depolarizing noise
Deep Q-learning decoder for depolarizing noise on the toric code [47]	Toric code	9	0.165	Deep reinforcement learning	Depolarizing noise
Comparing neural network based decoders for the surface code [48]	Rotated surface code	9	0.146 (depol.), 0.0032(circ.)	Supervised learning, Feed forward neural networks, Recurrent neural nets with LSTMs	Depolarizing and circuit noise
Symmetries for a High Level Neural Decoder on the Toric Code [49]	Toric code	7	Not reported	Supervised learning, Feed forward neural net	Depolarizing noise
Quantum error correction for the toric code using deep reinforcement learning [50]	Toric code	7	Not reported	Deep reinforcement learning	Bit-flip
Decoding surface code with a distributed neural network based decoder [51]	Rotated surface code	9	Not reported	Supervised learning, Neural network	Depolarizing noise
Reinforcement Learning Decoders for Fault-Tolerant Quantum Computation [52]	Rotated surface code	5	Not reported	Reinforcement learning, Convolutional neural network	Bit-flip, Depolarizing, Phenomenological noise
Deep neural decoders for near term fault-tolerant experiments [24]	Rotated surface code	5	Not reported	Supervised learning, Deep neural networks, Single layer neural networks	Circuit noise
Scalable Neural Network Decoders for Higher Dimensional Quantum Codes [53]	3D toric code, 4D toric code	12	0.175 (3D), 0.071 (4D)	Supervised learning, Convolutional neural network	Bit-flip, Phenomenological noise
Machine-learning-assisted correction of correlated qubit errors in a topological code [21]	Rotated surface code	3	Not reported	Supervised learning, Recurrent neural net with LSTMs	Depolarizing noise and Measurement errors
Decoding small surface codes with feedforward neural networks [17]	Rotated surface code	7	Not reported	Supervised learning, Feed forward neural network	Bit-flip, Depolarizing, Phenomenological and Circuit noise
Deep Neural Network Probabilistic Decoder for Stabilizer Codes [20]	Toric code	9	0.164	Neural net with 15-18 hidden layers	Depolarizing noise
Neural Decoder for Topological Codes [19]	Toric code	6	0.109	Restricted Boltzmann machine	Phase-flip errors

Neural Decoder for Topological Codes

Giacomo Torlai and Roger G. Melko
Phys. Rev. Lett. **119**, 030501 – Published 18 July 2017

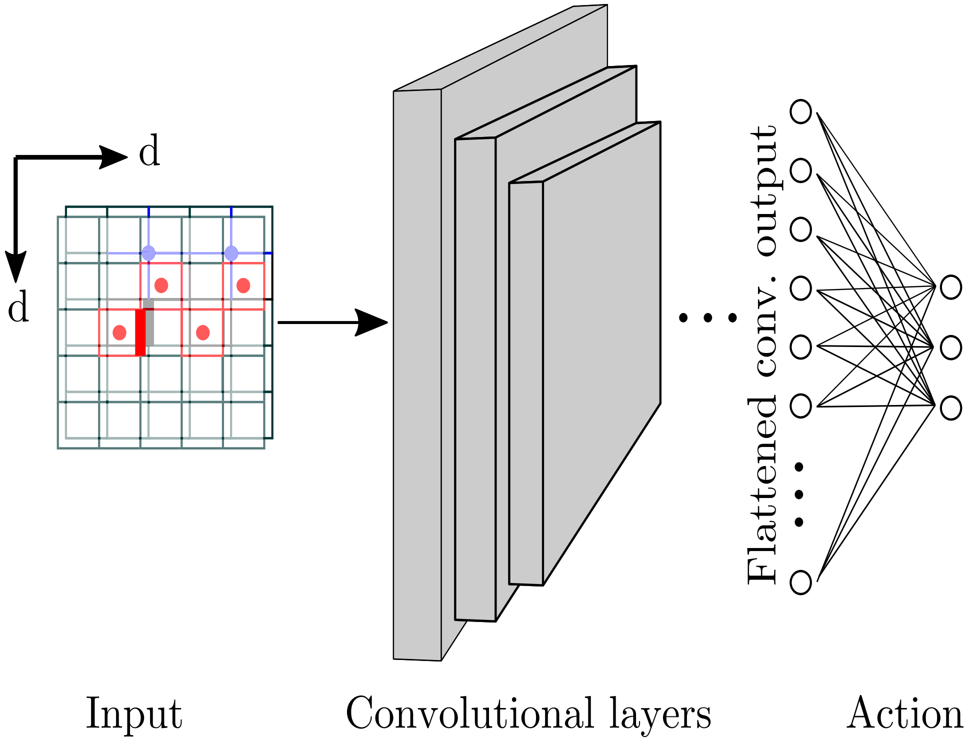
A scalable and fast artificial neural network syndrome decoder for surface codes

Spiro Gicev,[1,*] Lloyd C.L. Hollenberg,[1,†] and Muhammad Usman[1,2,‡]
[1] *Center for Quantum Computation and Communication Technology, School of Physics, University of Melbourne, Parkville, 3010, VIC, Australia.*
[2] *School of Computing and Information Systems, Melbourne School of Engineering, University of Melbourne, Parkville, 3010, VIC, Australia*

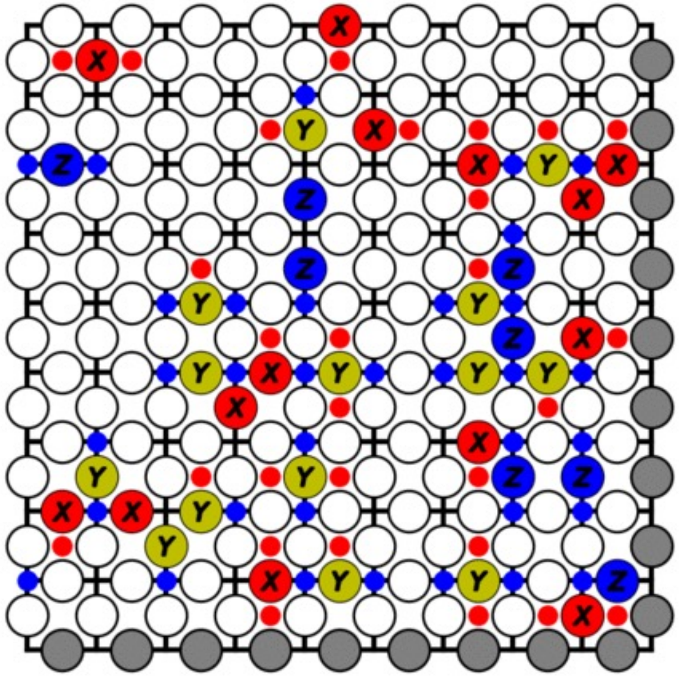
Mostly conceptual, with simplified error models

Our early attempt: Deep Reinforcement Learning

Convolutional network outputs action values for single qubit corrections



Step-by-step correction



Quantum error correction for the toric code using deep reinforcement learning

Philip Andreasson, Joel Johansson, Simon Liljestrand, and Mats Granath

Accepted in Quantum 2019-08-24, click title to verify

Reinforcement learning decoders for fault-tolerant quantum computation

Ryan Sweke[1], Markus S Kesselring[1], Evert P L van Nieuwenburg[2] and Jens Eisert[1,3]

Published 28 December 2020 • © 2020 The Author(s). Published by IOP Publishing Ltd
Machine Learning: Science and Technology, Volume 2, Number 2

PHYSICAL REVIEW RESEARCH 2, 023230 (2020)

Deep Q-learning decoder for depolarizing noise on the toric code

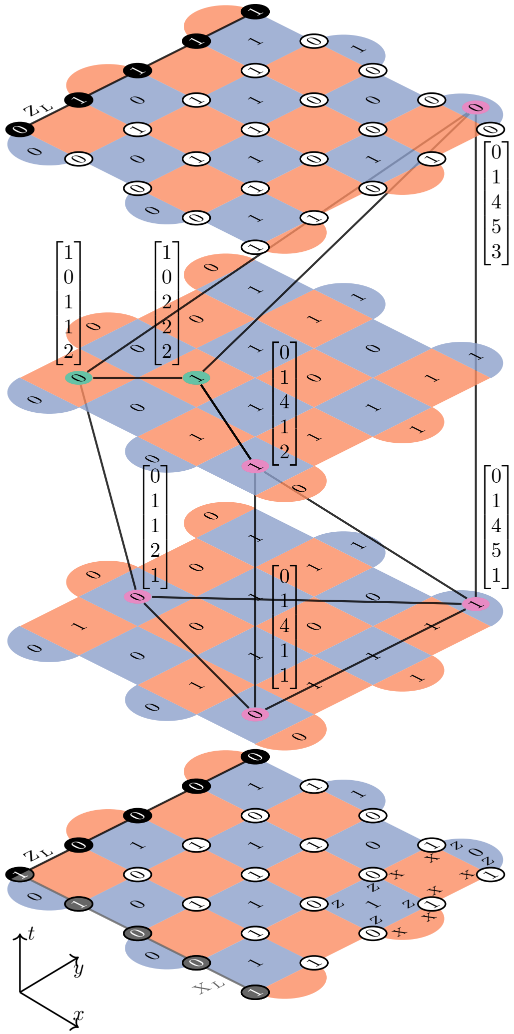
David Fitzek,[1,2,*] Mattias Eliasson,[3] Anton Frisk Kockum,[1] and Mats Granath[3,†]
[1]Wallenberg Centre for Quantum Technology, Department of Microtechnology and Nanoscience, Chalmers University of Technology, SE-41296 Gothenburg, Sweden
[2]Volvo Group Trucks Technology, 405 08 Gothenburg, Sweden
[3]Department of Physics, University of Gothenburg, SE-41296 Gothenburg, Sweden

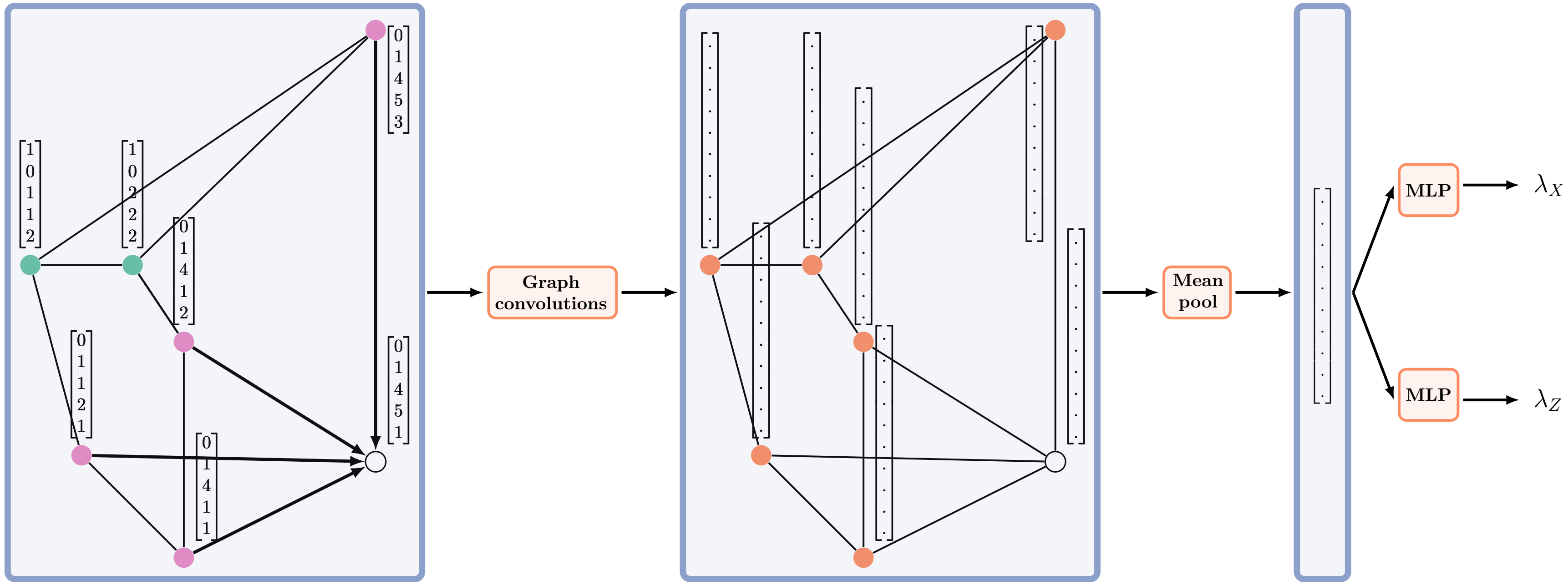
- Inefficient
- Difficult to scale
- Misses the point

Recent work: Graph neural network decoder

Tailored to experimental input

Fast inference? => in time error correction
High Accuracy => high logical fidelity





Data-driven decoding of quantum error correcting codes using graph neural networks

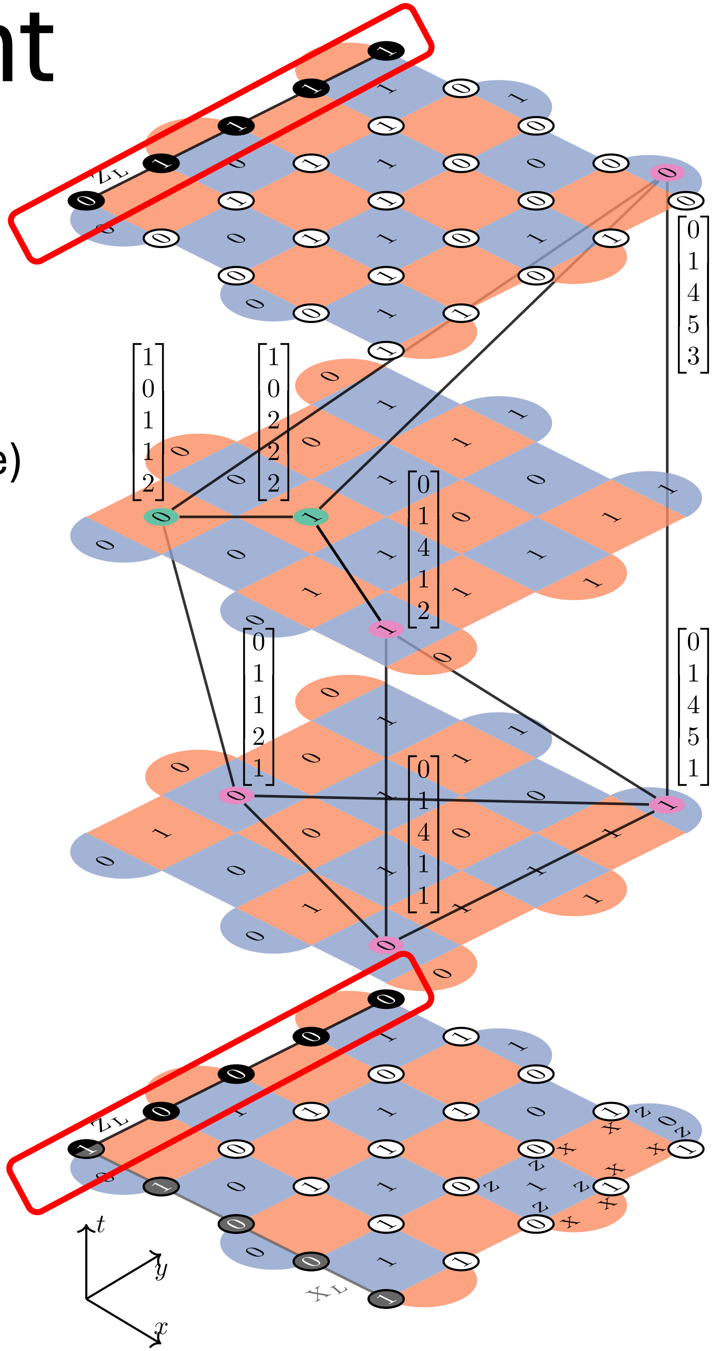
Moritz Lange,[1] Pontus Havström,[1] Basudha Srivastava,[1] Valdemar Bergentall,[1] + Isak Bengtsson
Karl Hammar,[1] Olivia Heuts,[1] Evert van Nieuwenburg,[2,*] and Mats Granath[1,†]
[1] *Department of Physics, University of Gothenburg, SE-41296 Gothenburg, Sweden*
[2] *Leiden Inst. of Advanced Computer Science, Leiden University, Leiden, Netherlands*

arXiv:2307.01241

Memory-Z experiment

1. Simple product state prepared
2. Stabilizers are measured over several rounds => changes=detector events
   Perfect Z-stabilizers in first and final rounds
3. Final individual qubit measurement
4. Measured logical coset of error (given by parity change on designated edge)
   compared to decoder prediction

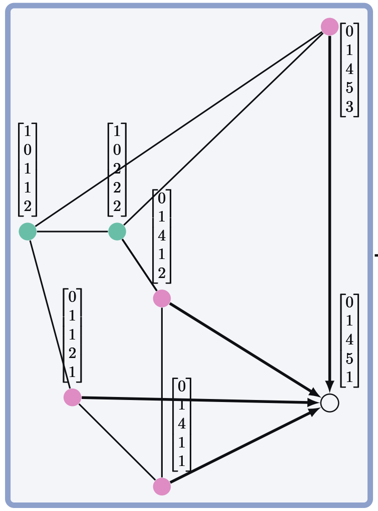
Gives logical fidelity of the quantum memory



GNN decoder data

- **Detector events = graph nodes**
  space-time location and type as node feature vector
- **Edges** ~ inverse euclidean (or manhattan) distance
- **Label**: **binary class, logical bit-flip or not**
  (or logical phase-flip or not)

Pruning of edges based on edge weights

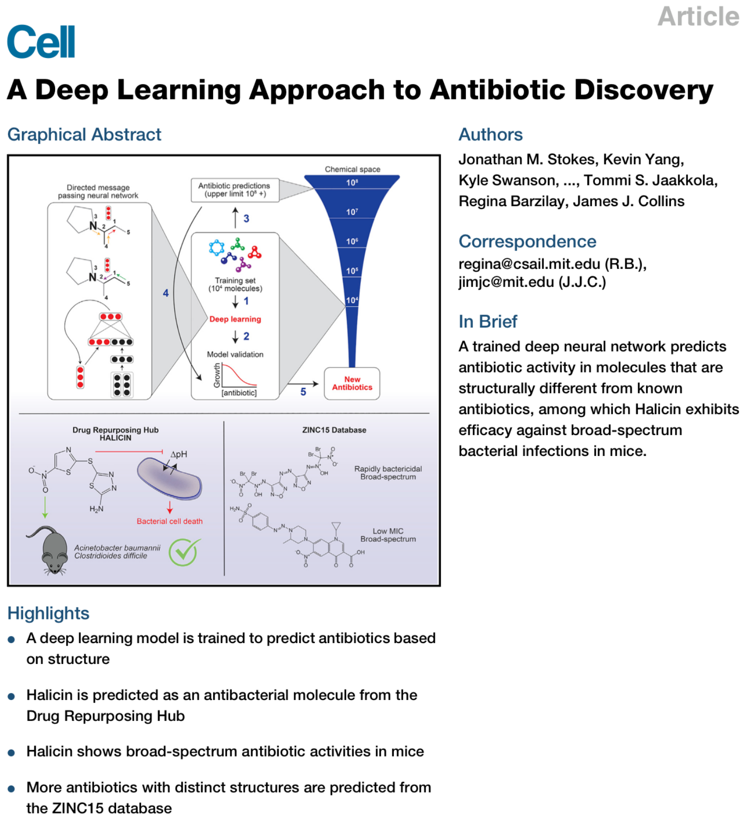


Graph neural networks (GNN)

- **Neural networks suited for graph structured data**

Examples

Antibiotic discovery
graph regression



Article

Cell

A Deep Learning Approach to Antibiotic Discovery

Graphical Abstract

Authors

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, ..., Tommi S. Jaakkola, Regina Barzilay, James J. Collins

Correspondence

regina@csail.mit.edu (R.B.), jimjc@mit.edu (J.J.C.)

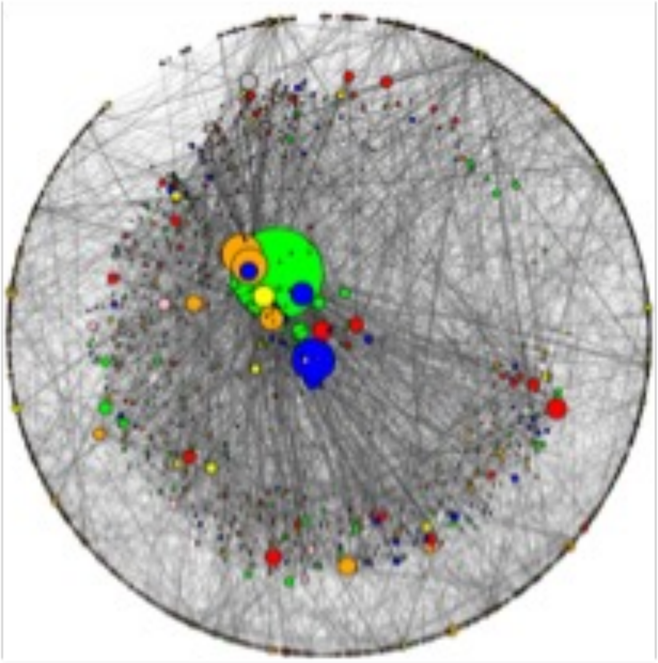
In Brief

A trained deep neural network predicts antibiotic activity in molecules that are structurally different from known antibiotics, among which Halicin exhibits efficacy against broad-spectrum bacterial infections in mice.

Highlights

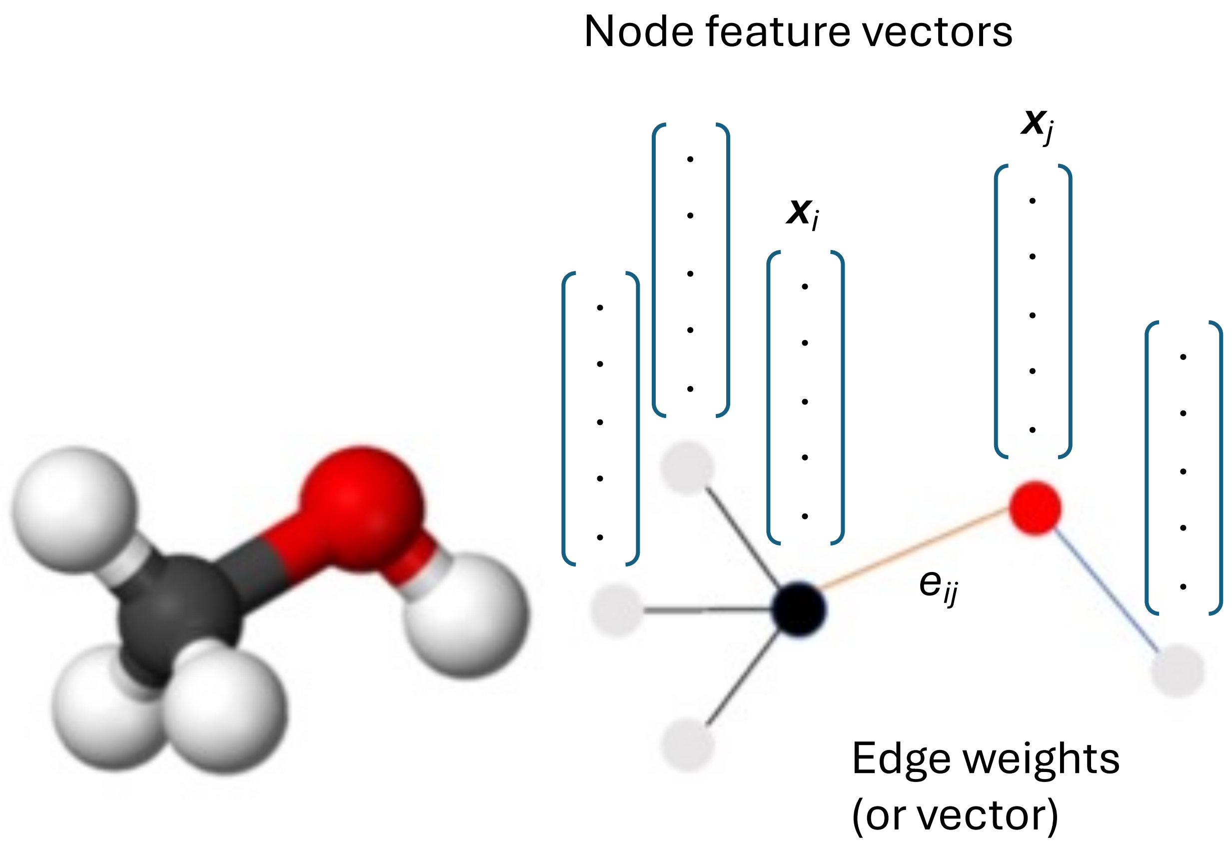
- A deep learning model is trained to predict antibiotics based on structure
- Halicin is predicted as an antibacterial molecule from the Drug Repurposing Hub
- Halicin shows broad-spectrum antibiotic activities in mice
- More antibiotics with distinct structures are predicted from the ZINC15 database

Cora dataset, citation network
Node classification



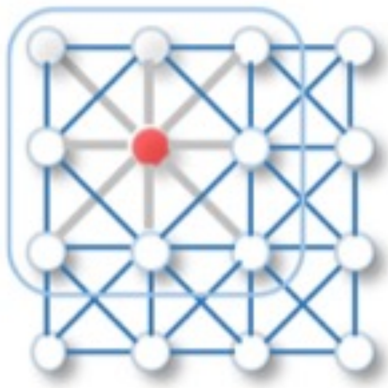
McCallum et al. 2000

Data object: Decorated graph



Graph convolutional layers

Grid: Standard convolutional filter of fixed size neighborhood



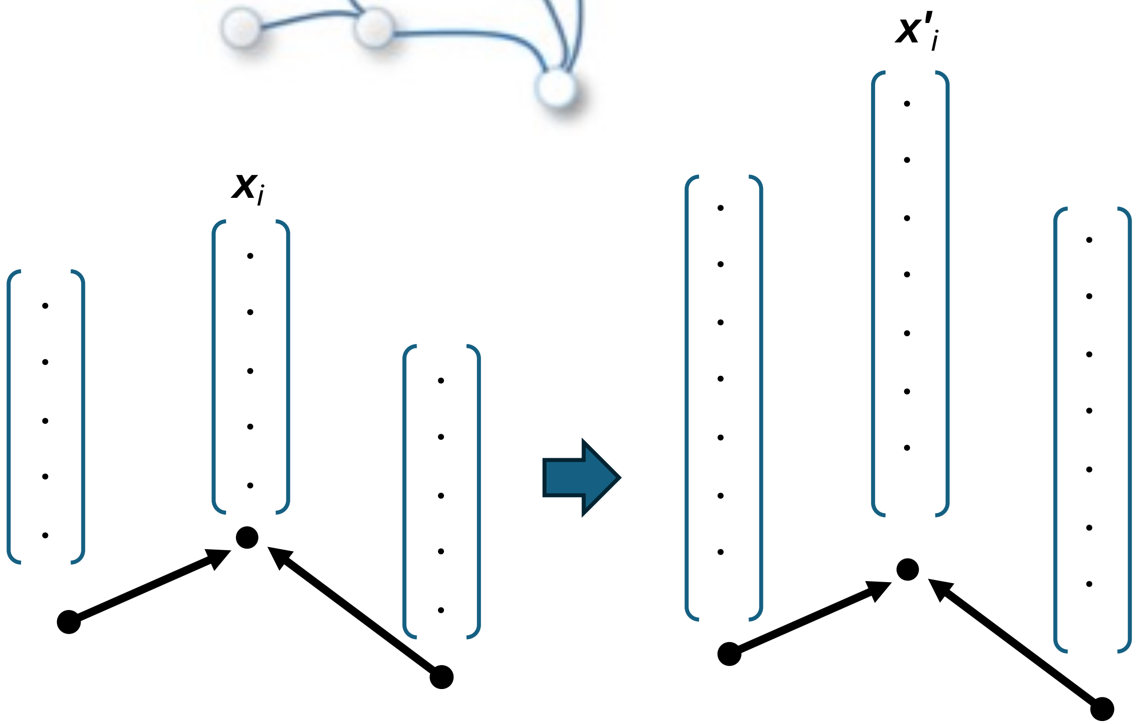
Graph: Convolutional filter adapted to varying neighborhood

A comprehensive survey on graph neural networks
Wu et al. 2019

Simple graph convolution:

$$\mathbf{x}'_i = \sigma(W_1\mathbf{x}_i + W_2 \sum_j e_{ij}\mathbf{x}_j)$$

W_1 and W_2: n' x n trainable weight matrices

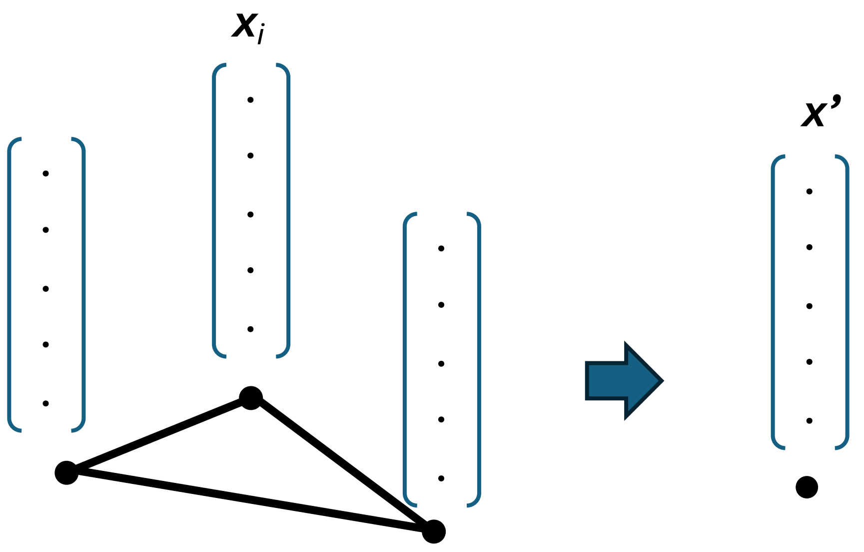


Semi-supervised classification with graph convolutional networks
Kipf and Wellling, 2016

Graph pooling

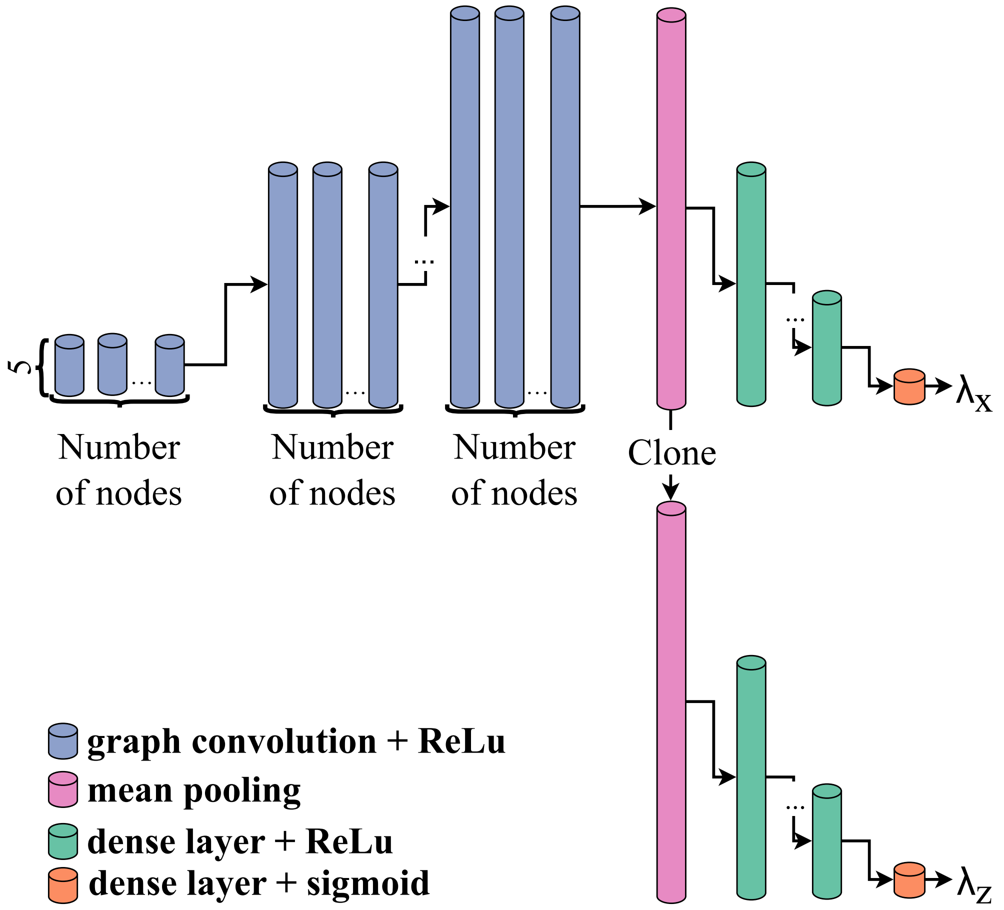
For graph classification output should be independent of number of nodes

$$\mathbf{x}' = \frac{1}{\#nodes} \sum_i \mathbf{x}_i$$



Graph embedding +
Standard dense network classifier

GNN decoder network architecture



Layer	d_{in}	d_{out}
GraphConv_1	5	32
GraphConv_2	32	128
GraphConv_3	128	256
GraphConv_4	256	512
GraphConv_5	512	512
GraphConv_6	512	256
GraphConv_7	256	256
Dense_1 X	256	128
Dense_2 X	128	181
Dense_3 X	128	32
Dense_4 X	32	1
Dense_1 Z	256	128
Dense_2 Z	128	181
Dense_3 Z	128	32
Dense_4 Z	32	1

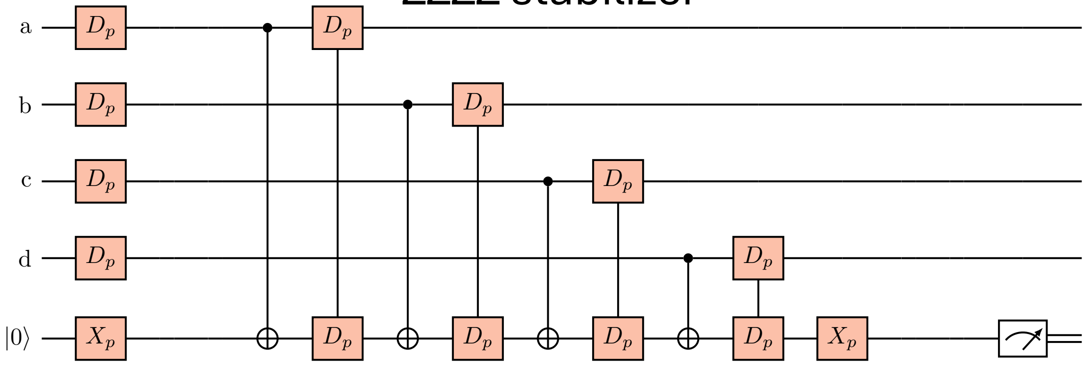
PyG *(PyTorch Geometric)* is a library built upon PyTorch to easily write and train Graph Neural Networks (GNNs) for a wide range of applications related to structured data.

GNN on circuit-level noise

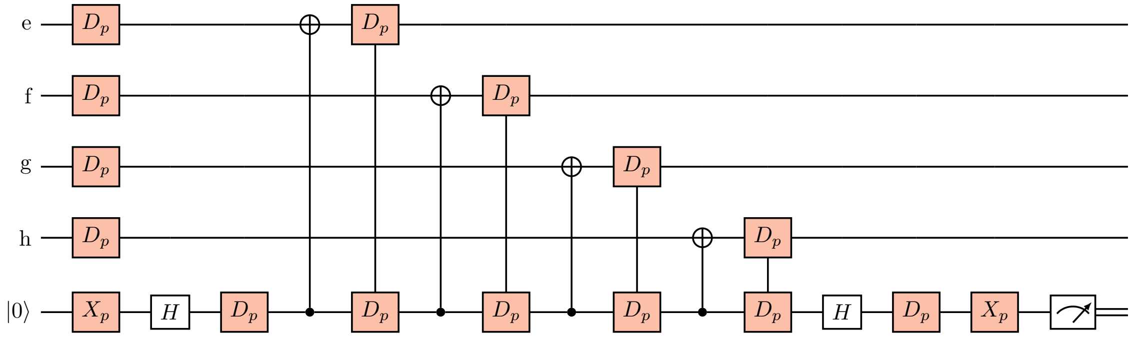
Lacking sufficient experimental data we use simulated "experiments"

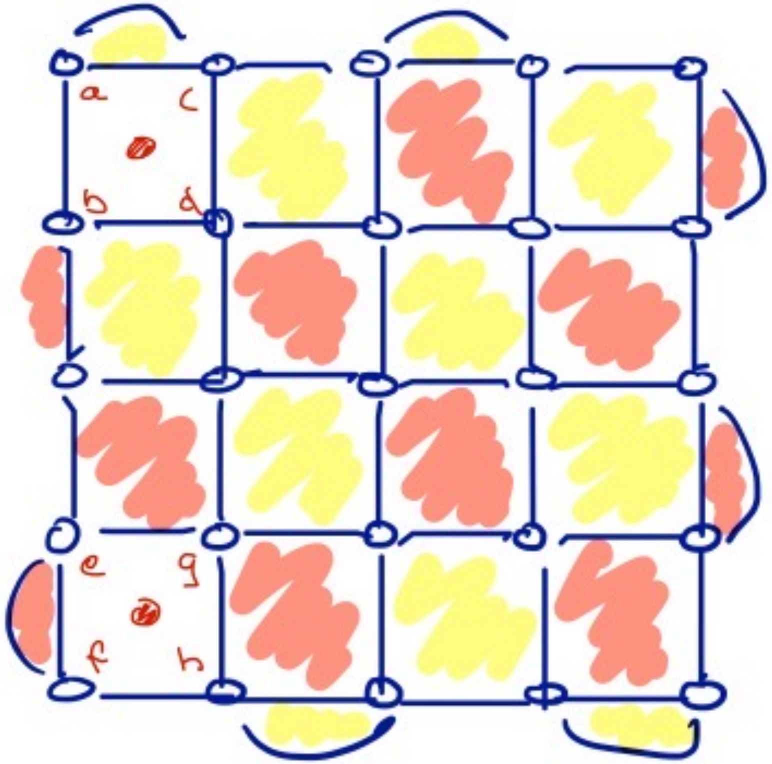
Depolarizing single qubit idle and two-qubit gate errors, and reset and readout errors

ZZZZ stabilizer



XXXX stabilizer

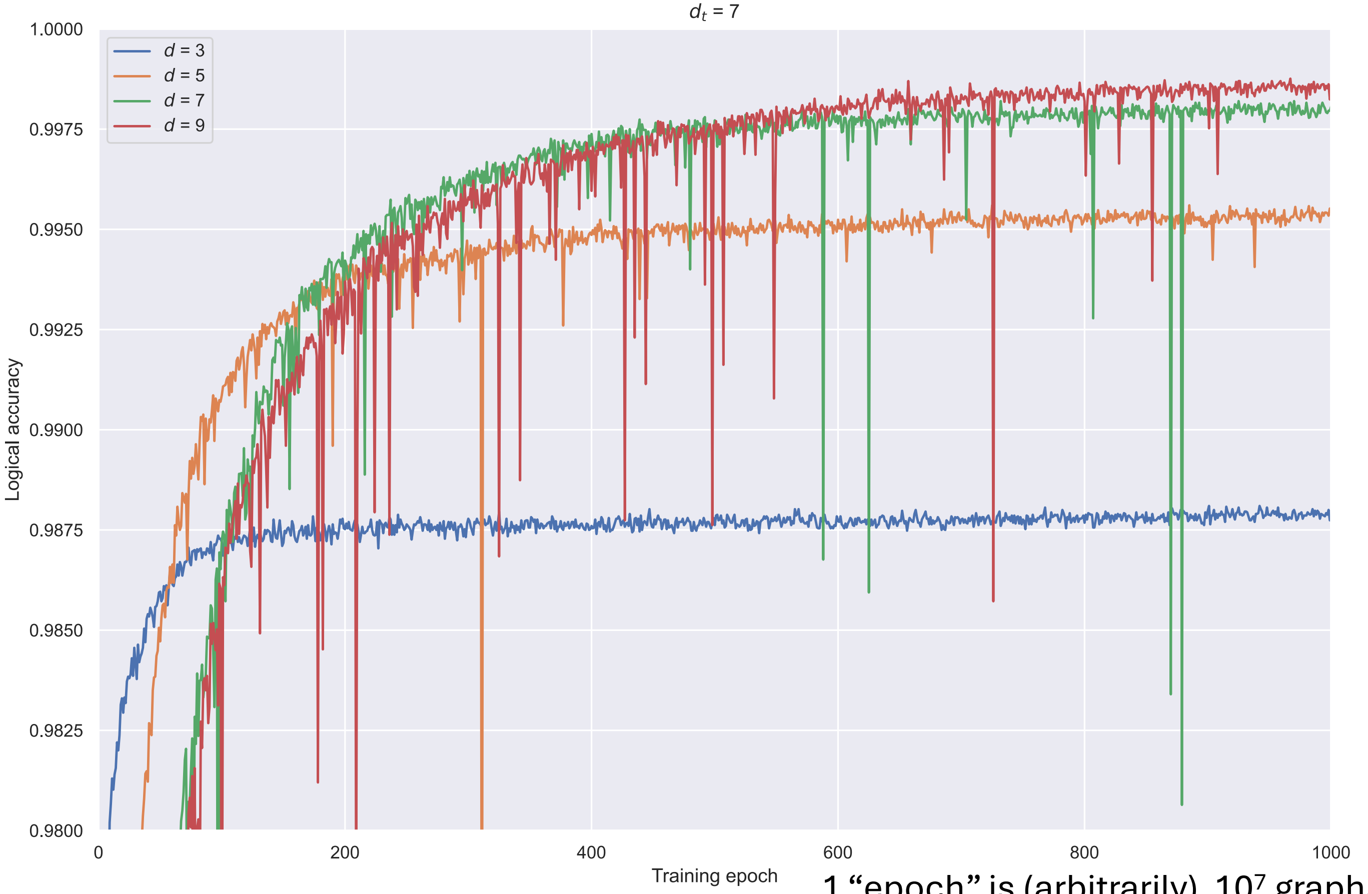




Training

- One network for each code distance *d* and number of cycles d_t
- **Range of training error rates** p=1.0-5.0 x 10^{-3}
- Data generated in **large batches** of 10,000-25,000 graphs (as much as can fit on the GPU memory)
- **No reuse of data** (no risk to overfit)
- Up to one week training on one Nvidia A100
- **Up to 10^{10} datapoints**

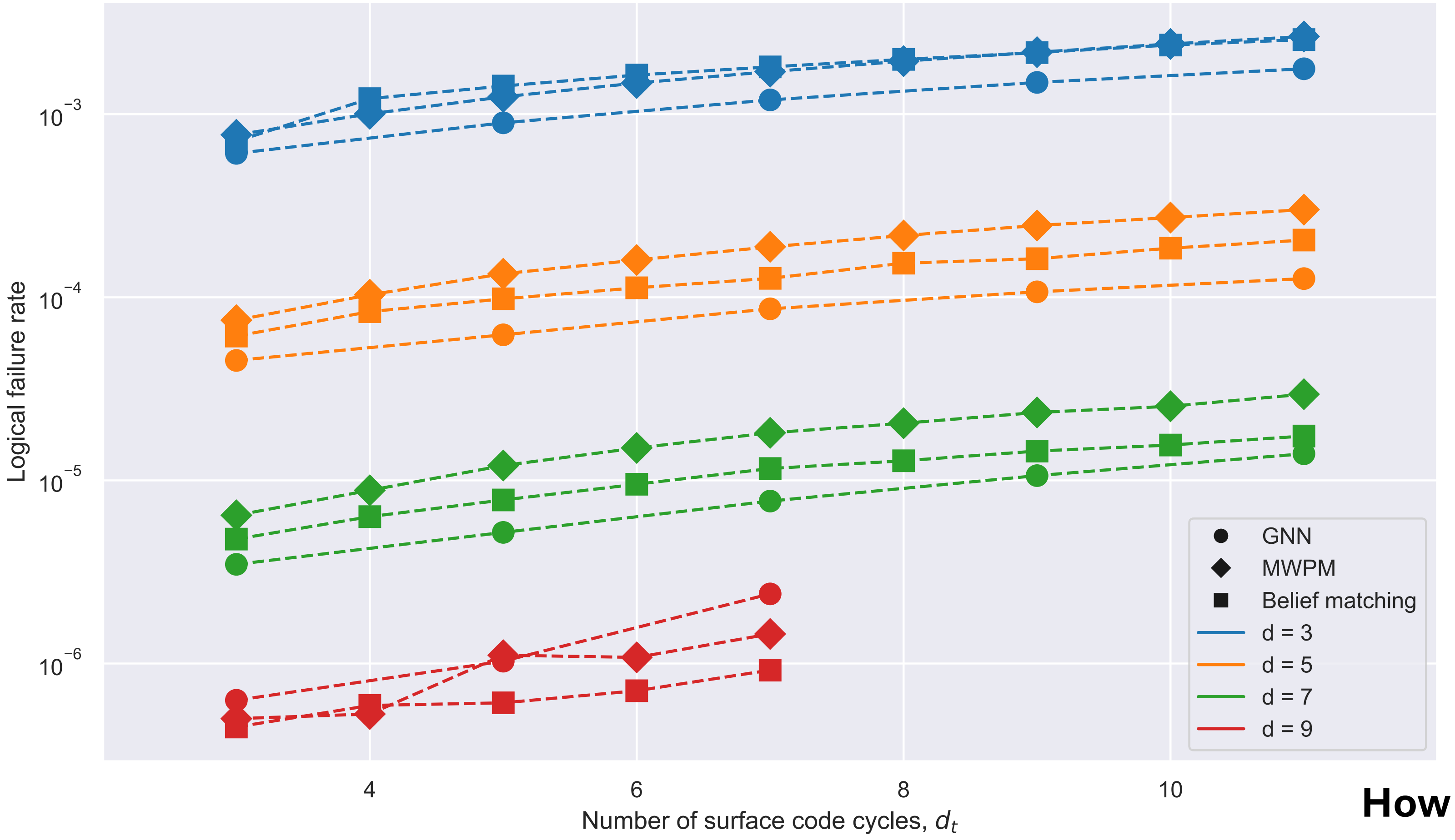
Examples of a few training curves



1 "epoch" is (arbitrarily) 10^7 graphs

Test

Benchmarked against matching (MWPM) and belief-matching at $p=1.0 \times 10^{-3}$

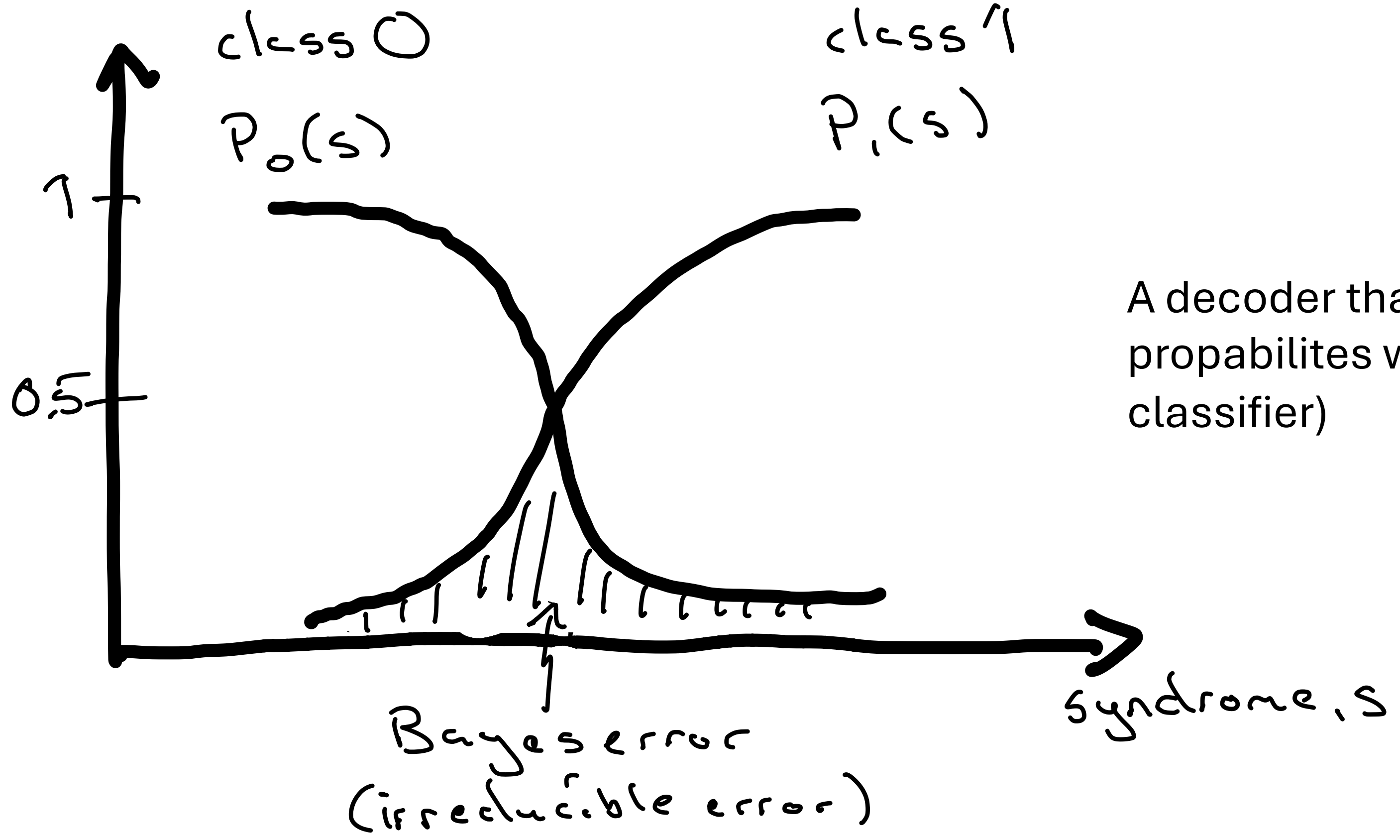


How close to optimal is it?

- GNN probably close to optimal (maximum-likelihood) decoder for small d
- **Matching decoders know the error model, the GNN decoder does not**
- Scaling to larger d is challenging, work in progress using larger networks

GNN as a maximum-likelihood decoder

work in progress

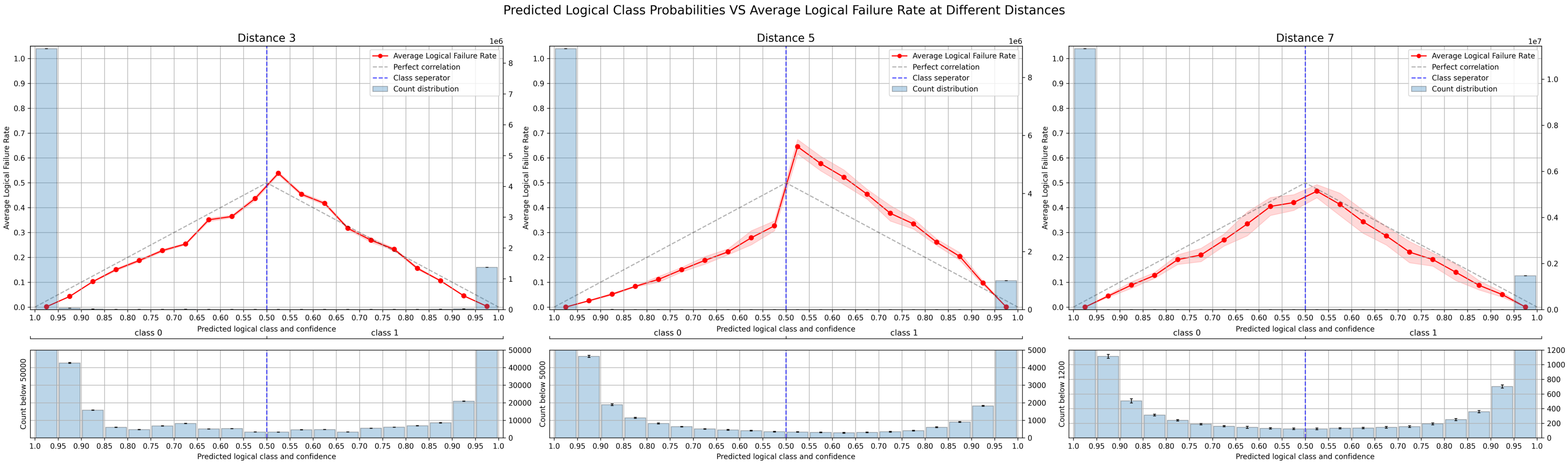


A decoder that outputs correct class propabilites will be optimal (Bayes classifier)

Test optimality

preliminary results

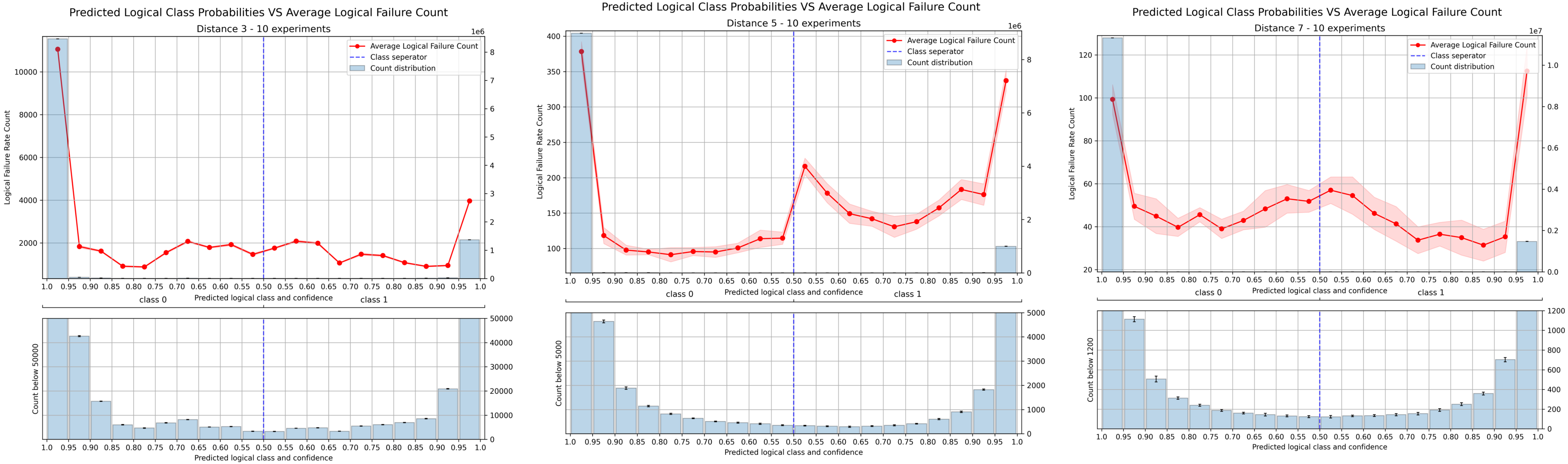
1. Bin GNN data according to predicted class probability
2. Compare to actual failure rates



GNN estimates class probabilities accurately. Indication that it’s close to optimal.

Most likely failures

Failure count versus decoder confidence



Interestingly, syndromes with high and low decoder confidence all contribute significantly to logical errors

Potential use-case: GNN as soft-output decoder for concatenated codes

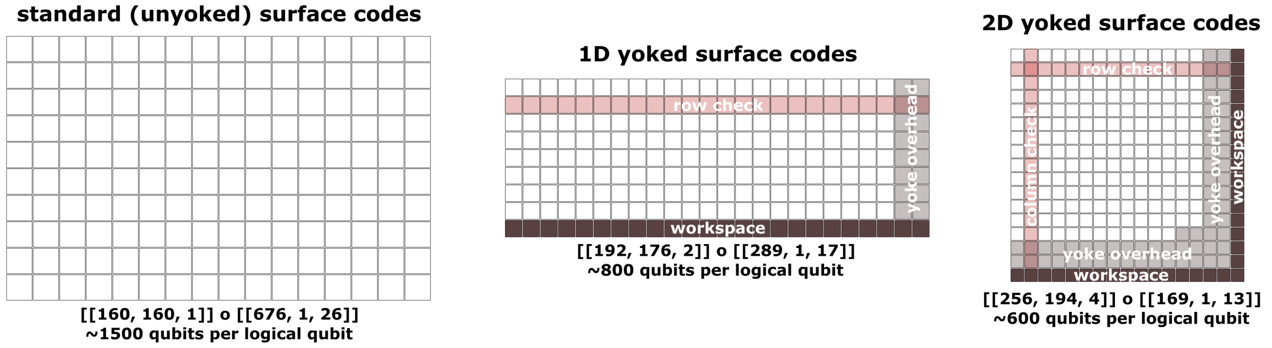
Of interest to output not only most likely class, but also the probability of failure

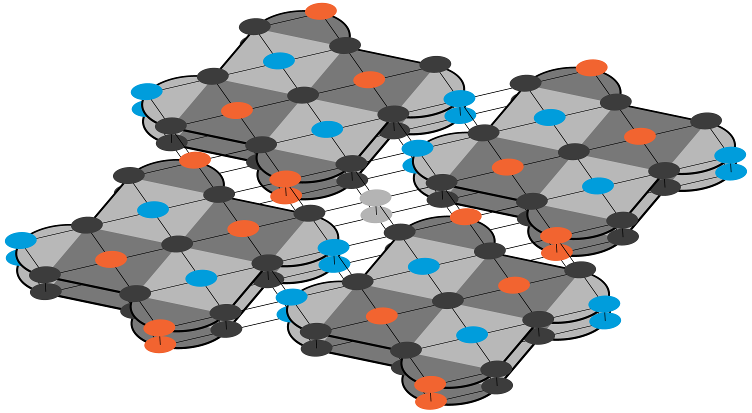
Yoked surface codes

Craig Gidney[1], Michael Newman[1], Peter Brooks[2], and Cody Jones[1]

Hierarchical memories: Simulating quantum LDPC codes with local gates

Christopher A. Pattison[1], Anirudh Krishna[2,3], and John Preskill[1,4]





- Surface code concatenated with other low-density parity check (LDPC) code
- Outer code decoder (with matching or belief propagation) can use conditional inner code error propabilities

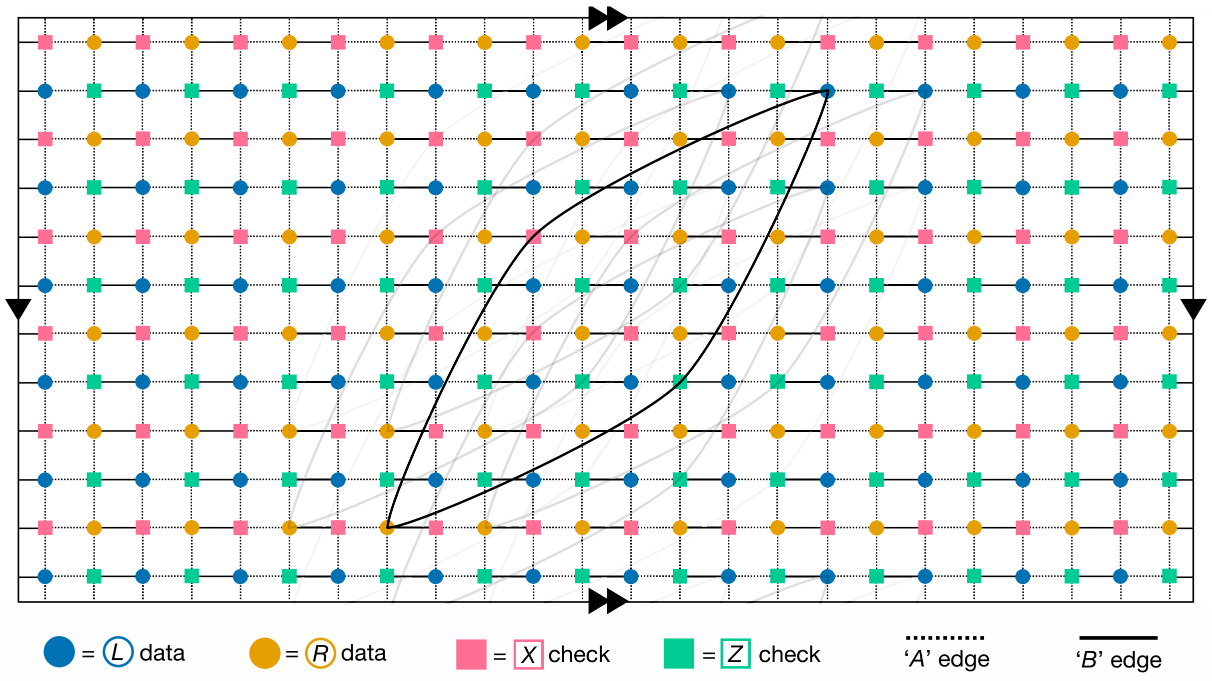
Decoding low-density parity check (LDPC) code

work in progress

High-threshold and low-overhead fault-tolerant quantum memory

778 | Nature | Vol 627 | 28 March 2024

[[144,12,12]]



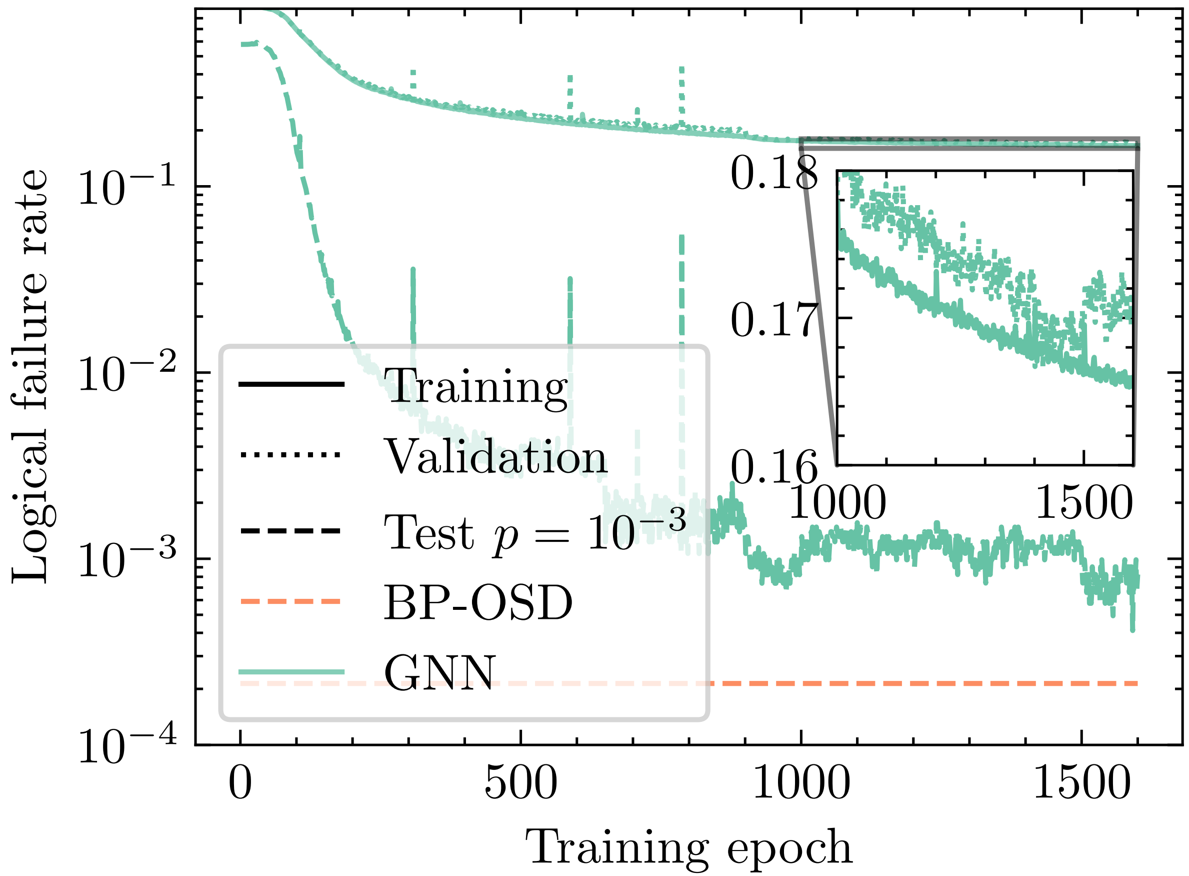
- Encodes 12 logical qubits in 144 physical qubits
- Non-local stabilizers in 2D
- Non-matchable (hyperedges)

GNN decoder for LDPC codes

preliminary results

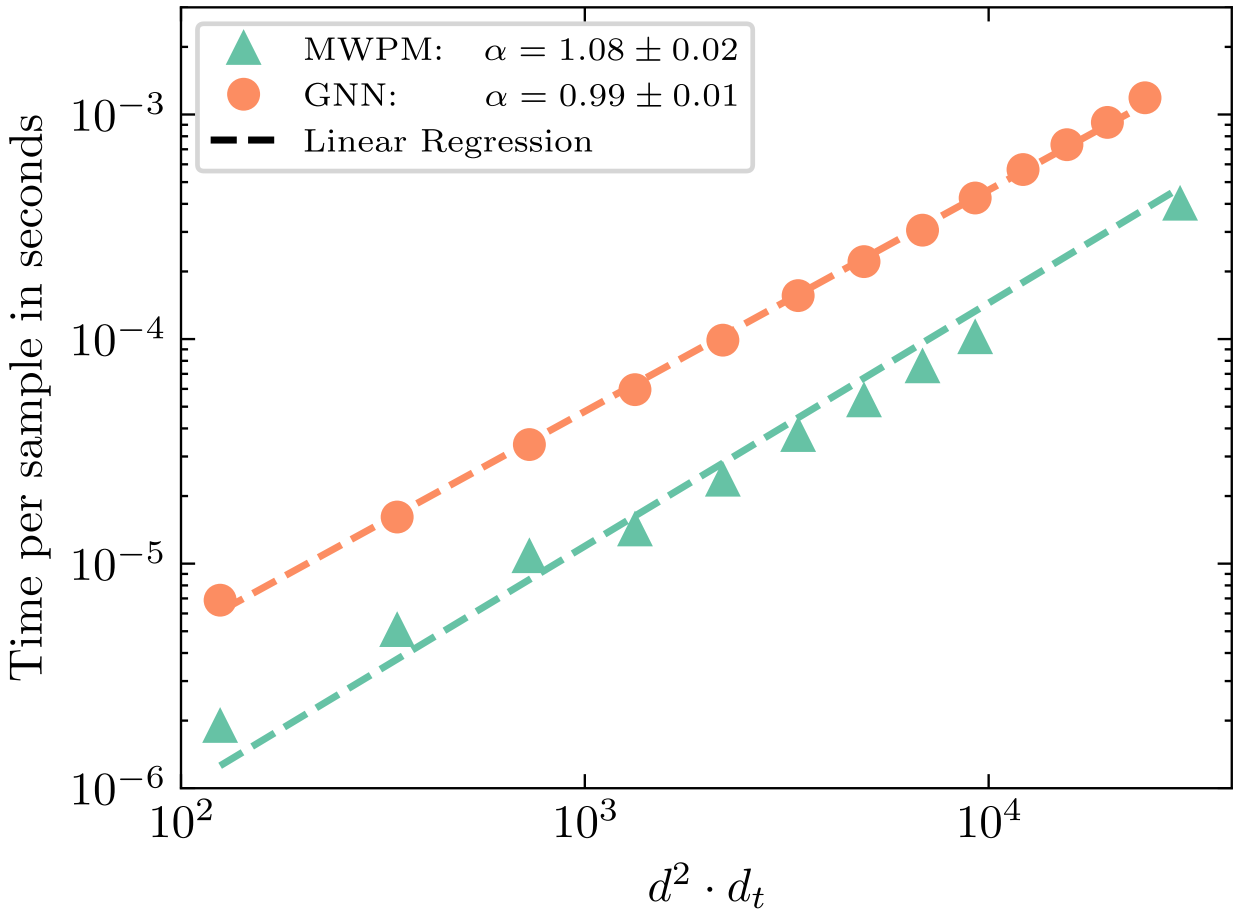
- Multiple logical qubits => multiple output layer nodes
- All graph nodes (stabilizers) are proximate due to long-range connectivity

[[72, 12, 6]] Smallest "IBM" code



Still a little way to go!

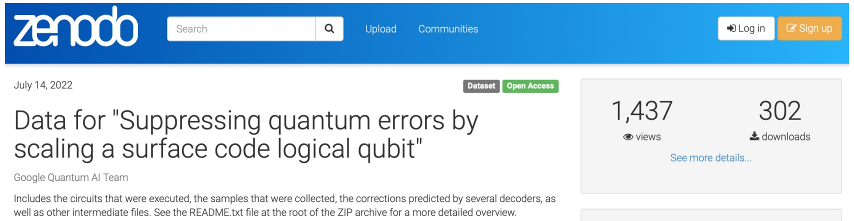
Inference time per syndrome



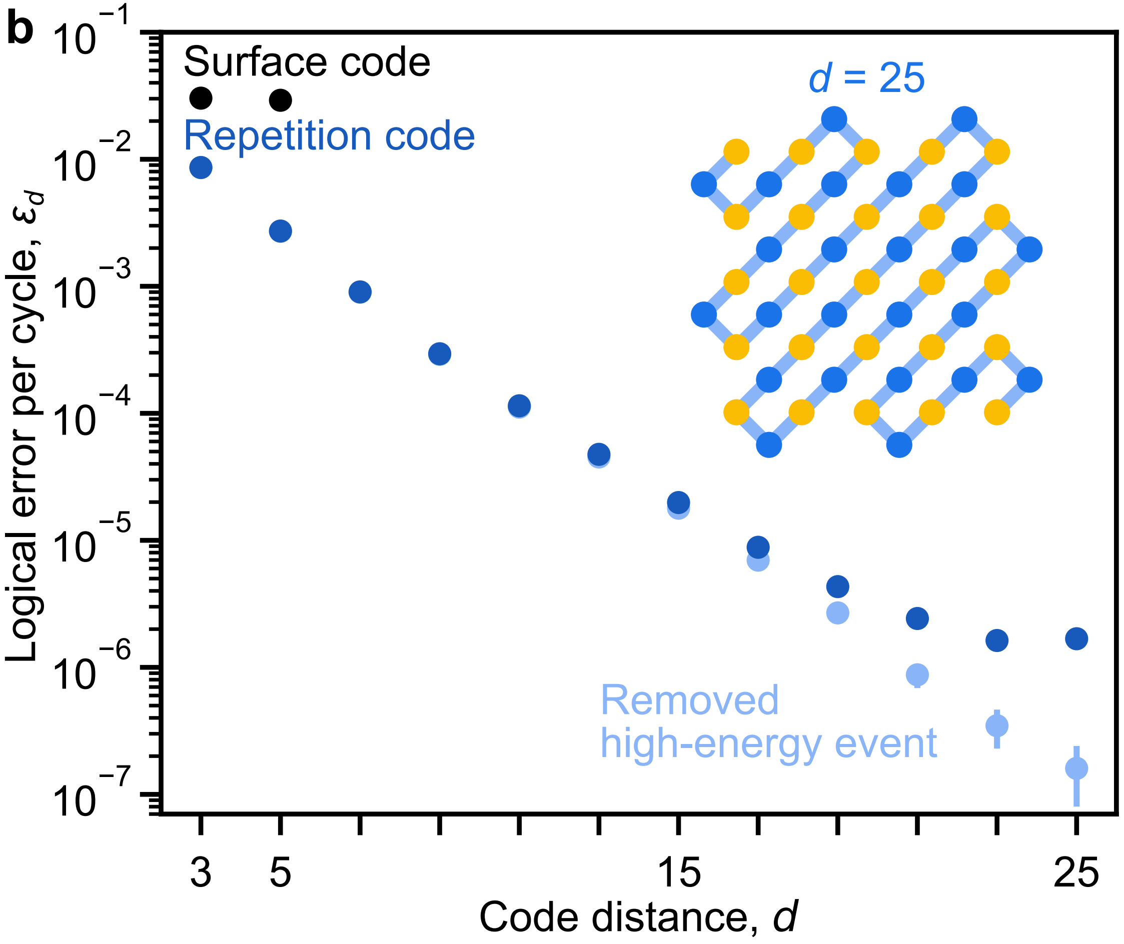
- Fixed size network
- Hardware and implementation dependent

- Decoding time scales linearly with code "volume"
- NB Both MWPM (Pymatching) and GNN are batched/parallelized

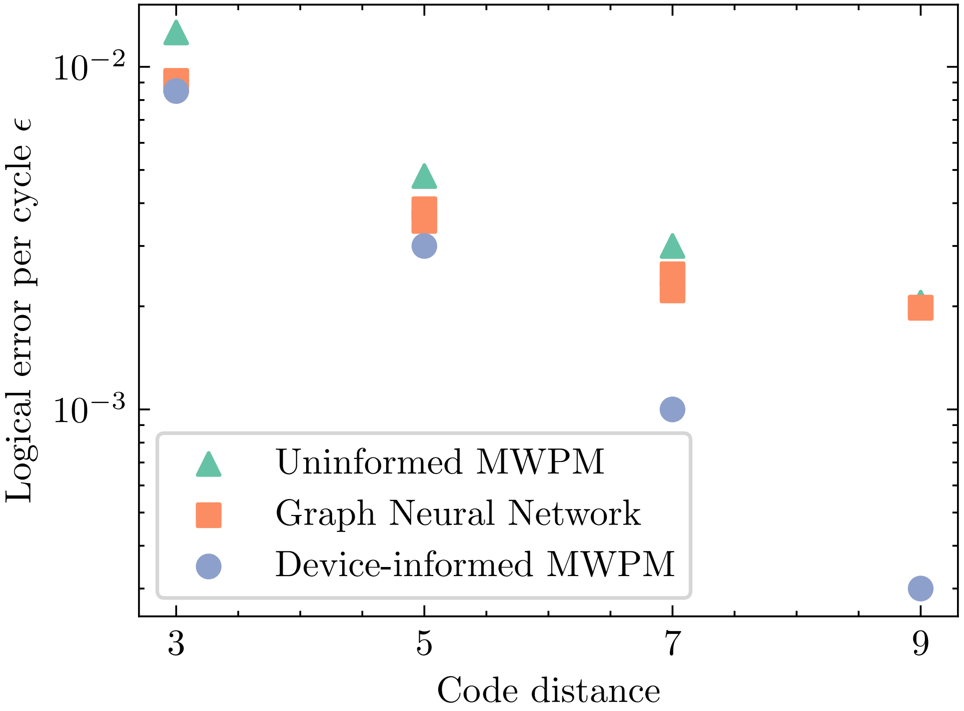
GNN on real experimental data



Repetition code, 25 stabilizer cycles
Datsets of around 10^7



GNN decoder on par with "informed" matching decoder for d=3 and 5.



Surface code dataset is too small, and/or error rates too high.

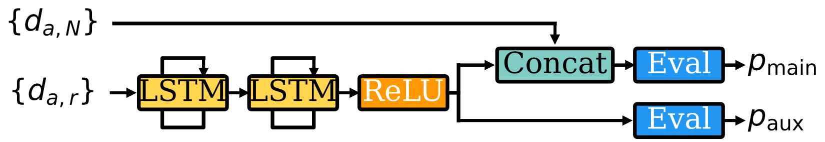
Other recent related work

Neural network decoder for near-term surface-code experiments

Boris M. Varbanov,[1,*] Marc Serra-Peralta,[1,2] David Byfield,[3] and Barbara M. Terhal[1,2]

[1] *QuTech, Delft University of Technology, P.O. Box 5046, 2600 GA Delft, The Netherlands*
[2] *Delft Institute of Applied Mathematics, Technische Universiteit Delft, 2628 CD Delft, The Netherlands*
[3] *Riverlane, Cambridge, CB2 3BZ, United Kingdom*

(Dated: October 24, 2023)

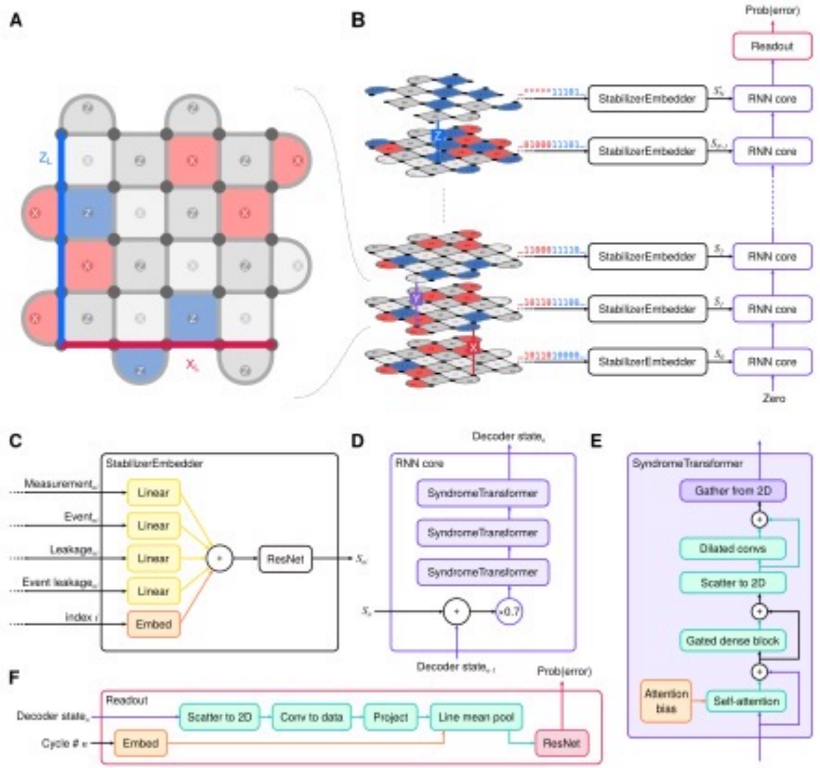


Learning to Decode the Surface Code with a Recurrent, Transformer-Based Neural Network

Johannes Bausch[1*†], Andrew W Senior[1*†], Francisco J H Heras[1†], Thomas Edlich[1†], Alex Davies[1†], Michael Newman[2†], Cody Jones[2], Kevin Satzinger[2], Murphy Yuezhen Niu[2], Sam Blackwell[1], George Holland[1], Dvir Kafri[2], Juan Atalaya[2], Craig Gidney[2], Demis Hassabis[1], Sergio Boixo[2], Hartmut Neven[2], Pushmeet Kohli[1]

[1] Google DeepMind & [2] Google Quantum AI

9 Oct 2023

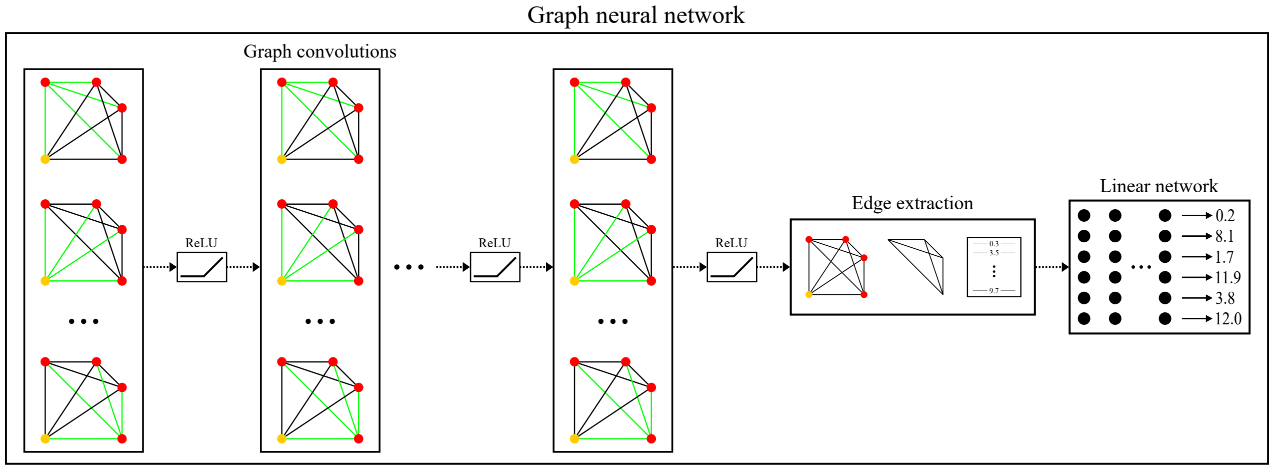


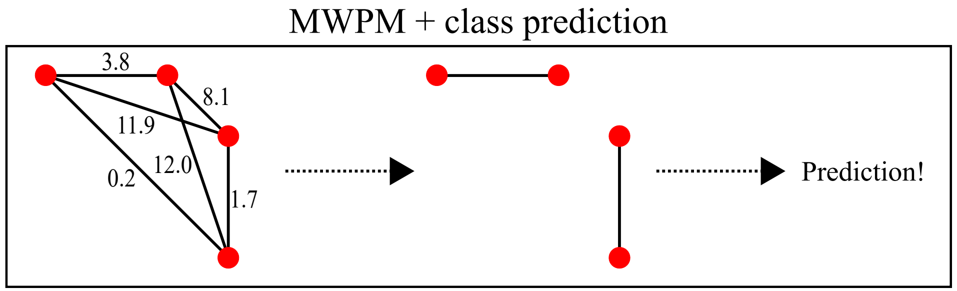
Work in progress: “Neural belief-matching”

- A pure neural network decoder is very data-hungry
- Can we combine a smaller graph network with a matching decoder?

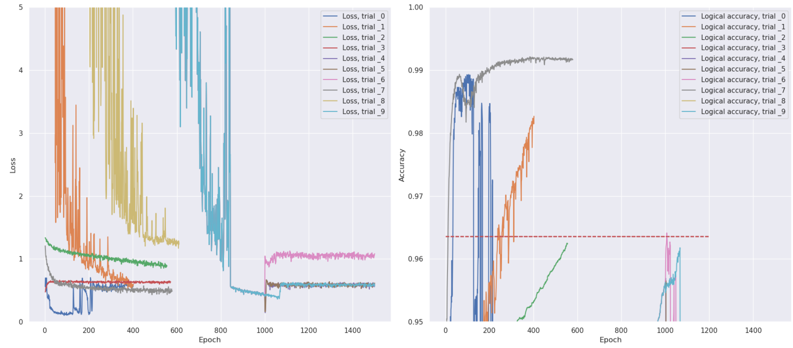
  Still **data-driven/model-free**

GNN provides edge-weights to a matching decoder





- Challenge: Loss is non-differentiable as matching gives discrete output



Approaching MWPM with error informed edge-weights

Conclusions

- Data-driven model-free approach to decoding using graph neural network
- Competitive to matching decoders for accuracy and speed
- Approaches maximum-likelihood decoder, soft-output decoder
- Challenging to scale to larger code-distances, more data/larger networks?
- Outlook: Generate training data (using IBM hardware)
- Outlook: Decode sliding window in time to scale (Google is doing this)
- Outlook: Move from GPU to FPGA for fast inference
- Outlook: Hybrid ”neural matching decoder”

Collaborators: **Basudha Srivastava** (GU and Quantinuum), **Moritz Lange**, **Isak Bengtsson**, **Blaž Pridgar**, Frida Fjelddahl, Pontus Havström, Valdemar Bergentall, Karl Hammar, Olivia Heuts, David Fitzek (Volvo), Ben Criger (Quantinuum), Anton Frisk Kockum (Chalmers), Evert van Nieuwenburg (Leiden)