Machine-learning-based decoders for quantum error correcting codes

Mats Granath
Department of Physics
University of Gothenburg

NAISS User Forum 2025
Chalmers

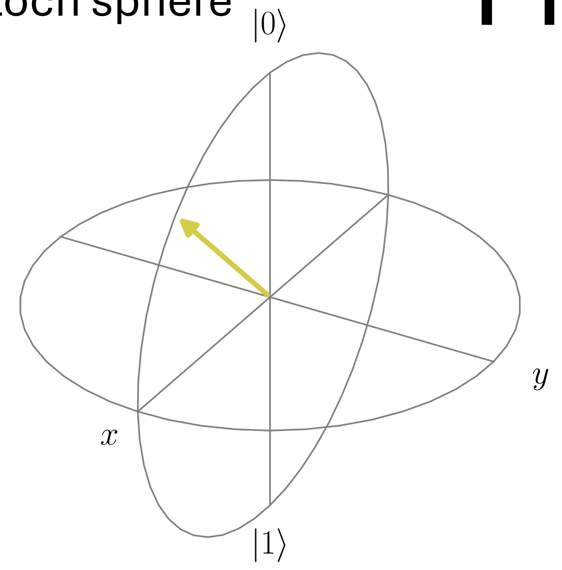
October 21, 2025

WACQT | Wallenberg Centre for Quantum Technology

UNIVERSITY OF GOTHENBURG

Prelude: Qubit – quantum bit

Bloch sphere



Superposition of 0 and 1

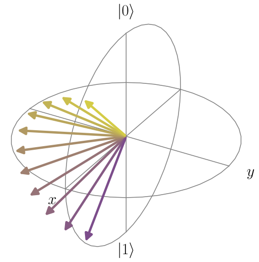
With normalized complex coefficients

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Basic operations (Pauli gates) on a qubit

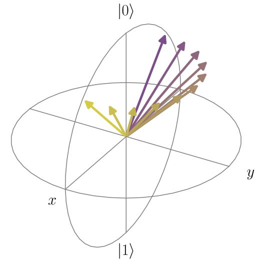
Bit flip

$$X|\Psi\rangle = \alpha|1\rangle + \beta|0\rangle$$



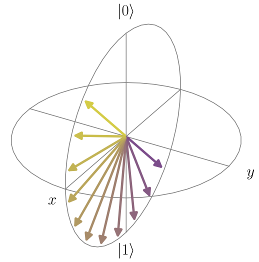
Phase flip

$$Z|\Psi\rangle = \alpha|0\rangle - \beta|1\rangle$$



Bit + Phase flip

$$Y|\Psi\rangle = i\alpha|1\rangle - i\beta|0\rangle$$

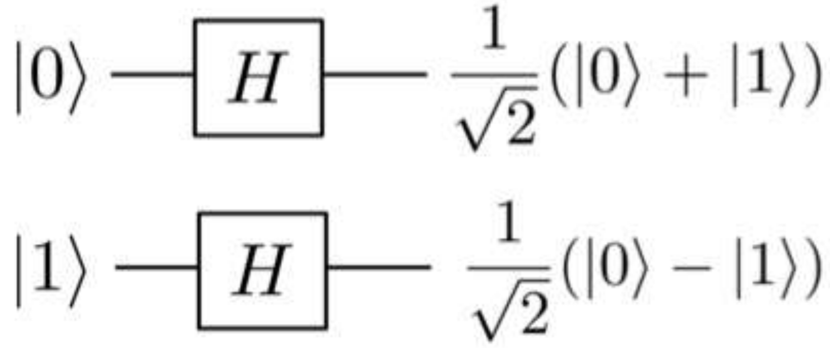


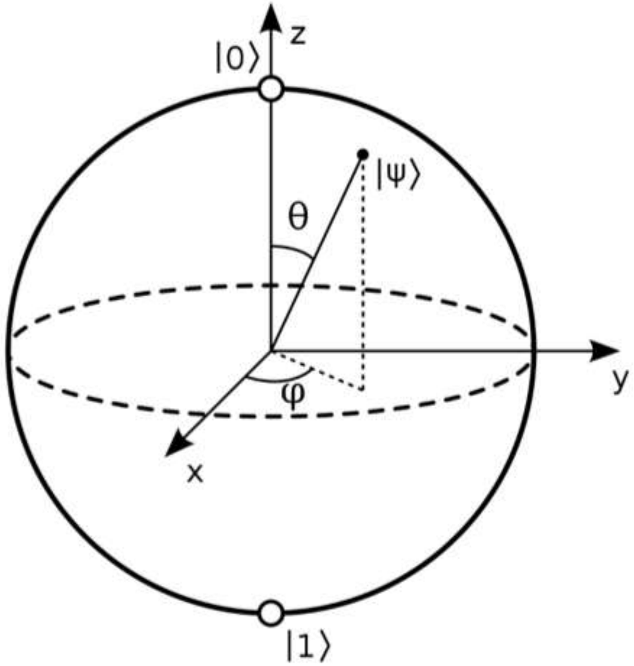
Rotations on the Bloch sphere are unitary (norm preserving) operations

Prelude: Programmable quantum computer

Small set of **quantum logic gates** gives universal quantum computer
Any unitary operation on *n* qubits (2^n by 2^n unitary matrix)

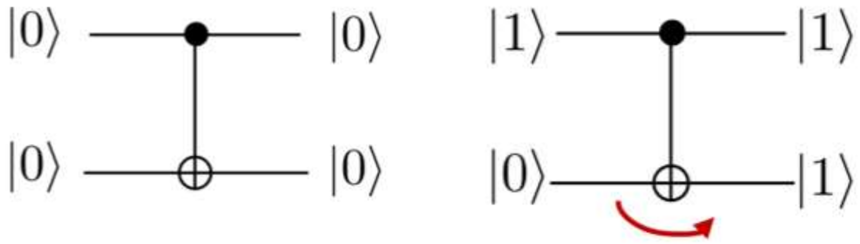
Hadamard gate – creates superposition





$$|\psi\rangle = \cos\frac{\Theta}{2}|0\rangle + e^{i\phi}\sin\frac{\Theta}{2}|1\rangle$$

CNOT gate – creates entanglement (2-qubit gate)

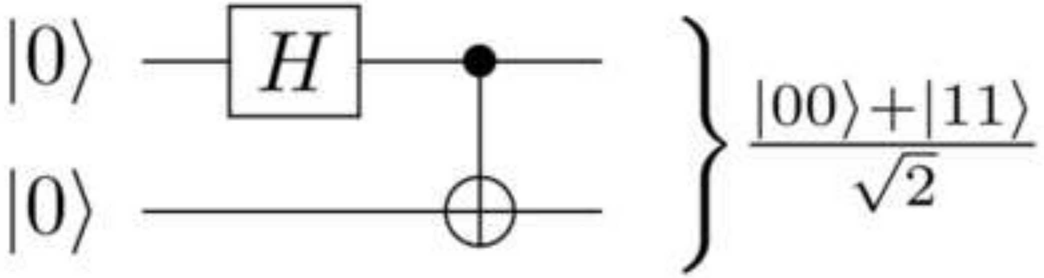


(+ additional single qubit rotation)

Prelude: Programmable quantum computer

quantum code=quantum circuit

For example. Circuit to create a "Bell-pair"



Entangled non-classical state
(not a product state)

Potential use cases for quantum computers

Shor's algoritm. Prime factorization

1789 x 1801= ? Easy
3221989= ? x ? Hard

Chryptography/Digital security

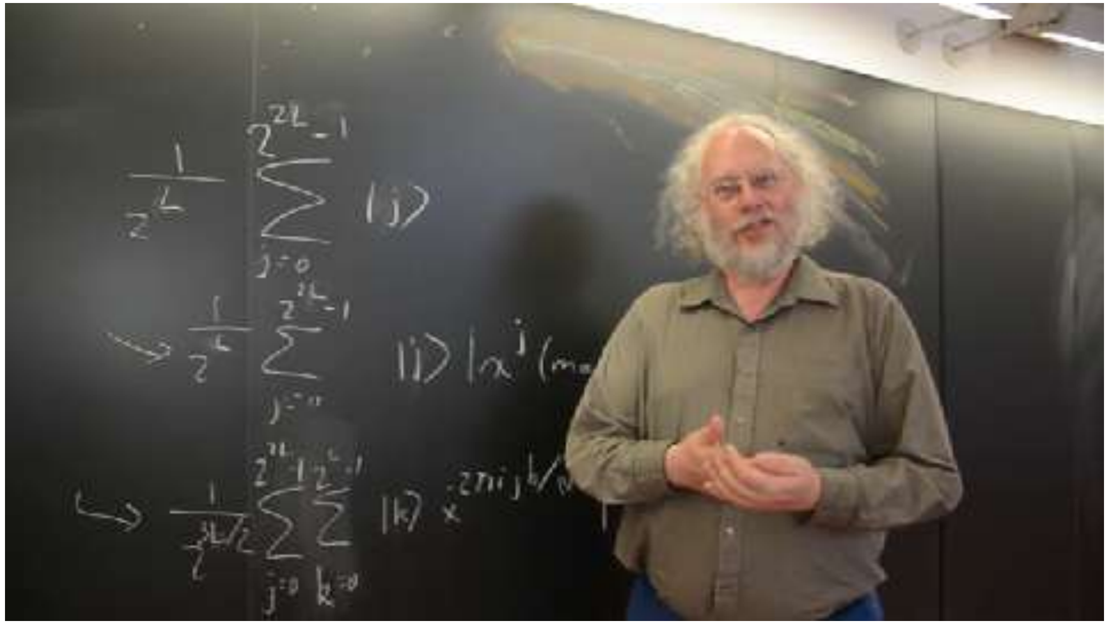


Image: physicsworld

Undiscovered Algorithms!
(no deep learning without GPU's)

Quantum chemistry: New molecules, drugs, etc.

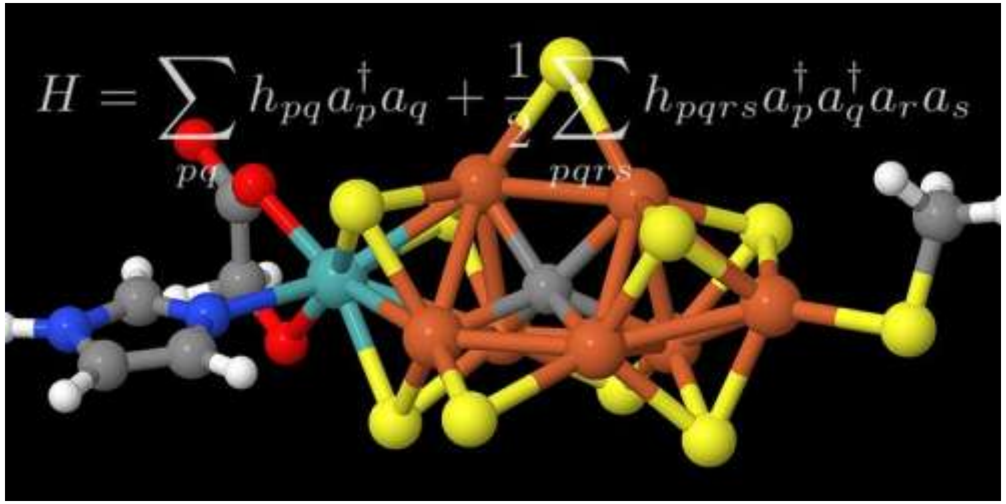


Image: Tech Explorist

Optimization: Logistics, planning, etc.

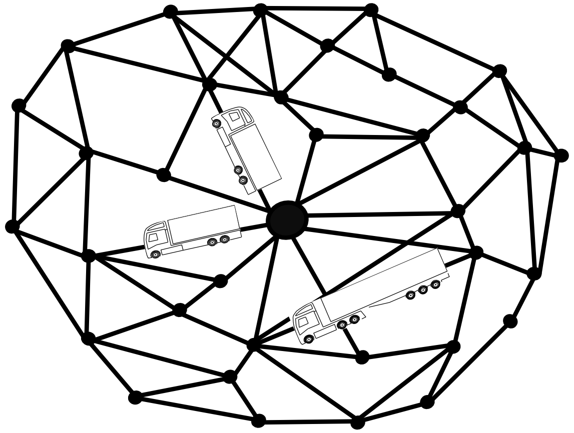
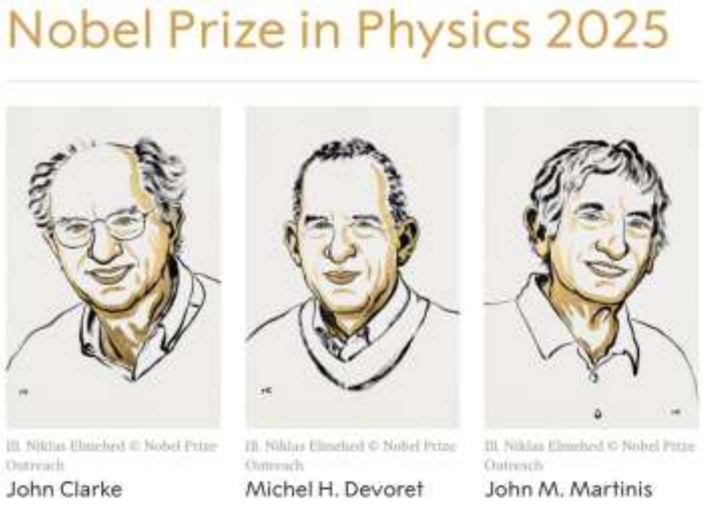


Image: David Fitzek

Quantum computing limited by decoherence



"for the discovery of macroscopic quantum mechanical tunnelling and energy quantisation in an electric circuit"

Superconducting (transmon) qubits:

- lifetime ~ 100 microsecond
- two qubit gate times ~ 100 nanoseconds

Maximum few 100 gates deep circuits
(error rates ~ 10^{-2} - 10^{-3})

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

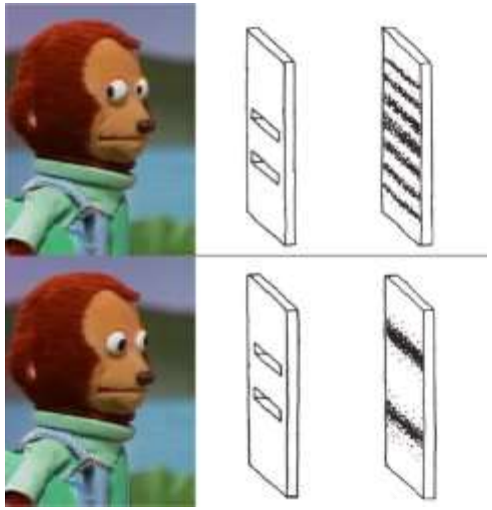
Assuming all to all connectivity and universal gate set!

Quantum error correction needed!

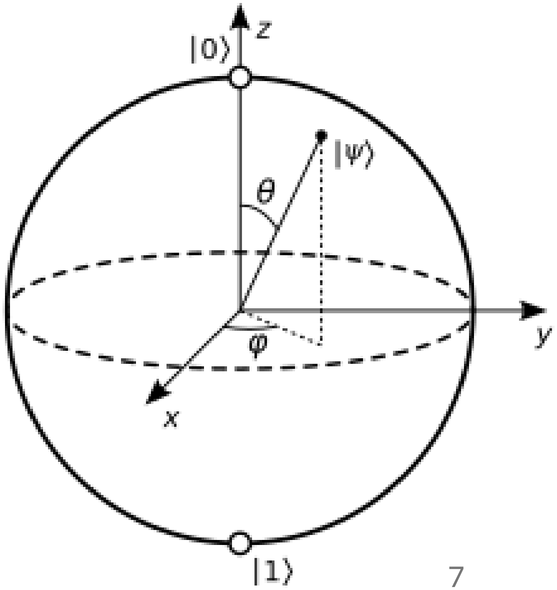
Quantum error correction

Major challenges compared to classical error correction using redundancy, e.g. 0 => 000..., 1 => 111...

1. No cloning of quantum state
2. Measurement are destructive (projective)
3. Continuous set of errors (rotations on the Bloch sphere)

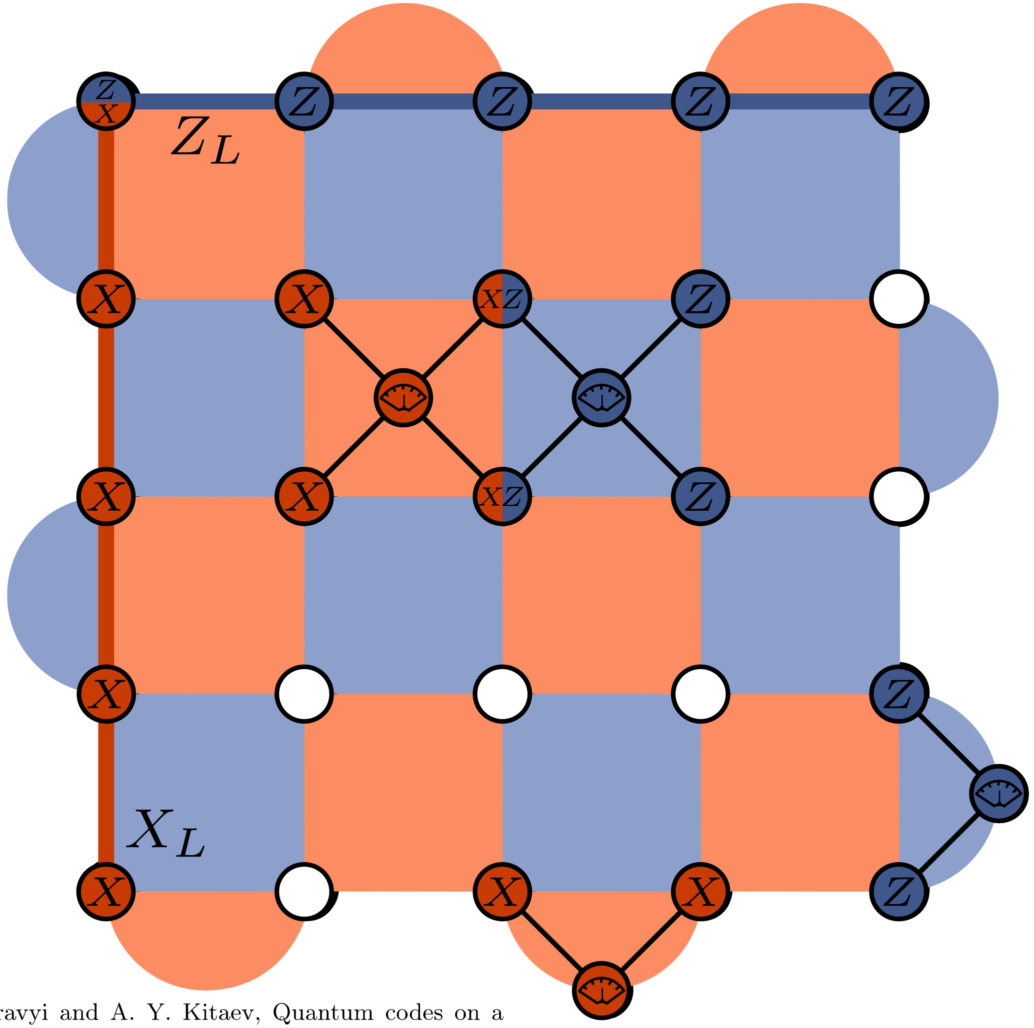


Stabilizer codes resolves these issues!



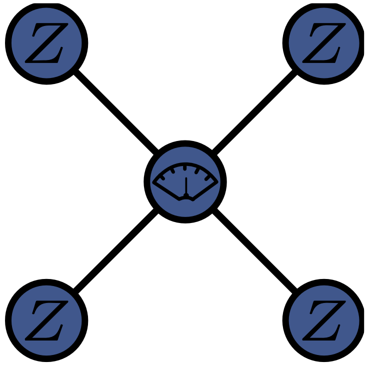
Peter W. Shor, "Scheme for reducing decoherence in quantum computer memory," Physical Review A **52**, R2493–R2496 (1995).
A. M. Steane, "Error Correcting Codes in Quantum Theory," Physical Review Letters **77**, 793–797 (1996).
Daniel Gottesman, "Stabilizer Codes and Quantum Error Correction," (1997),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

The surface code

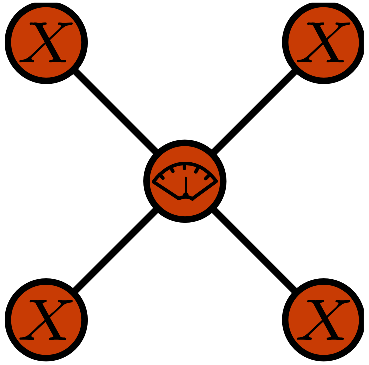


S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary (1998), arXiv:quant-ph/9811052.
E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, Journal of Mathematical Physics **43**, 4452 (2002).
A. Kitaev, Fault-tolerant quantum computation by anyons, Annals of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

- Lattice of d^2 ***data qubits***
- Measure $d^2 - 1$ stabilizers:



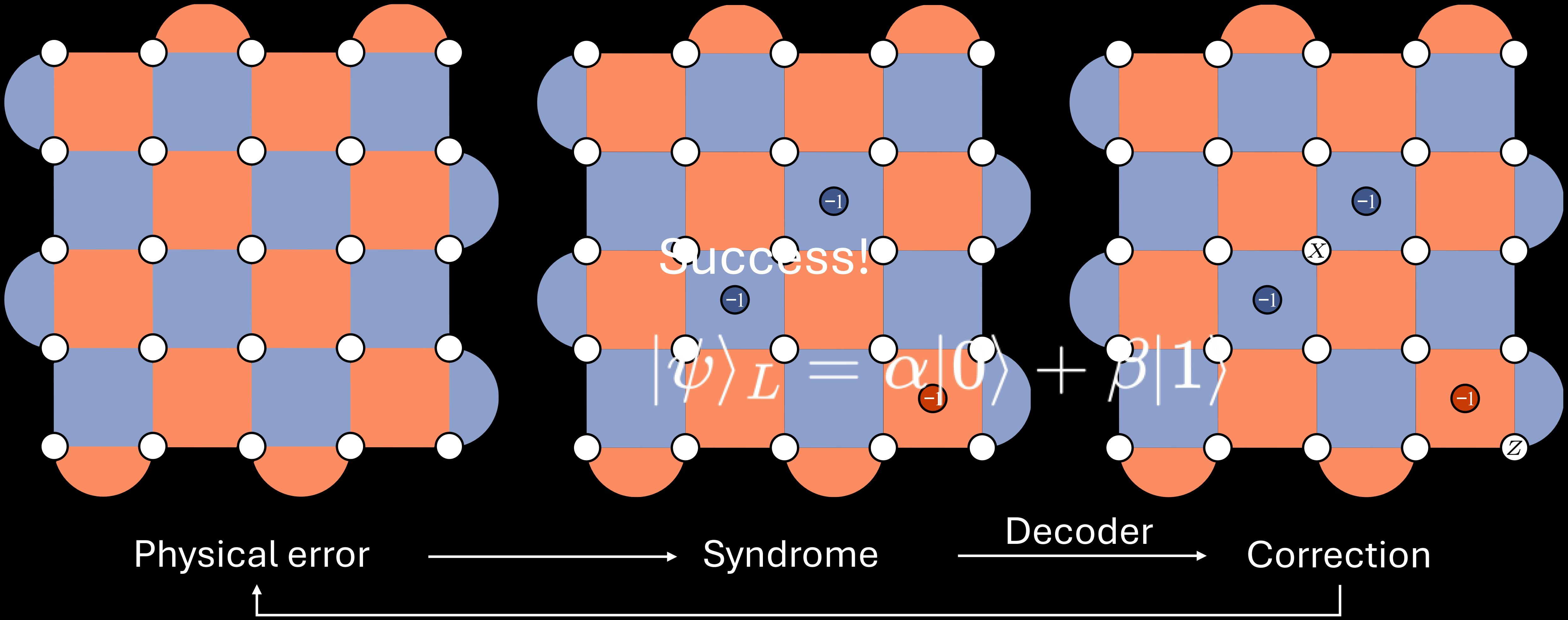
Detects bit errors



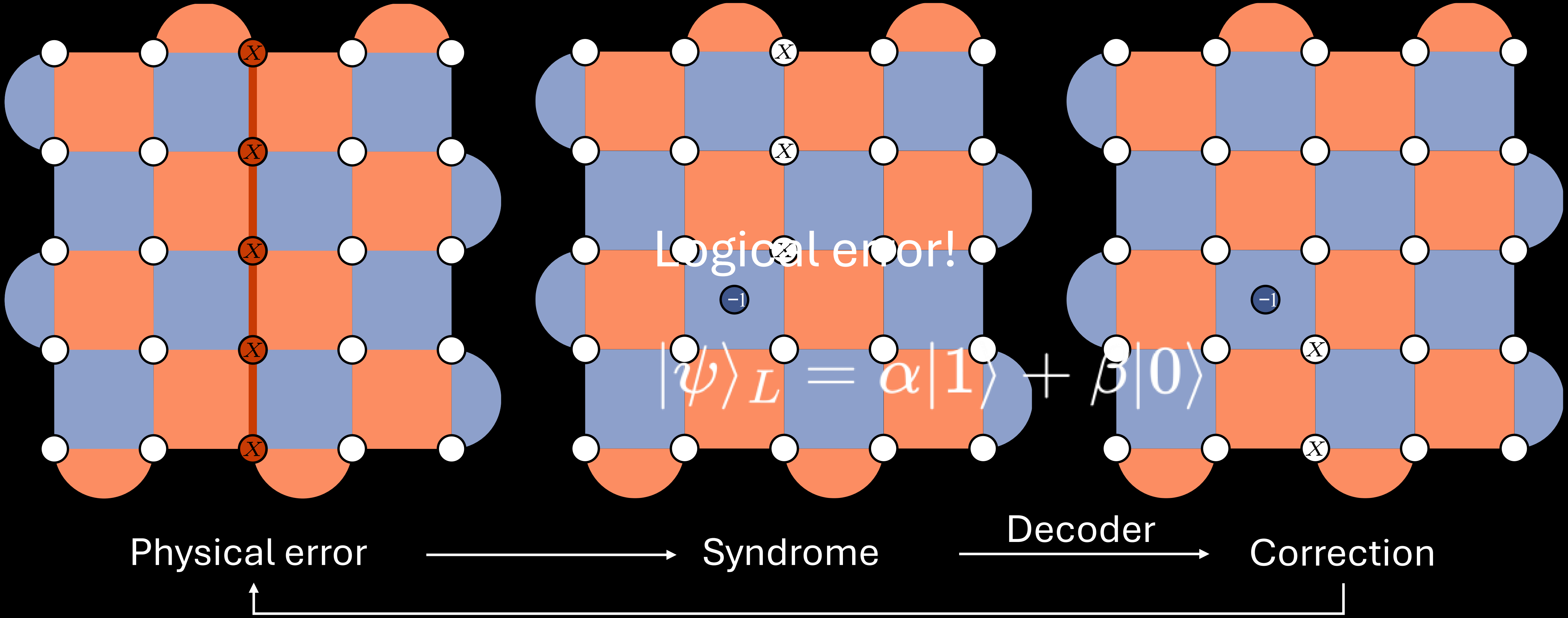
Detects phase errors

- Encodes one logical qubit: Z_L and X_L

Decoder is essential



Decoder is essential



Experimental realization: Google Quantum AI

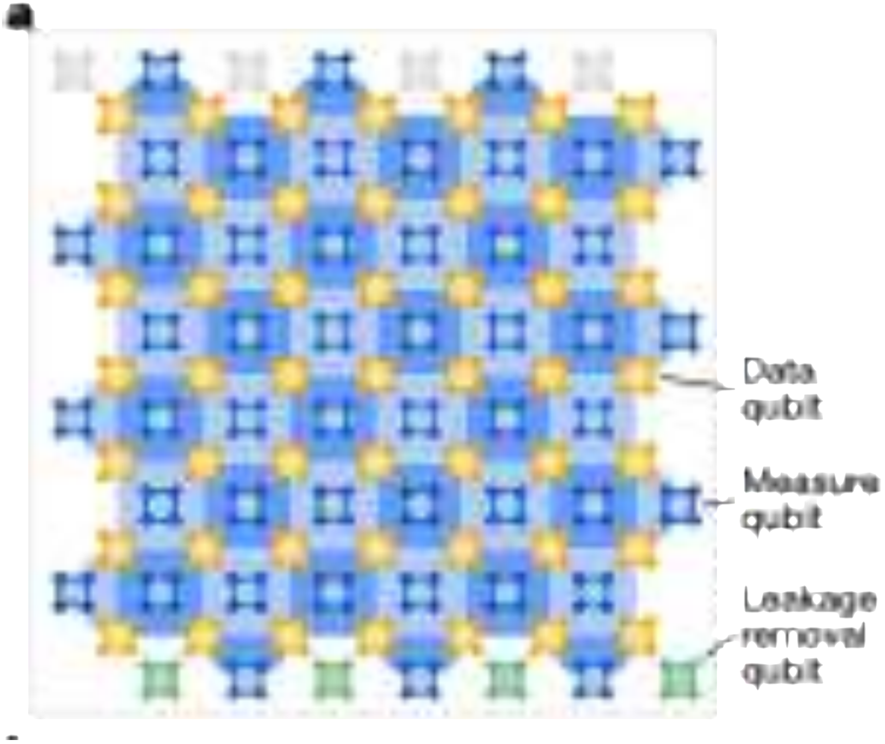
Article

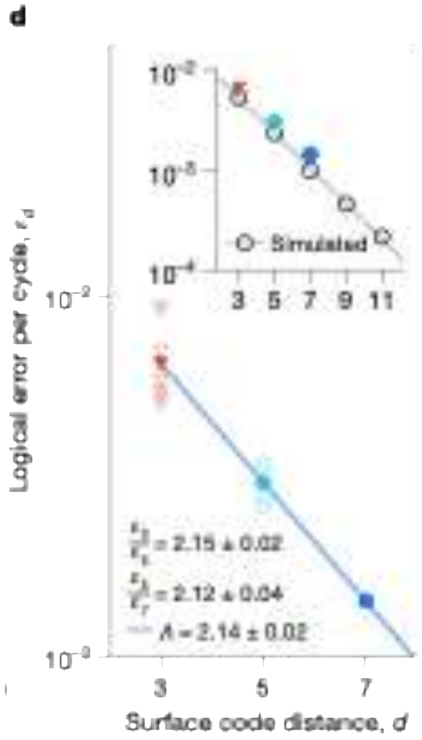
Quantum error correction below the surface code threshold

https://doi.org/10.1038/s41586-024-08449-y
Google Quantum AI and Collaborators*
Received: 24 August 2024

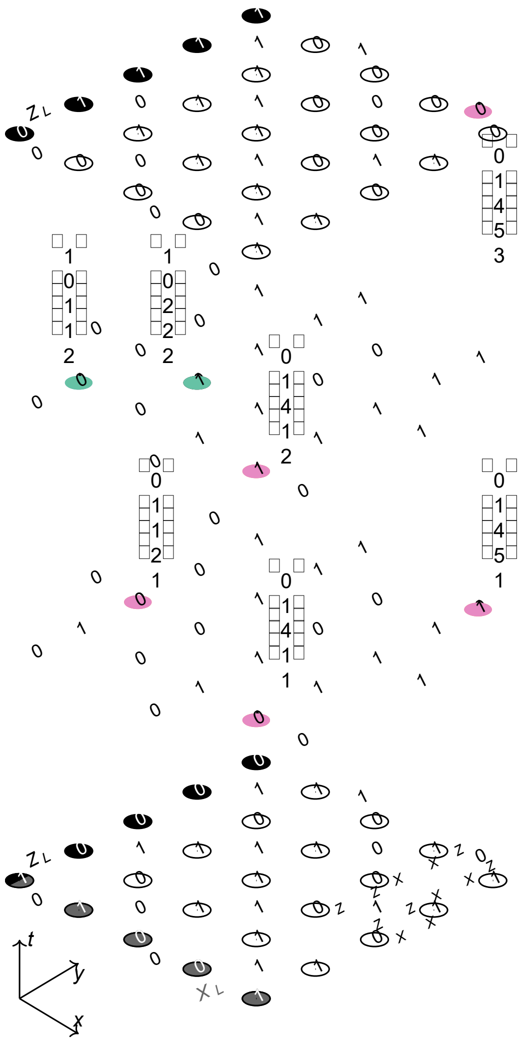
Exponential suppression of logical error rate with code-distance

$$P_L \uparrow (p/p_{th.})^{\uparrow d/2 \downarrow}$$



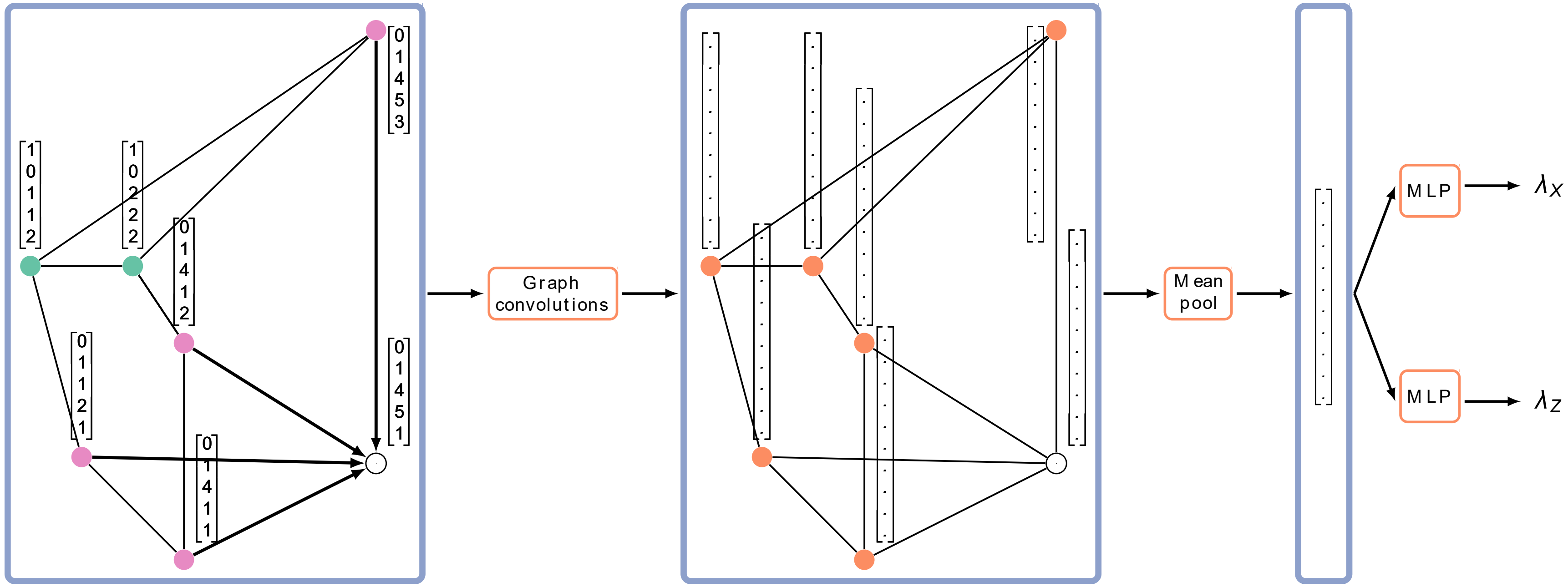


Graph neural network decoder



Tailored to experimental input

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

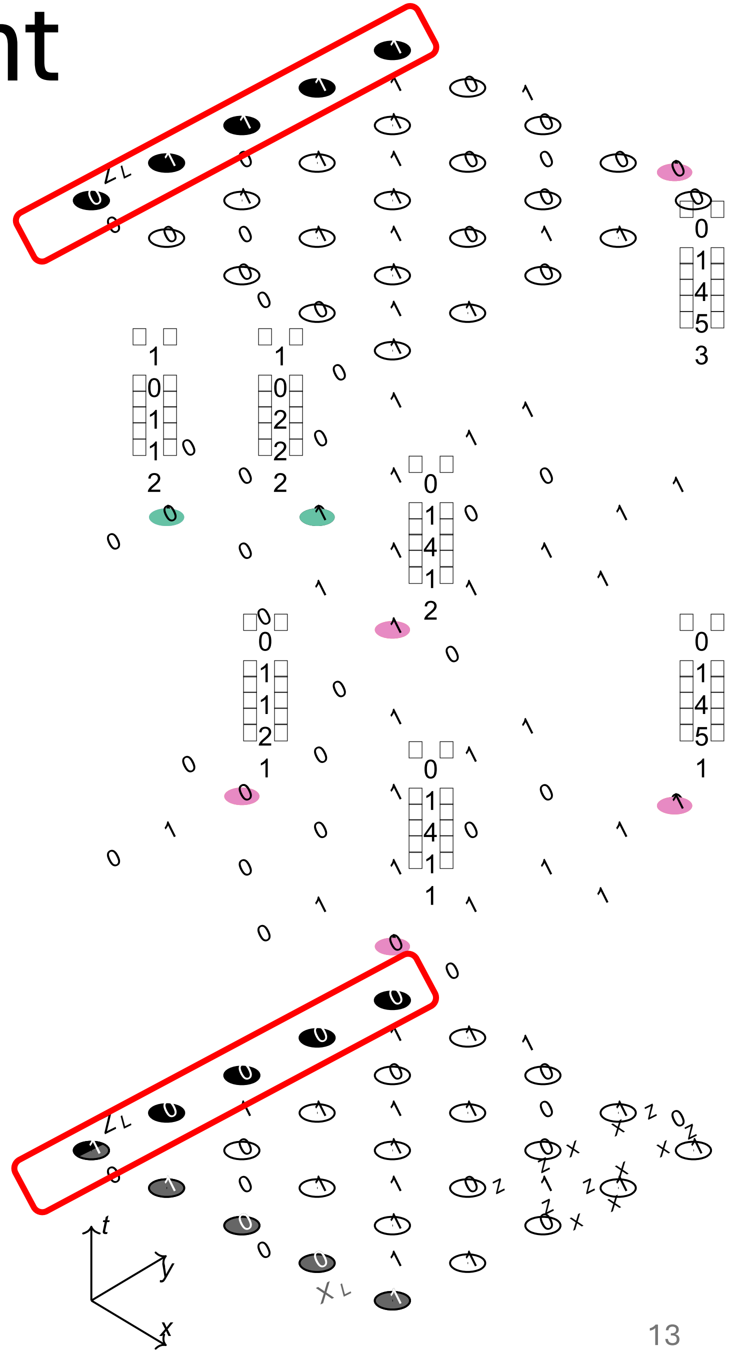


PHYSICAL REVIEW RESEARCH 7, 023181 (2025)

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

Moritz Lange[1,*] Pontus Havström[1] Basudha Srivastava[1,2] Isak Bengtsson[3] Valdemar Bergentall,[1] Karl Hammar,[1] Olivia Heuts,[1] Evert van Nieuwenburg,[4,†] and Mats Granath[1,‡]
[1]Department of Physics, University of Gothenburg, Gothenburg, Sweden
[2]Quantinuum, Terrington House, 13-15 Hills Rd, Cambridge CB2 1NL, United Kingdom
[3]Department of Physics, Chalmers University of Technology, Gothenburg, Sweden
[4]Leiden Inst. of Advanced Computer Science, Leiden University, Leiden, Netherlands

Memory-Z experiment

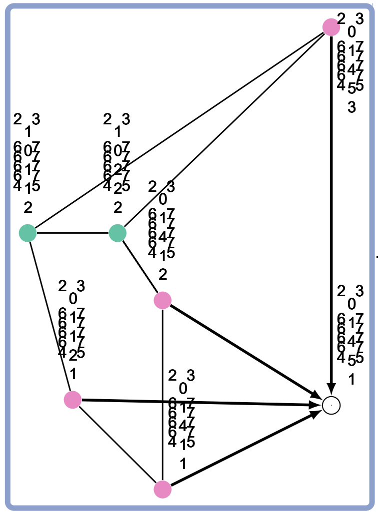


1. Simple product state prepared
2. Stabilizers are measured over several rounds
   => changes=detector events
3. Final individual qubit measurement
4. Measured error compared to decoder prediction

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**

Pruning of edges based on edge weights

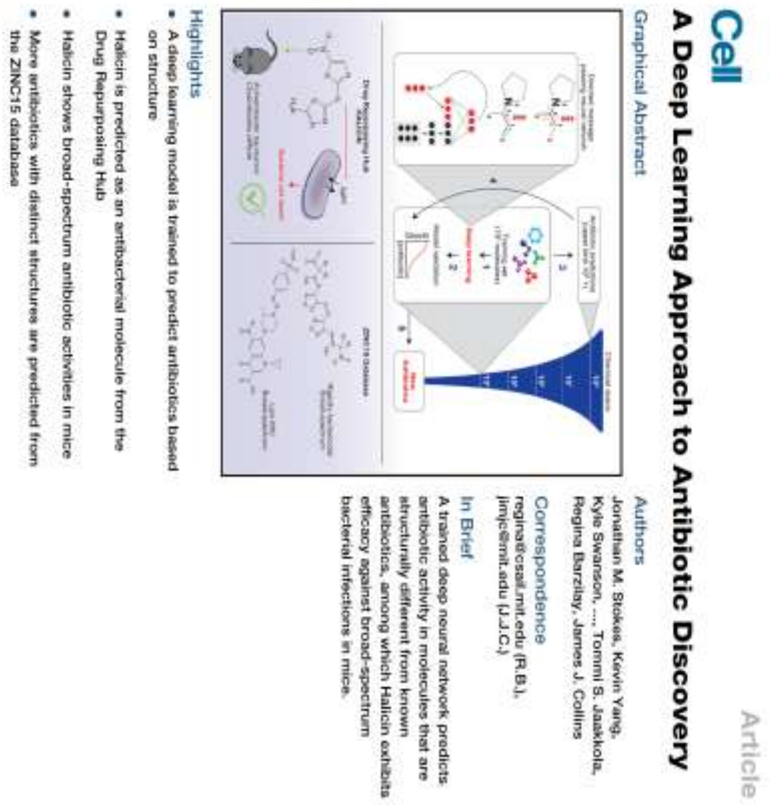


Graph neural networks (GNN)

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

Examples

Antibiotic discovery
graph regression



Cell

Article

A Deep Learning Approach to Antibiotic Discovery

Graphical Abstract

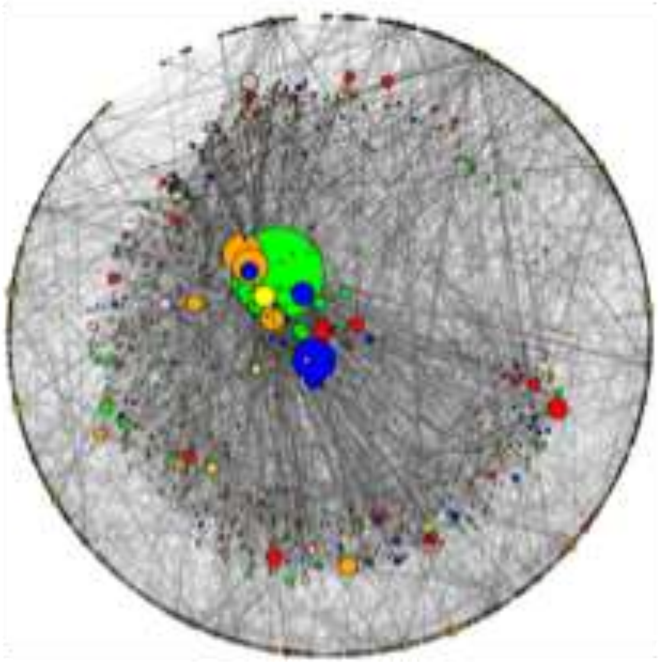
Authors
Jonathan M. Stokes, Kevin Yang, Kyle Swanson, ..., Tommi S. Jaakkola, Regina Barzilay, James J. Collins

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.

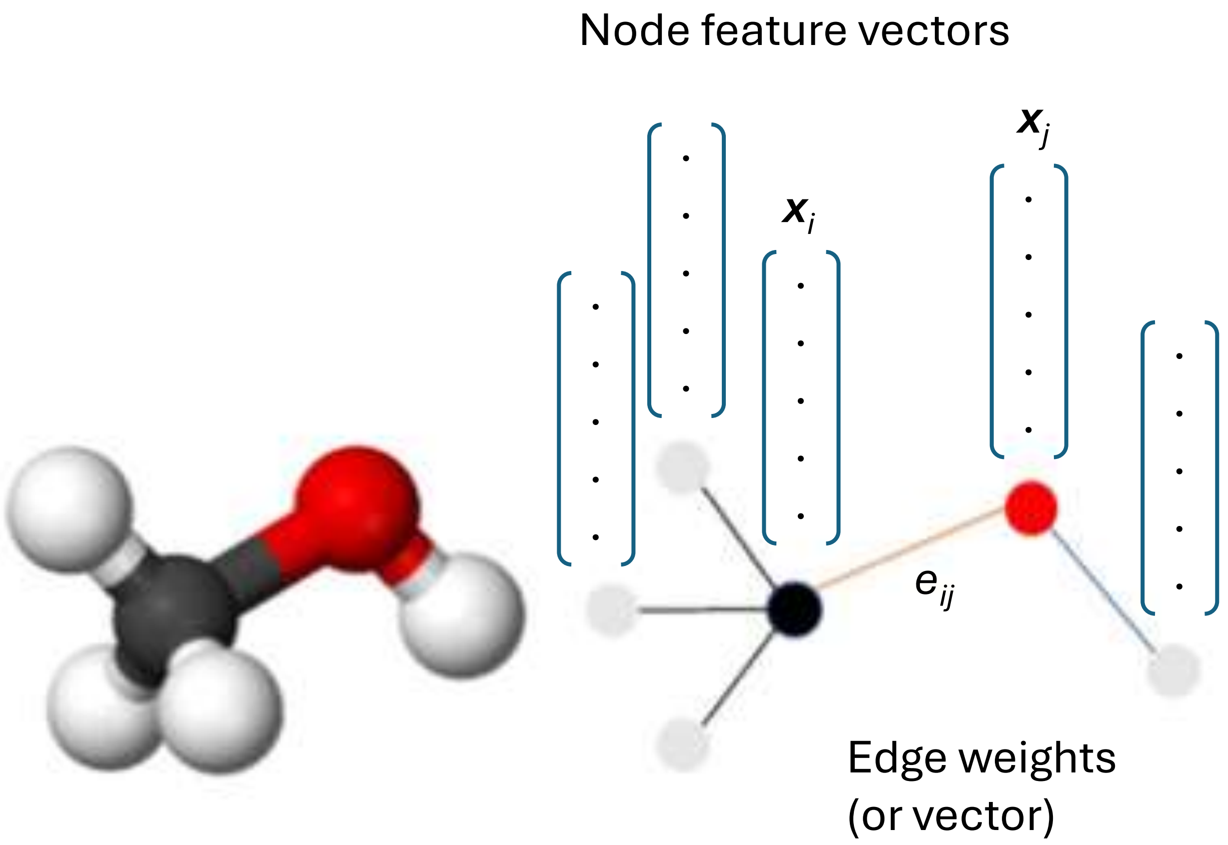
Highlights
- A deep learning model is trained to predict antibiotics based on structure
- Halicin is predicted as an antibacterial molecule from the Drug Repurposing Hub
- Halicin shows broad-spectrum antibiotic activities in mice
- More antibiotics with distinct structures are predicted from the ZINC15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

Cora dataset, citation network
Node classification



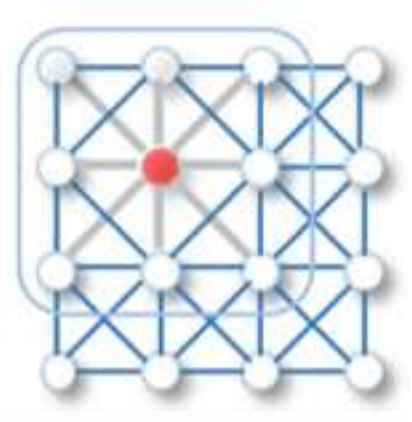
McCallum et al. 2000

Data object: Decorated graph



Graph convolutional layers

Grid: Standard convolutional filter of fixed size neighborhood



Graph: Convolutional filter adapted to varying neighborhood

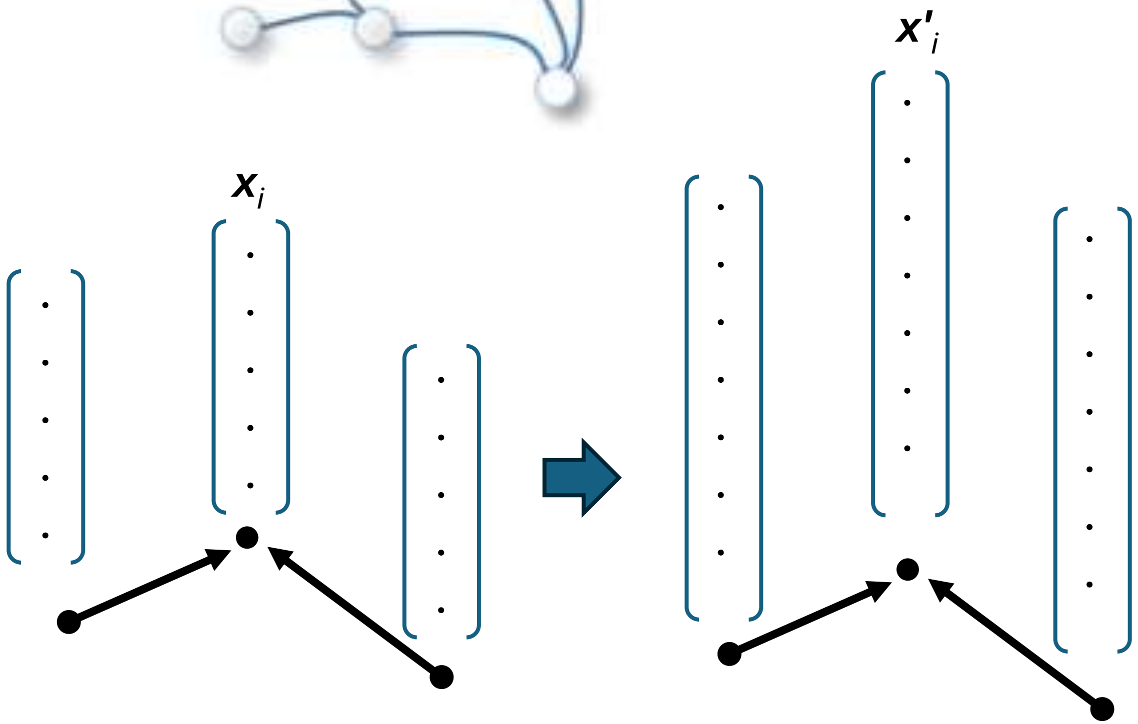
A comprehensive survey on graph neural networks
Wu et al. 2019

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

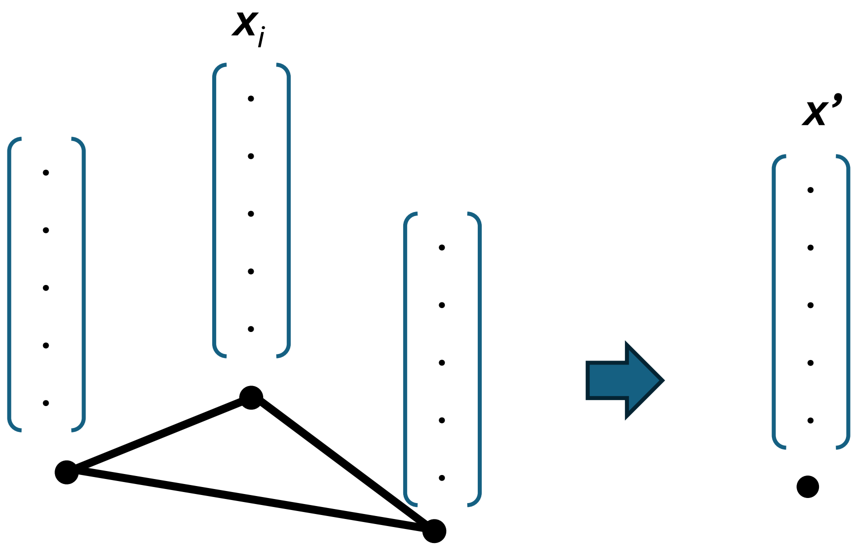
Semi-supervised classification with graph convolutional networks
Kipf and Wellling, 2016



Graph pooling

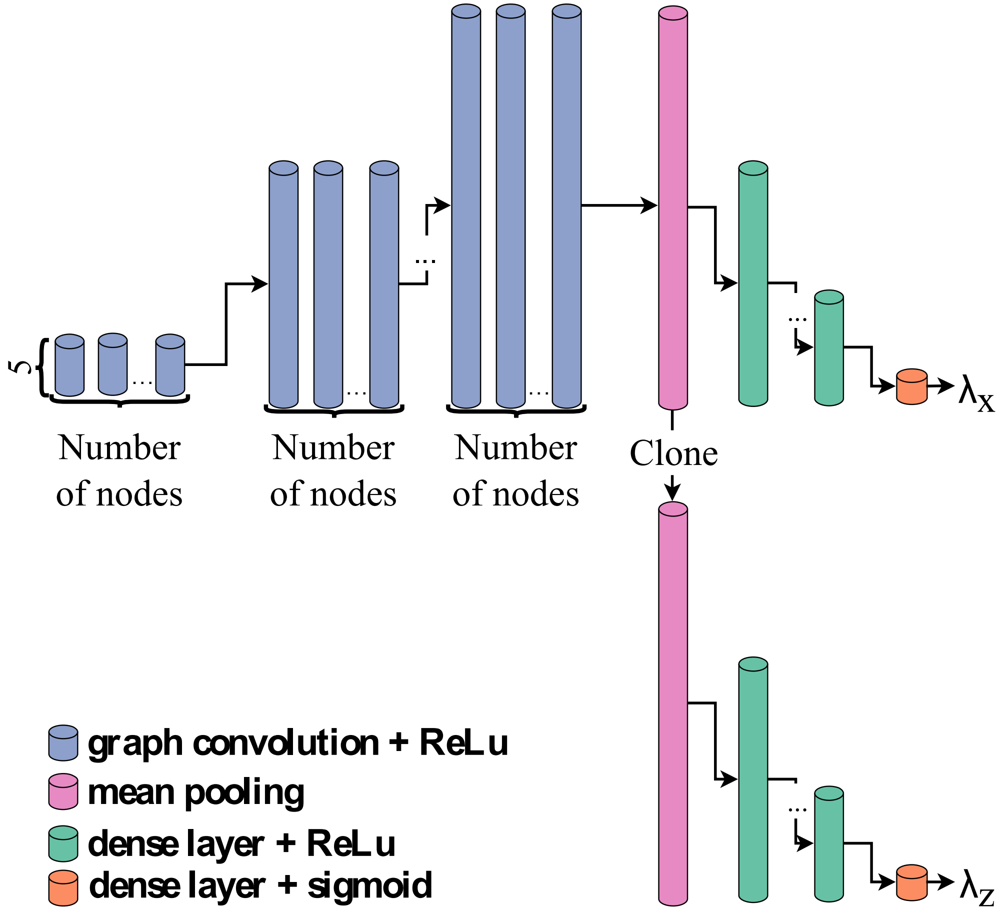
For graph classification output should be
independent of number of nodes

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



Graph embedding +
Standard dense network classifier

GNN decoder network architecture



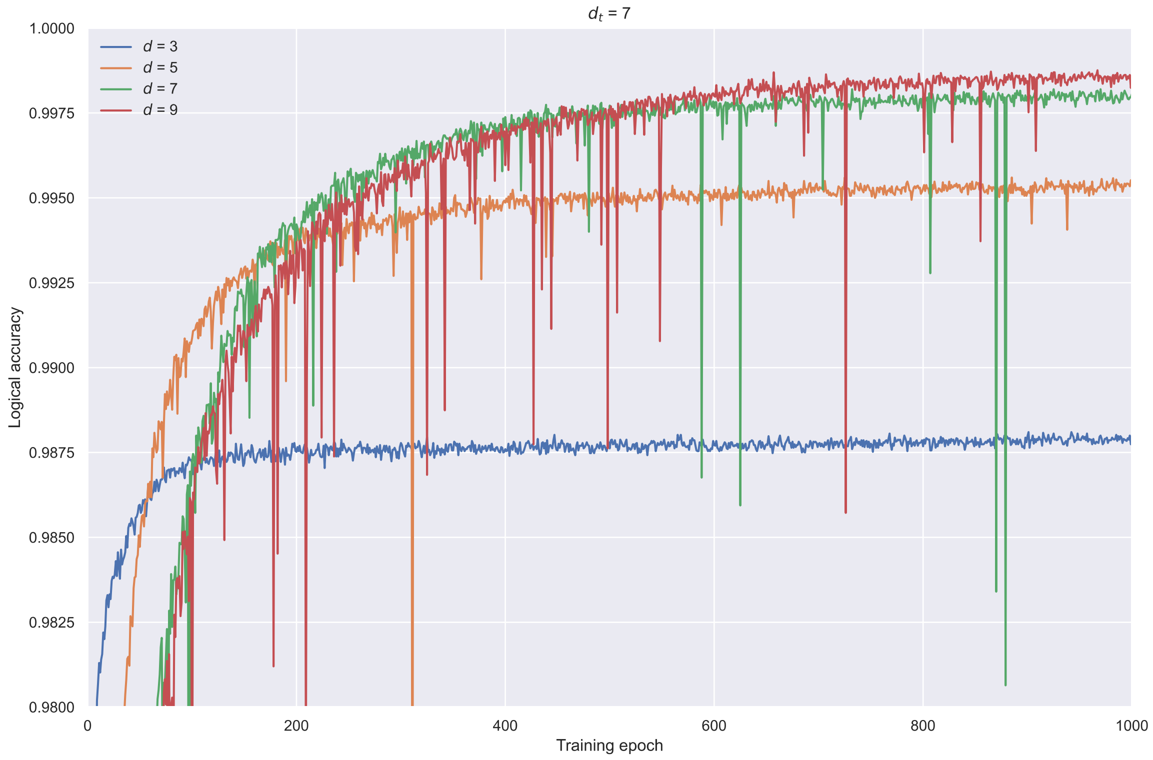
Layer	d_{in}	d_{out}
GraphConv_1	5	32
GraphConv_2	32	128
GraphConv_3	128	256
GraphConv_4	256	512
GraphConv_5	512	512
GraphConv_6	512	256
GraphConv_7	256	256
Dense_1 X	256	128
Dense_2 X	128	181
Dense_3 X	128	32
Dense_4 X	32	1
Dense_1 Z	256	128
Dense_2 Z	128	181
Dense_3 Z	128	32
Dense_4 Z	32	1

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.

Training

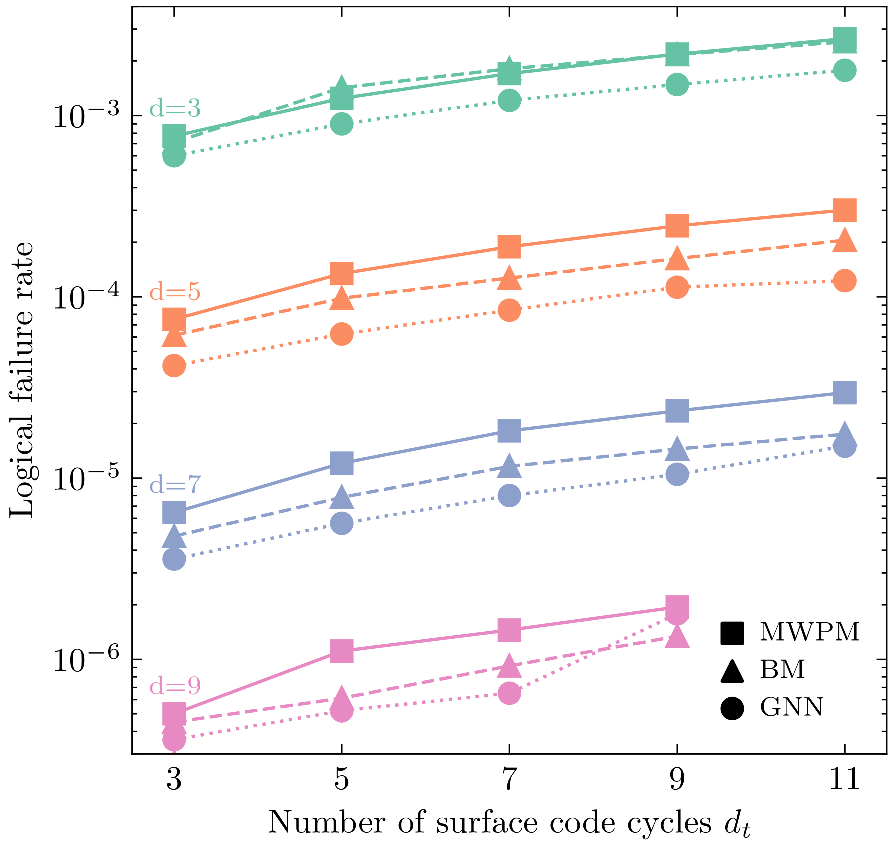
- The code is simulated with "semi-realistic" error model
- Data generated in **large batches** of 10,000-25,000 graphs (as much as can fit on the GPU memory)
- **No reuse of data** (no risk to overfit)
- Week(s) of training on one Nvidia A100
- **Up to 10^{10} datapoints**

Examples of a few training curves

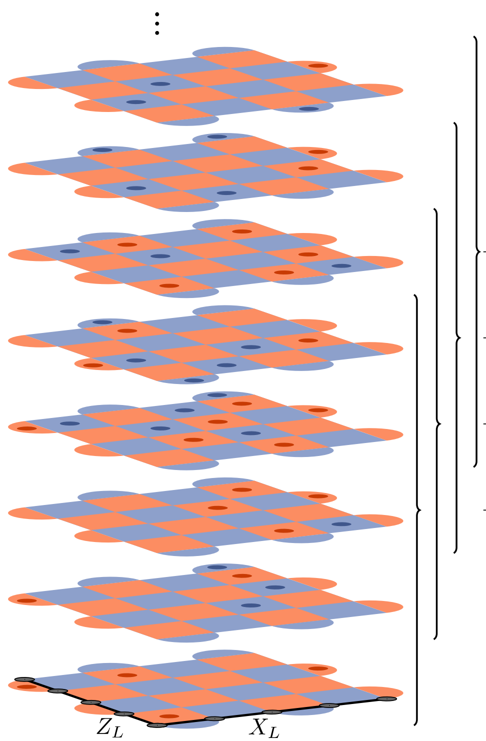


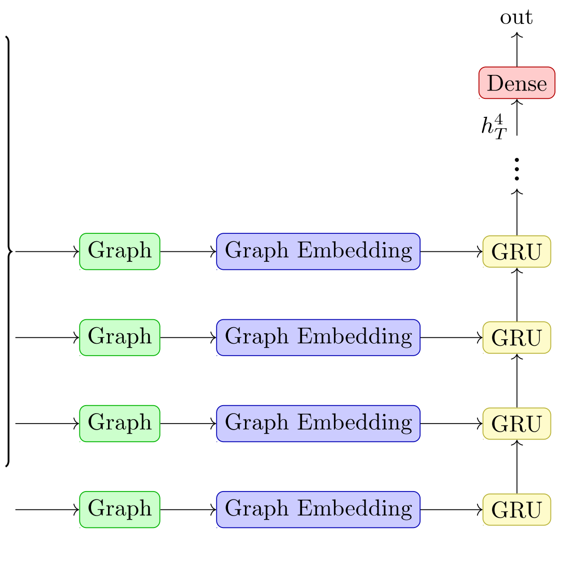
1 "epoch" is 10^7 graphs

Outperforms sota decoders



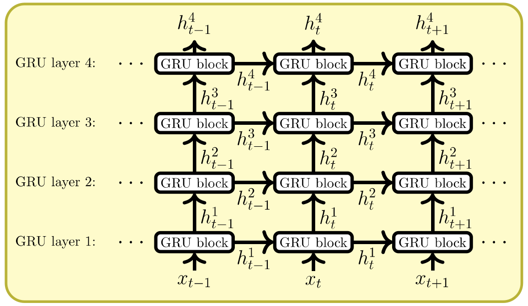
New sliding window decoder: GNN+RNN

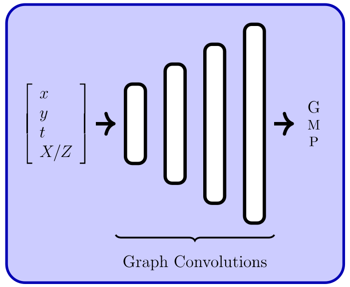


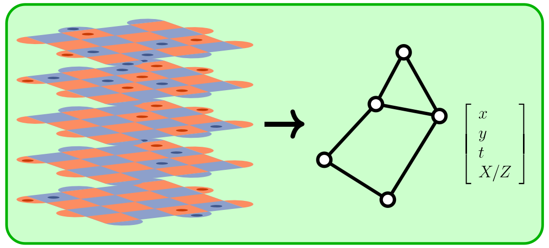


Master thesis: Gustaf Jonasson + Ole Fjeldså

Graph neural network +
Recurrent neural network







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

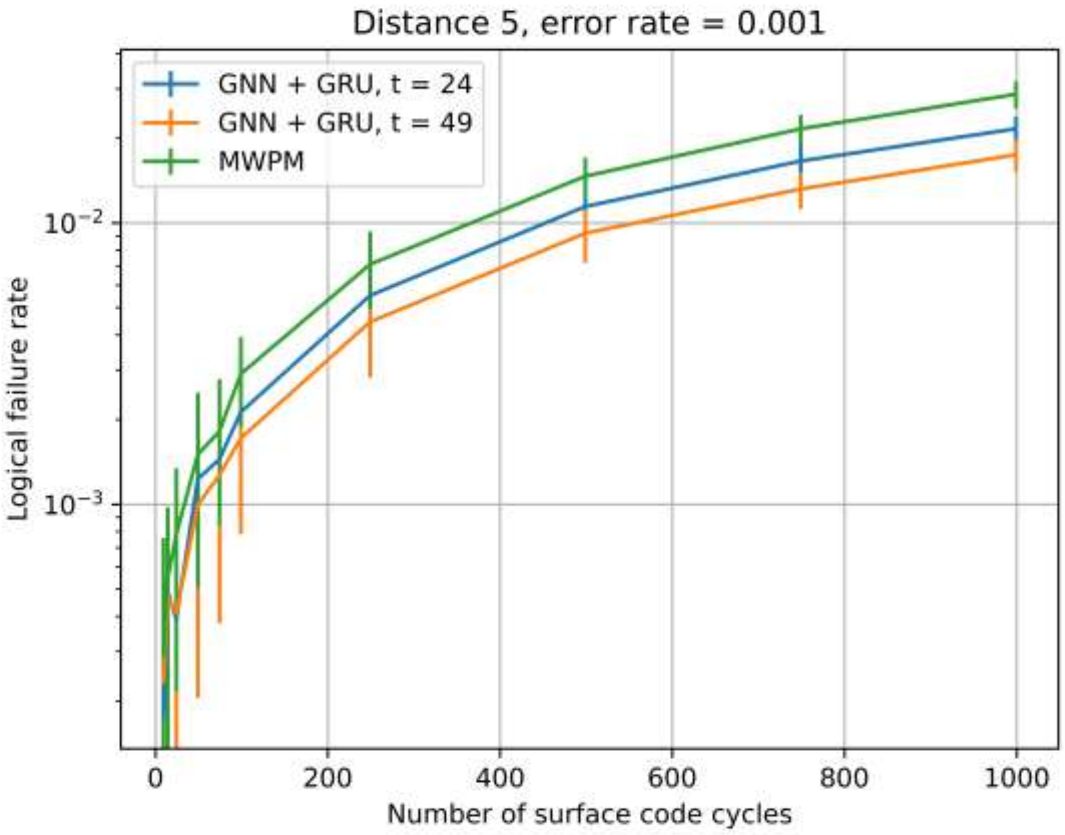
Johannes Bausch[1*†], Andrew W Senior[1*†], Francisco J H Heras[1†], Thomas Edlich[1†], Alex Davies[1†], Michael Newman[2†], Cody Jones[2], Kevin Satzinger[2], Murphy Yuezhen Niu[2], Sam Blackwell[1], George Holland[1], Dvir Kafri[2], Juan Atalaya[2], Craig Gidney[2], Demis Hassabis[1], Sergio Boixo[2],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

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

9 Oct 2023

Decoding multiple cycles

Trained on few cycles, generalizes to many cycles



Collecting training data using IBM devices

through
WACQT QT Testbed

Kvantfelskorrektion av bitfel i repetitionskoder med maskininlärning

Hugo Modin Asklid, Nils Kalmnäs Drakenfors, Caroline Paulis, Tova Hedmar, Emil Olofsson, Jacob Sandström

Datadriven avkodning av experimentella syndrommätningar från kvantfasfelkorrigerande repetitionskoder med neurala grafnätverk

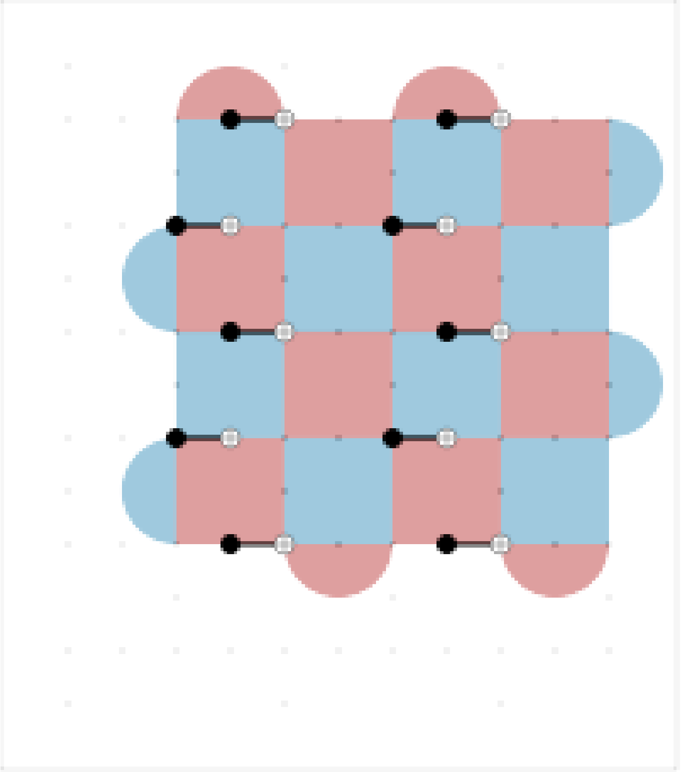
Kandidatarbete inom civilingenjörsprogrammet Teknisk fysik

Simon Benstorp, Liam Bulut Falkenström, Simon Eriksson, Alexander Jonsson, Lovisa Petersson, Vidar Petersson

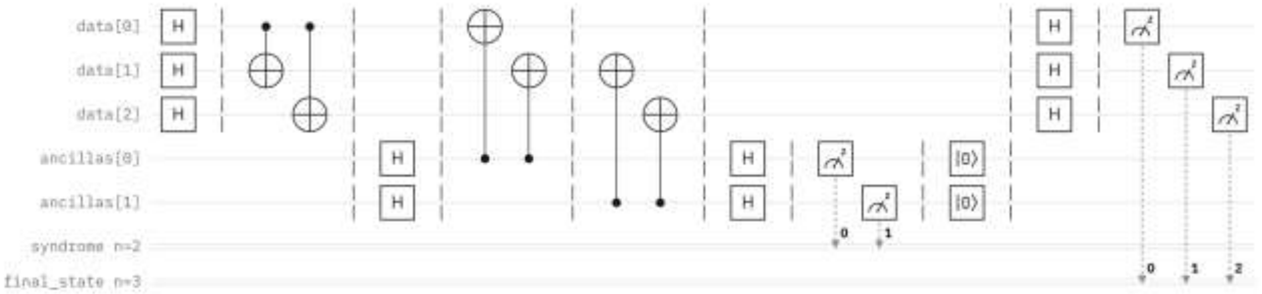
Summer Project 2024

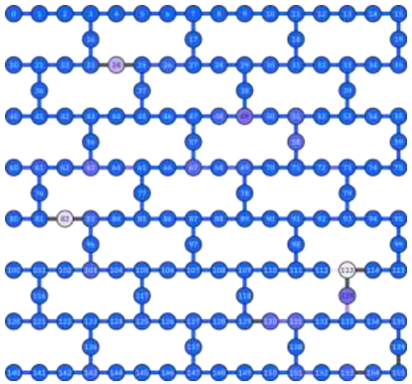
Jacob Olsson

Surface code on heavy hex



Repetition code





Fas	Kodavstånd d	Tidsrepetitioner d_t	Antal körningar s
+	3	3	2000000
+	3	5	2000000
+	5	5	2000000
+	7	7	1000000
+	9	9	1000000

Summary

- Quantum error correction is crucial for quantum computing
- Machine learning can be an important tool for QEC
- Outlook:
    - Move from GPU to FPGA for fast inference (work in progress)
    - Running code on IBM hardware for real training data (work in progress)
    - Error correction on the local WACQT (Chalmers) hardware

Thanks to! **Moritz Lange**, *Isak Bengtsson, Pontus Havström, Valdemar Bergentall, Karl Hammar, Vidar Petersson, Jacob Olsson, Ole Fjeldså, Gustaf Jonasson Johansson,* Evert van Nieuwenburg (Leiden), Basudha Srivastava (Quantinuum/PsiQuantum)

WACQT Wallenberg Centre for Quantum Technology

NAISS

UNIVERSITY OF GOTHENBURG