Designing neural decoders for large toric codes

Xiaotong Ni (TU Delft)
Erlangen, May 9

arXiv: 1809.06640
Toric code

Assuming only Pauli-X error on qubits
Syndrome extraction is noiseless

10% error rate
Toric code

Assuming only Pauli-X error on qubits
Syndrome extraction is noiseless

- Correction is sensitive to the small change of syndrome, and involves a lot of parity computation.

10% error rate
Some previous works of neural decoders

S. Krastanov and L. Jiang, Scientific Reports 7 (2017)
Repeatedly sample error configurations using RBM / NN

Keep track of the Pauli frames in a simulated circuit level noise model
Main result

• Introduced a quite reliable way to build neural decoders for large distance toric code (and likely many other topological codes)

• Previous works: d~10 —> This work d=64,…

• Source code (and a Google Colab script) can be found at https://github.com/XiaotongNi/toric-code-neural-decoder
Motivation
Motivation

• Same as several previous works
Motivation

• Same as several previous works
  • A convenient way to adapt to experimental noise models
Motivation

• Same as several previous works
  • A convenient way to adapt to experimental noise models
  • Offer a nice combination of speed and accuracy for topological codes.
Motivation

• Same as several previous works
  • A convenient way to adapt to experimental noise models
  • Offer a nice combination of speed and accuracy for topological codes.
• Go to quite large input size / code distance
Motivation

- Same as several previous works
  - A convenient way to adapt to experimental noise models
  - Offer a nice combination of speed and accuracy for topological codes.
- Go to quite large input size / code distance
  - Traditional decoders works for arbitrary size.
Motivation

- Same as several previous works
  - A convenient way to adapt to experimental noise models
  - Offer a nice combination of speed and accuracy for topological codes.
- Go to quite large input size / code distance
  - Traditional decoders works for arbitrary size.
  - Test versatility of NN / Find ways to overcome difficulties
Implementation
“Imitation learning”

1. Approximate the decoder in the paper.
"Imitation learning"

1. Approximate the decoder in the paper.

2. Further optimization with (syndrome, correction) pairs

Image from Colah's blog
RG decoder

coarse graining
• Replace detailed syndrome information with educated guess on the error rates of “coarse-grained qubits”.

• Educated guess done by belief propagation.
• Replace detailed syndrome information with educated guess on the error rates of “coarse-grained qubits”.
• Educated guess done by belief propagation.
• Replace detailed syndrome information with educated guess on the error rates of “coarse-grained qubits”.
• Educated guess done by belief propagation.
Stack NN together and train

$\begin{align*}
x_i & \quad \xrightarrow{\text{BP NN}} \\
0.1 & \quad \xrightarrow{\text{BP NN}} \\
0.9 & 
\end{align*}$
Stack NN together and train

We can then train all layers altogether, using (syndrome, correction) pairs.
Performance

With less than 2 hours of training time, we can get similar performance compared to minimum-weight perfect matching on bit-flip noise.
About training

Offline training

Neural decoder

Any reasonable amount of training data and time

Goal: good decoder on noise model similar to experiments, or decoders that can utilize error rate information
About training

**Offline training**

- Neural decoder

**Training with experimental data**

- Neural decoder

*Any reasonable amount of training data and time*

*Limited amount of training data (not focused in this work)*

Goal: good decoder on noise model similar to experiments, or decoders that can utilize error rate information
offline trained model

parameter space of neural networks

optimal decoder for experiment
Adapt to other noise models

- Small changes in the noise model likely can be compensated by small changes in the early layers (need to break the translational symmetry in some way)
Spatially varying error rates

• Proof of principle for a toy noise model
How it is done
How it is done
How it is done
How it is done

Roughly speaking: fix parameters of the neural decoder and only train \( \{w_i\} \)
Roughly speaking: fix parameters of the neural decoder and only train \( \{w_i\} \)

- Increase the logical accuracy from 0.967 to 0.993 after trained on this noise model (\( d=16 \))
General takeaway message
“Imitation learning”

1. Approximate
“Imitation learning”

1. Approximate

2. Further optimization with data

A Better heuristic algorithm

Heuristic algorithm
Heuristic algorithm

- Subroutine 1
- Subroutine 2
- Subroutine 3
Lazy way of injecting our knowledge about the problem to the machine learning model
Better way to utilize knowledge?
Better way to utilize knowledge?

Model

- Data
- Physical law
- Previous works