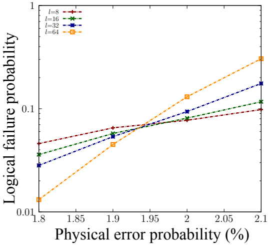
Surprising facts about quantum error correction

Andrew Darmawan, Nicolas Delfosse, Pavithran Iyer,
& David Poulin
Funded by ARO QCVV W911NF-14-C-0048

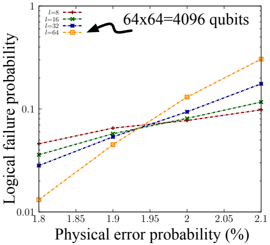
Institut Quantique & Département de Physique
Université de Sherbrooke

Sydney Quantum Information Theory Workshop
Coogee, NSW, Australia, January 2017

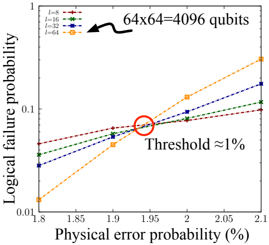
The accuracy threshold theorem



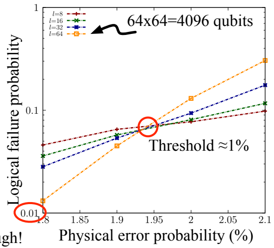
The accuracy threshold theorem



The accuracy threshold theorem

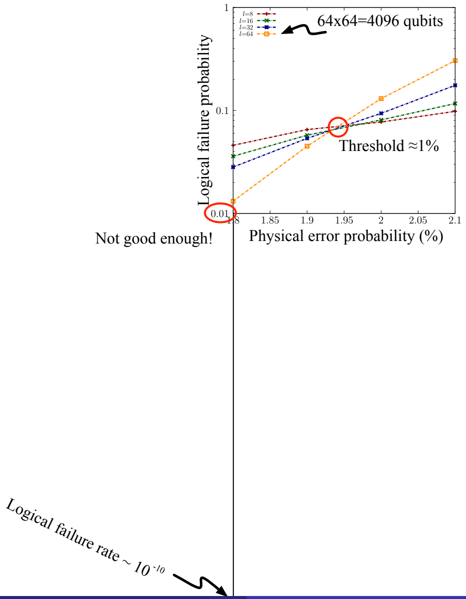


The accuracy threshold theorem

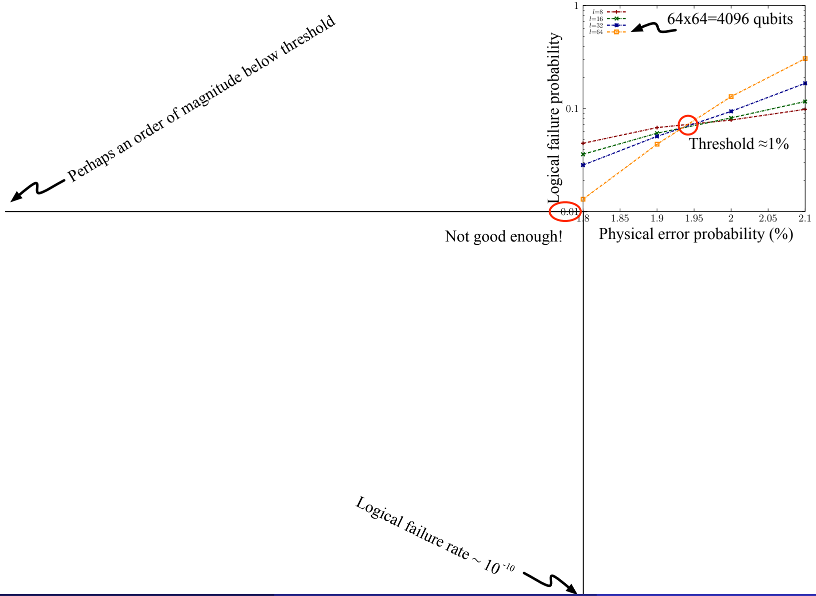


Not good enough!

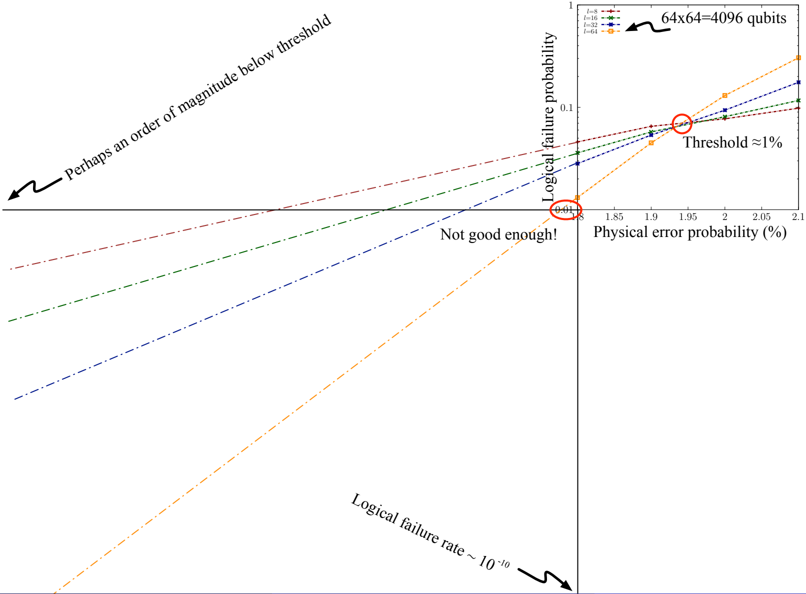
The accuracy threshold theorem



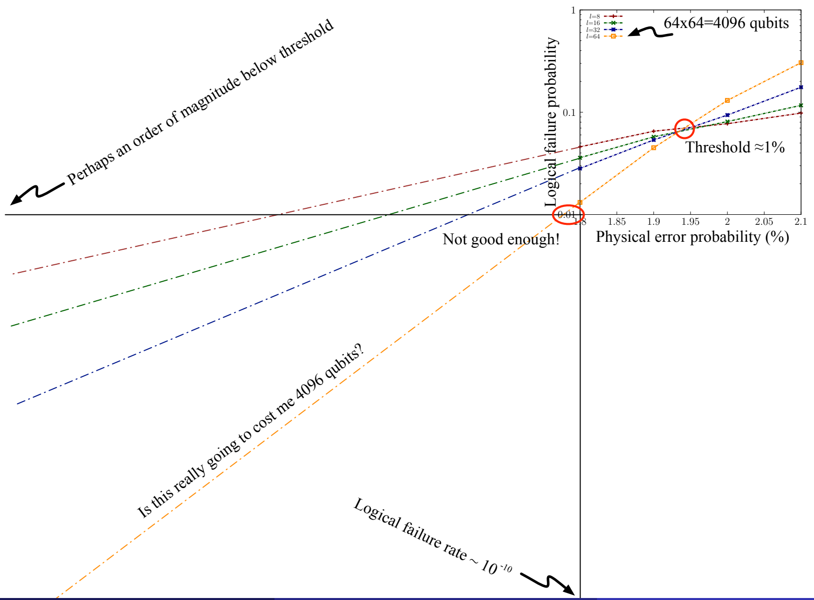
The accuracy threshold theorem



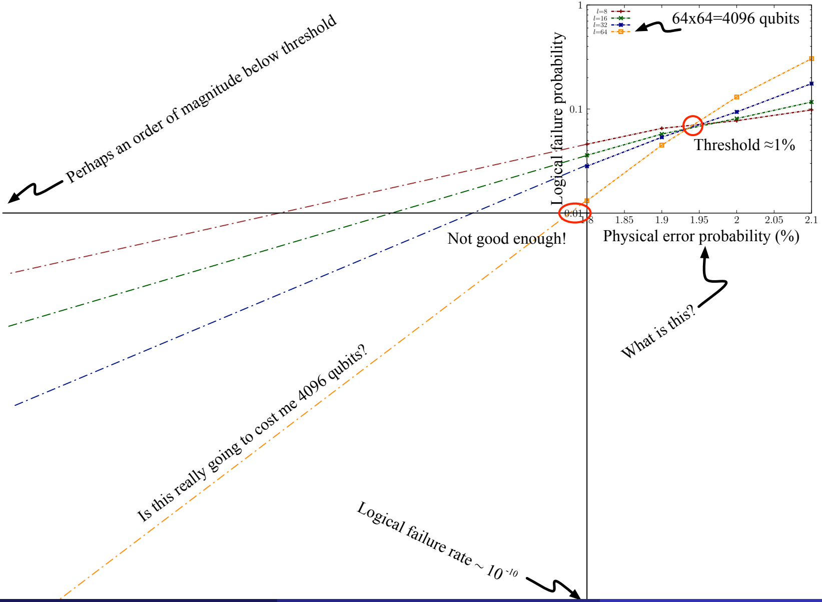
The accuracy threshold theorem



The accuracy threshold theorem



The accuracy threshold theorem



Outline

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog($\frac{1}{\delta}$).

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog$(\frac{1}{\delta})$.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like $Fe_2 S_2$.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog$(\frac{1}{\delta})$.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog$(\frac{1}{\delta})$.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog($\frac{1}{\delta}$).

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog($\frac{1}{\delta}$).

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog$(\frac{1}{\delta})$.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog($\frac{1}{\delta}$).

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Fault-tolerant overhead

- We want to execute a quantum algorithm with N logical gates.
  - $N \sim 10^{12}$-10^{15} to simulate a small molecule like Fe_2S_2.
- Each gate is error-corrected to accuracy δ, so errors build up to
  - $N\delta$ if they add coherently (worst case, systematic bias).
  - $\sqrt{N}\delta$ if they add stochastically.
- δ needs to be $\sim 1/\sqrt{N}$ to $1/N$ to prevent harmful error build up.
  - 10^{-6} to 10^{-15} for quantum chemistry (pretty vague).
- If the physical noise rate ϵ is sub threshold, then fault-tolerant error correction can produce logical gates of accuracy δ with overhead polylog$(\frac{1}{\delta})$.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

Outline

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
  - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
  - Density matrix renormalization group (DMRG).
  - Projected entangled pairs state (PEPS).
  - Multi-scale entanglement renormalization ansatz (MERA).
  - etc.

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
    - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
    - Density matrix renormalization group (DMRG).
    - Projected entangled pairs state (PEPS).
    - Multi-scale entanglement renormalization ansatz (MERA).
    - etc.

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
  - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
  - Density matrix renormalization group (DMRG).
  - Projected entangled pairs state (PEPS).
  - Multi-scale entanglement renormalization ansatz (MERA).
  - etc.

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
  - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
  - Density matrix renormalization group (DMRG).
  - Projected entangled pairs state (PEPS).
  - Multi-scale entanglement renormalization ansatz (MERA).
  - etc.

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
  - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
  - Density matrix renormalization group (DMRG).
  - Projected entangled pairs state (PEPS).
  - Multi-scale entanglement renormalization ansatz (MERA).
  - etc.

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
  - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
  - Density matrix renormalization group (DMRG).
  - Projected entangled pairs state (PEPS).
  - Multi-scale entanglement renormalization ansatz (MERA).
  - etc.

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
  - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
  - Density matrix renormalization group (DMRG).
  - Projected entangled pairs state (PEPS).
  - Multi-scale entanglement renormalization ansatz (MERA).
  - etc.

Quantum many-body methods

- Realistic noise models cannot be efficiently simulated.
  - Interacting quantum many-body problem.

Our contribution

Study fault-tolerance with realistic noise models using numerical many-body techniques

- Tensor network methods
  - Density matrix renormalization group (DMRG).
  - Projected entangled pairs state (PEPS).
  - Multi-scale entanglement renormalization ansatz (MERA).
  - etc.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\text{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

- Prepare some known code state $|\bar{\psi}\rangle$
- Applying some noise $\mathcal{E}$ to $\rho = |\bar{\psi}\rangle\langle\bar{\psi}|$.
  - When $\mathcal{E}$ is some stochastic noise, we can sample the noise instead of applying $\mathcal{E}$.
- Sample the syndrome bits $\mathrm{pr}_j(\pm) = \frac{1}{2}(1 \pm \mathrm{Tr}[\mathcal{E}(\rho)S_j])$.
- Decode, i.e., find a correction operation C based on the observed syndrome.
- Apply the correction to the post-measurement state ρ'.
- Evaluate the logical transformation that has been applied to the logical state.
- Repeat for different input states $\bar{\psi}$ to perform logical process tomography.
  - We actually use Jamilkowski isomorphism instead.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome pr(s).
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)}\mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)}\|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\mathcal{E}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome pr(s).
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)}\mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)}\|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\mathcal{E}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome pr(s).
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)}\mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)}\|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\mathcal{E}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome pr(s).
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)}\mathcal{E}_{\mathrm{s}}^{\mathrm{L}}$
- Average logical error $\sum_s \mathrm{pr(s)}\|\mathcal{E}_{\mathrm{s}}^{\mathrm{L}} - \mathrm{id}\|$
- Error of logical average $\|\mathcal{E}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome pr(s).
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)} \mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)} \|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\overline{\mathcal{E}}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome pr(s).
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)} \mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)} \|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\mathcal{E}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome pr(s).
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)}\mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)}\|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\overline{\mathcal{E}}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome $\mathrm{pr}(s)$.
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)}\mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)}\|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\mathcal{E}_L - \mathrm{id}\|$
- etc.

What goes into a simulation?

If we can do all of this...

Simulation

INPUT

- Noise $\mathcal{E}$.

OUTPUT

- A syndrome s.
- The probability of that syndrome $\mathrm{pr}(s)$.
- The logical channel conditioned on that syndrome $\mathcal{E}_s^L$.

Given this we can estimate...

- Average channel $\overline{\mathcal{E}}_L = \sum_s \mathrm{pr(s)}\mathcal{E}_\mathrm{s}^\mathrm{L}$
- Average logical error $\sum_s \mathrm{pr(s)}\|\mathcal{E}_\mathrm{s}^\mathrm{L} - \mathrm{id}\|$
- Error of logical average $\|\mathcal{E}_L - \mathrm{id}\|$
- etc.

Surface code

- The surface code logical basis states $|\bar{\psi}\rangle = |\bar{0}\rangle, |\bar{1}\rangle, |\bar{+}\rangle$, or $|\bar{-}\rangle$ are PEPS and are complete for logical process tomography.
- $\rho = \mathcal{E}(|\bar{\psi}\rangle\langle\bar{\psi}|)$ is a PEPO for any PEPSO CPTP map $\mathcal{E}$.
- Computing the probability of a syndrome bit $\frac{1}{2}(1 \pm \mathrm{Tr}[\rho S_j])$ boils down to contracting a 2D TN.
- The post-measurement state $\rho' \propto (1 \pm S_j)\rho(1 \pm S_j)$ is also a PEPO.
- Applying a Pauli correction operator to ρ' preserves the PEPO structure.
- Computing the logical probabilities $\mathrm{Tr}(\rho\bar{L})$ required for process tomography boils down to contracting a 2D TN.

While efficient, approximate contraction schemes are known, we choose to contract exactly, so the scaling is exponential with the (least) linear size of the lattice, e.g. $W \times L$ lattice, Le^W vs e^{LW}.

Surface code

- The surface code logical basis states $|\bar{\psi}\rangle = |\bar{0}\rangle,\ |\bar{1}\rangle,\ |\bar{+}\rangle$, or $|\bar{-}\rangle$ are PEPS and are complete for logical process tomography.
- $\rho = \mathcal{E}(|\bar{\psi}\rangle\langle\bar{\psi}|)$ is a PEPO for any PEPSO CPTP map $\mathcal{E}$.
- Computing the probability of a syndrome bit $\frac{1}{2}(1 \pm \mathrm{Tr}[\rho S_j])$ boils down to contracting a 2D TN.
- The post-measurement state $\rho' \propto (1 \pm S_j)\rho(1 \pm S_j)$ is also a PEPO.
- Applying a Pauli correction operator to ρ' preserves the PEPO structure.
- Computing the logical probabilities $\mathrm{Tr}(\rho\bar{L})$ required for process tomography boils down to contracting a 2D TN.

While efficient, approximate contraction schemes are known, we choose to contract exactly, so the scaling is exponential with the (least) linear size of the lattice, e.g. $W \times L$ lattice, Le^{W} vs e^{LW}.

Surface code

- The surface code logical basis states $|\bar{\psi}\rangle = |\bar{0}\rangle,\ |\bar{1}\rangle,\ |\bar{+}\rangle$, or $|\bar{-}\rangle$ are PEPS and are complete for logical process tomography.
- $\rho = \mathcal{E}(|\bar{\psi}\rangle\langle\bar{\psi}|)$ is a PEPO for any PEPSO CPTP map $\mathcal{E}$.
- Computing the probability of a syndrome bit $\frac{1}{2}(1 \pm \mathrm{Tr}[\rho S_j])$ boils down to contracting a 2D TN.
- The post-measurement state $\rho' \propto (1 \pm S_j)\rho(1 \pm S_j)$ is also a PEPO.
- Applying a Pauli correction operator to ρ' preserves the PEPO structure.
- Computing the logical probabilities $\mathrm{Tr}(\rho\bar{L})$ required for process tomography boils down to contracting a 2D TN.

While efficient, approximate contraction schemes are known, we choose to contract exactly, so the scaling is exponential with the (least) linear size of the lattice, e.g. $W \times L$ lattice, Le^W vs e^{LW}.

Surface code

- The surface code logical basis states $|\bar{\psi}\rangle = |\bar{0}\rangle,\ |\bar{1}\rangle,\ |\bar{+}\rangle$, or $|\bar{-}\rangle$ are PEPS and are complete for logical process tomography.
- $\rho = \mathcal{E}(|\bar{\psi}\rangle\langle\bar{\psi}|)$ is a PEPO for any PEPSO CPTP map $\mathcal{E}$.
- Computing the probability of a syndrome bit $\frac{1}{2}(1 \pm \mathrm{Tr}[\rho S_j])$ boils down to contracting a 2D TN.
- The post-measurement state $\rho' \propto (1 \pm S_j)\rho(1 \pm S_j)$ is also a PEPO.
- Applying a Pauli correction operator to ρ' preserves the PEPO structure.
- Computing the logical probabilities $\mathrm{Tr}(\rho\bar{L})$ required for process tomography boils down to contracting a 2D TN.

While efficient, approximate contraction schemes are known, we choose to contract exactly, so the scaling is exponential with the (least) linear size of the lattice, e.g. $W \times L$ lattice, Le^W vs e^{LW}.

Surface code

- The surface code logical basis states $|\bar{\psi}\rangle = |\bar{0}\rangle,\ |\bar{1}\rangle,\ |\bar{+}\rangle$, or $|\bar{-}\rangle$ are PEPS and are complete for logical process tomography.
- $\rho = \mathcal{E}(|\bar{\psi}\rangle\langle\bar{\psi}|)$ is a PEPO for any PEPSO CPTP map $\mathcal{E}$.
- Computing the probability of a syndrome bit $\frac{1}{2}(1 \pm \mathrm{Tr}[\rho S_j])$ boils down to contracting a 2D TN.
- The post-measurement state $\rho' \propto (1 \pm S_j)\rho(1 \pm S_j)$ is also a PEPO.
- Applying a Pauli correction operator to ρ' preserves the PEPO structure.
- Computing the logical probabilities $\mathrm{Tr}(\rho\bar{L})$ required for process tomography boils down to contracting a 2D TN.

While efficient, approximate contraction schemes are known, we choose to contract exactly, so the scaling is exponential with the (least) linear size of the lattice, e.g. $W \times L$ lattice, Le^W vs e^{LW}.

Surface code

- The surface code logical basis states $|\bar{\psi}\rangle = |\bar{0}\rangle,\ |\bar{1}\rangle,\ |\bar{+}\rangle$, or $|\bar{-}\rangle$ are PEPS and are complete for logical process tomography.
- $\rho = \mathcal{E}(|\bar{\psi}\rangle\langle\bar{\psi}|)$ is a PEPO for any PEPSO CPTP map $\mathcal{E}$.
- Computing the probability of a syndrome bit $\frac{1}{2}(1 \pm \mathrm{Tr}[\rho S_j])$ boils down to contracting a 2D TN.
- The post-measurement state $\rho' \propto (1 \pm S_j)\rho(1 \pm S_j)$ is also a PEPO.
- Applying a Pauli correction operator to ρ' preserves the PEPO structure.
- Computing the logical probabilities $\mathrm{Tr}(\rho\bar{L})$ required for process tomography boils down to contracting a 2D TN.

While efficient, approximate contraction schemes are known, we choose to contract exactly, so the scaling is exponential with the (least) linear size of the lattice, e.g. $W \times L$ lattice, Le^W vs e^{LW}.

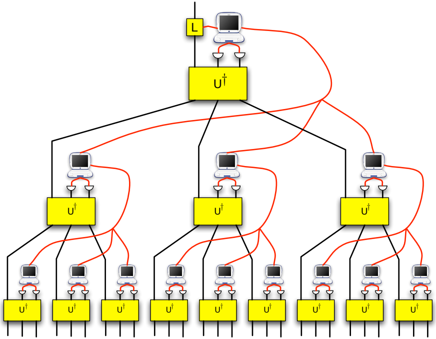
Surface code

- The surface code logical basis states $|\bar{\psi}\rangle = |\bar{0}\rangle$, $|\bar{1}\rangle$, $|\bar{+}\rangle$, or $|\bar{-}\rangle$ are PEPS and are complete for logical process tomography.
- $\rho = \mathcal{E}(|\bar{\psi}\rangle\langle\bar{\psi}|)$ is a PEPO for any PEPSO CPTP map $\mathcal{E}$.
- Computing the probability of a syndrome bit $\frac{1}{2}(1 \pm \mathrm{Tr}[\rho S_j])$ boils down to contracting a 2D TN.
- The post-measurement state $\rho' \propto (1 \pm S_j)\rho(1 \pm S_j)$ is also a PEPO.
- Applying a Pauli correction operator to ρ' preserves the PEPO structure.
- Computing the logical probabilities $\mathrm{Tr}(\rho\bar{L})$ required for process tomography boils down to contracting a 2D TN.

While efficient, approximate contraction schemes are known, we choose to contract exactly, so the scaling is exponential with the (least) linear size of the lattice, e.g. $W \times L$ lattice, Le^W vs e^{LW}.

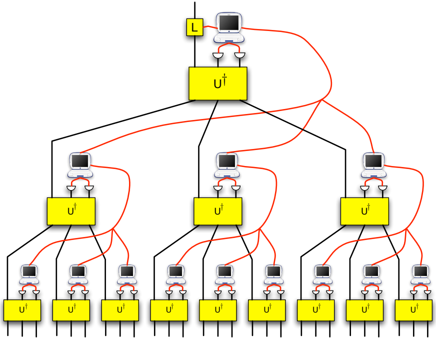
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise
  - MPS noise
  - etc.



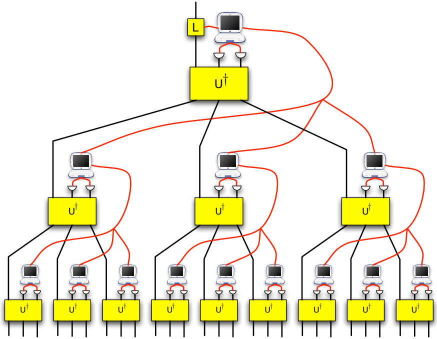
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



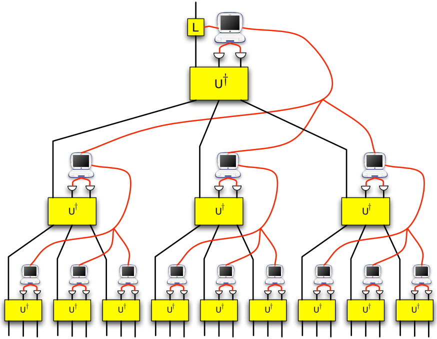
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



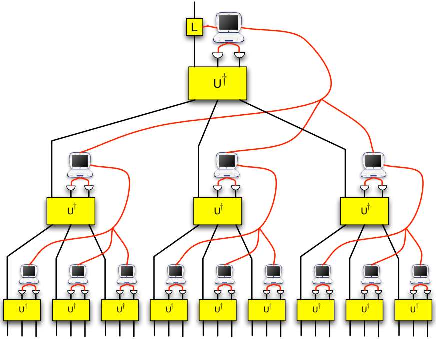
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



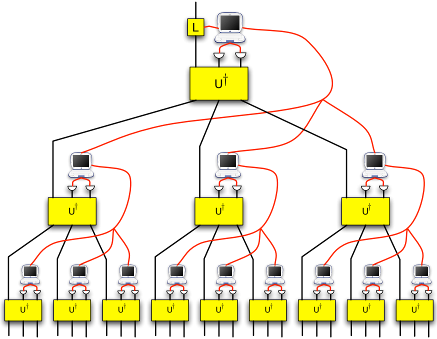
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



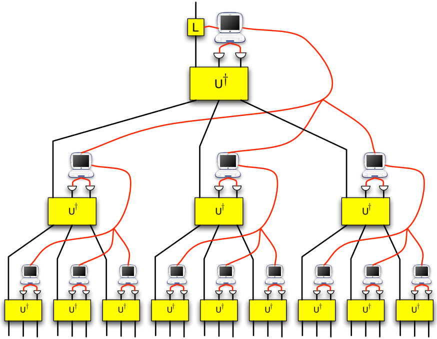
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



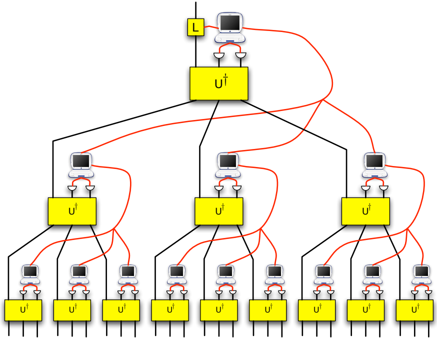
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
    - Uncorrelated noise.
    - Tree-like TN noise.
    - MPS noise.
    - etc.



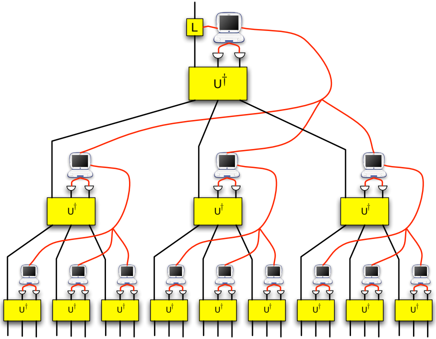
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



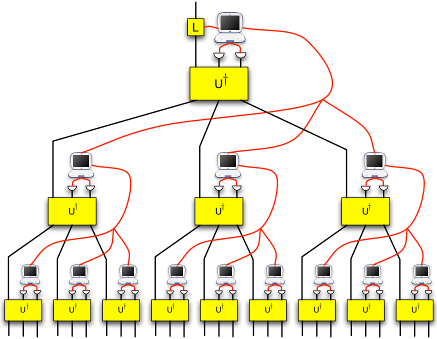
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
    - Uncorrelated noise.
    - Tree-like TN noise.
    - MPS noise.
    - etc.



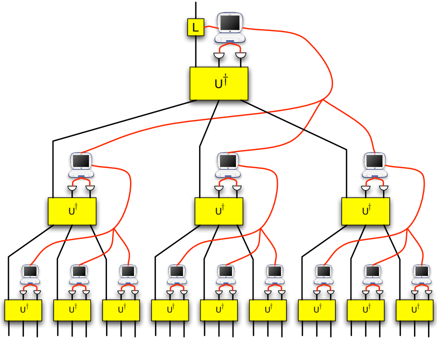
Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



Concatenated codes

- Use brute force simulate a finite block with physical noise model $\mathcal{E}$.
- Yields a logical channel conditioned on syndrome $\mathcal{E}_s^L$.
- Use $\mathcal{E}_s^L$ as input to finite block simulation.
- Compatible with
  - Uncorrelated noise.
  - Tree-like TN noise.
  - MPS noise.
  - etc.



Outline

Does this question even make sense?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.
  - Hilbert-Schmidt norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_2$ is easy to manipulate.
  - etc.
  - These metrics can differ significantly, e.g., ϵ vs $\sqrt{\epsilon}$.

Does this question even make sense?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.
  - Hilbert-Schmidt norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_2$ is easy to manipulate.
  - etc.
  - These metrics can differ significantly, e.g., ϵ vs $\sqrt{\epsilon}$.

Does this question even make sense?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.
  - Hilbert-Schmidt norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_2$ is easy to manipulate.
  - etc.
  - These metrics can differ significantly, e.g., ϵ vs $\sqrt{\epsilon}$.

Does this question even make sense?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.
  - Hilbert-Schmidt norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_2$ is easy to manipulate.
  - etc.
  - These metrics can differ significantly, e.g., ϵ vs $\sqrt{\epsilon}$.

Does this question even make sense?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.
  - Hilbert-Schmidt norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_2$ is easy to manipulate.
  - etc.
  - These metrics can differ significantly, e.g., ϵ vs $\sqrt{\epsilon}$.

Does this question even make sense?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.
  - Hilbert-Schmidt norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_2$ is easy to manipulate.
  - etc.
  - These metrics can differ significantly, e.g., ϵ vs $\sqrt{\epsilon}$.

Does this question even make sense?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.
  - Hilbert-Schmidt norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_2$ is easy to manipulate.
  - etc.
  - These metrics can differ significantly, e.g., ϵ vs $\sqrt{\epsilon}$.

Better approximation to the diamond norm

$\|\cdot\|_\diamond$ used for FT analysis and $1 - F$ accessed by randomized benchmarking differ significantly.

- The diamond norm could always be extracted from full process tomography of the channel.
- Can we combine randomized benchmarking results with other easily accessible quantities to estimate diamond norm?
- Yes, e.g., unitarity can provide tighter bounds.

Kueng, Long, Doherty, & Flammia arXiv:1510.05653

A useful norm is one which would enables us to predict the logical failure rate after error correction.

Better approximation to the diamond norm

$\|\cdot\|_\diamond$ used for FT analysis and $1-F$ accessed by randomized benchmarking differ significantly.

- The diamond norm could always be extracted from full process tomography of the channel.
- Can we combine randomized benchmarking results with other easily accessible quantities to estimate diamond norm?
- Yes, e.g., unitarity can provide tighter bounds.

Kueng, Long, Doherty, & Flammia arXiv:1510.05653

A useful norm is one which would enables us to predict the logical failure rate after error correction.

Better approximation to the diamond norm

$\|\cdot\|_\diamond$ used for FT analysis and $1-F$ accessed by randomized benchmarking differ significantly.

- The diamond norm could always be extracted from full process tomography of the channel.
- Can we combine randomized benchmarking results with other easily accessible quantities to estimate diamond norm?
- Yes, e.g., unitarity can provide tighter bounds.

Kueng, Long, Doherty, & Flammia arXiv:1510.05653

A useful norm is one which would enables us to predict the logical failure rate after error correction.

Better approximation to the diamond norm

$\| \cdot \|_\diamond$ used for FT analysis and $1 - F$ accessed by randomized benchmarking differ significantly.

- The diamond norm could always be extracted from full process tomography of the channel.
- Can we combine randomized benchmarking results with other easily accessible quantities to estimate diamond norm?
- Yes, e.g., unitarity can provide tighter bounds.

Kueng, Long, Doherty, & Flammia arXiv:1510.05653

A useful norm is one which would enables us to predict the logical failure rate after error correction.

Better approximation to the diamond norm

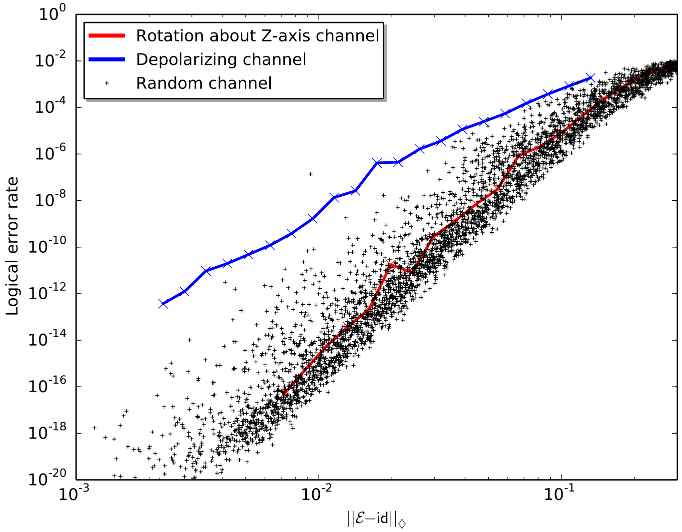
$\|\cdot\|_\diamond$ used for FT analysis and $1 - F$ accessed by randomized benchmarking differ significantly.

- The diamond norm could always be extracted from full process tomography of the channel.
- Can we combine randomized benchmarking results with other easily accessible quantities to estimate diamond norm?
- Yes, e.g., unitarity can provide tighter bounds.

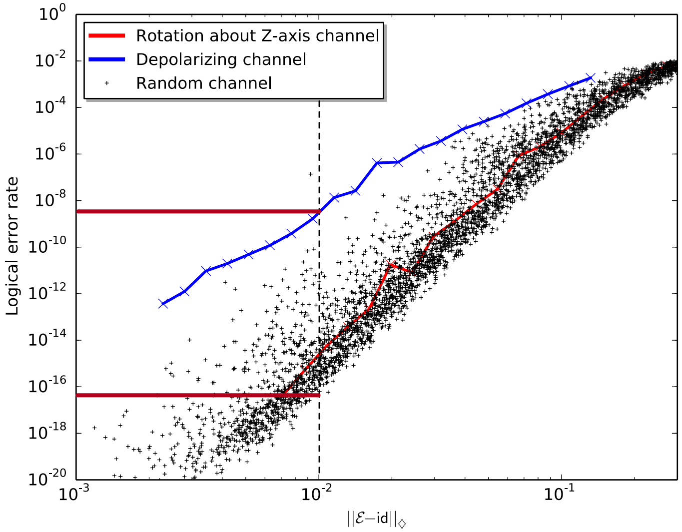
Kueng, Long, Doherty, & Flammia arXiv:1510.05653

A useful norm is one which would enables us to predict the logical failure rate after error correction.

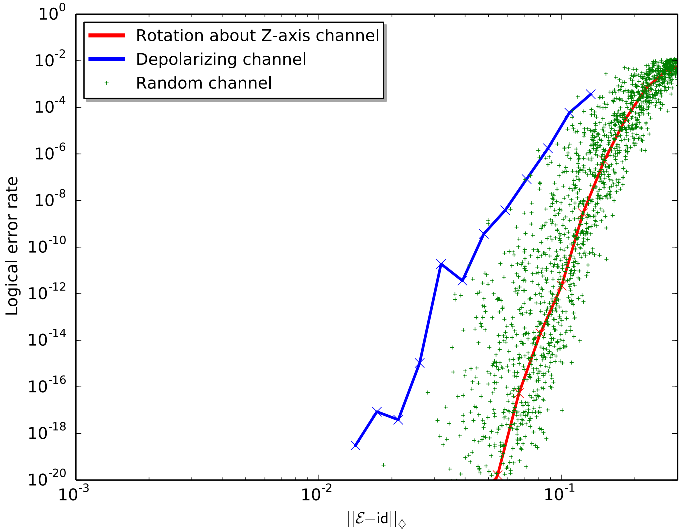
Predictability illustrated with Steane's code



Predictability illustrated with Steane's code



Predictability illustrated with Steane's code



Predictability of noise metrics

Conclusion

It is not possible to even very crudely predict the logical failure rate of a FT scheme given only the noise rate of the physical channel, as measured by any of the standard error metrics (Infidelity, Diamond norm, Channel entropy, Error probability, Euclidian norm, Trace norm).

- The Diamond norm does not stand out in any way.
- Incoherent noise is the worst, the opposite of widespread belief??
  - This is good news: most numerics done to date should be seen as worst case scenarios.
  - This statement is norm-dependent!!

Predictability of noise metrics

Conclusion

It is not possible to even very crudely predict the logical failure rate of a FT scheme given only the noise rate of the physical channel, as measured by any of the standard error metrics (Infidelity, Diamond norm, Channel entropy, Error probability, Euclidian norm, Trace norm).

- The Diamond norm does not stand out in any way.
- Incoherent noise is the worst, the opposite of widespread belief??
  - This is good news: most numerics done to date should be seen as worst case scenarios.
  - This statement is norm-dependent!!

Predictability of noise metrics

Conclusion

It is not possible to even very crudely predict the logical failure rate of a FT scheme given only the noise rate of the physical channel, as measured by any of the standard error metrics (Infidelity, Diamond norm, Channel entropy, Error probability, Euclidian norm, Trace norm).

- The Diamond norm does not stand out in any way.
- Incoherent noise is the worst, the opposite of widespread belief??
  - This is good news: most numerics done to date should be seen as worst case scenarios.
  - This statement is norm-dependent!!

Predictability of noise metrics

Conclusion

It is not possible to even very crudely predict the logical failure rate of a FT scheme given only the noise rate of the physical channel, as measured by any of the standard error metrics (Infidelity, Diamond norm, Channel entropy, Error probability, Euclidian norm, Trace norm).

- The Diamond norm does not stand out in any way.
- Incoherent noise is the worst, the opposite of widespread belief??
  - This is good news: most numerics done to date should be seen as worst case scenarios.
  - This statement is norm-dependent!!

Predictability of noise metrics

Conclusion

It is not possible to even very crudely predict the logical failure rate of a FT scheme given only the noise rate of the physical channel, as measured by any of the standard error metrics (Infidelity, Diamond norm, Channel entropy, Error probability, Euclidian norm, Trace norm).

- The Diamond norm does not stand out in any way.
- Incoherent noise is the worst, the opposite of widespread belief??
    - This is good news: most numerics done to date should be seen as worst case scenarios.
    - This statement is norm-dependent!!

Outline

How to predict the logical failure rate?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Analytical approach: stochastic adversarial noise model with rate ϵ should provide an upper bound.
  - Pros: Simple, widely applicable.
  - Cons: very loose bounds.
- Numerical approach: Monte Carlo simulation of FT scheme with depolarizing noise of rate ϵ.
  - Pros: More accurate estimates
  - Cons: Simulate 4096 qubits?!?

How to predict the logical failure rate?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Analytical approach: stochastic adversarial noise model with rate ϵ should provide an upper bound.
  - Pros: Simple, widely applicable.
  - Cons: very loose bounds.
- Numerical approach: Monte Carlo simulation of FT scheme with depolarizing noise of rate ϵ.
  - Pros: More accurate estimates
  - Cons: Simulate 4096 qubits?!?

How to predict the logical failure rate?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Analytical approach: stochastic adversarial noise model with rate ϵ should provide an upper bound.
  - Pros: Simple, widely applicable.
  - Cons: very loose bounds.
- Numerical approach: Monte Carlo simulation of FT scheme with depolarizing noise of rate ϵ.
  - Pros: More accurate estimates
  - Cons: Simulate 4096 qubits?!?

How to predict the logical failure rate?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Analytical approach: stochastic adversarial noise model with rate ϵ should provide an upper bound.
  - Pros: Simple, widely applicable.
  - Cons: very loose bounds.
- Numerical approach: Monte Carlo simulation of FT scheme with depolarizing noise of rate ϵ.
  - Pros: More accurate estimates
  - Cons: Simulate 4096 qubits?!?

How to predict the logical failure rate?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Analytical approach: stochastic adversarial noise model with rate ϵ should provide an upper bound.
  - Pros: Simple, widely applicable.
  - Cons: very loose bounds.
- Numerical approach: Monte Carlo simulation of FT scheme with depolarizing noise of rate ϵ.
  - Pros: More accurate estimates
  - Cons: Simulate 4096 qubits?!?

How to predict the logical failure rate?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Analytical approach: stochastic adversarial noise model with rate ϵ should provide an upper bound.
  - Pros: Simple, widely applicable.
  - Cons: very loose bounds.
- Numerical approach: Monte Carlo simulation of FT scheme with depolarizing noise of rate ϵ.
  - Pros: More accurate estimates
  - Cons: Simulate 4096 qubits?!?

How to predict the logical failure rate?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Analytical approach: stochastic adversarial noise model with rate ϵ should provide an upper bound.
  - Pros: Simple, widely applicable.
  - Cons: very loose bounds.
- Numerical approach: Monte Carlo simulation of FT scheme with depolarizing noise of rate ϵ.
  - Pros: More accurate estimates
  - Cons: Simulate 4096 qubits?!?

Clifford simulations

- Stabilizer codes use only Clifford operations $(X, Y, Z, H, S, \& CNOT)$.
- These can be simulated efficiently
  - For N qubits, need to manipulate $2N \times 2N$ matrices.
- If the noise consists of Clifford operations too, the whole process can be efficiently simulated.
  - In particular Pauli noise models (e.g. depolarizing).
  - These are the 'free-fermion' models of fault-tolerance.

Clifford simulations

- Stabilizer codes use only Clifford operations $(X, Y, Z, H, S, \& CNOT)$.
- These can be simulated efficiently
  - For N qubits, need to manipulate $2N \times 2N$ matrices.
- If the noise consists of Clifford operations too, the whole process can be efficiently simulated.
  - In particular Pauli noise models (e.g. depolarizing).
  - These are the 'free-fermion' models of fault-tolerance.

Clifford simulations

- Stabilizer codes use only Clifford operations $(X, Y, Z, H, S, \& CNOT)$.
- These can be simulated efficiently
  - For N qubits, need to manipulate $2N \times 2N$ matrices.
- If the noise consists of Clifford operations too, the whole process can be efficiently simulated.
  - In particular Pauli noise models (e.g. depolarizing).
  - These are the 'free-fermion' models of fault-tolerance.

Clifford simulations

- Stabilizer codes use only Clifford operations $(X, Y, Z, H, S, \& CNOT)$.
- These can be simulated efficiently
  - For N qubits, need to manipulate $2N \times 2N$ matrices.
- If the noise consists of Clifford operations too, the whole process can be efficiently simulated.
  - In particular Pauli noise models (e.g. depolarizing).
  - These are the 'free-fermion' models of fault-tolerance.

Clifford simulations

- Stabilizer codes use only Clifford operations ($X, Y, Z, H, S, \& CNOT$).
- These can be simulated efficiently
  - For N qubits, need to manipulate $2N \times 2N$ matrices.
- If the noise consists of Clifford operations too, the whole process can be efficiently simulated.
  - In particular Pauli noise models (e.g. depolarizing).
  - These are the 'free-fermion' models of fault-tolerance.

Clifford simulations

- Stabilizer codes use only Clifford operations $(X, Y, Z, H, S, \& CNOT)$.
- These can be simulated efficiently
  - For N qubits, need to manipulate $2N \times 2N$ matrices.
- If the noise consists of Clifford operations too, the whole process can be efficiently simulated.
  - In particular Pauli noise models (e.g. depolarizing).
  - These are the 'free-fermion' models of fault-tolerance.

Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.

Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.

Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.

Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.

Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.

Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.

- To probe $\delta \sim 10^{-10}$, need $M \sim 10^{10}$.
- Limited to unphysical noise models.

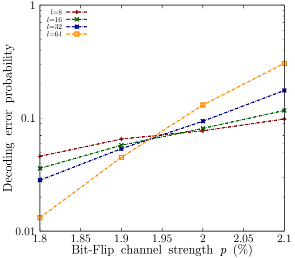
Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.



- To probe $\delta \sim 10^{-10}$, need $M \sim 10^{10}$.
- Limited to unphysical noise models.

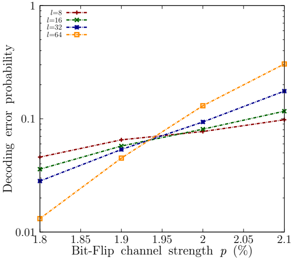
Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
   - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.



- To probe $\delta \sim 10^{-10}$, need $M \sim 10^{10}$.
- Limited to unphysical noise models.

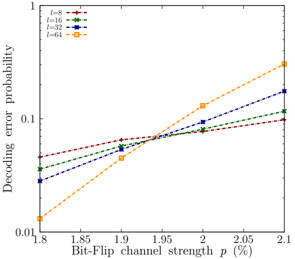
Monte Carlo simulations

FOR t=1:M

1. Assume the system is initially in a valid code state.
2. Execute a fault tolerant circuit on this encoded state
    - Sprinkle noise operations at a rate ϵ over the circuit.
3. Verify if the logical information has been altered by the noise.

END

Report logical failure rate $\delta =$ Number of reported failures/M.



- To probe $\delta \sim 10^{-10}$, need $M \sim 10^{10}$.
- Limited to unphysical noise models.

Low noise rate & non-Pauli noise

Our simulations methods are based on the idea of simulating the full dynamics $\mathcal{E}$ and sampling the syndromes, not the errors, just like in a real experiment.

- Thus, they are not limited to Pauli noise.
- When sampling Pauli errors, the correction is either right or wrong.
- When sampling syndromes, and simulating $\mathcal{E}$ entirely, there is always some residual logical error.
  - This is why we are able to report 10^{-18} logical error rates.
- We also used importance sampling methods to decrease statistical fluctuations:

$$\sum_s \mathrm{pr}(s) \|\mathcal{E}_s^L - \mathrm{id}\| = \sum_s q_s \left(\frac{\mathrm{pr}(s)}{q_s} \|\mathcal{E}_s^L - \mathrm{id}\| \right)$$

  - The intrinsic variance of the logical error is very large: outliers are dominant.

Low noise rate & non-Pauli noise

Our simulations methods are based on the idea of simulating the full dynamics $\mathcal{E}$ and sampling the syndromes, not the errors, just like in a real experiment.

- Thus, they are not limited to Pauli noise.
- When sampling Pauli errors, the correction is either right or wrong.
- When sampling syndromes, and simulating $\mathcal{E}$ entirely, there is always some residual logical error.
    - This is why we are able to report 10^{-18} logical error rates.
- We also used importance sampling methods to decrease statistical fluctuations:

$$\sum_s \mathrm{pr}(s)\|\mathcal{E}_s^L - \mathrm{id}\| = \sum_s q_s\left(\frac{\mathrm{pr}(s)}{q_s}\|\mathcal{E}_s^L - \mathrm{id}\|\right)$$

    - The intrinsic variance of the logical error is very large: outliers are dominant.

Low noise rate & non-Pauli noise

Our simulations methods are based on the idea of simulating the full dynamics $\mathcal{E}$ and sampling the syndromes, not the errors, just like in a real experiment.

- Thus, they are not limited to Pauli noise.
- When sampling Pauli errors, the correction is either right or wrong.
- When sampling syndromes, and simulating $\mathcal{E}$ entirely, there is always some residual logical error.
  - This is why we are able to report 10^{-18} logical error rates.
- We also used importance sampling methods to decrease statistical fluctuations:

$$\sum_s \mathrm{pr}(s)\|\mathcal{E}_s^L - \mathrm{id}\| = \sum_s q_s \left(\frac{\mathrm{pr}(s)}{q_s}\|\mathcal{E}_s^L - \mathrm{id}\|\right)$$

  - The intrinsic variance of the logical error is very large: outliers are dominant.

Low noise rate & non-Pauli noise

Our simulations methods are based on the idea of simulating the full dynamics $\mathcal{E}$ and sampling the syndromes, not the errors, just like in a real experiment.

- Thus, they are not limited to Pauli noise.
- When sampling Pauli errors, the correction is either right or wrong.
- When sampling syndromes, and simulating $\mathcal{E}$ entirely, there is always some residual logical error.
    - This is why we are able to report 10^{-18} logical error rates.
- We also used importance sampling methods to decrease statistical fluctuations:

$$\sum_s \mathrm{pr}(s) \|\mathcal{E}_s^L - \mathrm{id}\| = \sum_s q_s \left(\frac{\mathrm{pr}(s)}{q_s} \|\mathcal{E}_s^L - \mathrm{id}\| \right)$$

    - The intrinsic variance of the logical error is very large: outliers are dominant.

Low noise rate & non-Pauli noise

Our simulations methods are based on the idea of simulating the full dynamics $\mathcal{E}$ and sampling the syndromes, not the errors, just like in a real experiment.

- Thus, they are not limited to Pauli noise.
- When sampling Pauli errors, the correction is either right or wrong.
- When sampling syndromes, and simulating $\mathcal{E}$ entirely, there is always some residual logical error.
  - This is why we are able to report 10^{-18} logical error rates.
- We also used importance sampling methods to decrease statistical fluctuations:

$$\sum_s \mathrm{pr}(s)\|\mathcal{E}_s^L - \mathrm{id}\| = \sum_s q_s \left(\frac{\mathrm{pr}(s)}{q_s} \|\mathcal{E}_s^L - \mathrm{id}\| \right)$$

  - The intrinsic variance of the logical error is very large: outliers are dominant.

Low noise rate & non-Pauli noise

Our simulations methods are based on the idea of simulating the full dynamics $\mathcal{E}$ and sampling the syndromes, not the errors, just like in a real experiment.

- Thus, they are not limited to Pauli noise.
- When sampling Pauli errors, the correction is either right or wrong.
- When sampling syndromes, and simulating $\mathcal{E}$ entirely, there is always some residual logical error.
  - This is why we are able to report 10^{-18} logical error rates.
- We also used importance sampling methods to decrease statistical fluctuations:

$$\sum_s \mathrm{pr}(s) \|\mathcal{E}_s^L - \mathrm{id}\| = \sum_s q_s \left(\frac{\mathrm{pr}(s)}{q_s} \|\mathcal{E}_s^L - \mathrm{id}\| \right)$$

  - The intrinsic variance of the logical error is very large: outliers are dominant.

Outline

Twirled approximation

MC simulations are limited to unphysical Pauli noise models.

- We need to approximate the physical channel $\mathcal{E}$ by a Pauli channel $\mathcal{P}$.
  - Ignore the non-Pauli contributions to the channel.
- E.g. Rotation $R_z(\theta) = e^{i\theta Z} = \cos\theta I + i\sin\theta Z$ error

$$\rho \to (\cos\theta I + i\sin\theta Z)\rho(\cos\theta I - i\sin\theta Z)$$

is approximated by a stochastic Z error (dephasing)

$$\rho \to \cos^2\theta\ \rho + \sin^2\theta\ Z\rho Z.$$

- This twirled, or 'tight-binding' approximation might be much nicer than the original model.

Twirled approximation

MC simulations are limited to unphysical Pauli noise models.

- We need to approximate the physical channel $\mathcal{E}$ by a Pauli channel $\mathcal{P}$.
  - Ignore the non-Pauli contributions to the channel.
- E.g. Rotation $R_z(\theta) = e^{i\theta Z} = \cos\theta I + i\sin\theta Z$ error

$$\rho \to (\cos\theta I + i\sin\theta Z)\rho(\cos\theta I - i\sin\theta Z)$$

is approximated by a stochastic Z error (dephazing)

$$\rho \to \cos^2\theta\ \rho + \sin^2\theta\ Z\rho Z.$$

- This twirled, or 'tight-binding' approximation might be much nicer than the original model.

Twirled approximation

MC simulations are limited to unphysical Pauli noise models.

- We need to approximate the physical channel $\mathcal{E}$ by a Pauli channel $\mathcal{P}$.
  - Ignore the non-Pauli contributions to the channel.
- E.g. Rotation $R_z(\theta) = e^{i\theta Z} = \cos\theta I + i\sin\theta Z$ error

$$\rho \to (\cos\theta I + i\sin\theta Z)\rho(\cos\theta I - i\sin\theta Z)$$

is approximated by a stochastic Z error (dephazing)

$$\rho \to \cos^2\theta\ \rho + \sin^2\theta\ Z\rho Z.$$

- This twirled, or 'tight-binding' approximation might be much nicer than the original model.

Twirled approximation

MC simulations are limited to unphysical Pauli noise models.

- We need to approximate the physical channel $\mathcal{E}$ by a Pauli channel $\mathcal{P}$.
    - Ignore the non-Pauli contributions to the channel.
- E.g. Rotation $R_z(\theta) = e^{i\theta Z} = \cos\theta I + i\sin\theta Z$ error

$$\rho \to (\cos\theta I + i\sin\theta Z)\rho(\cos\theta I - i\sin\theta Z)$$

is approximated by a stochastic Z error (dephazing)

$$\rho \to \cos^2\theta\ \rho + \sin^2\theta\ Z\rho Z.$$

- This twirled, or 'tight-binding' approximation might be much nicer than the original model.

Twirled approximation

MC simulations are limited to unphysical Pauli noise models.

- We need to approximate the physical channel $\mathcal{E}$ by a Pauli channel $\mathcal{P}$.
  - Ignore the non-Pauli contributions to the channel.
- E.g. Rotation $R_z(\theta) = e^{i\theta Z} = \cos\theta I + i\sin\theta Z$ error

$$\rho \to (\cos\theta I + i\sin\theta Z)\rho(\cos\theta I - i\sin\theta Z)$$

is approximated by a stochastic Z error (dephazing)

$$\rho \to \cos^2\theta\ \rho + \sin^2\theta\ Z\rho Z.$$

- This twirled, or 'tight-binding' approximation might be much nicer than the original model.

Twirled approximation

MC simulations are limited to unphysical Pauli noise models.

- We need to approximate the physical channel $\mathcal{E}$ by a Pauli channel $\mathcal{P}$.
  - Ignore the non-Pauli contributions to the channel.
- E.g. Rotation $R_z(\theta) = e^{i\theta Z} = \cos\theta I + i\sin\theta Z$ error

$$\rho \rightarrow (\cos\theta I + i\sin\theta Z)\rho(\cos\theta I - i\sin\theta Z)$$

is approximated by a stochastic Z error (dephazing)

$$\rho \rightarrow \cos^2\theta\ \rho + \sin^2\theta\ Z\rho Z.$$

- This twirled, or 'tight-binding' approximation might be much nicer than the original model.

Twirled approximation

MC simulations are limited to unphysical Pauli noise models.

- We need to approximate the physical channel $\mathcal{E}$ by a Pauli channel $\mathcal{P}$.
  - Ignore the non-Pauli contributions to the channel.
- E.g. Rotation $R_z(\theta) = e^{i\theta Z} = \cos\theta I + i\sin\theta Z$ error

$$\rho \rightarrow (\cos\theta I + i\sin\theta Z)\rho(\cos\theta I - i\sin\theta Z)$$

is approximated by a stochastic Z error (dephazing)

$$\rho \rightarrow \cos^2\theta\ \rho + \sin^2\theta\ Z\rho Z.$$

- This twirled, or 'tight-binding' approximation might be much nicer than the original model.

Honest Pauli approximation

MC simulations are limited to unphysical Pauli noise models.

- Find Pauli channel $\mathcal{P}$ that approximates physical channel $\mathcal{E}$ well, yet don't outperform it.
- This is a simple optimization problem

$$\arg\min_{\mathcal{P}} \|\mathcal{E}-\mathcal{P}\|_\diamond \quad \text{such that} \quad \|\mathcal{E}(\rho)-\mathcal{I}(\rho)\|_1 \leq \|\mathcal{P}(\rho)-\mathcal{I}(\rho)\|_1 \quad \forall\rho.$$

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

- Numerical simulations with $\mathcal{P}$ are efficient (Clifford).
- In principle the idea generalizes to, e.g.
  - to Clifford noise — Janardan, Tomita, Gutierrez & Brown arXiv:1512.06284
  - to spatio-temporally correlated noise models, with a correlated Pauli or Clifford noise model.

Honest Pauli approximation

MC simulations are limited to unphysical Pauli noise models.

- Find Pauli channel $\mathcal{P}$ that approximates physical channel $\mathcal{E}$ well, yet don't outperform it.
- This is a simple optimization problem

$$\arg\min_{\mathcal{P}} \|\mathcal{E}-\mathcal{P}\|_\diamond \quad \text{such that} \quad \|\mathcal{E}(\rho)-\mathcal{I}(\rho)\|_1 \le \|\mathcal{P}(\rho)-\mathcal{I}(\rho)\|_1 \quad \forall\rho.$$

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

- Numerical simulations with $\mathcal{P}$ are efficient (Clifford).
- In principle the idea generalizes to, e.g.
  - to Clifford noise — Janardan, Tomita, Gutierrez & Brown arXiv:1512.06284
  - to spatio-temporally correlated noise models, with a correlated Pauli or Clifford noise model.

Honest Pauli approximation

MC simulations are limited to unphysical Pauli noise models.

- Find Pauli channel $\mathcal{P}$ that approximates physical channel $\mathcal{E}$ well, yet don't outperform it.
- This is a simple optimization problem

$$\arg\min_{\mathcal{P}} \|\mathcal{E}-\mathcal{P}\|_\diamond \quad \text{such that} \quad \|\mathcal{E}(\rho)-\mathcal{I}(\rho)\|_1 \leq \|\mathcal{P}(\rho)-\mathcal{I}(\rho)\|_1 \quad \forall \rho.$$

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

- Numerical simulations with $\mathcal{P}$ are efficient (Clifford).
- In principle the idea generalizes to, e.g.
  - to Clifford noise — Janardan, Tomita, Gutierrez & Brown arXiv:1512.06284
  - to spatio-temporally correlated noise models, with a correlated Pauli or Clifford noise model.

Honest Pauli approximation

MC simulations are limited to unphysical Pauli noise models.

- Find Pauli channel $\mathcal{P}$ that approximates physical channel $\mathcal{E}$ well, yet don't outperform it.
- This is a simple optimization problem

$$\arg\min_{\mathcal{P}} \|\mathcal{E}-\mathcal{P}\|_\diamond \quad \text{such that} \quad \|\mathcal{E}(\rho)-\mathcal{I}(\rho)\|_1 \leq \|\mathcal{P}(\rho)-\mathcal{I}(\rho)\|_1 \quad \forall\rho.$$

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

- Numerical simulations with $\mathcal{P}$ are efficient (Clifford).
- In principle the idea generalizes to, e.g.
  - to Clifford noise — Janardan, Tomita, Gutierrez & Brown arXiv:1512.06284
  - to spatio-temporally correlated noise models, with a correlated Pauli or Clifford noise model.

Honest Pauli approximation

MC simulations are limited to unphysical Pauli noise models.

- Find Pauli channel $\mathcal{P}$ that approximates physical channel $\mathcal{E}$ well, yet don't outperform it.
- This is a simple optimization problem

$$\arg\min_{\mathcal{P}} \|\mathcal{E} - \mathcal{P}\|_\diamond \quad \text{such that} \quad \|\mathcal{E}(\rho) - \mathcal{I}(\rho)\|_1 \leq \|\mathcal{P}(\rho) - \mathcal{I}(\rho)\|_1 \quad \forall \rho.$$

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

- Numerical simulations with $\mathcal{P}$ are efficient (Clifford).
- In principle the idea generalizes to, e.g.
  - to Clifford noise — Janardan, Tomita, Gutierrez & Brown arXiv:1512.06284
  - to spatio-temporally correlated noise models, with a correlated Pauli or Clifford noise model.

Honest Pauli approximation

MC simulations are limited to unphysical Pauli noise models.

- Find Pauli channel $\mathcal{P}$ that approximates physical channel $\mathcal{E}$ well, yet don't outperform it.
- This is a simple optimization problem

$$\arg\min_{\mathcal{P}} \|\mathcal{E}-\mathcal{P}\|_\diamond \quad \text{such that} \quad \|\mathcal{E}(\rho)-\mathcal{I}(\rho)\|_1 \leq \|\mathcal{P}(\rho)-\mathcal{I}(\rho)\|_1 \quad \forall\rho.$$

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

- Numerical simulations with $\mathcal{P}$ are efficient (Clifford).
- In principle the idea generalizes to, e.g.
  - to Clifford noise — Janardan, Tomita, Gutierrez & Brown arXiv:1512.06284
  - to spatio-temporally correlated noise models, with a correlated Pauli or Clifford noise model.

Honest Pauli approximation

MC simulations are limited to unphysical Pauli noise models.

- Find Pauli channel $\mathcal{P}$ that approximates physical channel $\mathcal{E}$ well, yet don't outperform it.
- This is a simple optimization problem

$$\arg\min_{\mathcal{P}} \|\mathcal{E}-\mathcal{P}\|_\diamond \quad \text{such that} \quad \|\mathcal{E}(\rho)-\mathcal{I}(\rho)\|_1 \leq \|\mathcal{P}(\rho)-\mathcal{I}(\rho)\|_1 \quad \forall \rho.$$

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

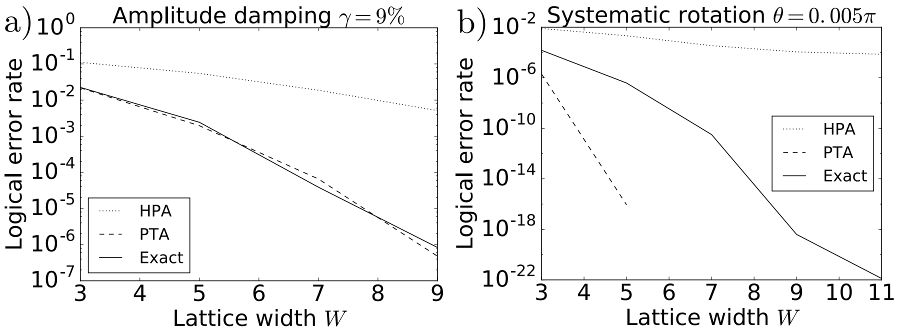
- Numerical simulations with $\mathcal{P}$ are efficient (Clifford).
- In principle the idea generalizes to, e.g.
  - to Clifford noise — Janardan, Tomita, Gutierrez & Brown arXiv:1512.06284
  - to spatio-temporally correlated noise models, with a correlated Pauli or Clifford noise model.

Pauli approximations, surface code threshold

	Exact	Pauli twirl	Honest Pauli
Depolarising (ϵ)	$18.5 \pm 1.5\%$	$18.5 \pm 1.5\%$	$18.5 \pm 1.5\%$
Damping (γ)	$39 \pm 2\%$	$39 \pm 2\%$	$6 \pm 1\%$
Z-rotation (θ)	$> 0.40\pi$	0.34π	0.11π

- Depolarizing, lattice up to size $9 \times 9 = 81$ qubits.
- Amplitude damping, lattice up to size $9 \times 17 = 153$ qubits.
- Depolarizing, lattice up to size $11 \times 11 = 121$ qubits.

Pauli approximations, surface code overhead



Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the z-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the z-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the z-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the z-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the z-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the *z*-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the z-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the *z*-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the *z*-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Usefulness of Pauli approximations

Conclusions

- It is not possible to even very crudely predict the logical failure rate of a FT scheme from known Pauli approximations.
- The twirl approximation gets a good threshold estimate in the examples we looked at.

- For the amplitude damping channel:
  - the twirled gave a good prediction,
  - the honest Pauli approximation was grossly overestimating the logical failure.
- For the *z*-rotation channel:
  - both approximations are totally off.

It is essential to develop simulation methods adapted to non-Pauli noise models to get a reliable estimate of the FT threshold.

Outline

Pauli approximations for the purpose of decoding

There are two levels of difficulty: decoding and simulating.

- Even for Pauli noise, decoding is in general a hard problem, but there are efficient algorithms for some classes of codes.
- For non-Pauli noise, the problem becomes even harder.
- For decoding purposes, one could assume one of the Pauli approximations.
  - Suboptimal but efficient.
- Same comments apply to correlated noise models (either Pauli or not).

Our simulations methods, combined to efficient (approximate) contraction schemes of TN provide efficient decoders for a wide variety of non-Pauli and/or correlated noise models.

Pauli approximations for the purpose of decoding

There are two levels of difficulty: decoding and simulating.

- Even for Pauli noise, decoding is in general a hard problem, but there are efficient algorithms for some classes of codes.
- For non-Pauli noise, the problem becomes even harder.
- For decoding purposes, one could assume one of the Pauli approximations.
  - Suboptimal but efficient.
- Same comments apply to correlated noise models (either Pauli or not).

Our simulations methods, combined to efficient (approximate) contraction schemes of TN provide efficient decoders for a wide variety of non-Pauli and/or correlated noise models.

Pauli approximations for the purpose of decoding

There are two levels of difficulty: decoding and simulating.

- Even for Pauli noise, decoding is in general a hard problem, but there are efficient algorithms for some classes of codes.
- For non-Pauli noise, the problem becomes even harder.
- For decoding purposes, one could assume one of the Pauli approximations.
  - Suboptimal but efficient.
- Same comments apply to correlated noise models (either Pauli or not).

Our simulations methods, combined to efficient (approximate) contraction schemes of TN provide efficient decoders for a wide variety of non-Pauli and/or correlated noise models.

Pauli approximations for the purpose of decoding

There are two levels of difficulty: decoding and simulating.

- Even for Pauli noise, decoding is in general a hard problem, but there are efficient algorithms for some classes of codes.
- For non-Pauli noise, the problem becomes even harder.
- For decoding purposes, one could assume one of the Pauli approximations.
  - Suboptimal but efficient.
- Same comments apply to correlated noise models (either Pauli or not).

Our simulations methods, combined to efficient (approximate) contraction schemes of TN provide efficient decoders for a wide variety of non-Pauli and/or correlated noise models.

Pauli approximations for the purpose of decoding

There are two levels of difficulty: decoding and simulating.

- Even for Pauli noise, decoding is in general a hard problem, but there are efficient algorithms for some classes of codes.
- For non-Pauli noise, the problem becomes even harder.
- For decoding purposes, one could assume one of the Pauli approximations.
  - Suboptimal but efficient.
- Same comments apply to correlated noise models (either Pauli or not).

Our simulations methods, combined to efficient (approximate) contraction schemes of TN provide efficient decoders for a wide variety of non-Pauli and/or correlated noise models.

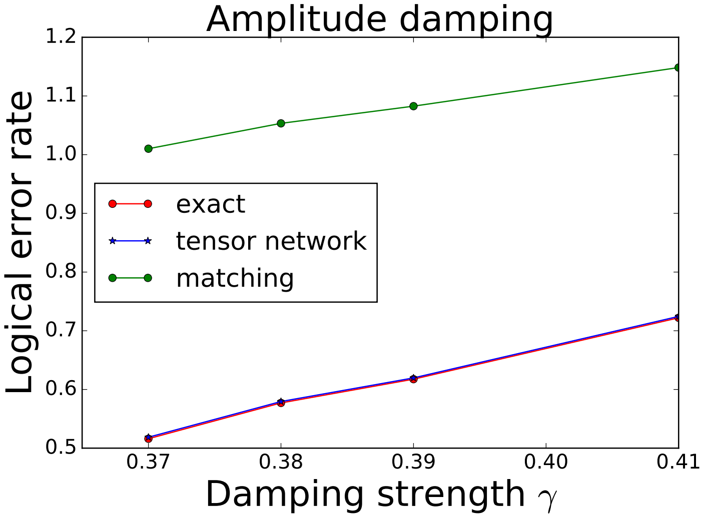
Pauli approximations for the purpose of decoding

There are two levels of difficulty: decoding and simulating.

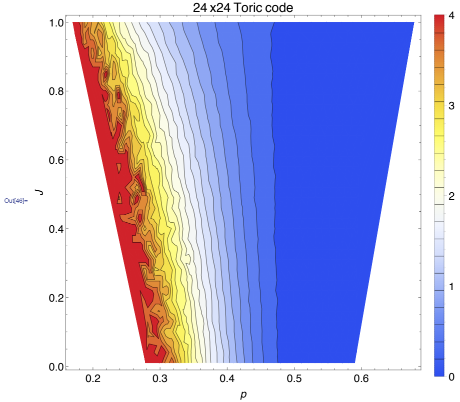
- Even for Pauli noise, decoding is in general a hard problem, but there are efficient algorithms for some classes of codes.
- For non-Pauli noise, the problem becomes even harder.
- For decoding purposes, one could assume one of the Pauli approximations.
  - Suboptimal but efficient.
- Same comments apply to correlated noise models (either Pauli or not).

Our simulations methods, combined to efficient (approximate) contraction schemes of TN provide efficient decoders for a wide variety of non-Pauli and/or correlated noise models.

Pauli approximations for surface code decoding



Correlated erasures on surface code



The erasure pattern is given by spin down configuration of a classical ferromagnetic Ising model in a magnetic field favoring spin ups.

Color shows
$-\log_{10}$(logical error rate)

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- We use methods from quantum many-body physics to address this question.
  - Surface code.
  - Concatenated code.
- We have found that:
  - for a fixed physical noise rate, as measured by any standard metric, the logical failure rate can fluctuate by several orders of magnitude.
  - the diamond norm is no better than other norms in that regard.
  - the logical failure rate of Pauli approximations can differ by several orders of magnitude.
  - for a given diamond norm physical error rate, incoherent noise is the worst.
  - the twirled approximation can give a reasonable threshold estimate.
- Also developed an efficient decoder for non-Pauli and correlated noise on surface code.

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Given a physical noise rate ϵ, how much error correction do I need to achieve a logical noise rate δ?

- Given an experimentally motivated noise model have the tools to predict its FT performances.
- These predictions can guide experimentalists:
  - If I changed this parameter, would it have a noticeable effect?

What features of a noise model are critical to FT QEC?

- We are using machine learning techniques, combined to our simulations, to find these parameters.
  - Do these features have an intuitive meaning?
  - Can these features be measured efficiently experimentally?

Institue Quantique @ Sherbrooke

Newly founded institute to conduct research at the frontier of quantum information and quantum materials, and to go from quantum science to quantum technologies.

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.

Institue Quantique @ Sherbrooke

Newly founded institute to conduct research at the frontier of quantum information and quantum materials, and to go from quantum science to quantum technologies.

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.

Institue Quantique @ Sherbrooke

Newly founded institute to conduct research at the frontier of quantum information and quantum materials, and to go from quantum science to quantum technologies.

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.

Institue Quantique @ Sherbrooke

Newly founded institute to conduct research at the frontier of quantum information and quantum materials, and to go from quantum science to quantum technologies.

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.

Institue Quantique @ Sherbrooke

Newly founded institute to conduct research at the frontier of quantum information and quantum materials, and to go from quantum science to quantum technologies.

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.

Institue Quantique @ Sherbrooke

Newly founded institute to conduct research at the frontier of quantum information and quantum materials, and to go from quantum science to quantum technologies.

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.

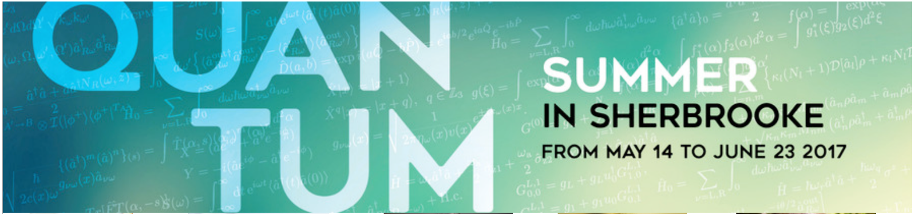
Institue Quantique @ Sherbrooke

Newly founded institute to conduct research at the frontier of quantum information and quantum materials, and to go from quantum science to quantum technologies.

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.




Glen Evenbly
Sherbrooke


Andreas Wallraff
ETH Zurich


Stephanie Simmons
Simon Fraser


Jason Petta
Princeton


David Reilly
Sydney

Stephen Bartlett
Sydney

Sergey Bravyi
IBM

Matthias Troyes
Microsoft

Gilles Brassard
Montréal

Krysta Svore
Microsoft