Numerical Simulations of Quantum Error Correction

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

Institut Quantique & Département de Physique
Université de Sherbrooke

Quantum Computing Theory in Practice
Bristol, UK, April 2019

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 $\mathrm{polylog}(\frac{1}{\delta})$.

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?

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

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

Pauli noise

- Noise modeled by some (perhaps correlated) probability distribution $P(E)$, over $E \in \mathcal{P}^{\otimes N}$.
- To numerically simulate:
  - Sample $E \sim P(E)$ (only randomness).
  - Compute associated syndrome $s(E)$.
  - Decode: guess $\hat{E}$ from s.
  - Check if $\hat{E} \simeq E$ (up to stabilizer).
  - Repeat to estimate logical error probability.
- For generalized noise models (think of systematic error $U^{\otimes N}$):
  - Errors are not element of the Pauli group: CPTP map.
  - Syndrome is not determined by error: Born's rule.
  - Correction is not right or wrong: some fidelity.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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.

Stuff in red is numerically hard.

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 $\mathrm{pr}(s)$.
- 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 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 $\|\mathcal{E}_L - \mathrm{id}\|$
- 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.

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 $\sqrt{\epsilon}$.

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

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 $\sqrt{\epsilon}$.

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

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 $\sqrt{\epsilon}$.

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

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 $\sqrt{\epsilon}$.

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

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 $\sqrt{\epsilon}$.

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

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 $\sqrt{\epsilon}$.

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

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 $\sqrt{\epsilon}$.

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

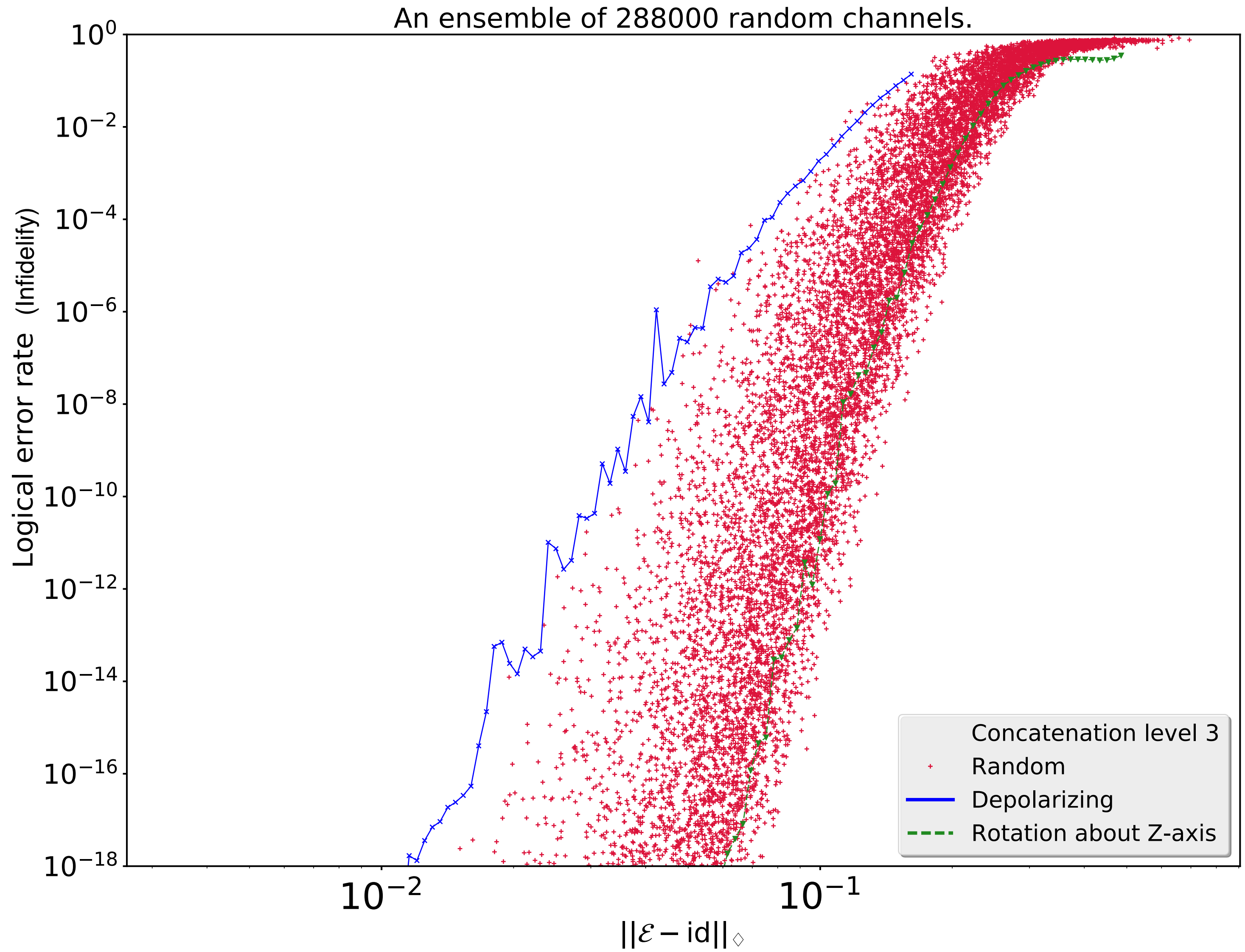
Does this question even make sense?

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

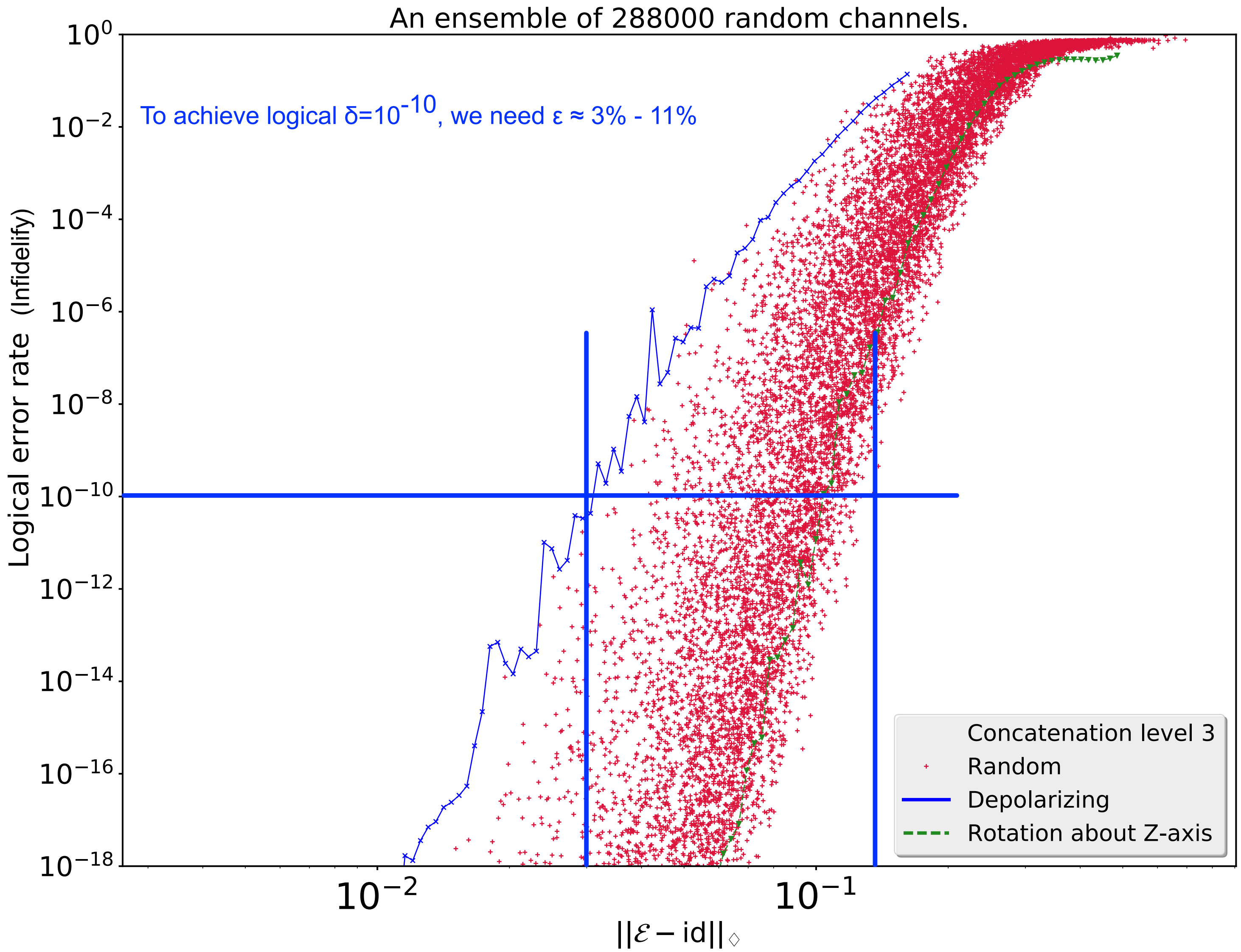
- How do I quantify the physical noise rate?
  - Infidelity $\epsilon = 1 - \int d\psi F[\psi, \mathcal{E}(\psi)]$ has a nice statistical interpretation, measured by randomized benchmarking.
  - Diamond norm $\epsilon = \|\mathcal{E} - \mathcal{I}\|_\diamond$ composes well, used in analytical FT studies.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

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

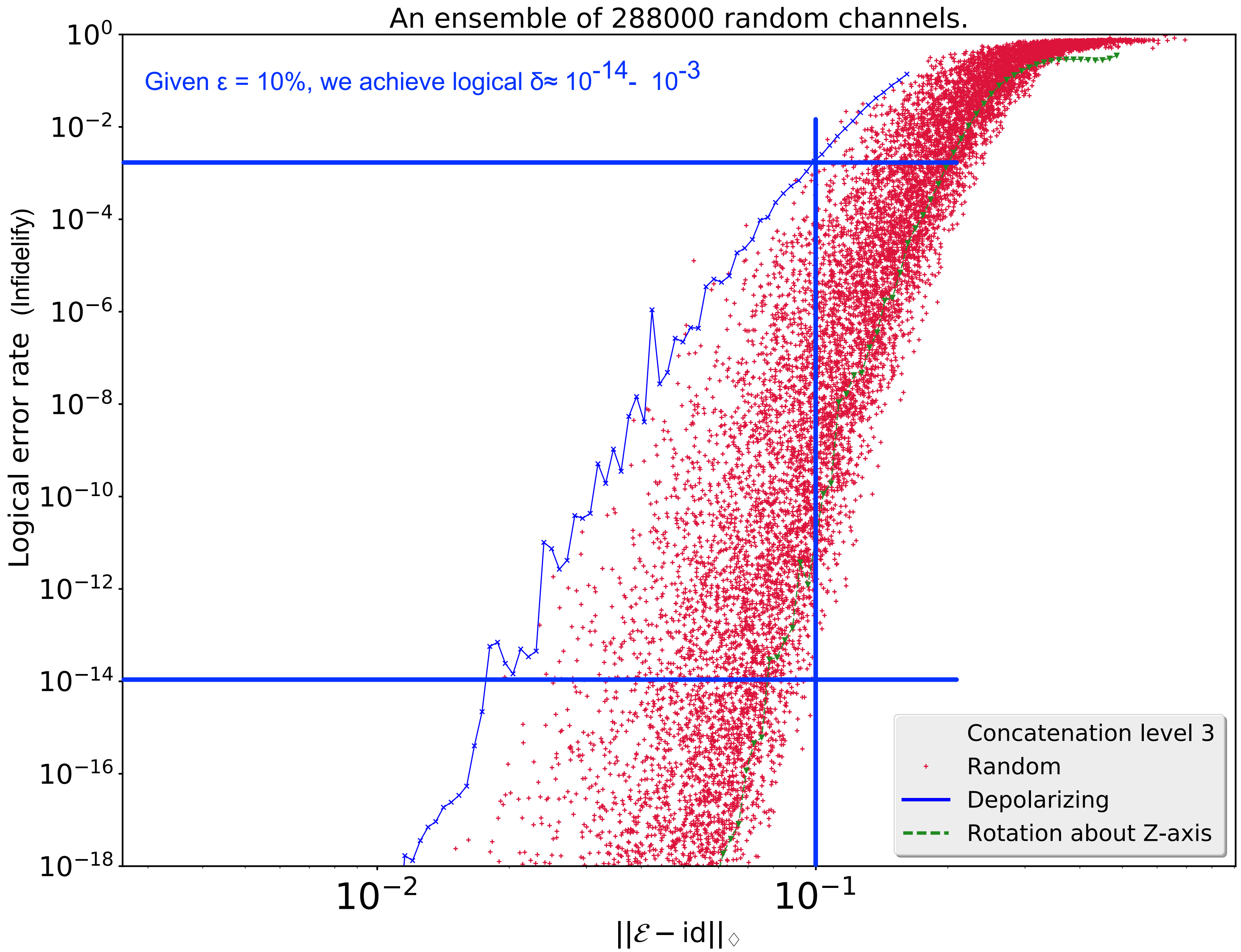
Predictability illustrated with Steane's code



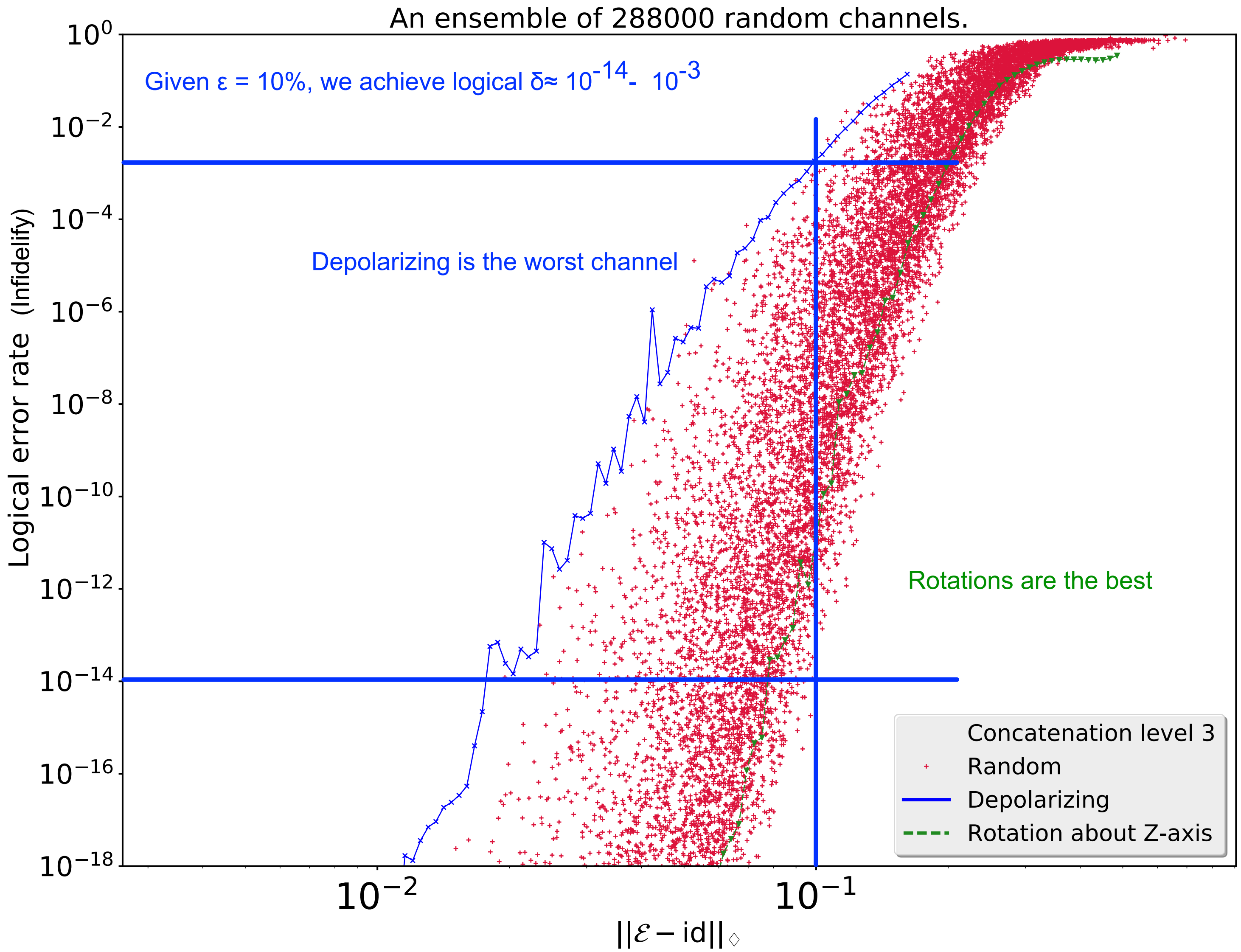
Predictability illustrated with Steane's code



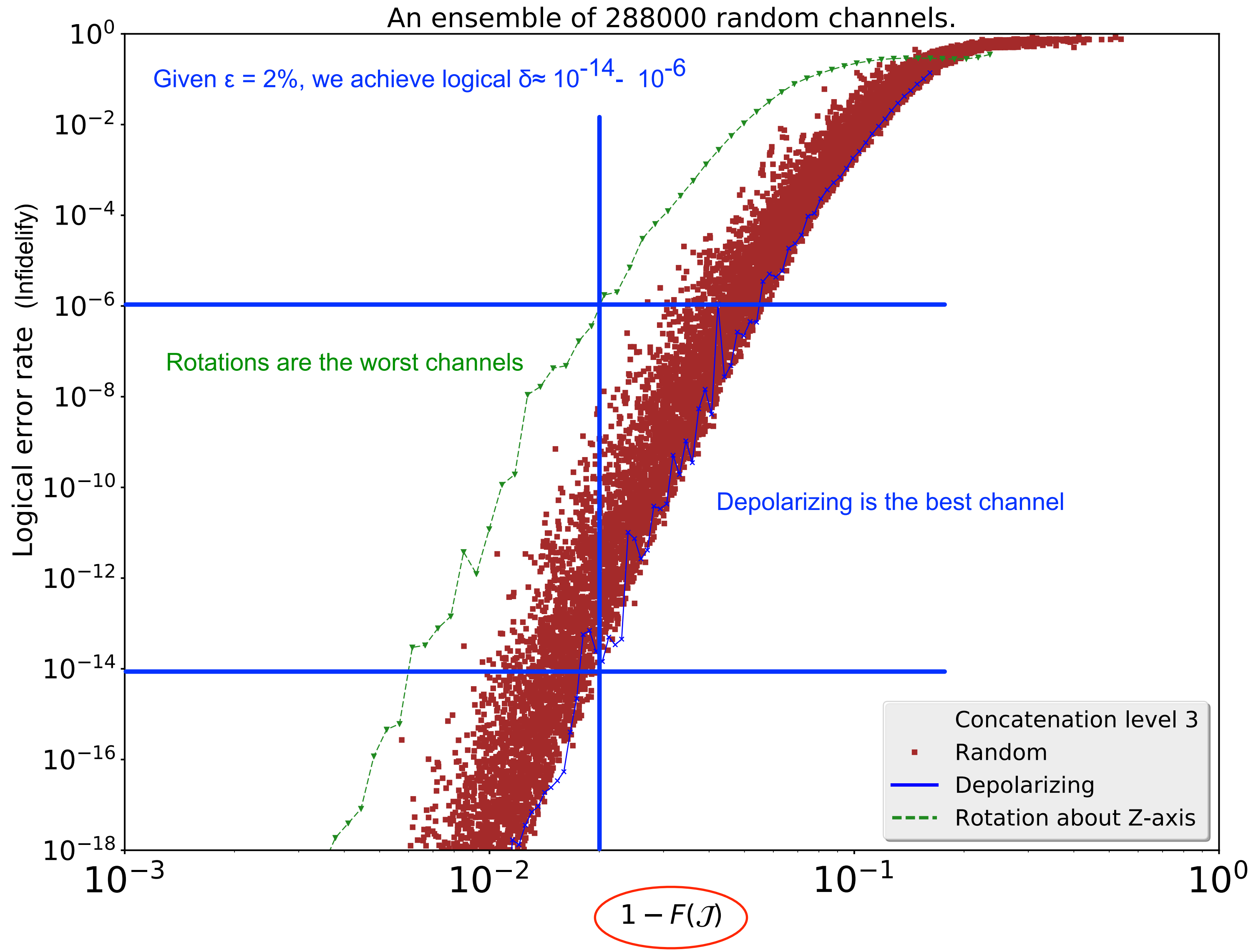
Predictability illustrated with Steane's code



Predictability 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).

Outline

Pauli approximations

MC simulations are numerically very efficient, but limited to unphysical Pauli noise models.

- Let's 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)

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

- Or we could find the closest Pauli channel which is no better than $\mathcal{E}$: Honest Pauli Approximation

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

Pauli approximations

MC simulations are numerically very efficient, but limited to unphysical Pauli noise models.

- Let's 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)

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

- Or we could find the closest Pauli channel which is no better than $\mathcal{E}$: Honest Pauli Approximation

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

Pauli approximations

MC simulations are numerically very efficient, but limited to unphysical Pauli noise models.

- Let's 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)

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

- Or we could find the closest Pauli channel which is no better than $\mathcal{E}$: Honest Pauli Approximation

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

Pauli approximations

MC simulations are numerically very efficient, but limited to unphysical Pauli noise models.

- Let's 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)

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

- Or we could find the closest Pauli channel which is no better than $\mathcal{E}$: Honest Pauli Approximation

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

Pauli approximations

MC simulations are numerically very efficient, but limited to unphysical Pauli noise models.

- Let's 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)

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

- Or we could find the closest Pauli channel which is no better than $\mathcal{E}$: Honest Pauli Approximation

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

Pauli approximations

MC simulations are numerically very efficient, but limited to unphysical Pauli noise models.

- Let's 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)

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

- Or we could find the closest Pauli channel which is no better than $\mathcal{E}$: Honest Pauli Approximation

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

Pauli approximations

MC simulations are numerically very efficient, but limited to unphysical Pauli noise models.

- Let's 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

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

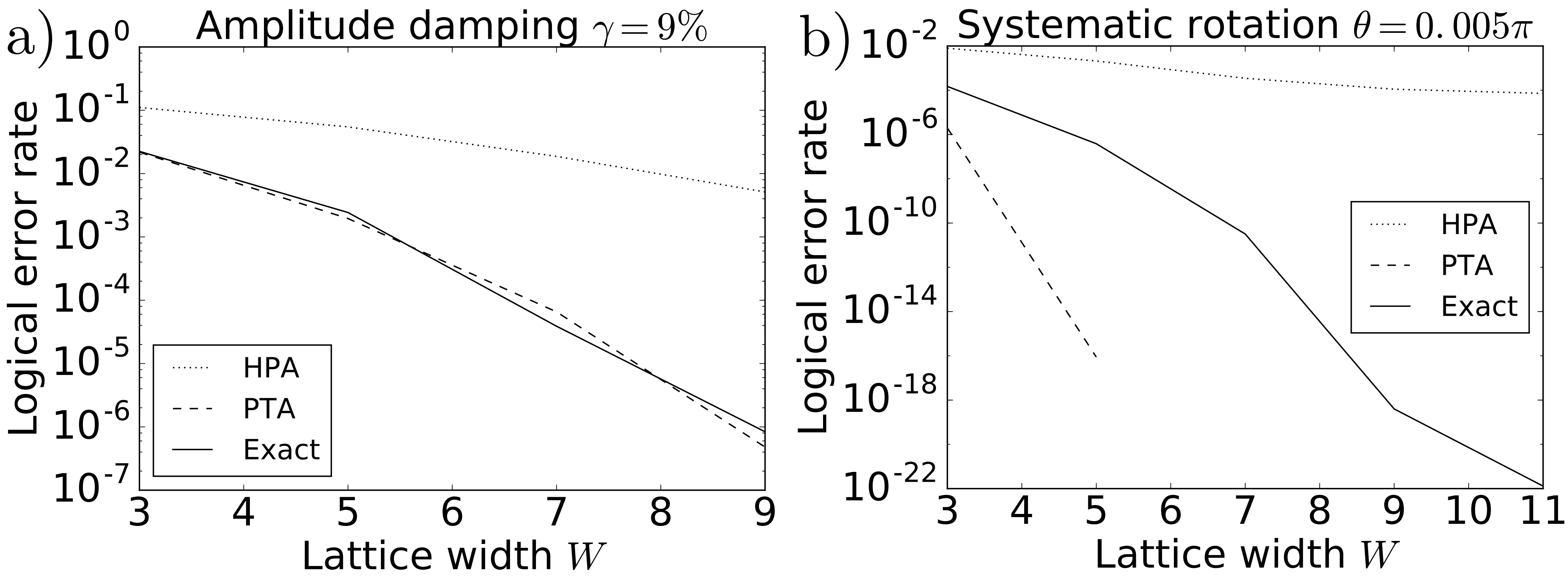
is approximated by a stochastic Z error (dephazing)

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

- Or we could find the closest Pauli channel which is no better than $\mathcal{E}$: Honest Pauli Approximation

Magasan, Puzzuoli, Granade, & Cory arXiv:1206.5407

Pauli approximations, surface code overhead



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

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

Outline

Decoding non-Pauli noise

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.

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.

Decoding non-Pauli noise

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.

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.

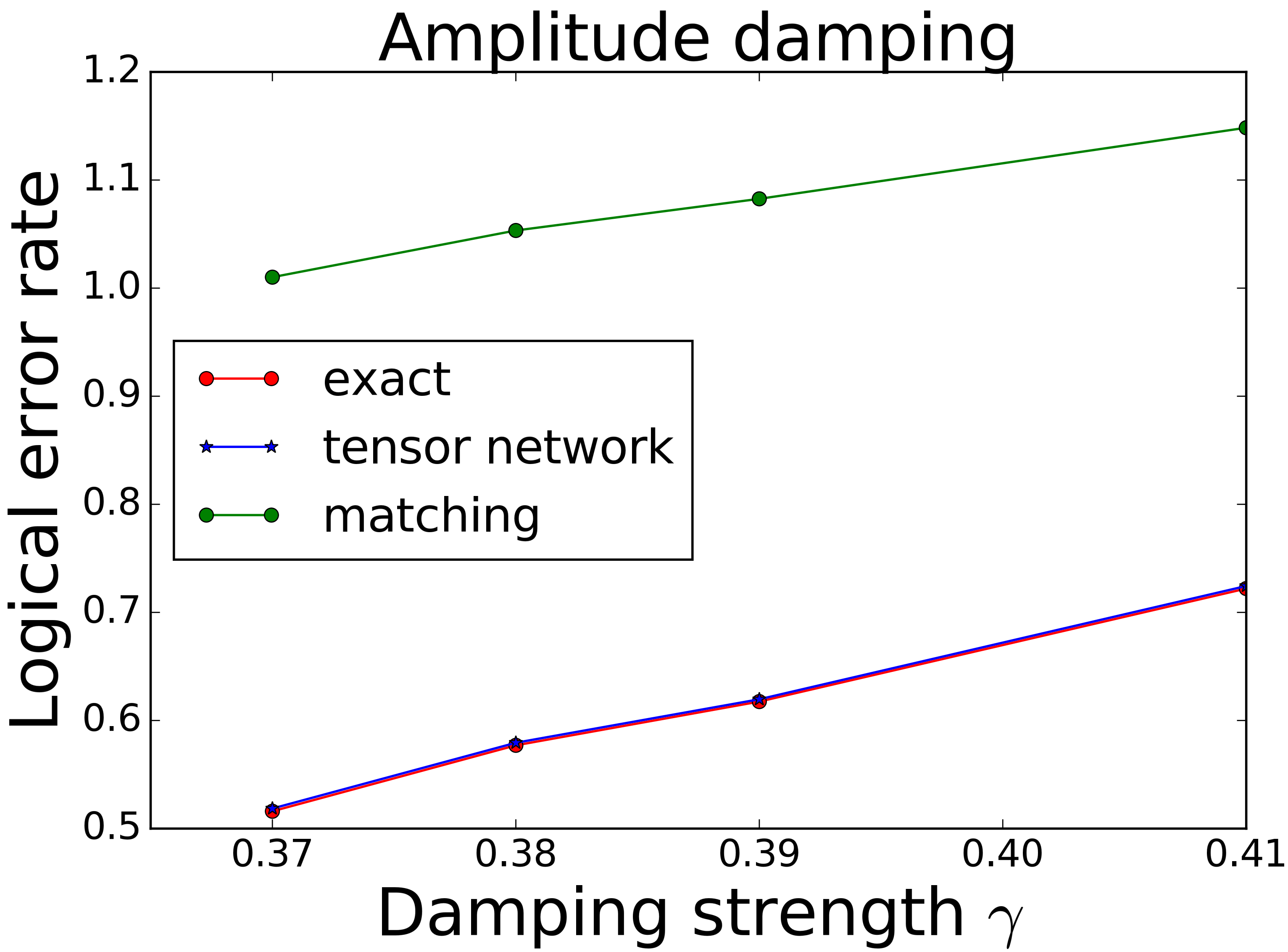
Decoding non-Pauli noise

There are two levels of difficulty: decoding and simulating.

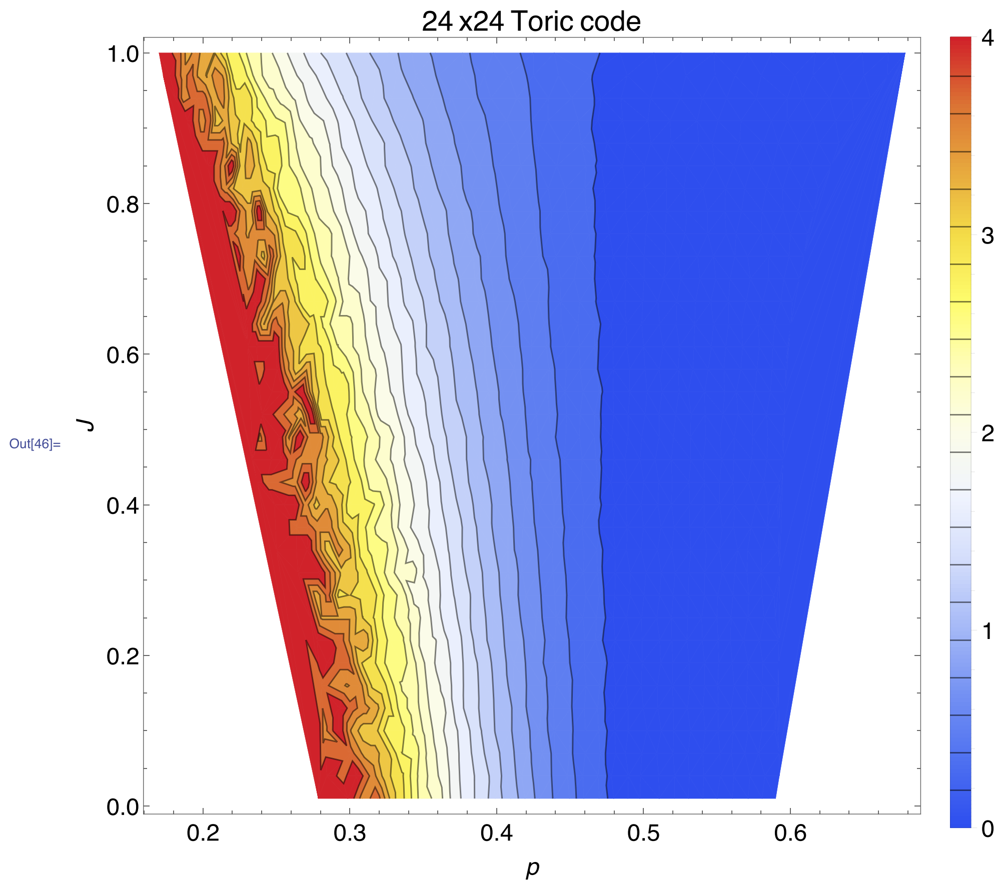
- Even for Pauli noise, decoding is in general a hard problem, but there are efficient algorithms for some classes of codes.
- For non-Pauli noise, the problem becomes even harder.

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

Known vs unknown inhomogeneity

- Different errors affect the performance of a fault tolerance scheme differently, so additional efforts should be assign to reduce the critical noise parameters.
- But for a given device, do I need to know the noise model?
- Knowing T_1 and T_2 is not important, but T_1/T_2 matters.
- What else?

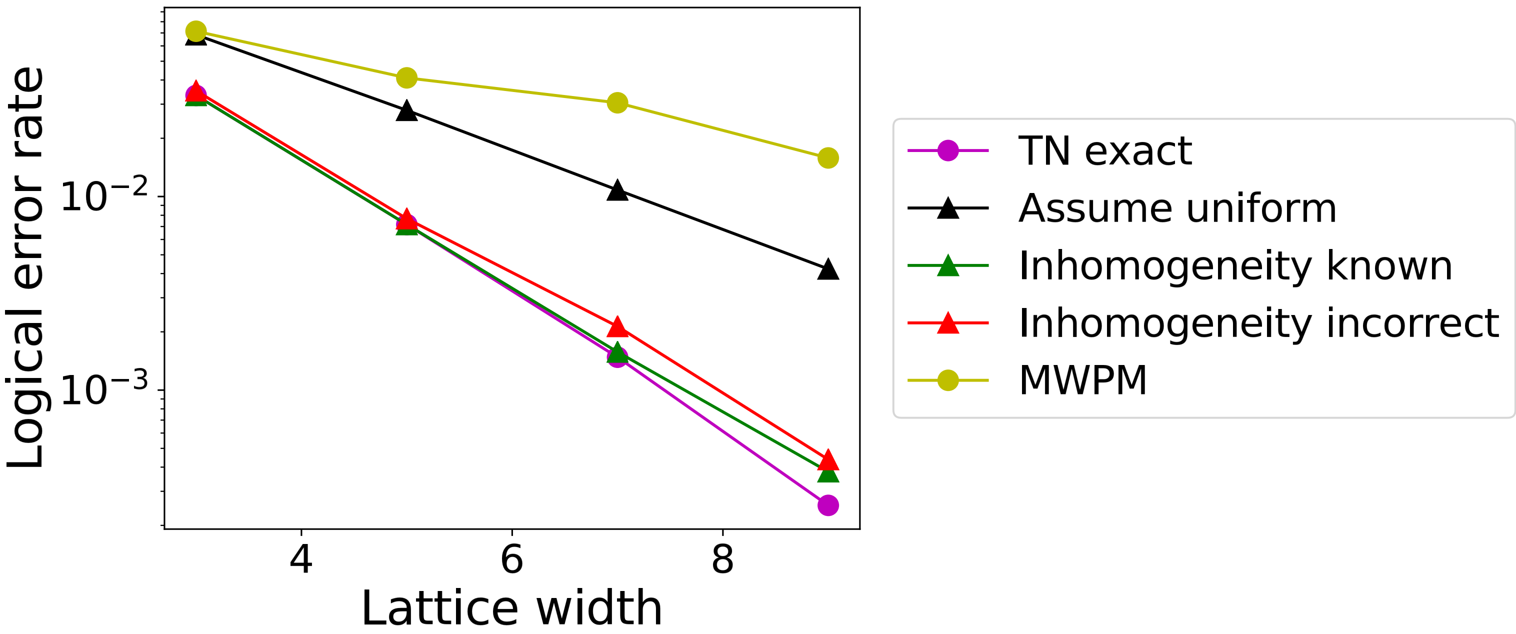
Known vs unknown inhomogeneity

- Different errors affect the performance of a fault tolerance scheme differently, so additional efforts should be assign to reduce the critical noise parameters.
- But for a given device, do I need to know the noise model?

- Knowing T_1 and T_2 is not important, but T_1/T_2 matters.
- What else?

Known vs unknown inhomogeneity

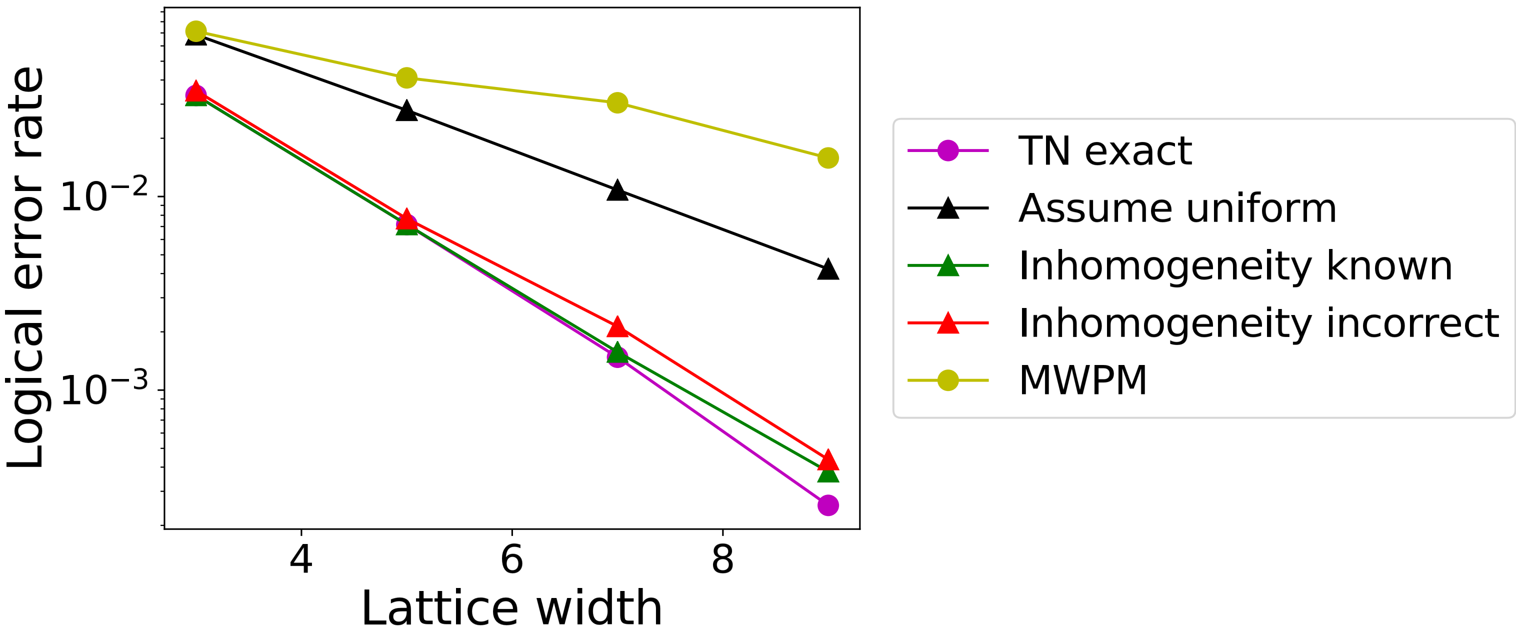
- Different errors affect the performance of a fault tolerance scheme differently, so additional efforts should be assign to reduce the critical noise parameters.
- But for a given device, do I need to know the noise model?



- Knowing T_1 and T_2 is not important, but T_1/T_2 matters.
- What else?

Known vs unknown inhomogeneity

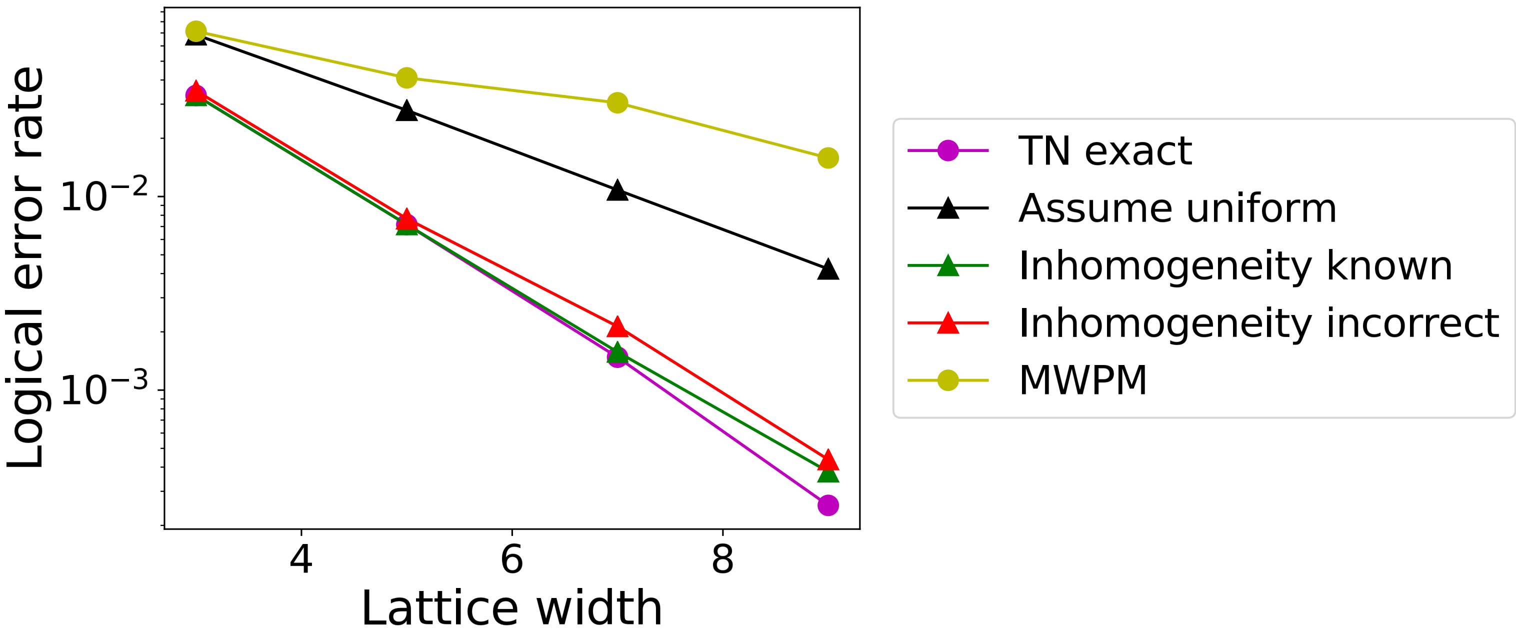
- Different errors affect the performance of a fault tolerance scheme differently, so additional efforts should be assign to reduce the critical noise parameters.
- But for a given device, do I need to know the noise model?



- Knowing T_1 and T_2 is not important, but T_1/T_2 matters.
- What else?

Known vs unknown inhomogeneity

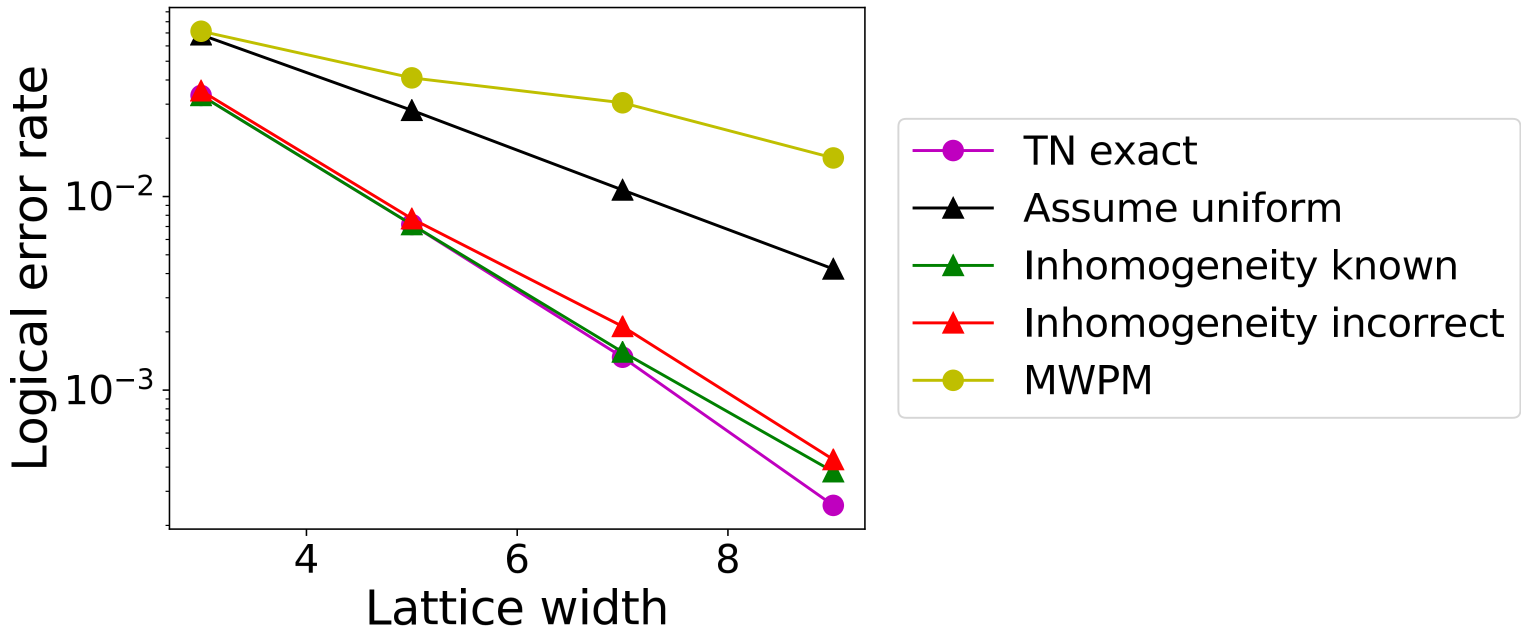
- Different errors affect the performance of a fault tolerance scheme differently, so additional efforts should be assign to reduce the critical noise parameters.
- But for a given device, do I need to know the noise model?



- Knowing T_1 and T_2 is not important, but T_1/T_2 matters.
- What else?

Known vs unknown inhomogeneity

- Different errors affect the performance of a fault tolerance scheme differently, so additional efforts should be assign to reduce the critical noise parameters.
- But for a given device, do I need to know the noise model?



- Knowing T_1 and T_2 is not important, but T_1/T_2 matters.
- What else?

Channel knowledge

Conclusions

- Incorporating knowledge of the noise model in the decoding process can be very beneficial.
- But an accurate channel description is an overkill: only a subset of the noise parameters matter.

Channel knowledge

Conclusions

- Incorporating knowledge of the noise model in the decoding process can be very beneficial.
- But an accurate channel description is an overkill: only a subset of the noise parameters matter.

Channel knowledge

Conclusions

- Incorporating knowledge of the noise model in the decoding process can be very beneficial.
- But an accurate channel description is an overkill: only a subset of the noise parameters matter.

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.
- Reporting the quality of a qubit with a single number is not useful to predict how it will respond in a fault-tolerant scheme.

Fault-tolerance critical parameters

In a given experiment, what is the leading noise source which limits the logical accuracy?

- Developed machine learning methods to identify such critical parameters.
- Decoding non-Pauli noise with tensor networks.

Decoding critical parameters

What information must I know about the device to decode it optimally?

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.
- Reporting the quality of a qubit with a single number is not useful to predict how it will respond in a fault-tolerant scheme.

Fault-tolerance critical parameters

In a given experiment, what is the leading noise source which limits the logical accuracy?

- Developed machine learning methods to identify such critical parameters.
- Decoding non-Pauli noise with tensor networks.

Decoding critical parameters

What information must I know about the device to decode it optimally?

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.
- Reporting the quality of a qubit with a single number is not useful to predict how it will respond in a fault-tolerant scheme.

Fault-tolerance critical parameters

In a given experiment, what is the leading noise source which limits the logical accuracy?

- Developed machine learning methods to identify such critical parameters.
- Decoding non-Pauli noise with tensor networks.

Decoding critical parameters

What information must I know about the device to decode it optimally?

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.
- Reporting the quality of a qubit with a single number is not useful to predict how it will respond in a fault-tolerant scheme.

Fault-tolerance critical parameters

In a given experiment, what is the leading noise source which limits the logical accuracy?

- Developed machine learning methods to identify such critical parameters.
- Decoding non-Pauli noise with tensor networks.

Decoding critical parameters

What information must I know about the device to decode it optimally?

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.
- Reporting the quality of a qubit with a single number is not useful to predict how it will respond in a fault-tolerant scheme.

Fault-tolerance critical parameters

In a given experiment, what is the leading noise source which limits the logical accuracy?

- Developed machine learning methods to identify such critical parameters.
- Decoding non-Pauli noise with tensor networks.

Decoding critical parameters

What information must I know about the device to decode it optimally?

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.
- Reporting the quality of a qubit with a single number is not useful to predict how it will respond in a fault-tolerant scheme.

Fault-tolerance critical parameters

In a given experiment, what is the leading noise source which limits the logical accuracy?

- Developed machine learning methods to identify such critical parameters.
- Decoding non-Pauli noise with tensor networks.

Decoding critical parameters

What information must I know about the device to decode it optimally?

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.
- Reporting the quality of a qubit with a single number is not useful to predict how it will respond in a fault-tolerant scheme.

Fault-tolerance critical parameters

In a given experiment, what is the leading noise source which limits the logical accuracy?

- Developed machine learning methods to identify such critical parameters.
- Decoding non-Pauli noise with tensor networks.

Decoding critical parameters

What information must I know about the device to decode it optimally?

Institue Quantique @ Sherbrooke

We are looking for talented

- Graduate students
- Postdocs
- Visiting faculty/scientists

Talk to me if you have any interest.