Bridging Quantum Error Correction and Mitigation

Zhenyu Cai

QEC & QEM

QEM:

- Low or no qubit overhead.
- Low or no requirement on gate fidelity (few additional quantum operations needed)
- Fast computation on unencoded (or low-distance) qubits

QEC:

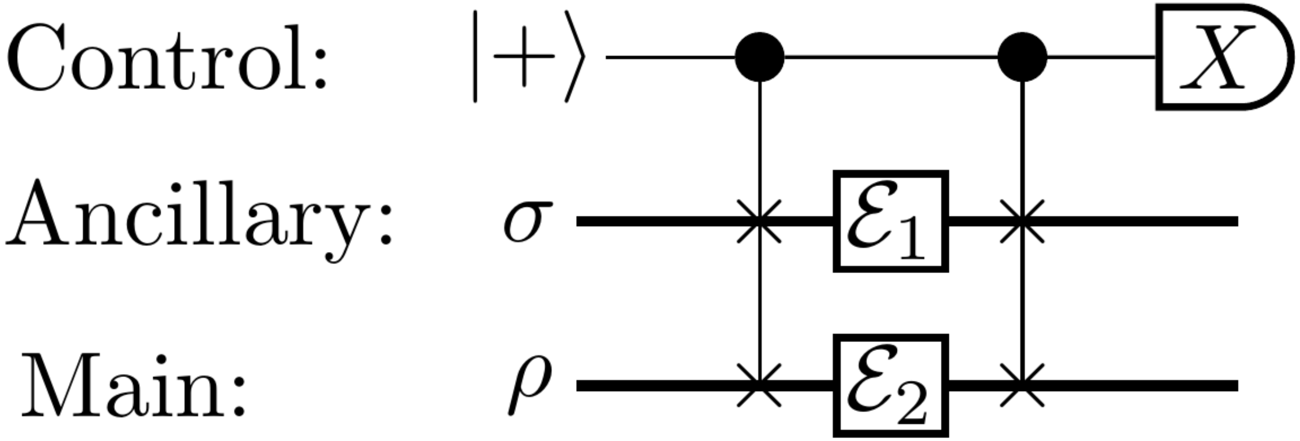
- Exponential suppression of error with increased qubit overhead and without sampling overhead.
- Universal applicability to all algorithms

Virtual Quantum Error Correction

Liu et al, "Virtual Channel Purification", PRX Quantum 6 (2), 020325

Prior Work:
Piveteau et al, PRL 127, 200505 (2021),
Suzuki et al, PRX Quantum 3, 010345 (2022).

Virtual Channel Entanglement



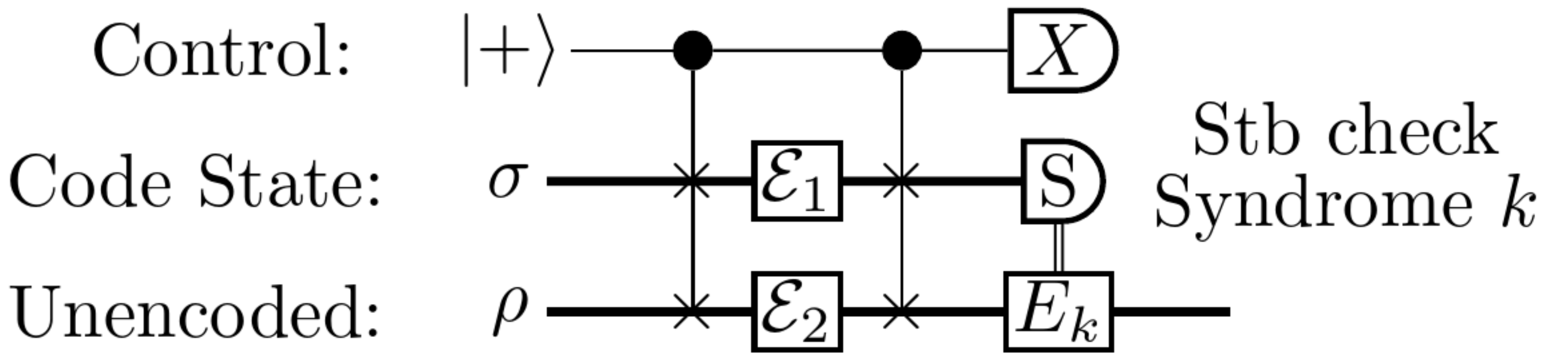
$$\mathcal{E}_{1/2}(\sigma) = \sum_i p_i \, E_i \sigma E_i^\dagger$$

- After post-processing based on the X measurement, the effective output state for the two unmeasured registers is

$$\sum_{i,j} p_i^2 \left(E_j \sigma E_i^\dagger\right) \otimes \left(E_i \rho E_j^\dagger\right)$$

where the two noise channels are virtually entangled.

Virtual Quantum Error Correction



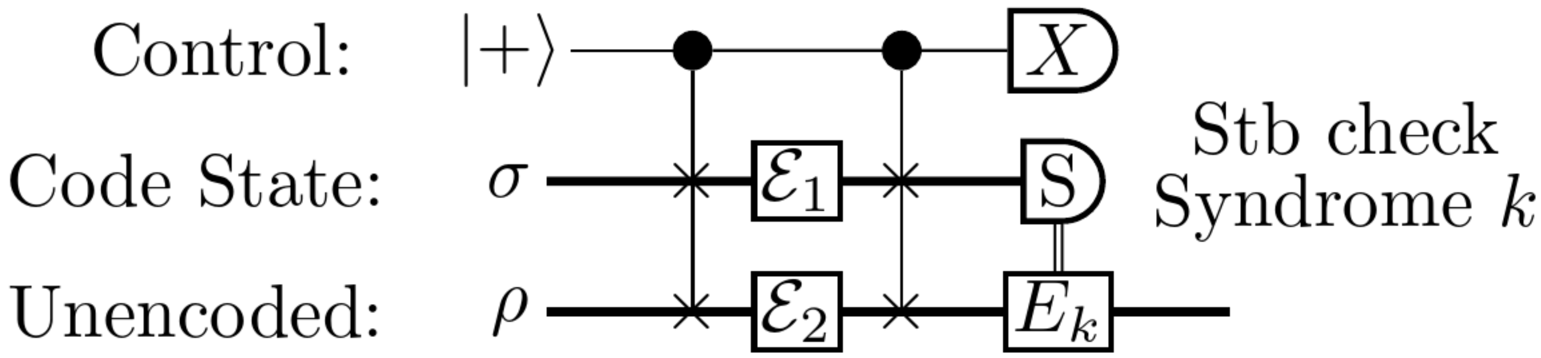
- Choosing σ to be the code state of a non-degenerate QEC code that can correct $\mathcal{E}_{1/2}$
- Stb measurements → syndrome k, →collapse incoming errors into E_k
- We are performing QEC on the unencoded register!

$$\sum_{i,j} p_i^2 (E_j \sigma E_i^\dagger) \otimes (E_i \rho E_j^\dagger)$$

$E_j \sigma E_i^\dagger$ → $E_k \sigma E_k^\dagger$

$E_i \rho E_j^\dagger$ → $E_k \rho E_k^\dagger$ → (Correction E_k) → ρ

Virtual Quantum Error Correction

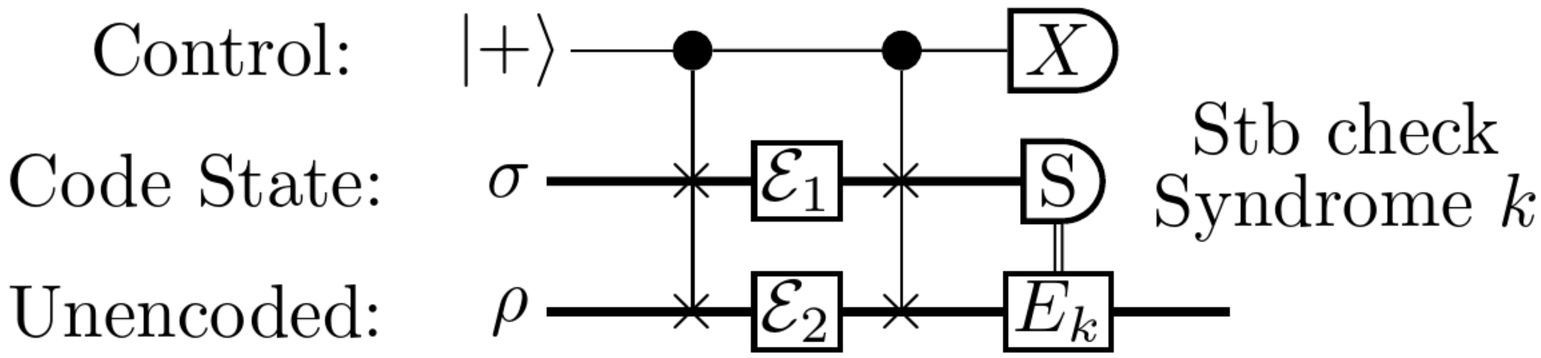


- The error correction is virtual because it requires post-processing based on control-qubit X measurement
- The post-processing comes with a sampling overhead of around

$$\sim \left(\sum_k p_k^2\right)^{-2}$$

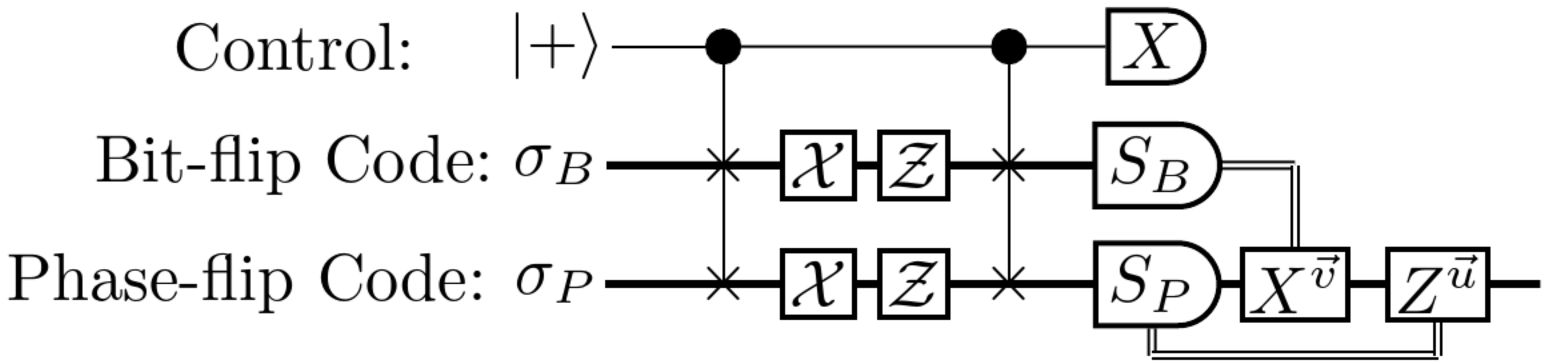
which is similar to virtual state purification (distillation), but now we can achieve the same noise suppression power as QEC.

Comparison to pure QEC



- Goal: send a K-qubit state ρ through the noise channel $\mathcal{E}_2$.
- Code overhead: suppose we need K physical qubits per logical qubit
- QEC: the total number of qubit required is K^2.
- VEC: The total number of qubits required is $2K + 1$.
- Limitation: VEC only works when the errors in $\mathcal{E}_{1/2}$ is non-degenerate (distinct syndromes for distinct errors) for the given code.

Combining two codes



- Instead of an encoded and an unencoded register, we put one register in bit-flip code and the other in phase-flip code
- Bit-flip check collapses the bit-flip noise on both registers, similarly for phase noise.
- So we only need two classical codes to correct quantum noise (both bit-flip and phase-flip)

Comparison to surface code

- Using distance-d bit-flip and phase-flip repetition codes for VEC, we can correct the same set of errors as distance-d surface codes.
- Advantages:
  - Qubit overhead: $2d+1$ (VEC) VS d^2 (surface code)
- Limitation:
  - Sampling overhead: same as virtual channel purification (two-copies)
  - Assuming noiseless CSWAP and stb checks. (Can be further mitigated using PEC)
  - Further work needed for computation.

QEM for Sampling Algorithm

Liu & Cai, “Quantum Error Mitigation for Sampling Algorithms”, arXiv:2502.11285

QEM for Sampling Algorithm

- QEM use post-processing to combine the output from multiple noisy circuits to obtain the error-mitigated expectation values.
  - The effective damage from noise is only reduced for the entire ensemble of circuit runs.
  - The noise remains unchanged or even increases when zoom individual circuit runs.

- Sampling algorithms (e.g. quantum phase estimation): rely on accurate results for every circuit run, thus seems to be inherently incompatible with QEM (except for those uses post-selection).

Error-mitigated State

- QEM can also be viewed as trying to extract the error-mitigated "states" ρ_{em} out of the noisy circuit runs.

- The error-mitigated expectation value is given as $\mathrm{Tr}(O\rho_{em})$. (O is the observable of interests)

- The error-mitigated states ρ_{em} is obtained via linear combination of output states from different circuit configurations.

- This covers most mainstream QEM techniques.

Cai *et al*, arXiv:2110.05389.

Examples of Error-mitigated States

- Linear Zero-noise extrapolation (can be generalized to Richardson):

$$\rho_p = (1-p)\rho_0 + p\rho_{err} \quad \Rightarrow \quad \rho_0 = \rho_{em} \propto p_2\rho_{p_1} - p_1\rho_{p_2}$$

- Probabilistic error cancellation for bit-flip noise:

$$\rho = (1-p)\rho_0 + pX\rho_0X \quad \Rightarrow \quad \rho_0 = \rho_{em} \propto (1-p)\rho - pX\rho X$$

- Also applicable to other major QEM techniques like virtual purification and symmetry verification.

Cai *et al*, arXiv:2110.05389.

State to Distribution is Taking Expectation.

- The output probability of a string z is the expectation value of the projector $\Pi_z = |z\rangle\langle z|$.

$$p(\mathrm{z}) = \mathrm{Tr}(\Pi_z \rho)$$

- All probabilities of different z can be measured simultaneously.

In a given circuit run, by measuring in the computational basis which output the string z', we have obtained one sample for all $\{\Pi_z\}$ with

  - one sample of 1 for the Π_z with $z = z'$
  - one sample of 0 for the Π_z with $z \neq z'$

QEM for Recovering Output Distribution

- Obtaining error-mitigated distributions $p_{em}(\mathrm{z}) = \mathrm{Tr}(\Pi_z \rho_{em})$ from error-mitigated states ρ_{em} is efficient.

- Existing mainstream QEM techniques can be used to extract error-mitigated "states" ρ_{em}, thus can be straightforwardly extended to extract error-mitigated distributions.

PEC Example

- Probabilistic error cancellation for bit-flip noise:

$$\rho = (1-p)\rho_0 + pX\rho_0X \quad \Rightarrow \quad \rho_0 = \rho_{em} = \frac{(1-p)\rho - pX\rho X}{1-2p}$$

- Implementation:

1. Sample ρ and $X\rho X$ with probability $(1-p)$ and p, respectively.
2. Measure in computation basis $\{Z_i\}$, post-process to obtain the set of observables $\{\Pi_z\}$.
3. If $X\rho X$ is sampled, attach minus sign to the output.
4. $p_{em}(\mathrm{z})$ is estimated by taking the average over all samples of Π_z and renormalise the result with the $(1-2p)^{-1}$ factor.

Sampling overhead

- Let us consider the trivial observable I:

$$\hat{I} = \sum_z \hat{\Pi}_z \quad \Rightarrow \quad \mathrm{Var}\left[\hat{I}\right] \approx \sum_z \mathrm{Var}\left[\hat{\Pi}_z\right] \qquad\qquad (|\mathrm{Cov}| < 1)$$

- i.e. the variance of estimating a single observable I is similar to the total variance of estimating the probability of all z, i.e. the entire probability distribution.

- For a given number of circuit runs, the total variance achieved for all entries in the entire estimated distribution is actually similar to the variance of one single observable.

Application to Quantum Phase Estimation

- Considering using quantum phase estimation for obtaining ground state energy.

- Instead of trying to obtain the whole distribution, we are trying to obtain the smallest string from the output distribution.

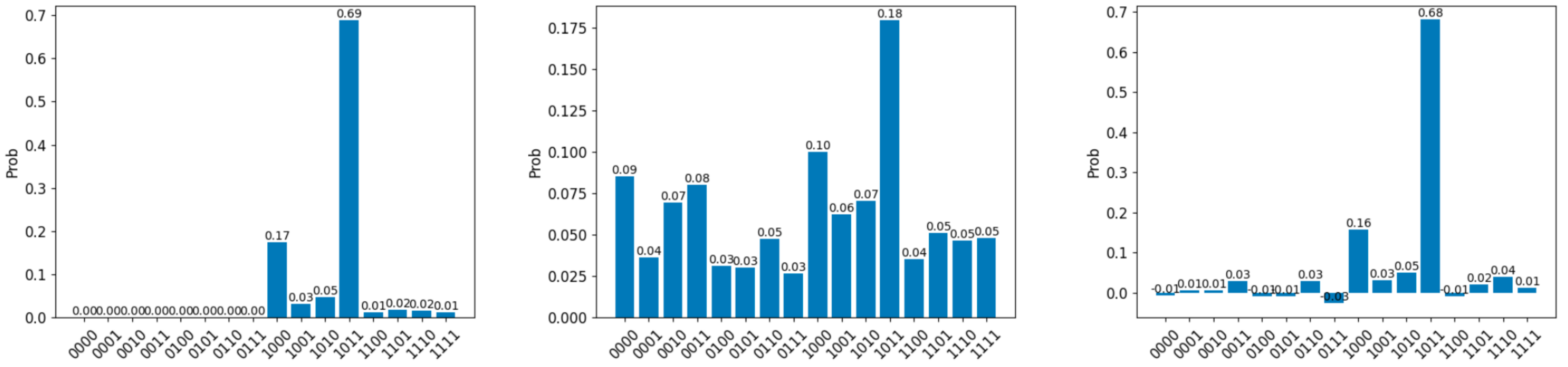
- Cannot simply output the smallest string from the estimated error-mitigated distribution, since shot noise can turn zero-probability entries to non-zero.

Application to Quantum Phase Estimation

- An additional step to test whether an entry is likely to be zero or not.
- Set a threshold probability $p_{\mathrm{th}}(z)$ for each entry such that
  - $\hat{p}_{\mathrm{em}}(z) \leq p_{\mathrm{th}}(z) \quad \Rightarrow \text{Accept null: } p_{\mathrm{em}}(z) = 0$
  - $\hat{p}_{\mathrm{em}}(z) > p_{\mathrm{th}}(z) \quad \Rightarrow \text{Accept null: } p_{\mathrm{em}}(z) > 0$

- $p_{\mathrm{th}}(z)$ can be set using:
  - Proportional to the sample standard deviation of the $\hat{p}_{\mathrm{em}}(z)$ estimator.
  - Known lower bound of the probability of the smallest string.

Numerical Simulation



(a) Ideal $p_0(z)$

(b) Noisy $p(z)$

(c) Mitigated $\hat{p}_{\text{em}}(z)$, 10^6 circuit runs

- QPE with 4-bit precision
- Circuit error rate ~ 0.6
- 10^6 runs

- Total square errors reduced from 0.297 to 0.004
- Valid threshold: $0.03 < p_{\text{th}} < 0.16$

How to sample from the QEM distribution?

- Without QEM, when measure z in a circuit run, we put one sample into the “bucket” corresponding to outcome z.

- With QEM, when measure z in a circuit run, there is also a additional sign associated with the circuit configuration we are running:
  - +ve sign: add one sample into the “bucket” corresponding to outcome z
  - -ve sign: remove one sample from the “bucket” corresponding to outcome z

How to sample from the QEM distribution?

- There can be negative number of samples! Esp. when the number of circuit run is small.

- When comes to interpretation of results, these negative number can effectively be treated as zero since any components below zero are entirely due to shot noise.

Summary

- VEC is a framework to combine the power of different QEC codes using post-processing.
- QEM can be used for recovering the output distribution and also sampling from it, and this is as cheap as one observable.

Open questions:

- More general frameworks to combine QEC and QEM, hopefully introducing a range of techniques with different trade-offs.
- What are the other (practical) algorithms that still fall outside the remit of QEM? How to include them?