Efficient Mitigation of Error Floors in Quantum Error Correction using Non-Binary LDPC Code

Kenta Kasai

Institute of Science Tokyo

ISIT 2025, Ann Arbor (Michigan), USA, June 26th, 2025

Background

- Recent studies[1] have reported that binary CSS codes based on $\mathbb{F}_q$-valued LDPC codes[2] exhibit **near-hashing-bound** decoding performance over the depolarizing channel using **joint BP** decoding.
- However, for codes with a **low coding rate R**, a **significant error floor** has been observed.
- This study aims to mitigate or eliminate the error floor—**ideally achieving a target frame error rate (FER) of 10^{-4}** near the hashing bound.

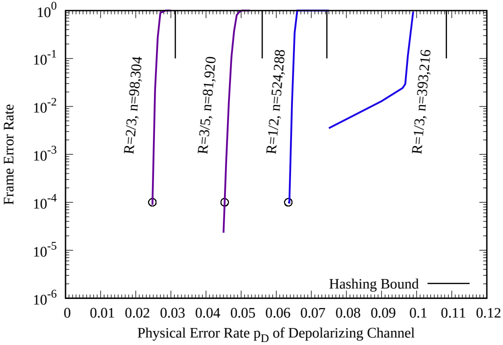

Figure: Joint BP peformance of QEC with non-binary LDPC codes with $\mathbb{F}_q$ $(q = 2^8)$.

[1] Komoto and Kasai, **under minor revision**, *npj Quantum Information*, 2025.
[2] Kasai, Hagiwara, Imai and Sakaniwa, IEEE Trans. Information Theory, 2011.

Code Construction

- We employ orthogonal $\mathbb{F}_q$-valued parity-check matrices H_X and H_Z with **column weight two** and **girth 12**.
- The matrices H_X and H_Z are constructed to be orthogonal by leveraging the structure of **circulant permutation matrices** or **affine permutation matrices**.
- By using companion matrices, the matrices H_X and H_Z can also be regarded as binary parity-check matrices.

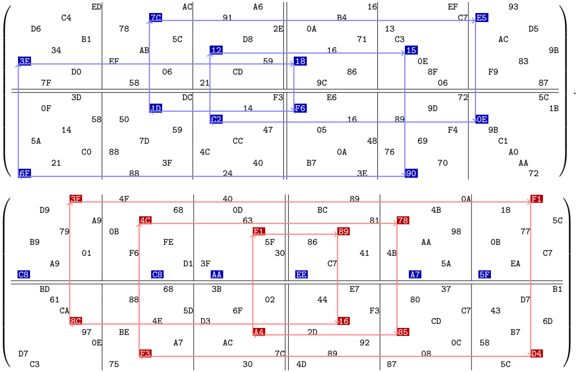

Figure: Parity-check matrics H_Z and H_X over $\mathbb{F}_q$ $(q = 256)$.

Joint BP decoding

- **Joint BP decoding** is a belief propagation algorithm that **simultaneously estimates** $\underline{x}$ and $\underline{z}$.
- Joint BP decoding begins by measuring the syndromes $\underline{s}$ and $\underline{t}$ corresponding to the noise vectors $\underline{x}$ and $\underline{z}$.

$$\underline{s} = H_Z \underline{x} \text{ and } \underline{t} = H_X \underline{z}.$$

- Joint BP iteratively estimates $\underline{\hat{x}}^{(\ell)}$ and $\underline{\hat{z}}^{(\ell)}$ at each iteration ℓ.

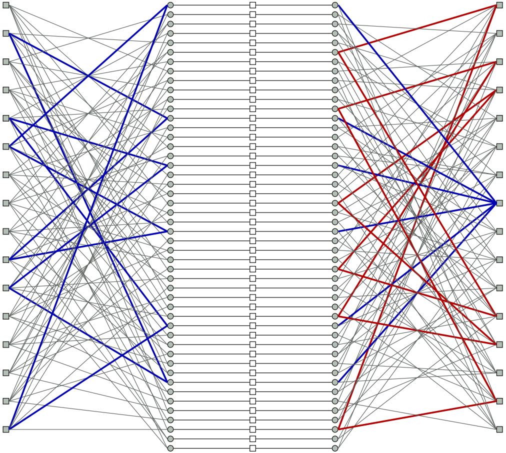

Figure: Factor graph of joint BP.

Insights into Decoding Failures in the Error Floor Regime

- In the error floor regime, joint BP decoding is sufficient to correctly estimate the noise in most cases. However, it occasionally fails to do so.
- In such failure cases, the joint BP algorithm becomes **trapped in a union of length-12 cycles** on the Tanner graph.
- In our experiments, decoding failures in the error floor regime caused by combined cycles involving both the X- and Z-side factor graphs were not observed.

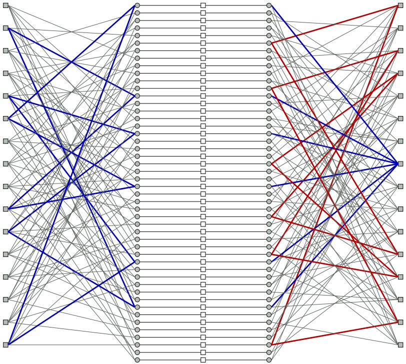

Figure: Factor graph of joint BP.

Overview of Proposed Decoding Algorithm

1. First, we run joint belief propagation for a sufficiently large number of iterations. In most cases, this step alone is enough to correctly estimate the noise.
2. However, if joint BP decoding fails—typically due to being trapped in cycles of length 12—we then proceed to estimate the trapping cycles.
3. Once the trapping cycles are identified, we estimate the remaining undetermined noise by solving a linear system of equations.

Method: Estimation of Trapping Cycles

- At each iteration, we keep track of the locations where the estimated noise and the syndrome values have recently changed:

  $K_d^{(\ell)}$: Estimated noise that have changed within the past d iterations

  $I_d^{(\ell)}$: Syndromes of the estimated noise that have changed within the past d iterations

- For sufficiently large ℓ and d, it was observed that $K_d^{(\ell)}$ and $I_d^{(\ell)}$ tend to **cover the columns and rows of trapping cycles**, respectively.
- This observation enables us to **efficiently identify the trapping cycles**.

	Estimation for $\underline{x}$				Estimation for $\underline{z}$			
ℓ	$\lvert K_{\mathrm{err}}^{(\ell)}\rvert$	$\lvert K_d^{(\ell)}\rvert$	$\lvert I_{\mathrm{err}}^{(\ell)}\rvert$	$\lvert I_d^{(\ell)}\rvert$	$\lvert K_{\mathrm{err}}^{(\ell)}\rvert$	$\lvert K_d^{(\ell)}\rvert$	$\lvert I_{\mathrm{err}}^{(\ell)}\rvert$	$\lvert I_d^{(\ell)}\rvert$
0	14944	0	9689	9689	15017	0	9741	9741
1	13731	4270	8618	10371	13845	4165	8677	10399
2	12875	6986	7676	10631	12959	6864	7791	10656
3	12108	8776	7036	10757	12306	8660	7178	10791
4	11693	10053	6558	10852	11765	10017	6717	10883
5	11297	11035	6221	10907	11370	11022	6304	10941
6	10866	11808	5862	10951	11043	11808	6028	10986
7	10542	12446	5640	10974	10667	12518	5745	11027
8	10300	12950	5464	10044	10364	13119	5537	10141
9	10069	11536	5216	9337	10099	11796	5334	9442
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
41	466	5682	462	3625	846	5956	684	3755
42	221	5088	204	3167	473	5405	436	3421
43	90	4337	103	2742	227	4822	225	3053
44	15	3633	25	2307	81	4243	95	2664
45	2	2980	2	1856	21	3575	27	2210
46	3	2257	4	1389	5	2882	7	1755
47	2	1595	2	909	0	2197	0	1300
48	2	973	2	565	0	1538	0	897
49	3	538	4	261	0	998	0	531
50	2	250	2	118	0	542	0	264
51	2	101	2	29	0	256	0	108
52	3	19	4	6	0	87	0	30
53	2	6	2	6	0	23	0	7
54	2	6	2	6	0	5	0	0
55	3	6	4	6	0	0	0	0
56	2	6	2	6	0	0	0	0
57	2	6	2	6	0	0	0	0
58	3	6	4	6	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure: Transition of the joint BP decoding state over iterations ($d = 8$).

Method: Post-Processing Algorithm

- For estimating X-noise, **solve a linear system** involving the syndrome $\underline{s}$ and the noise vector $\underline{x}_K$ **associated with the trapping cycles**. This is done using a submatrix of H_Z restricted to the set of column positions K corresponding to the trapping cycles:

$$\underline{s} = (H_Z)_K \underline{x}_K + (H_Z)_{\overline{K}} \underline{\hat{x}}_{\overline{K}}$$

  The size of K is independent of the code length, and the system can be solved by Gaussian elimination with computational complexity $O(|K|^3)$.
- Similarly, estimate the Z-noise vector by solving the corresponding linear system.
- Error correction is regarded as successful if and only if

$$\underline{x} + \underline{\hat{x}} \in C_X^{\perp} \text{ and } \underline{z} + \underline{\hat{z}} \in C_Z^{\perp}.$$

Results

- The error floor was mitigated to some extent.
- A relatively high error floor still remained.
- The remaining error floor is attributed to the presence of length-12 cycles that contain non-zero codewords in

$$C_Z \setminus C_X^{\perp} \quad \text{and} \quad C_X \setminus C_Z^{\perp}.$$

These codewords lead to logical errors in the decoding process.

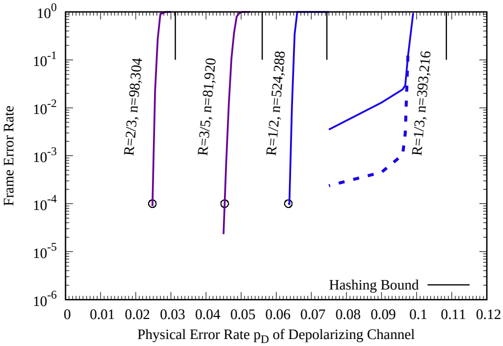

Figure: Comparison: joint BP (solid) with joint BP+post-processin (dashed)

Recent Result: Code Construction to Avoid Small Logical Errors[3]

- In binary codes, any cycle with column weight 2 always contains a **nonzero codeword**. However, this is **not necessarily the case** in the non-binary setting.
- By ensuring that the determinant of each cycle is nonzero, we can eliminate nonzero codewords from the cycles.
- We modified the $\mathbb{F}_q$-valued entries in the length-12 cycles so that the corresponding codewords are necessarily the **zero codeword**.
- As a result, the **error floor disappeared** at least down to FER $= 10^{-4}$.

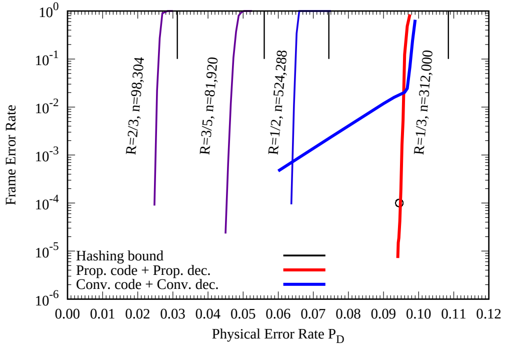

Figure: Comparison: Conventional method (blue) with proposed method (red).

[3] K. Kasai, “Quantum error correction exploiting degeneracy to approach the hashing bound,” *arXiv:2506.15636*, 2025.

Conclusions and Future work

- We successfully constructed binary CSS codes that **scale well across a wide range of coding rate**, achieving FER $= 10^{-4}$ by using non-binary LDPC codes.
- To further approach the hashing bound, we aim to **incorporate techniques originally developed for classical codes**, including:
  - Spatial Coupling
  - Multiplicative Repetition
  - Generalized LDPC Codes
- Currently, no upper bound on the girth of cycles leading to logical errors is known. This may open the possibility for applying **density evolution** analysis in future work.
- If you have ideas related to these directions, I would be very happy to hear them.

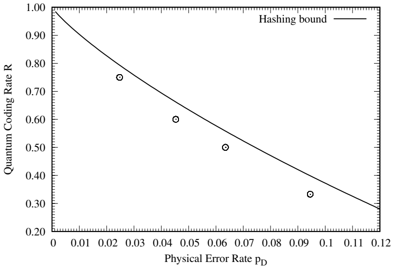

Figure: Physical Error Rate required for FER$=10^{-4}$ vs. Coding Rate