符号長に線形時間の復号で ハッシング限界に迫る量子誤り訂正

Kenta Kasai

東京科学大学

量子誤り訂正理論 若手研究会
2025 年 12 月 17 日–19 日

古典符号理論から量子誤り訂正へ

古典符号理論：すでに成熟した成功

- 古典符号理論は高度に成熟しており、BP 復号によって通信路容量に迫る性能をブロック長に線形な計算量で実現できる。
- そこで自然に生じる問いは、これらの強力な古典技術を量子誤り訂正へ持ち込めるか、という点である。

古典 LDPC 符号から学ぶこと

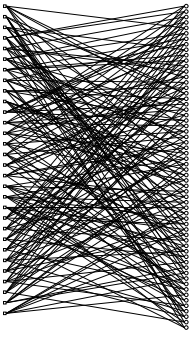
- **BP 復号**はパリティ検査行列 H の疎性を利用し、局所的なパリティ制約に基づいて各ビットの信頼度を反復更新する。
- **密度発展**は、H の正則次数分布の LDPC 符号はウォーターフォール領域で優れた BP 性能が得られることを示す。
- ランダム疎符号では、**列重み** ≥ 3 を課すことで最小距離が線形に成長することが保証される。
- Tanner グラフの **ガース**（最短閉路長）は、復号ダイナミクスと距離特性の両方に本質的である。
- **有限体拡大**上の非 2 値 LDPC 符号は、列重み 2 でも強い BP 性能を示し得る。
- 最終的には、最小距離がエラーフロア挙動を決める。

例：$(3, 6)$ 正則符号

$$H = \left(\right.$$

$\left. \right)$

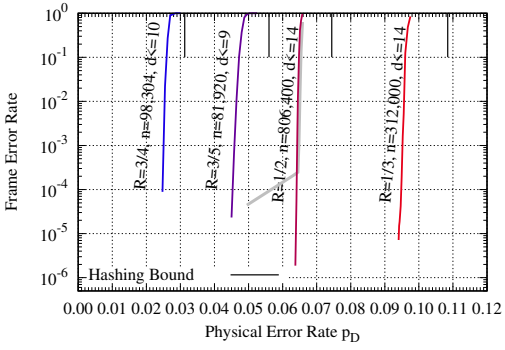
例：Tanner グラフ



数値結果：非 2 値 LDPC に基づく qubit 符号

まず提案方式の復号性能を示します。以下の性能が得られています。

- しきい値的な **FER 曲線** が明確に観測され、ハッシング限界に近づく。
- BP 復号は論理量子ビット数に対して**線形計算量**。OSD などの重い後処理は用いていない。
- フレーム誤り率が少なくとも 10^{-4} まで**エラーフロア**は観測されていない。
- 非常に高い **符号化率** を達成：$R \geq 1/3$。
- 論理量子ビット数は $k = 10^6$ に迫り、量子 LDPC 符号としては例外的に大規模である。
- GPU 実装により、復号スループットは約 **1M [qbps]**。



[a] https://github.com/kasaikenta/gd-css-decoder
[b] https://github.com/NagatsukiSep/gd-css-decoder

[a] Komoto and Kasai, *npj Quantum Information*, 2025.
[b] Kasai, Hagiwara, Imai, and Sakaniwa, IEEE Trans. IT, 2011.
[c] K. Kasai, *arXiv:2506.15636*, 2025.
[d] K. Kasai, IEEE ISTC 2025.

誤り訂正の一般手順

古典の誤り訂正

1. 古典 LDPC 符号は *1* つ のパリティ検査行列 H で定義される。
2. 物理ノイズは 2 値誤りベクトル $e \in \mathbb{F}_2^n$ で表す。
3. シンドロームを測定：
$$s = He.$$
4. 復号器は $\hat{e}$ を求める：
$$H\hat{e} = s.$$
5. **成功条件：**
$$\hat{e} + e \in \{0\}.$$

量子の誤り訂正

1. 量子 LDPC 符号では *2* つ の行列 H_X, H_Z が必要で、
$$H_X H_Z^{\mathsf{T}} = 0$$
を満たす。
2. 物理ノイズは (x, z) で表す。
3. シンドロームを測定：
$$s_X = H_X z, \qquad s_Z = H_Z x.$$
4. 復号器は $\hat{x}, \hat{z}$ を求める：
$$H_Z \hat{x} = s_Z, \quad H_X \hat{z} = s_X.$$
5. **成功条件：**
$$\hat{x} + x \in C_Z^{\perp},\ \hat{z} + z \in C_X^{\perp}.$$

量子 LDPC 符号の一般的構成

量子 LDPC 符号には *2* つ のパリティ検査行列 H_X と H_Z が必要で、直交条件

$$H_X H_Z^{\mathsf{T}} = 0 \quad (\text{over } \mathbb{F}_2).$$

を満たさなければならない。

- この直交条件は H_X と H_Z の設計に強い制約を与え、行・列重み分布や達成可能なガースを大きく制限する。
- その結果、多くの構成は高い構造性をもつ完全直交な疎 *full* 行列対から出発する。しかし、この段階では量子レート $R = (n - \mathrm{rank}(H_X) - \mathrm{rank}(H_Z))/n = 0$ となることが典型的である。

$$H_X^{(\mathrm{ful})} \big(H_Z^{(\mathrm{ful})}\big)^{\mathsf{T}} = 0.$$

- 望ましい符号化率を得るために、full 行列からチェック行の一部を削除して最終的な H_X と H_Z を得る。

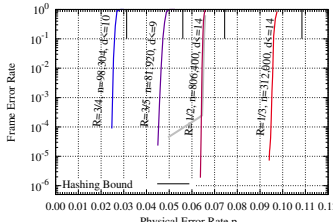
$$H_X^{(\mathrm{ful})} = \begin{bmatrix} H_X \\ H_X^{(\mathrm{res})} \end{bmatrix}, \qquad H_Z^{(\mathrm{ful})} = \begin{bmatrix} H_Z \\ H_Z^{(\mathrm{res})} \end{bmatrix}$$

$H_X^{(\mathrm{res})}$ の各行 z（典型的に低重み）は自動的に $H_Z z^{\mathsf{T}} = 0$ を満たし、したがって $z \in C_Z$ である。一般に $z \notin C_X^{\perp}$ なので、z は $C_Z \setminus C_X^{\perp}$ に属する Z 型の 論理 演算子として働く。

提案量子 LDPC 符号を支える鍵となる要素

提案性能は、以下の要素を組み合わせることで実現される。

- 一般化 **Hagiwara–Imai** 構成を採用する。
- アフィン置換行列（APM）により、H_X と H_Z の **可換性** を精密に制御する。
- 有限体上の非 2 値 LDPC では列重み $J = 2$ が有利であり、**超疎** なパリティ検査行列を実現し、**大きなガース（≥ 12）** を保証できる。
- 列重み $J = 2$ では、すべての符号語は Tanner グラフ上の閉路（または閉路の和）で生成される。**非 2 値シンボルを慎重に割り当てる**ことで、閉路由来の構造が有効な符号語を作れないようにし、小さな論理誤りを除去する。
- X と Z の相関を最大限に活かすため、**ジョイント信念伝播** 復号を用いる。
- エラーフロア領域で支配的な、長さ 12 閉路に対応するトラップ集合による誤り事象は、符号長に依存しない $O_n(1)$ 計算量の**後処理** アルゴリズムで補正する。



[a] Komoto and Kasai, *npj Quantum Information*, 2025.

[b] Kasai, Hagiwara, Imai, and Sakaniwa, IEEE Trans. IT, 2011.

[c] K. Kasai, *arXiv:2506.15636*, 2025.

[d] K. Kasai, ISTC 2025.

一般化 Hagiwara–Imai 構成：(J, L) 正則 LDPC CSS 符号

- 一般化 Hagiwara–Imai 構成は、置換ブロックを周期的に配置することで (J, L) 正則 LDPC CSS 符号を系統的に構成する方法である。
- $\{F_i\}_{i=0}^{L/2-1}$ と $\{G_i\}_{i=0}^{L/2-1}$ を $P \times P$ の置換行列とする。
- full パリティ検査行列 $H_X^{(\mathrm{ful})}$ と $H_Z^{(\mathrm{ful})}$ は $L/2 \times L$ のブロック巡回（block-circulant）行列であり、各ブロック行は 1 ステップ巡回シフトで得られる。

例：$J = 2,\ L = 6$

$$H_X^{(\mathrm{ful})} = \left(\begin{array}{cccc|cc} F_0 & F_1 & F_2 & G_0 & G_1 & G_2 \\ F_2 & F_0 & F_1 & G_2 & G_0 & G_1 \\ \hline F_1 & F_2 & F_0 & G_1 & G_2 & G_0 \end{array}\right), \qquad H_Z^{(\mathrm{ful})} = \left(\begin{array}{cccc|cc} G_0^{-1} & G_1^{-1} & G_2^{-1} & F_0^{-1} & F_1^{-1} & F_2^{-1} \\ G_2^{-1} & G_0^{-1} & G_1^{-1} & F_2^{-1} & F_0^{-1} & F_1^{-1} \\ \hline G_1^{-1} & G_2^{-1} & G_0^{-1} & F_1^{-1} & F_2^{-1} & F_0^{-1} \end{array}\right).$$

- 最終的な H_X と H_Z は、$H_X^{(\mathrm{ful})}$ と $H_Z^{(\mathrm{ful})}$ の上から J 個のブロック行を取り出すことで得られ、(J, L) 正則 LDPC 行列となる。

一般化 Hagiwara–Imai 構成：可換性が直交性を制御する

すべての置換ブロック $\{F_i\}, \{G_j\}$ に可換性を強制すれば直交性は保証されるが、同時に低重みの論理誤りを誘発し、最小距離を制限してしまう。

鍵となる発想：可換性を緩める。
すべて の置換ブロックに可換性を課すのではなく、H_X と H_Z の直交性を保証するのに本当に必要な ブロック対に対してのみ可換性を要求する。

可換性が直交性を制御する。
H_X と H_Z の直交性は、置換ブロック F_i と G_j の可換関係で決まる。ここで

$$\Delta_J := \{(k-i) \bmod L_2 \mid 0 \le i, k \le J-1\} \subseteq \{0, 1, \ldots, L_2 - 1\}.$$

とおく。任意の $r \in \Delta_J$ と任意の $u \in \{0, 1, \ldots, J-1\}$ に対して

$$F_u \, G_{r-u} = G_{r-u} \, F_u$$

が成り立てば、直交条件 $H_X H_Z^{\mathsf{T}} = 0$ が満たされる。

非可換性が距離制限を破る。
この必要部分集合の外側では意図的に非可換性を許すことで、硬直した代数構造が壊れる。
その結果、完全可換構成に内在する厳しい上界、例えば最小距離に対する $d_{\min} \le L$ やガースに対する $g \le 2L$ といった制限を緩和できる。

可換性制御のためのアフィン置換行列

- 置換ブロック間の可換性は直交条件 $H_X H_Z^{\mathsf{T}} = 0$ を保証する。一方、意図的に導入する 非可換性 は、過度に制限された構造を避け、最小距離に関する従来の上界を緩める鍵となる。
- しかし、一般の置換行列の中で、指定した可換・非可換パターンを満たすものを設計するのは難しい。

- **アフィン置換行列（APM）**：$\mathbb{Z}_P$ 上のアフィン置換は

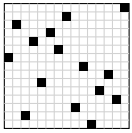
$$f(x) = ax + b \pmod{P}, \qquad a \in \mathbb{Z}_P^{\times},\ b \in \mathbb{Z}_P.$$

- 2 つのアフィン置換 $f(x) = a_1 x + b_1$ と $g(x) = a_2 x + b_2$ に対し、可換条件 $f \circ g = g \circ f$ は次の単純な代数条件に還元される：

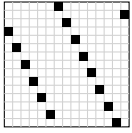
$$(a_1 - 1)b_2 = (a_2 - 1)b_1 \pmod{P}.$$

- したがって APM を用いると、可換性と非可換性を組合せ的探索ではなくパラメータで明示的に設計できる。

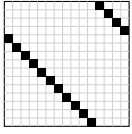
一般の置換：
$f(x) = (6, 2, \ldots, 8, 11, 0)$

$F =$ 

アフィン置換：
$f(x) = 2x + 3 \pmod{P = 15}$

$F =$ 

巡回置換：
$f(x) = x + 4 \pmod{P = 15}$

$F =$ 

APM の合成による閉路検出（$J = 2$）

$$H_X := \left(\begin{array}{ccc|ccc} F_0 & F_1 & F_2 & G_0 & G_1 & G_2 \\ F_2 & F_0 & F_1 & G_2 & G_0 & G_1 \end{array}\right), \qquad H_Z := \left(\begin{array}{ccc|ccc} G_0^{-1} & G_1^{-1} & G_2^{-1} & F_0^{-1} & F_1^{-1} & F_2^{-1} \\ G_2^{-1} & G_0^{-1} & G_1^{-1} & F_2^{-1} & F_0^{-1} & F_1^{-1} \end{array}\right) \quad (J = 2).$$

- すべてのブロック F_i と G_i が**アフィン置換行列（APM）** であると仮定する。
- 対応する 2×2 ブロック部分行列に長さ 4 閉路が存在することは、ある **合成アフィン置換** が不動点を持つことと同値である。すなわち

$$f_0^{-1} \circ f_1 \circ f_0^{-1} \circ f_2(x) = x \quad \text{を満たす } x \in [P] \text{ が存在する。}$$

- APM では合成写像も再びアフィンとなる：

$$f_0^{-1} \circ f_1 \circ f_0^{-1} \circ f_2(x) = ax + b \pmod{P},$$

  したがって不動点の存在は **gcd 条件** $\gcd(a - 1, P) \mid b$ で判定できる。
- よって 4 サイクルの有無は、対応する **APM の合成** が不動点を持つかどうかの判定に還元される。
- この手法は長さ ℓ の閉路にも自然に一般化できる。列重み $J = 2$ が与える **超疎** 構造を活かすことで、短い閉路を体系的に避け、ガース 12 を達成できる。

非 2 値拡張

- パリティ検査行列を有限体 $\mathrm{GF}(2^e)$ 上へ拡張する。本研究では $e = 8$ を用いる。
- 各有限体要素 α^i は、パリティ検査行列の中で $e \times e$ の 2 値行列 A^i により表現する。ここで α は原始多項式 $p(\alpha) = 0$ で定まる原始元、A は対応する コンパニオン行列である。
- 非 2 値シンボルを 2 値コンパニオン行列で表すため、提案符号は qudit 符号ではなく *qubit* 符号とみなせる。
- 列重み $J = 2$ が有利で、**超疎** なパリティ検査行列を実現し、**大きなガース（≥ 12）** を保証できる。
- BP 復号では $\mathrm{GF}(2^e)$ 上の確率分布を更新する。

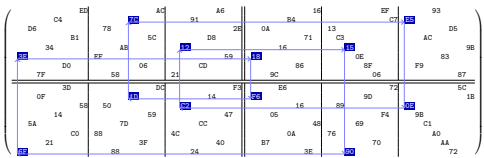


Figure: 例：$\mathrm{GF}(2^8)$ 上の非 2 値パリティ検査行列 H_Z.

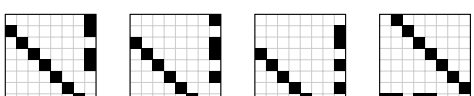


Figure: コンパニオン行列 A とその冪 $A^2, A^3, \ldots, A^{254} = A^{-1}$（GF(2) 上）.

$J = 2$ における小さな論理誤りの除去

- 列重みが $J = 2$ のとき、すべての符号語は Tanner グラフ上の単一の閉路または閉路の和で生成される。
- 削除した行 $H_X^{(\mathrm{res})}$ と $H_Z^{(\mathrm{res})}$ に含まれる閉路構造は、低重み符号語 を自然に生み、論理誤りの支配的要因になり得る。
- 辺（またはブロック）に対して**非 2 値シンボルを慎重に割り当てる**ことで、これらの閉路が フルランク になるようにする。
- その結果、閉路由来の構造は有効な符号語を作れなくなり、小さな論理誤りを効果的に除去できる。

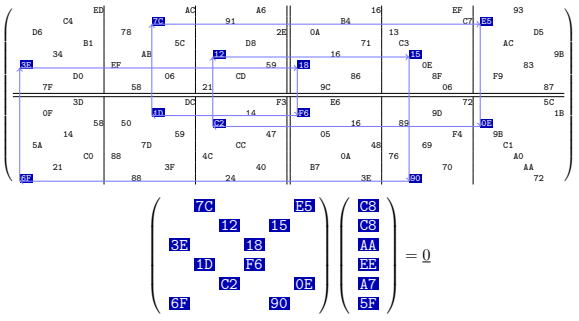


Figure: 長さ 2ℓ の閉路がランク欠損なら、重み ℓ の閉路符号語が存在する。

[2] Kasai, Hagiwara, Imai, and Sakaniwa, IEEE Trans. Information Theory, 2011.

ジョイント BP 復号 [5]

- 非 2 値化した行列：$\mathrm{GF}(2^8)$ 上の H_X と H_Z
- シンドロームを測定：

$$\boldsymbol{s} = H_Z \boldsymbol{x}, \quad \boldsymbol{t} = H_X \boldsymbol{z}$$

- ジョイント BP 復号はパリティ検査行列の疎性を利用し、局所的なパリティ制約に基づいて各ビットの信頼度を **反復更新**する。
- ジョイント BP は各反復で $\hat{\boldsymbol{x}}$ と $\hat{\boldsymbol{z}}$ を **同時に推定**する。
- 誤り訂正の成功は次と同値：

$$\boldsymbol{x} + \hat{\boldsymbol{x}} \in C_X^{\perp} \text{ かつ } \boldsymbol{z} + \hat{\boldsymbol{z}} \in C_Z^{\perp}.$$

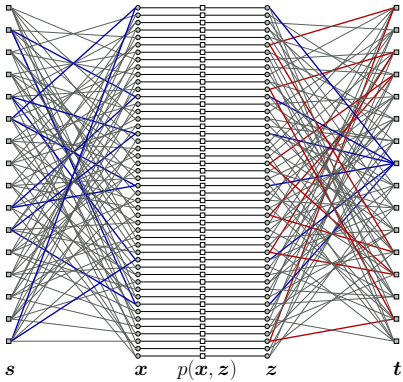
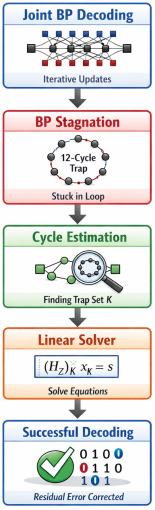


Figure: BP の因子グラフ.

[5] https://github.com/kasaikenta/gd-css-decoder

提案する後処理付き復号アルゴリズムの概要

1. まず、十分多い反復回数でジョイント BP 復号を実行する。ほとんどの場合、このステップだけでノイズを正しく推定できる。
2. 稀に（概ね 10^3 試行に 1 回程度）ジョイント BP 復号が収束しない。このとき、不満足なシンドローム数がゼロまで減らず、復号が停滞する。原因は典型的に短い閉路（例：長さ 12 閉路）へのトラップである。そこで対応するトラップ閉路を推定する。
3. トラップ閉路を同定できたら、残っている未決定のノイズ成分を小さな線形方程式系を解くことで推定する。



手法：トラップ閉路の推定

- 各反復で、推定ノイズおよびシンドローム値が最近変化した位置を追跡する：

  $K_d^{(\ell)}$: 過去 d 反復で変化した推定ノイズの位置

  $I_d^{(\ell)}$: 過去 d 反復で変化した推定ノイズのシンドローム位置

- 十分大きい ℓ と d を取ると、$K_d^{(\ell)}$ と $I_d^{(\ell)}$ がそれぞれトラップ閉路の列・行を **覆う** ことが経験的に観測される。
- これにより、トラップ閉路を **効率よく同定**できる。

	Estimation for x				Estimation for 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	X stagnation region			6	0	23	0	7
54	2	6	2	6	0	5	0	0
55	3	6	4	6	Z converged region			
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: ジョイント BP 復号状態の反復に伴う遷移 ($d = 8$) .

手法：後処理アルゴリズム

- X ノイズを推定するため、トラップ閉路に対応する集合 K 上の未知成分 $\boldsymbol{x}_K$ に対し、シンドローム $\boldsymbol{s}$ を用いた **線形系** を解く。具体的には、H_Z を K に対応する列へ制限した部分行列を用いる：

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

- $|K|$ は符号長に依存しないため、ガウス消去によって計算量 $O(|K|^3)$ で効率的に解ける。
- 実験では $|K| = 12$ とすればすべてのトラップ事象を解消できた。
- 同様に、Z ノイズも対応する線形系を解いて推定する。
- 誤り訂正の成功は次と同値：

$$\boldsymbol{x} + \hat{\boldsymbol{x}} \in C_X^{\perp} \quad \text{かつ} \quad \boldsymbol{z} + \hat{\boldsymbol{z}} \in C_Z^{\perp}.$$

今後の課題と結論

- 非 2 値 LDPC 符号に着想を得た量子 LDPC 符号化枠組みを提案し、**しきい値的** な復号性能を、**ハッシング限界** に近い領域で達成した。
- 列重み $J = 2$ の超疎構造、アフィン置換行列、そして非 2 値シンボルの慎重な割り当てにより、**小さな論理誤り** を除去できた。
- コンパニオン行列表現を用いることで、物理符号は **qubit 符号** のまま非 2 値代数の利点を活用できる。
- フォールトトレラント量子計算（FTQC）に向けた重要課題として、回路レベル雑音、相関誤り、**雑音を含む・繰り返し測定が必要なシンドローム測定**を取り込んだ現実的な雑音モデルの導入が不可欠である。
- （補足）現実的な雑音モデルの導入には、回路レベル雑音や相関誤りの扱いも重要である。
- 符号構成、復号アルゴリズム、雑音モデル、ならびに実験検証に関する共同研究を歓迎する。

（任意）量子 LDPC 符号に密度発展を適用する条件

- **密度発展（DE）の本質的要件：**
  - BP の反復回数を固定したとき、計算グラフが **局所的に木状** であること。
  - この条件の下では、入力 BP メッセージを独立とみなせ、DE 再帰が正当化される。
- **ガースの役割：**
  - ブロック長とともにガースが成長すれば、局所木構造性を保証する **十分条件** となる。
  - ただし、**ガース成長は必要条件ではない**。
    - 任意の固定深さにおいて、近傍が高確率で木状であれば十分である。
    - ランダム疎グラフ族では、ガースが有界でも局所木構造性が成り立ち得る。
- **量子 LDPC 符号に固有の構造制約：**
  - スタビライザ符号の full $\mathrm{GF}(4)$ Tanner グラフでは、可換条件により**短い閉路（特に 4 サイクル）** が必然的に導入される。
  - 一方、本研究で用いる因子グラフでは、縮退誤りに由来する長さ $2L$ の閉路は存在するが、それ自体が直接の復号失敗を引き起こすわけではない。
  - さらに、ガースの明確な上界は現在知られておらず、実験的には大きなガース（例：16）が得られている。
- **結論：**
  - 密度発展の本質は**ガースそのものではなく局所木構造性** にある。
  - 因子グラフを適切に設計できれば、量子固有の制約があっても、DE は量子 LDPC 符号に対して有意義な解析ツールであり続ける。