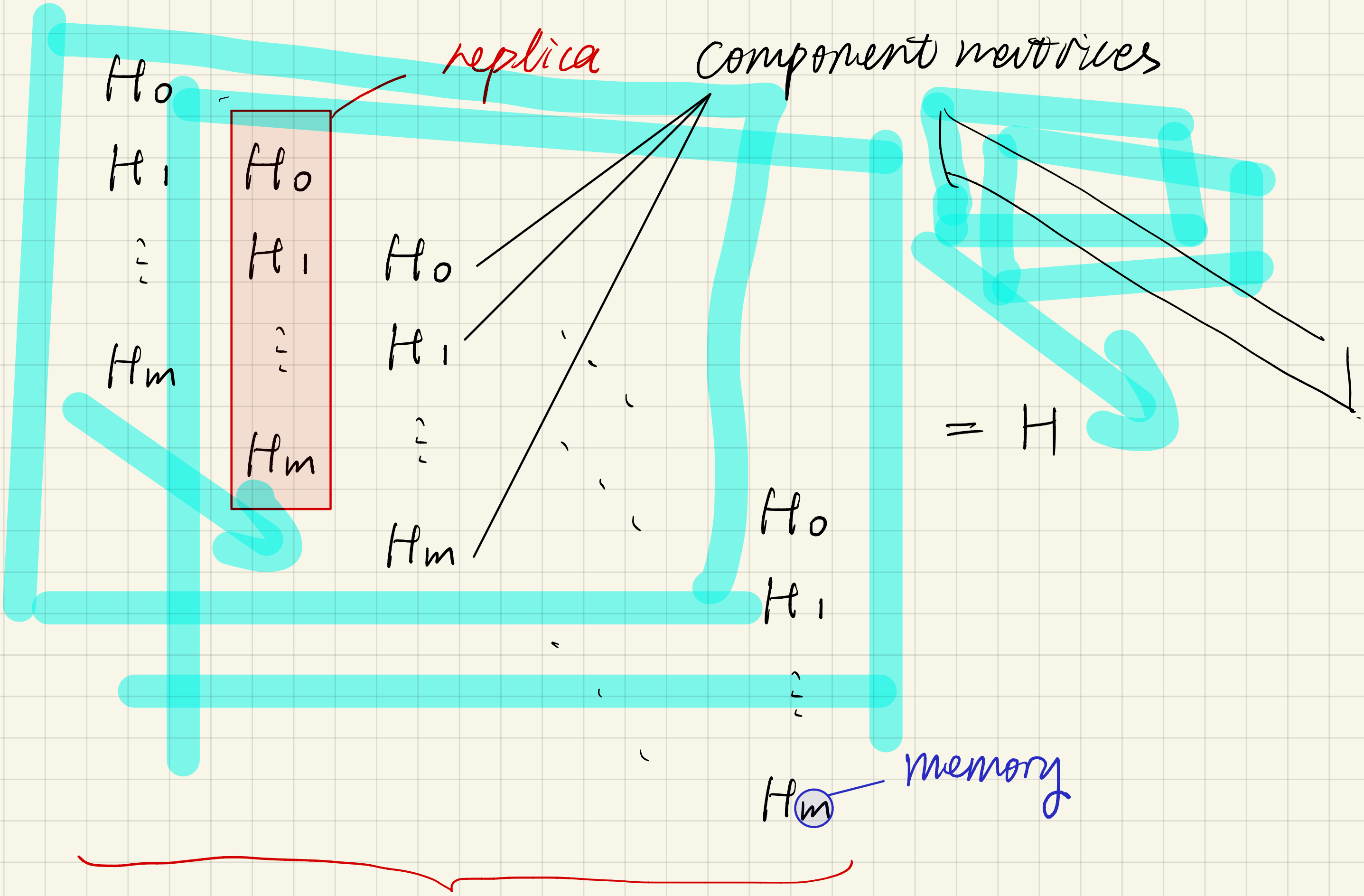
Lecture Spatially-Coupled Codes

Structure:



Coupling Length: L

QC-SC Codes

Rate: $R = 1 - (1+\frac{m}{L})\frac{\gamma}{\kappa}$

Component matrices are QC codes.

$H_0, H_1, \ldots, H_m$

$\uparrow \quad \uparrow \qquad\qquad \uparrow$ lifting

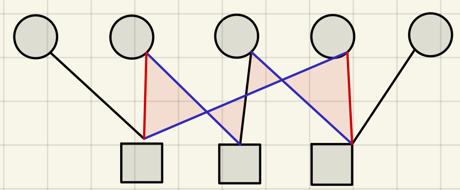
$H_0^P, H_1^P, \ldots, H_m^P \in \mathbb{F}_2^{\gamma\times\kappa}$ protograph

$\boxed{\Pi = H_0^P + H_1^P + \cdots + H_m^P \text{ : base matrix.}}$

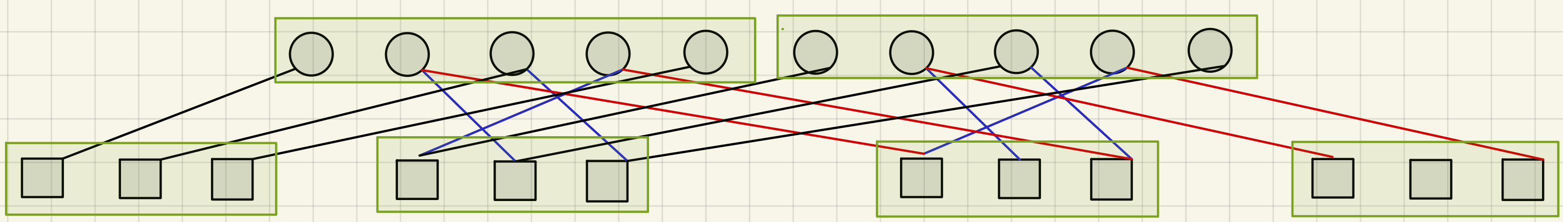
$$\begin{array}{ccccc} H_0^P & & & & \\ H_1^P & H_0^P & & & \\ \vdots & H_1^P & H_0^P & & \\ H_m^P & \vdots & H_1^P & \ddots & \\ & H_m^P & \vdots & \ddots & H_0^P \\ & & H_m^P & & H_1^P \\ & & & \ddots & \vdots \\ & & & & H_m^P \end{array} = H^P$$

$\longleftarrow$ protograph of QC-SC code

Graphical Interpretation:



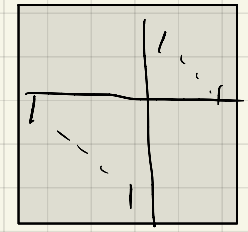
$$\Pi = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$



$$H_0^P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad H_1^P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$H_2^P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$
H^P = \left[\begin{array}{ccccc|ccccc}
1 & 0 & 0 & 0 & 0 & & & & & \\
0 & 0 & 1 & 0 & 0 & & & & & \\
0 & 0 & 0 & 0 & 1 & & & & & \\
\hline
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
\hline
 & & & & & 0 & 1 & 0 & 0 & 0 \\
 & & & & & 0 & 0 & 0 & 0 & 0 \\
 & & & & & 0 & 0 & 0 & 1 & 0
\end{array}\right]
$$



Partitioning Matrix:

$$
\Pi = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \longrightarrow P = \begin{bmatrix} 0 & 2 & * & 1 & * \\ * & 1 & 0 & * & * \\ * & * & 1 & 2 & 0 \end{bmatrix}
$$

Lifting Matrix:

$$
L = \begin{bmatrix} l_1 & l_2 & * & l_3 & * \\ * & l_4 & l_5 & * & * \\ * & * & l_6 & l_7 & l_8 \end{bmatrix}
$$

$$
l_i \in [0, Z-1]
$$

Cycle Condition

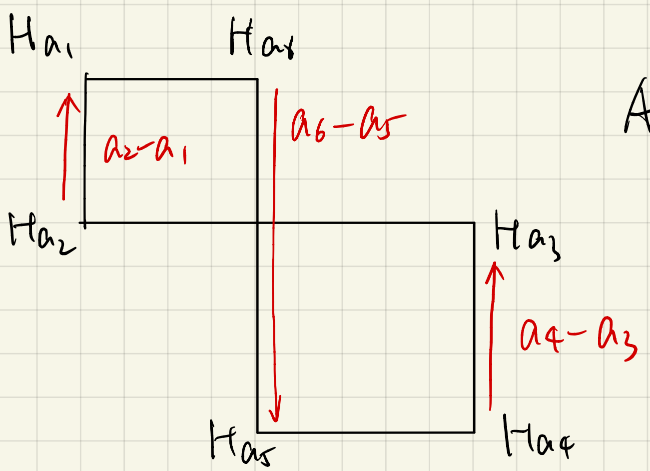
$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\Pi \longrightarrow H^P \longrightarrow H$$

$$\downarrow$$

$$P_{12} - P_{14} + P_{34} - P_{33} + P_{23} - P_{22} = 0$$

$$L_{12} - L_{14} + L_{34} - L_{33} + L_{23} - L_{22} = 0 \mod Z$$



Alternating Sum:

$$a_2 - a_1 + a_4 - a_3 + a_6 - a_5$$

Optimization

Removing Cycles: <u>increasing m, Z.</u>

$\downarrow$

also improves threshold / waterfall performance

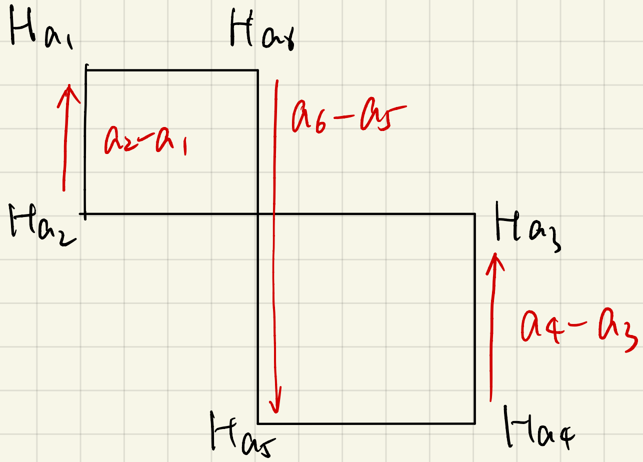
Step 1: Optimize the number of cycles of protograph by optimizing P.

Step 2: Randomly assign lifting parameters to L and apply greedy algorithm to further optimize.

low-memory: Combinatorial Optimization is applicable

high-memory: high complexity $\swarrow$

$\longrightarrow$ probablistic method.



$$\mathbb{P}[P_{i,j}=k]=p_k. \quad 0\le k\le m$$

$$p=(p_0,p_1,\cdots,p_m)$$

$$f_p(x)=\sum_{k=0}^{m} p_k x^k$$

$$[f_p(x)f_p(x^{-1})]^3$$

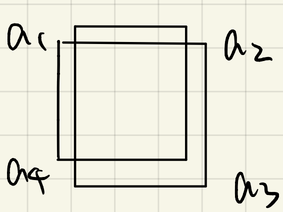
$$=\sum_{k}\sum_{a_6-a_5+a_4-a_3+a_2-a_1=k}\left(\prod_{i=0}^{m} p_{a_i}\right)x^k$$

$$=\sum_{k}\mathbb{P}[a_2-a_1+a_4-a_3+a_6-a_5=k]\,x^k$$

Probability of a cycle in base matrix becomes cycles in the protograph: $[(f_p(x)f_p(x^{-1}))^3]_0$

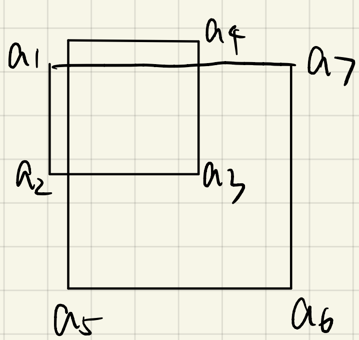
coefficient of x^0/constant term

Cycles 8.



$$2(a_1-a_2+a_3-a_4)=0$$

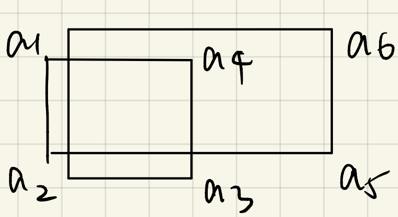
$$\Rightarrow [f^2(x)f^2(x^{-1})]_0 \qquad N_1$$



$$a_1-a_2+a_3-a_4+a_1-a_5+a_6-a_7=0$$

$$\Rightarrow 2a_1-a_2+a_3-a_4-a_5+a_6-a_7=0$$

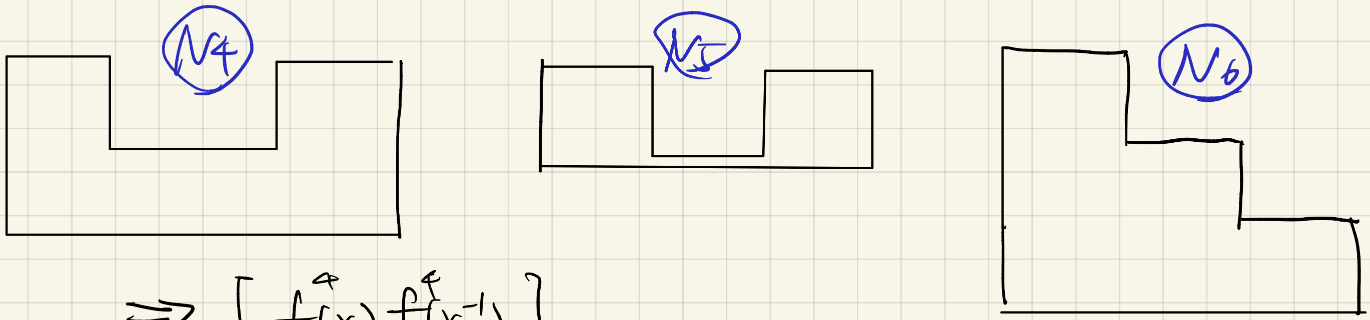
$$\Rightarrow [f(x^2)f^2(x)f^4(x^{-1})]_0 \qquad N_2$$



$$a_1-a_2+a_3-a_4+a_1-a_2+a_5-a_6=0$$

$$\Rightarrow 2a_1-2a_2+a_3-a_4+a_5-a_6=0$$

$$\Rightarrow [f(x^2)f(x^{-2})f^2(x)f^2(x^{-1})]_0 \qquad N_3$$



$$\Rightarrow [f^4(x) f^4(x^{-1})]_0$$

$$F_8(p) = N_1 [f^2(x) f^2(x^{-1})]_0 + N_2 [f(x^2) f^2(x) f^4(x^{-1})]_0$$
$$+ N_3 [f(x^2) f(x^{-2}) f^2(x) f^2(x^{-1})]_0 + (N_4+N_5+N_6) [f^4(x) f^4(x^{-1})]_0$$

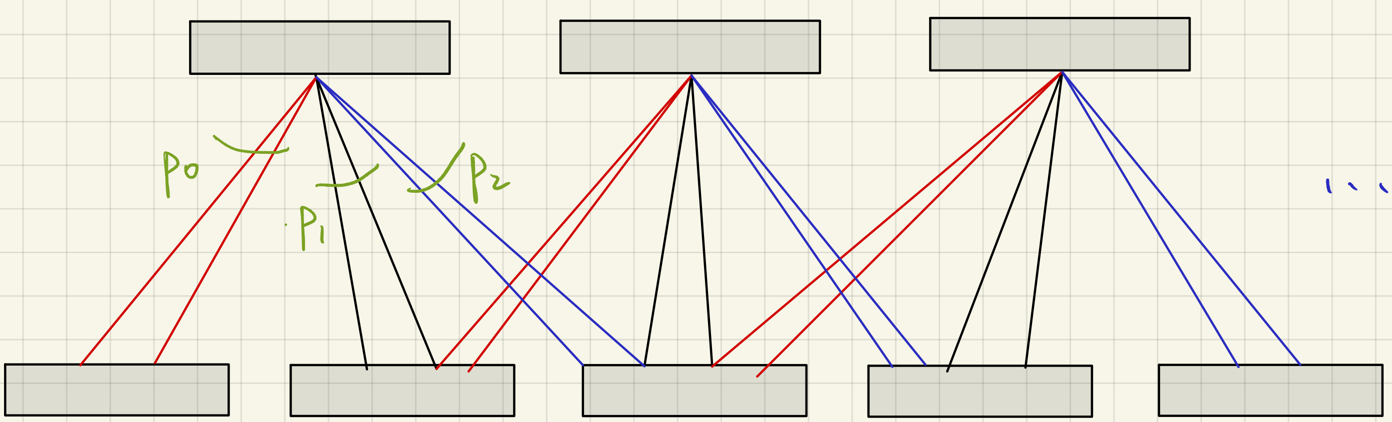
$$F_6(p) = N [f^3(x) f^3(x^{-1})]_0.$$

$$F(p) = w F_6(p) + F_8(p).$$
$$\nabla_p F(p) = w \nabla_p F_6(p) + \nabla_p F_8(p)$$

$>$ gradient desent.

$$\nabla_p F_6(p) = N \nabla_p [f^3(x) f^3(x^{-1})]_0$$
$$= N [3f^2(x) f^3(x^{-1}) \nabla_p f(x) + 3 f^3(x) f^2(x^{-1}) \nabla_p f(x^{-1})]_0$$
$$= N [3f^2(x) f^3(x^{-1}) (1, x, \cdots, x^m)$$
$$+ 3f^3(x) f^2(x^{-1}) (1, x^{-1}, \cdots, x^{-m})]_0$$

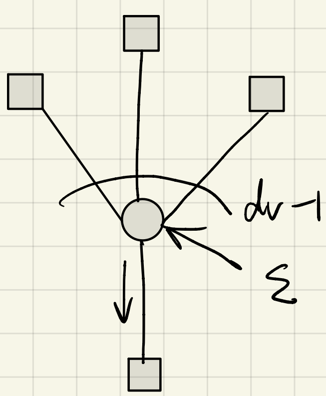
Threshold Saturation on BEC(ε)



$$\begin{array}{c|ccccc} & x_1^{(l)} & x_2^{(l)} & x_3^{(l)} & & x_L^{(l)} \\ y_1^{(l)} & H_0\, p_0 & & & & \\ y_2^{(l)} & H_1\, p_1 & H_0\, p_0 & & & \\ \vdots & \vdots & H_1\, p_1 & H_0 & \ddots & \\ y_{m+1}^{(l)} & H_m\, p_m & \vdots & H_1 & \ddots & \\ & & H_m\, p_m & \vdots & \ddots & \\ \vdots & & & H_m & \ddots & H_0\, p_0 \\ & & & & \ddots & H_1\, p_1 \\ & & & & \ddots & \vdots \\ y_{L+m}^{(l)} & & & & & H_m\, p_m \end{array} = H$$

$$\vec{x}^{(l)} = \left(x_1^{(l)}, x_2^{(l)}, \cdots, x_L^{(l)} \right)$$

VN-2-CN Update

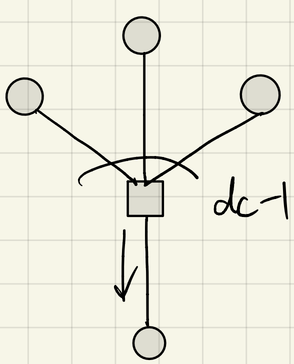


$$x_{k,d_v}^{(l+1)} = \Sigma \left(p_0 y_k^{(l)} + p_1 y_{k+1}^{(l)} + \cdots + p_m y_{k+m}^{(l)} \right)^{d_v - 1}$$

$$x_k^{(l+1)} = \mathbb{E}_{d_v} \left[x_{k,d_v}^{(l)} \right]$$

$$= \boxed{\Sigma \lambda \left(\sum_{i=0}^{m} p_i y_{k+i}^{(l)} \right)} \qquad k = 1, \cdots, L$$

CN-2-VN Update



$$y_{k,d_c}^{(l)} = 1 - \left(\sum_{i=0}^{\min(k,m)} p_i \left(1 - x_{k-i}^{(l)}\right) + \sum_{\min(k,m)+1}^{m} p_i \right)^{d_c - 1}$$

$$1 \le k \le L$$

$$y_{L+k,d_c}^{(l)} = 1 - \left(\sum_{i=k}^{m} p_i \left(1 - x_{L+k-i}^{(l)}\right) + \sum_{i=0}^{k-1} p_i \right)^{d_c - 1}$$

$$1 \le k \le m$$

Introduce: $x_{-m}^{(l)} = x_{-m+1}^{(l)} = \cdots = x_{-1}^{(l)} = 0$ $\quad \forall\ l \in N^*$

$x_{L+1}^{(l)} = x_{L+2}^{(l)} = \cdots = x_{L+m}^{(l)} = 0.$

$$y_{k,d_c}^{(l)} = 1 - \Big(\sum_{i=0}^{m} p_i (1 - x_{k-i}^{(l)})\Big)^{d_c - 1} \qquad 1 \le k \le L+m.$$

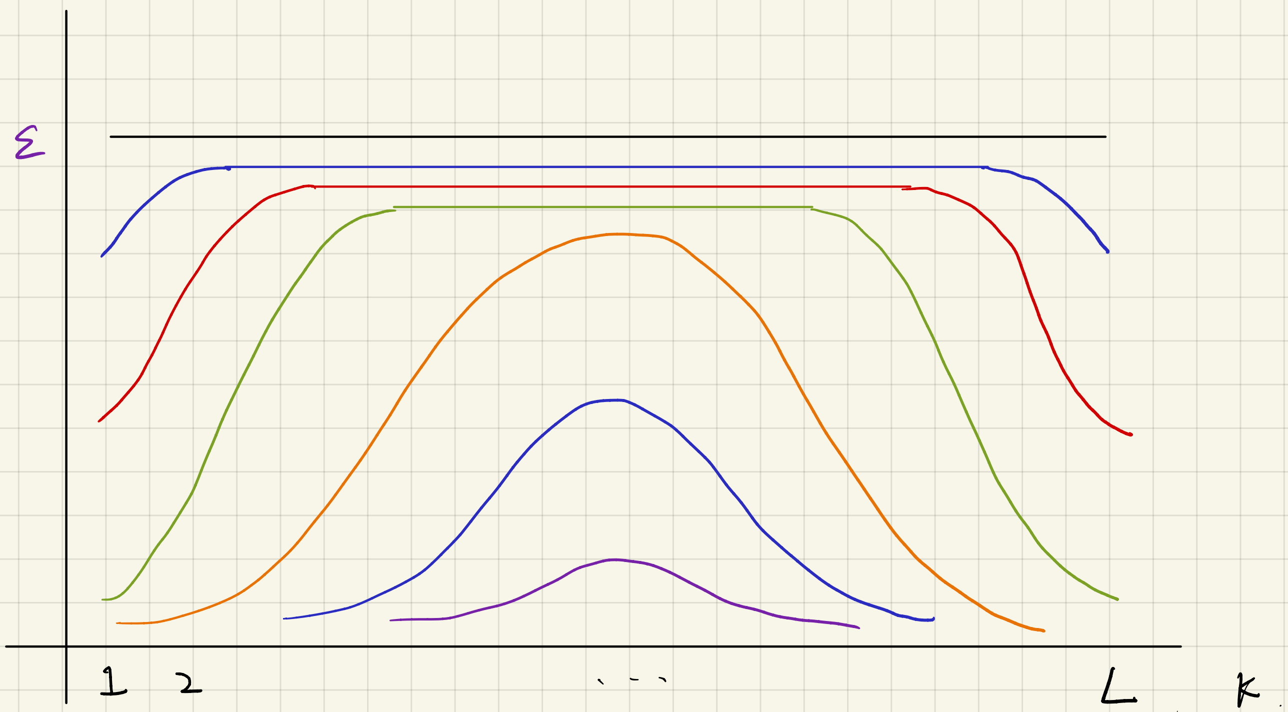
$$y_k^{(l)} = \mathbb{E}_{d_c}\big[y_{k,d_c}^{(l)}\big] = \boxed{1 - \rho\Big(1 - \sum_{i=0}^{m} p_i x_{k-i}^{(l)}\Big)}.$$

$$\boxed{x_k^{(l+1)} = \varepsilon \lambda \Big(\sum_{i=0}^{m} p_i \Big(1 - \rho\Big(1 - \sum_{j=0}^{m} p_i x_{k+i-j}^{(l)}\Big)\Big)\Big)}$$

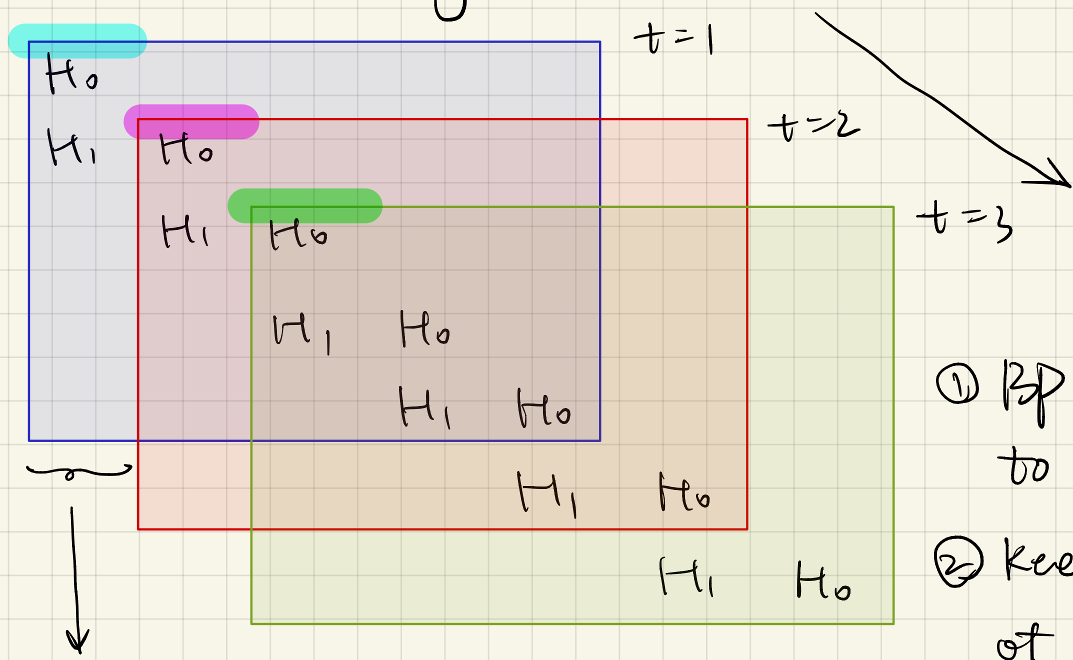
Density Evolution:

$\vec{x}^{(0)} = (\varepsilon, \varepsilon, \cdots, \varepsilon).$

obtain the maximal ε s.t. $\forall\ \varepsilon' < \varepsilon$ $\quad \vec{x}^{(l)} \to 0_L$.



Windowed Decoding.



targeted symbols

① BP within each WD. to decode target symbol

② keep the information of the last window.

Density Evolution

$$\boxed{x_k^{(l+1)} = \varepsilon \lambda \left(\sum_{i=0}^{m} p_i \left(1 - \rho\left(1 - \sum_{j=0}^{m} p_i x_{k+i-j}^{(l)}\right)\right)\right)}$$

$$\begin{cases} x_k^{(l+1)} = \varepsilon\, g(x_{k-m}, x_{k-m+1}, \cdots, x_{k+m-1}, x_{k+m}) \\ x_k^{(l)} = 0, \quad k \notin [L]. \end{cases}$$

$$\begin{aligned} y_{\{c\}} &= (y_{1,\{c\}}, y_{2,\{c\}}, \ldots, y_{w,\{c\}}) \\ &= (x_{c,\{c\}}, x_{c+1,\{c\}}, \ldots, x_{c+w-1,\{c\}}) \end{aligned}$$

$$x_{\{1\}} = (\underbrace{\varepsilon, \varepsilon, \cdots, \varepsilon}_{y_{\{1\}}^{(0)}}, \varepsilon, \varepsilon, \cdots, \varepsilon)$$

$$\big\downarrow g \quad y_{\{1\}}^{(\infty)}$$

$$x_{\{2\}} = (y_{1,\{1\}}^{(\infty)}, y_{2,\{1\}}^{(\infty)}, \cdots, y_{w,\{1\}}^{(\infty)}, \varepsilon, \varepsilon, \cdots, \varepsilon)$$

$$\underbrace{\qquad\qquad}_{y_{\{2\}}^{(0)}} \xrightarrow{g} y_{\{2\}}^{(\infty)}$$

$$x_{\{3\}} = \left(y_{1,\{1\}}^{(\infty)}, y_{1,\{2\}}^{(\infty)}, \underbrace{\cdots, y_{w,\{2\}}^{(\infty)}, \varepsilon,}_{y_{\{3\}}^{(0)}} \cdots, \varepsilon\right)$$

$$y_{\{3\}}^{(0)} \xrightarrow{g} y_{\{3\}}^{(\infty)}$$

$$x_{\{4\}} = \left(y_{1,\{1\}}^{(\infty)}, y_{1,\{2\}}^{(\infty)}, y_{1,\{3\}}^{(\infty)}, \underbrace{\cdots, y_{w,\{3\}}^{(\infty)}, \varepsilon,}\cdots, \varepsilon\right)$$

$$\vdots$$

$$x_{\{L\}}$$

<u>Threshold</u> $\quad x_{\{L\}} \preceq \delta 1_c.$

$\varepsilon_{WD}^{BP}(\lambda, \rho, \delta, W, L) \qquad W$ large enough, $\varepsilon_{WD}^{BP} \to \varepsilon^{BP}$

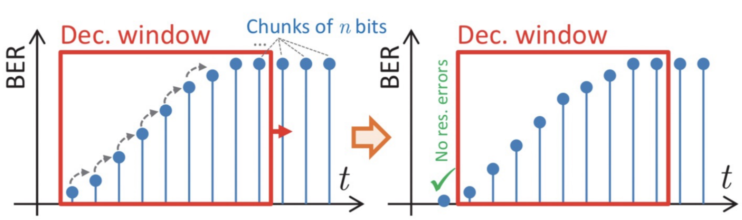


Fig. 2: Illustration of windowed decoding of SC-LDPC codes

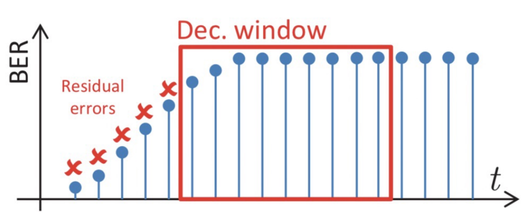


Fig. 3: Illustration of stalling of windowed decoding of SC-LDPC codes

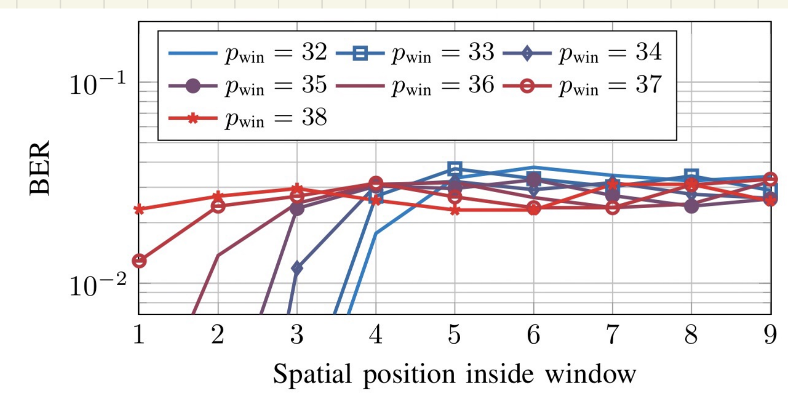


Fig. 1. BER within a window decoder with fixed $I = 3$ iterations of size $w = 9$ for several spatial positions during a decoder stall. The decoder gets stuck at around position 37.

Error propagation: Decoding "stall" if WD moves too fast

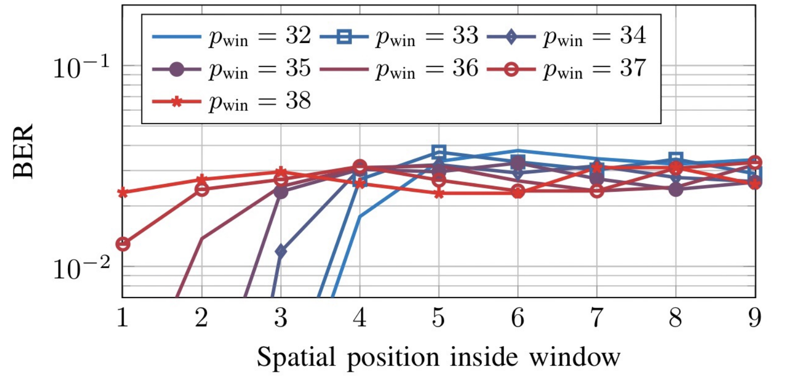


Fig. 1. BER within a window decoder with fixed $I = 3$ iterations of size $w = 9$ for several spatial positions during a decoder stall. The decoder gets stuck at around position 37.

Increasing # iterations per WD could improve the performance.

Stall Detection

① If the CNs are satisfied or not within current WD.

② Soft-decision:

$$P_{stall} = \frac{1}{K+1}\sum_{k=1}^{K}\frac{1}{1+\exp(k-1)}$$

Algorithm 1 Adaptive iterations decoder

Input:
- I_{min} min. number of iter. per window
- I_{max} max. number of iter. per window

```
for p_win = 1 : N_win do
    I ← 0
    while I < I_max do
        CN update, VN update and stall detection
        if I ≥ I_min and stall detection == false then
            break
        end if
        I ← I + 1
    end while
end for
```

Add number of iterations within each WD until no stall detected.

Algorithm 2 Window shift decoder

Input:
- I_{min} min. number of iter. per window
- I_{max} max. number of iter. per window
- n_b number of skipped blocks

```
for p_win = 1 : N_win do
    I ← 0
    while I < I_max do
        CN update, VN update and stall detection
        if I == I_min and stall detection == false then
            break
        else if I == I_min then
            P_stall (win) ← p_win − n_b
        end if
        I ← I + 1
    end while
end for
```

move back n_b positions & increase # of iterations within the WD

Algorithm 3 Wave tracking decoder

Input:
- I_{min} min. number of iter. per window
- I_{max} max. number of iter. per window
- n_b number of skipped blocks

```
for p_win = 1 : N_win do
    flag ← 0
    I ← 0
    while I < I_max do
        CN update, VN update and stall detection
        if I == I_min and stall detection == false then
            break
        else if I == I_min then
            p_win ← p_win − n_b
            flag=1
        end if
        if I > I_min and stall detection == false and flag==1
then
            p_win ← p_win + n_b
            flag=0
        else if I > I_min and stall detection == false then
            p_win ← p_win + 1
        end if
        I ← I + 1
    end while
end for
```

stall $\xrightarrow[-n_b]{I{+}{+}}$ stall $\xrightarrow{I{+}{+}}$ stall $\xrightarrow{I{+}{+}}$ ⋯ → no stall $\xrightarrow[+n_b]{I{+}{+}}$ $\xrightarrow[+1]{I{+}{+}}$

p = p+1