Design of Free-Ride Linear Block Codes: Puncturing Pattern Optimization and Graph Structure

GiGi Chou

Advisor: Prof. Jiun Hung Yu
Prof. Tofar Chih-Yuan Chang

Transmission and Networking Technologies Laboratory,
Institute of Communications Engineering
National Chiao Tung University, Hsinchu, Taiwan

Outline

- **Introduction**
- **Free-Ride Concept**
  - Encoding Scheme
  - Decoding Scheme(Hard/Soft)
  - Simulation
  - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
  - One–Step Recoverable Node Scheme
  - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Introduction

- In many communication systems, alongside the primary payload data, a small number of supplementary bits (e.g., device-ID fields or HARQ ACK/NACK flags) must also be transmitted. Traditionally, these extra bits are encoded and sent separately to meet their unique reliability and timing requirements, but at the cost of additional bandwidth and power. The recently proposed Partial-Superposition Incremental-Redundancy HARQ (PS-IR-HARQ) scheme[6] overcomes this overhead by partially superimposing previous code-block information onto the first HARQ retransmission—embedding both payload and control bits without any extra spectral or power resources, thereby boosting throughput and reliability simultaneously.
- This separation is either because the receiver is only interested in the additional bits or because the reliability requirement of the additional bits is higher. One main drawback of the separated transmission paradigm is that it will require additional bandwidth and power.

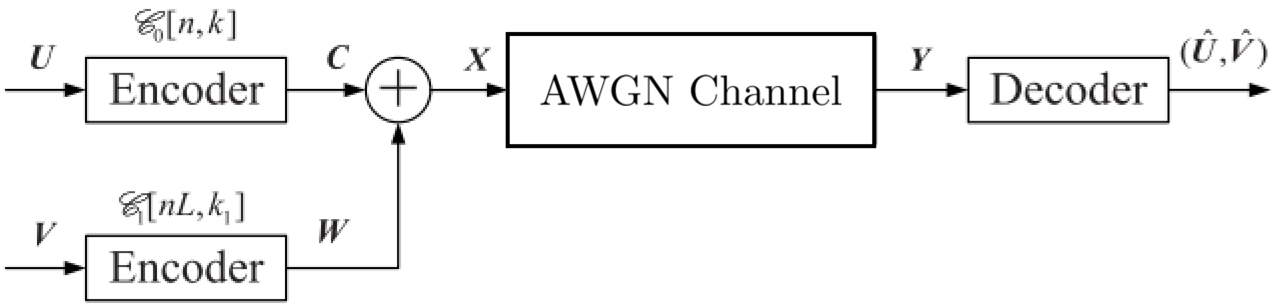
Outline

- **Introduction**
- **Free-Ride Concept**
  - Encoding Scheme
  - Decoding Scheme(Hard/Soft)
  - Simulation
  - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
  - One–Step Recoverable Node Scheme
  - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Free-Ride Concept

- One simple approach is to encode the extra bits and the payload data separately. This separation is either because the receiver is only interested in the extra bits or because the reliability requirement of the extra bits is higher. The main drawback of such an approach is that it will inevitably lead to an extra consumption of transmission power and bandwidth.
- To transmit extra bits along with LDPC coded data, whereby extra bits are randomly encoded and then superimposed on LDPC-coded payload data. Note that by “free-ride”.

Free-Ride – Encoding Concept



Encoding Scheme

$\mathbf{U}$: Payload Information Bits
$\mathbf{V}$: Extra Information Bits
$\mathbf{C}$: Payload CodeWord
$\mathbf{W}$: Extra CodeWord
$\mathbf{X}$: Transmit CodeWord
$\mathbf{Y}$: Receive signal
$\hat{\mathbf{U}}$: Estimated Payload Information bits
$\hat{\mathbf{V}}$: Estimated Extra Information bits

Algorithm 1 Encoding scheme

1: Encode the payload information sequence $\mathbf{U} = (u_0, u_1, \ldots, u_{k-1})$ with the encoder of the LDPC code $\xi_0[n,k]$ to obtain the codeword $\mathbf{C} = (c_0, c_1, \ldots, c_{n-1})$.
2: Encode the extra information sequence $\mathbf{V} = (v_0, v_1, \ldots, v_{l-1})$ with the encoder of the LDPC code $\xi_1[n,k]$ to obtain the codeword $\mathbf{W} = (w_0, w_1, \ldots, w_{n-1})$.
3: Superimpose $\mathbf{W}$ onto $\mathbf{C}$ to obtain the transmitted sequence $\mathbf{X} = (\mathbf{C} + \mathbf{W}) \bmod 2$.

Encoding Algorithm

Free-Ride – Hard Decoding

Algorithm 2 Hard-Decision Decoding of the Scheme With Randomly Coded Extra Bits

1: Make a hard-decision on the received sequence y, resulting in $\hat{y}$.
2: **for all** $V \in \xi_1$ **do** Compute

$$N(W) = W_H\left((\hat{y} + W)\, H^T\right).$$

3: Output $\hat{V}$ such that $N(\hat{V}G_1) = \min_{W \in \xi_1} N(W)$.
4: Remove the interference of $\hat{W} = \hat{V}G_1$ on y, obtaining $\tilde{y}_i = (-1)^{\hat{w}_i} y_i,\ i = 0, \ldots, n-1.$
5: Input $\tilde{y}$ into the LDPC decoder to obtain $\hat{u}$.

Try all extra codeword

Find min hamming distance

Remove extra interference

Hard Decoding Algorithm

Free-Ride – Soft Decoding

$$\Lambda_{\vec{\boldsymbol{x}},j} = \log \frac{P_{Y|X}(y_j \mid 0)}{P_{Y|X}(y_j \mid 1)}, \quad j = 0, 1, \ldots, n-1 \tag{1}$$

$$\Lambda_{\vec{\boldsymbol{x}}+\tilde{\boldsymbol{w}},\, j} = (-1)^{\tilde{w}_j}\, \Lambda_{\vec{\boldsymbol{x}},j}, \quad j = 0, 1, \ldots, n-1,\ \tilde{\boldsymbol{w}} \in \mathbb{F}_2^{\ell} \tag{2}$$

$$\Lambda_{(\vec{\boldsymbol{x}}+\tilde{\boldsymbol{w}})H^T,\, i} = 2\ \tanh^{-1}\left(\prod_{j:h_{ij}=1} \tanh\left(\tfrac{1}{2}\Lambda_{\vec{\boldsymbol{x}}+\tilde{\boldsymbol{w}},\, j}\right) \right), \quad i = 0, 1, \ldots, m-1 \tag{3}$$

$$\Lambda(W) = \sum_{i=0}^{m-1} \Lambda_{(\vec{\boldsymbol{x}}+\tilde{\boldsymbol{w}})H^T,\, i} \tag{4}$$

Algorithm 3 Soft-Decision Decoding of the Scheme With Randomly Coded Extra Bits

1: Compute the LLR $\Lambda_{\vec{x}}$ from the received sequence $\vec{y}$
2: **for all** $\vec{w} \in \xi_1$ **do** compute $\Lambda(W)$ according to equations (2)–(4)
3: Choose $\hat{V}$ that corresponds to the largest $\Lambda(W)$:

$$\Lambda(\hat{v}G_1) = \max_{\vec{w} \in \xi_1} \Lambda(W)$$

4: Remove the interference of $\hat{w} = \hat{v}G_1$ on $\vec{y}$, obtaining:

$$\tilde{y}_i = (-1)^{\hat{w}_i} y_i, \quad i = 0, 1, \dots, n{-}1$$

5: Input $\tilde{y}$ into the LDPC decoder to obtain the estimated payload data $\hat{u}$

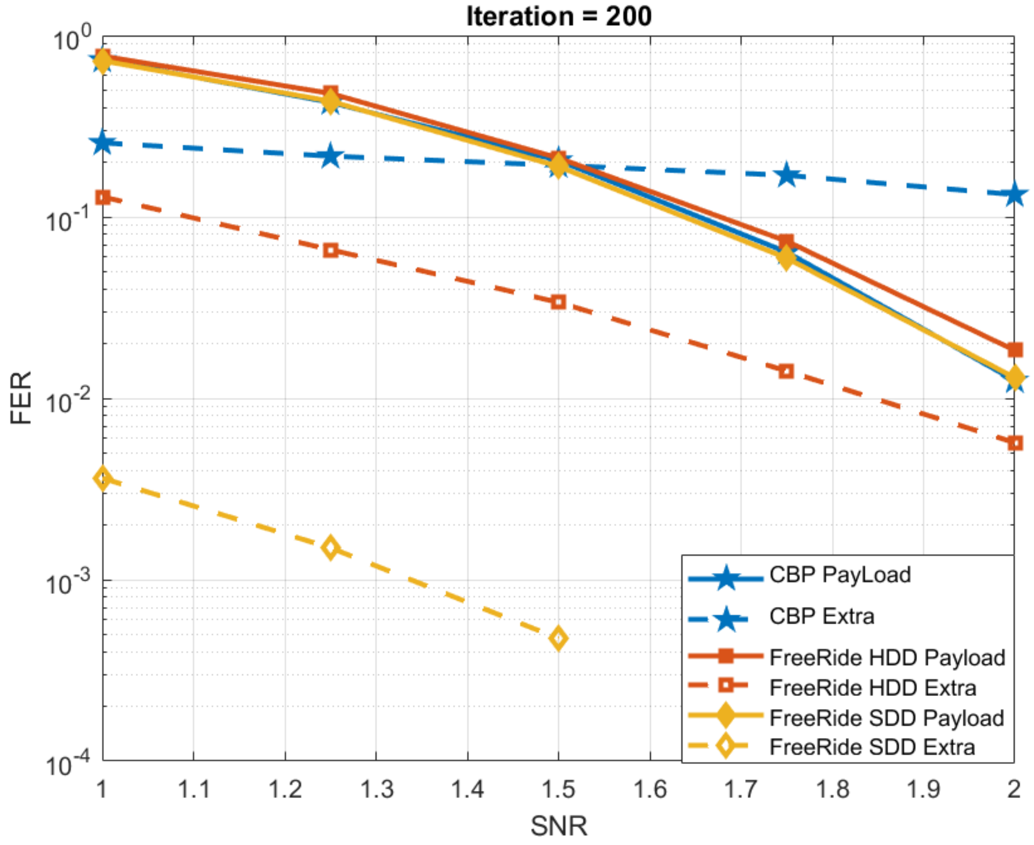
Soft Decoding Algorithm

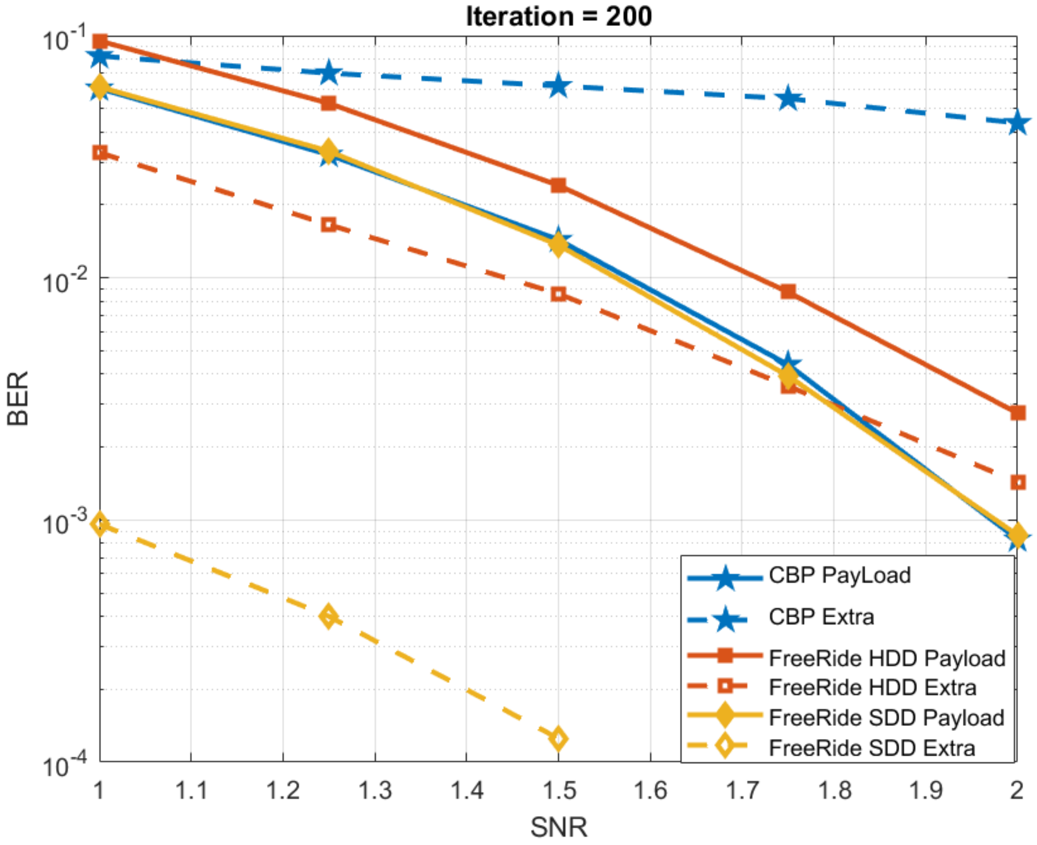
Free-Ride – Code Information

Code	Codeword Length	Information Length	Code Rate	Max Col Degree	Max Row Degree	Girth
PEG-P1	1008	504	0.5	3	8	8
LDPC-E1	10	5	0.5	2	4	6
BCH-E2	15	7	0.47	4	4	6

Free-Ride – Simulation

- Payload : PEG-P1
- Extra : LDPC-E1





Free-Ride – Pros & Cons

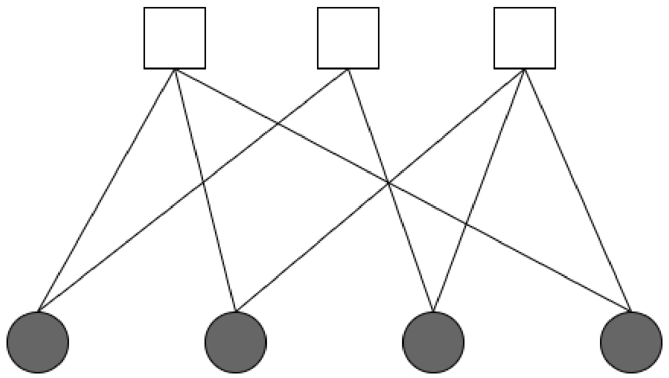
- **Pros**
  - With Free-Ride Decoding, the Extra performance even surpasses simply transmitting the extra bits.
  - Payload performance is almost unchanged.
- **Cons**
  - The Extra decoding employs a maximum likelihood–based decoding approach, so it isn't practical to include too many extra bits in the process.
  - The decoding process follows a strictly serial approach: it must fully complete the extra-bit decoding before it can begin payload decoding.

Outline

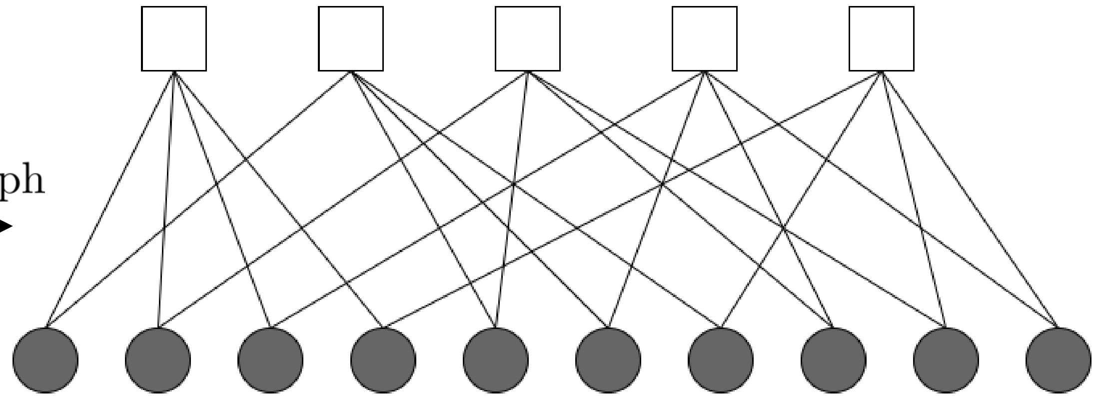
- **Introduction**
- **Free-Ride Concept**
  - Encoding Scheme
  - Decoding Scheme(Hard/Soft)
  - Simulation
  - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
  - One–Step Recoverable Node Scheme
  - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Tanner Graph Combination

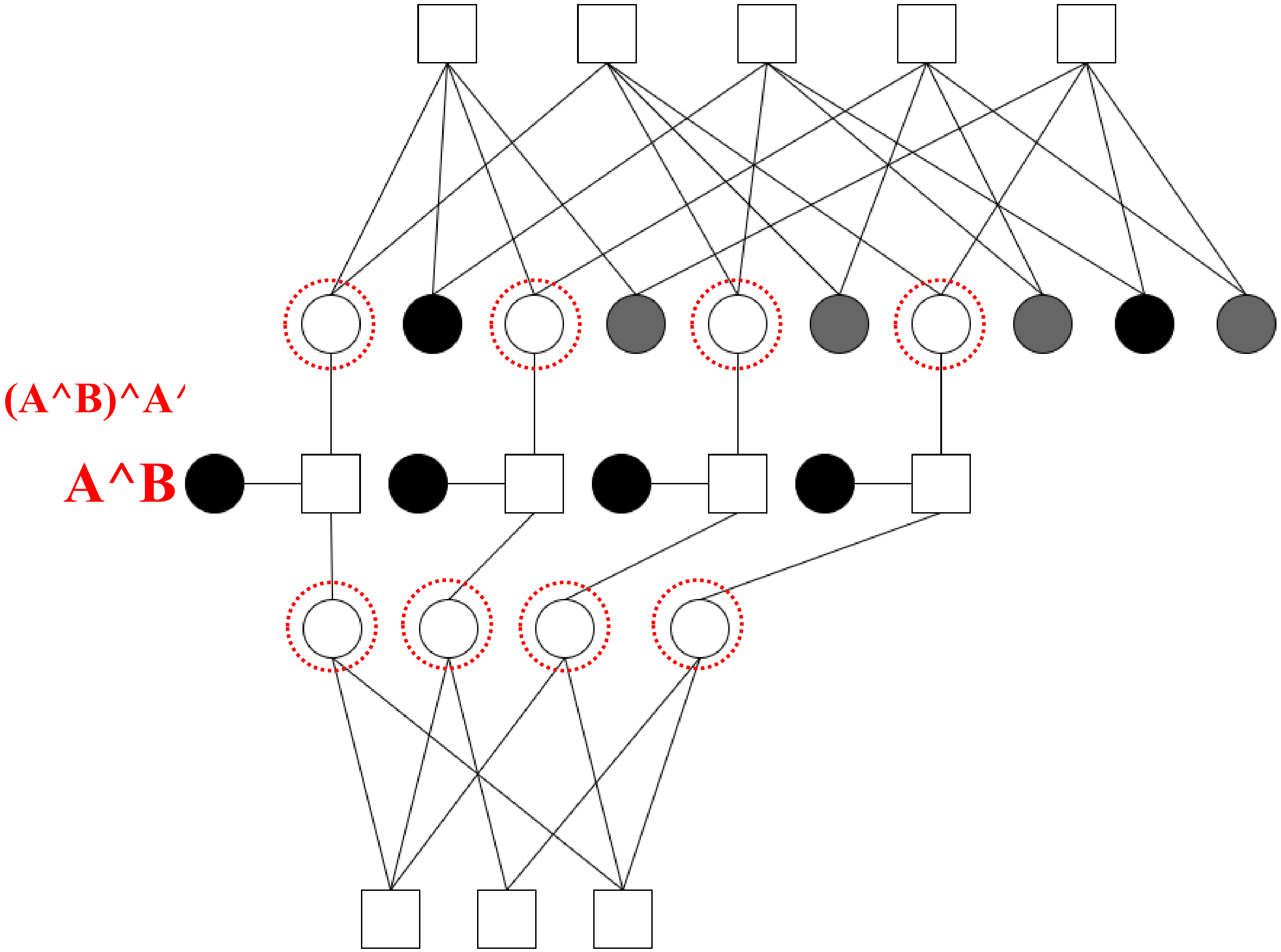
$$H_{extra} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Tanner Graph}}$$



$$H_{payload} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Tanner Graph}}$$



Tanner Graph Combination



Outline

- **Introduction**
- **Free-Ride Concept**
  - Encoding Scheme
  - Decoding Scheme(Hard/Soft)
  - Simulation
  - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
  - One–Step Recoverable Node Scheme
  - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Puncture – Concept

- Puncturing in LDPC (Low-Density Parity-Check) codes refers to the deliberate omission of certain coded bits before transmission to increase the effective code rate. After generating a full LDPC codeword according to a base code rate, the encoder simply removes (punctures) a predefined subset of bits and does not send them over the channel. At the decoder, these punctured positions are treated as erasures; the remaining received bits and the code's sparse parity-check structure are used to infer the missing values during iterative decoding.
- Puncturing lets our adapt the code rate without changing the parity-check matrix but reduces redundancy and degrades error-correction performance. It is typically combined with rate-matching in standards like LTE and 5G NR to optimize throughput and reliability.

Puncture – k-SR

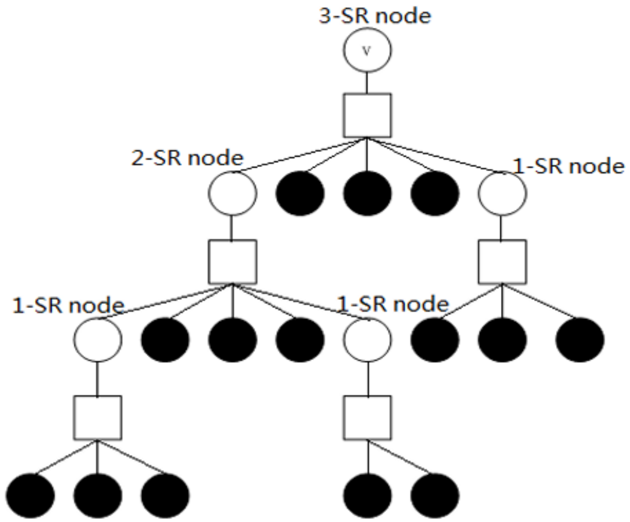
- A punctured variable node can be reliably recovered if it is connected to a sufficient number of check nodes, each of which is itself connected to reliable variable nodes.

- Generally, a punctured variable node V is called k-step recoverable (k-SR) if V has at least one neighbor C, such that

$$N(C) \setminus \{V\}$$

contains at least one $(k-1)$-SR node and all other nodes are m-SR for some $0 \leq m \leq k-1$. The k-SR node is recovered after the k-th iteration.

Recovery Tree



- filled circles: unpunctured nodes
- unfilled circles: punctured nodes

Recovery Error Probability

Let $G_{k>0}$ be the set of variable nodes at depth $\leq k$ in the recovery tree of v. Over a BEC with erasure probability ζ, the recovery error probability of a variable node v is

$$P_e(v) = \tfrac{1}{2}\big(1 - \Phi(v,\zeta)\big),$$

where

$$\Phi(v,\zeta) = (1-\zeta)^{S(v)},$$

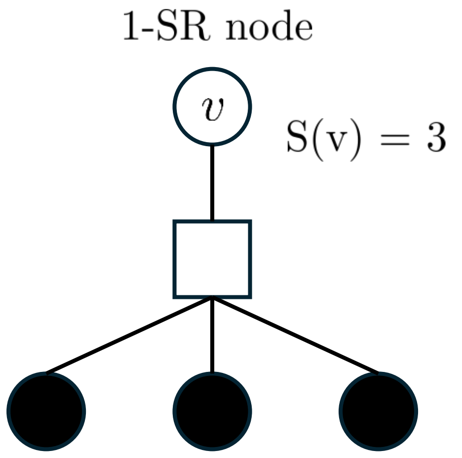
and $S(v)$ is the number of surviving neighbors of v in its recovery tree.

Base Case ($k = 1$)

For $k = 1$, all variable nodes in the recovery tree of v lie in G_0. Each surviving check node has degree d_c, so

$$P_e(v) = \tfrac{1}{2}\big(1 - (1-\zeta)^{d_c-1}\big) = \tfrac{1}{2}\big(1 - (1-\zeta)^{S(v)}\big) = \tfrac{1}{2}\big(1 - \Phi(v,\zeta)\big).$$

Here d_c is the degree of the surviving check node of v.



- filled circles: unpunctured nodes
- unfilled circles: punctured nodes

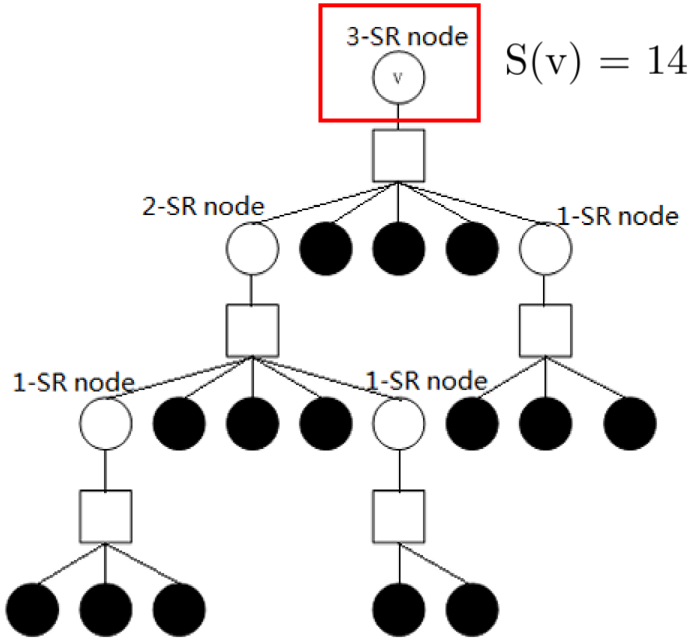
Induction Step $(k > 1)$

For any $v \in G_{k+1}$ with surviving check-node degree d_c and neighbors $\gamma_1, \ldots, \gamma_{d_c-1} \in G_h$ $(1 \le h \le k)$, define

$$S(v) = \sum_{j=1}^{d_c-1} S(\gamma_j), \quad \Phi(x,\zeta) = (1-\zeta)^{S(x)}.$$

Then

$$\begin{aligned} P_e(v) &= \tfrac{1}{2}\Big(1 - \prod_{j=1}^{d_c-1} \Phi(\gamma_j,\zeta)\Big) \\ &= \tfrac{1}{2}\Big(1 - (1-\zeta)^{\sum_{j=1}^{d_c-1} S(\gamma_j)}\Big) \\ &= \tfrac{1}{2}\Big(1 - (1-\zeta)^{S(v)}\Big) \\ &= \tfrac{1}{2}\big(1 - \Phi(v,\zeta)\big). \end{aligned}$$



- filled circles: unpunctured nodes
- unfilled circles: punctured nodes

Puncture – k-SR Algorithm[4]

Algorithm 4 Puncturing Node Grouping

1: **Input:** Parity-check matrix $H \in \{0,1\}^{m\times n}$

2: **[Initialize]**

$$k \leftarrow 1, \quad R_0 \leftarrow \emptyset,\ R_1 \leftarrow \emptyset,\ R_\infty \leftarrow \{1,2,\ldots,m\},$$

$$\Gamma^\rho \leftarrow \{\gamma : H_{\rho,\gamma} = 1, and\ 1 \le \gamma \le n\}, \quad \Lambda^\gamma \leftarrow \{\rho : H_{\rho,\gamma} = 1, and\ 1 \le \rho \le m\},$$

$$G_0 \leftarrow \emptyset,\ G_1 \leftarrow \emptyset,\ G_\infty \leftarrow \{1,2,\ldots,n\},\ S(j) \leftarrow 0\ for\ all\ 1 \le j \le n$$

3: **[Group Column Indices]** For each $\rho \in R_\infty$, set $\vartheta^\rho_\infty = \Gamma^\rho \cap G_\infty$

4: **[Find Candidate Rows]** Make a subset of R_∞ (call it Ω) such that

$$\forall \omega \in \Omega, \quad |\vartheta^\omega_\infty| = rw^{\min}_{\text{eff}} \ \le\ |\vartheta^\rho_\infty| = rw_{\text{eff}}(\rho, G_\infty) \quad \forall \rho \in R_\infty.$$

5: **[Group Row Indices]** Make a set C^ω_∞ such that

$$C^\gamma_\infty \ = \ \Lambda^\gamma \cap R_\infty, \quad \forall \gamma \in \vartheta^\omega_\infty,\ \omega \in \Omega.$$

6: **[Find Best Rows]** Find a subset of Ω (call it Ω^*) such that

$$\forall \omega^* \in \Omega^*, \ \exists c \in \vartheta^{\omega^*}_\infty \quad such\ that \quad |C^c_\infty| = cw^{\min}_{\text{eff}} \ \le\ |C^\gamma_\infty| = cw_{\text{eff}}(\gamma, R_\infty)$$

$$for\ any\ \omega \in \Omega\ and\ \gamma \in \vartheta^\omega_\infty.$$

7: **[Make a Set of Ordered Pairs]** Pick a column index

$$c^* \ \in\ G^{\omega^*}_\infty \quad with \quad cw^{\min}_{\text{eff}}$$

randomly , when there are more than one column index with $cw^{\min}_{\text{eff}}$ .Then, we will have an ordered pair

$$\mathcal{O} = \{(\omega^*_1, c^*_1),\ (\omega^*_2, c^*_2),\ \ldots,\ (\omega^*_p, c^*_p)\},$$

where each ω^*_j and c^*_j is a row- and column-index pair satisfying

$$rw^{\min}_{\text{eff}} \quad and \quad cw^{\min}_{\text{eff}}, \quad 1 \le j \le |\mathcal{O}| = p.$$

8: **[Find The Best Pair]** Pick a pair (ω^*, c^*) from $\mathcal{O}$ such that

$$\mathcal{W}(\omega^*) \le \mathcal{W}(\omega^*_j), \quad 1 \le j \le p, \quad \mathcal{W}(\omega^*_j) \ =\ \sum_{\gamma \in \Gamma^{\omega^*_j}} S(\gamma).$$

If there are more than one pair satisfying the inequality, pick one randomly.

9: **[Update]**

$$G_k \leftarrow G_k \cup \{c^*\}, \quad G_0 \leftarrow G_0 \cup (\vartheta^{\omega^*}_\infty \setminus \{c^*\}),$$

$$G_\infty \leftarrow G_\infty \setminus \vartheta^{\omega^*}_\infty, \quad R_k \leftarrow R_k \cup \{\omega^*\},$$

$$R_0 \leftarrow R_0 \cup (C^{c^*}_\infty \setminus \{\omega^*\}), \quad R_\infty \leftarrow R_\infty \setminus C^{c^*}_\infty,$$

$$S(\gamma) = 1 \quad \forall \gamma \in \vartheta^{\omega^*}_\infty \setminus \{c^*\}, \quad S(c^*) = \sum_{\gamma \in \Gamma^{\omega^*} \setminus \{c^*\}} S(\gamma).$$

10: **[Check Stop Condition]** If G_∞ is an empty set, then **STOP**.

11: **[Decision]** If $R_\infty \neq \emptyset$, go to 3.

12: **[No More Undetermined Rows]**

$$R_\infty = \{\rho : \rho \in R_0\ and\ rw_{\text{eff}}(\rho, G_\infty) > 0\}.$$

13: $k \leftarrow k+1$, and go to 3.

Find most 1-SR → 2-SR → … → k-SR

Puncture – RC Scheme Algorithm[3]

Algorithm 5 Rate-Compatible Puncturing Scheme

1: **Inputs:** Parity-check matrix (Tanner graph) of $C(r_0)$; sequence of rates $r_1, \ldots, r_m$; $\eta_{\max}$; $\ell_{\max}$. Set k =1.

2: Find the set UPset_k, and calculate P_k(node number). Also, for all $c \in V_c$, calculate $F(c)$ and $U(c)$, and set counter $\leftarrow 0$.

3: Let

$$\Psi = \{c^* \in V_c : F(c^*) \le F(c)\ \forall c \in V_c\},$$

then set

$$\text{CandidateCheck} = \{c^* \in \Psi : U(c^*) \le U(c)\ \forall c \in \Psi\}.$$

4: Let

$$\text{CandidateVar} = \left\{v : v \in N(c)\ \forall c \in \text{CandidateCheck}\right\} \cap \text{UPset}_k.$$

5: Let

$$\Gamma = \left\{v^* \in \text{CandidateVar} : H(v^*) \le H(v)\ \forall v \in \text{CandidateVar}\right\}.$$

If $|\Gamma| = 1$, $v_p = \Gamma$ and go to Step 10.

6: Let

$$\Delta = \left\{v^* \in \Gamma : d(v^*) \le d(v)\ \forall v \in \Gamma\right\}.$$

If $|\Delta| = 1$, $v_p = \Gamma$ and go to Step 10.

7: Let

$$\Theta = \left\{v^* \in \Delta : K(v^*) \le K(v)\ \forall v \in \Delta\right\}.$$

If $|\Theta| = 1$, $v_p = \Gamma$ and go to Step 10.

8: Set $\ell \leftarrow 4$ and $\Lambda \leftarrow \Theta$.
   (a) $\Lambda \leftarrow \{v^* \in \Lambda : S^{v^*}_{\eta_{\max}}(\ell) \le S^{v}_{\eta_{\max}}(\ell)\ \forall v \in \Lambda\}$.
   (b) **If** $|\Lambda| = 1$ **then** $v_p = \Gamma$ and go to Step 10.
   (c) **If** $\ell < \ell_{\max}$ **then** $\ell \leftarrow \ell + 2$ and **go to** Step 8(a).

9: Choose v_p at random from Λ.

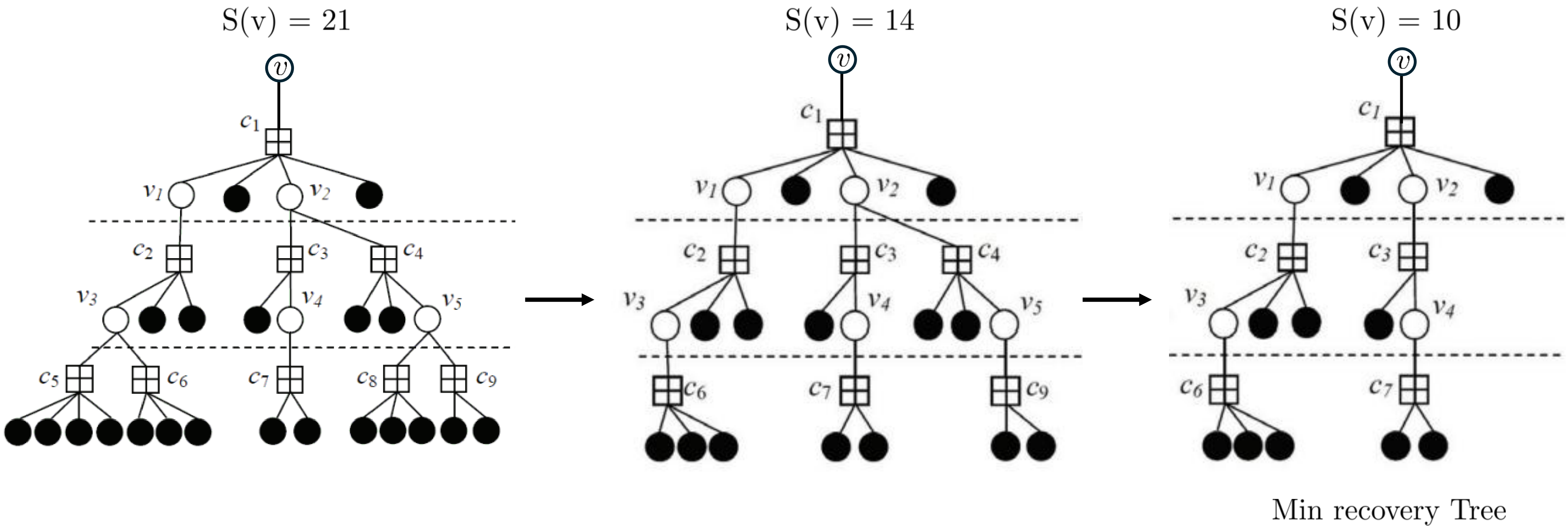
10: Puncture v_p, set

$$\text{counter} \leftarrow \text{counter} + 1, \quad \text{UPset}_k \leftarrow \text{UPset}_k \setminus \{v_p\}, \quad \text{Pset}_k \leftarrow \text{Pset}_k \cup \{v_p\}.$$

11: **if** counter $< P_k$, $\forall c \in N(v_p)$, set $F(c) \leftarrow F(c)+1$, and $\forall c \in \{c : v_p \in T(c)\}$, update $U(c)$. Also, $\forall v \in B(v_p)$, set $H(v) \leftarrow H(v) + 1$, and Go to Step 3. Otherwise, if $k < m$, then $k \leftarrow k + 1$, and go to Step 2.

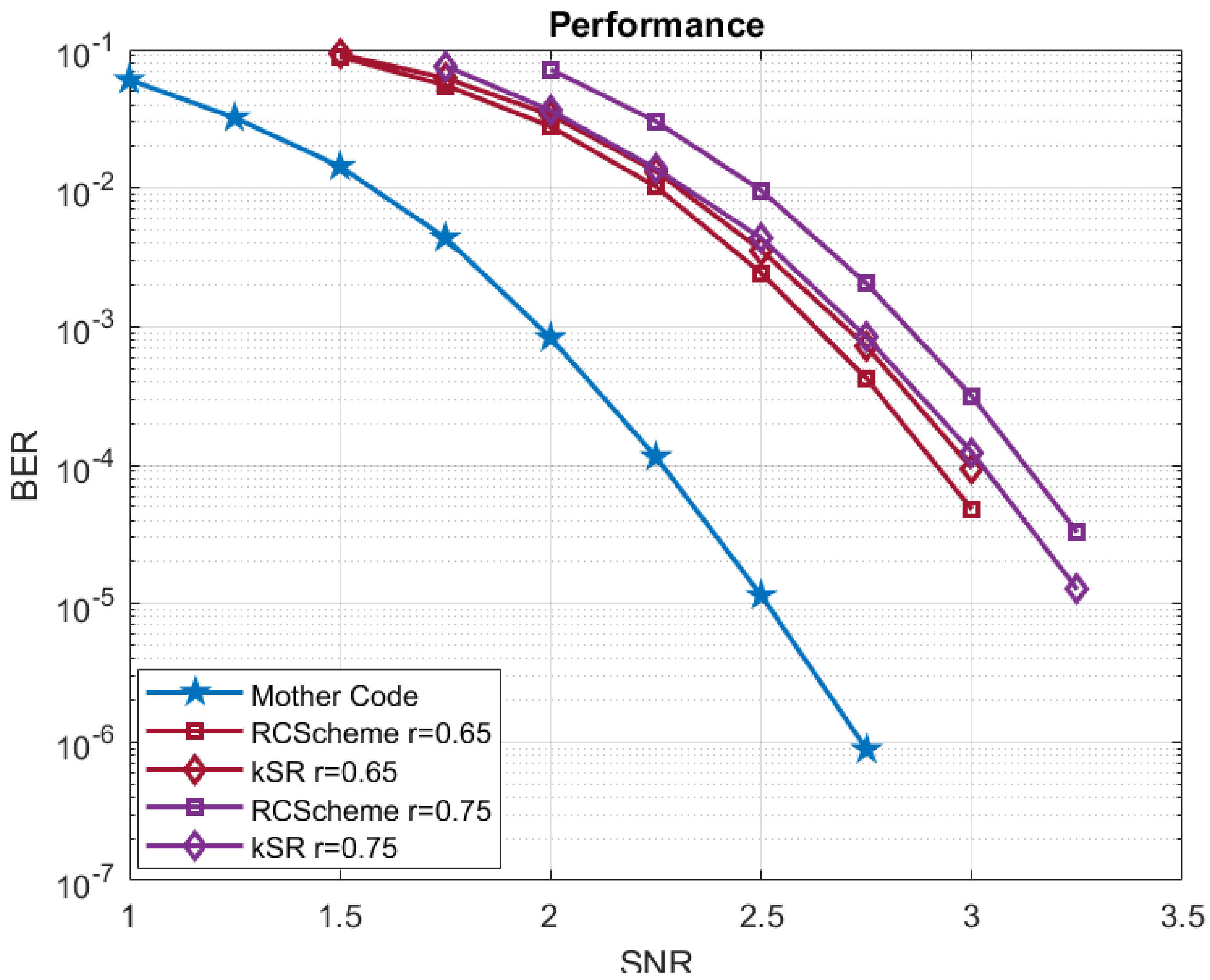
12: Stop

Puncture – RC Scheme Algorithm



Puncture – simulation

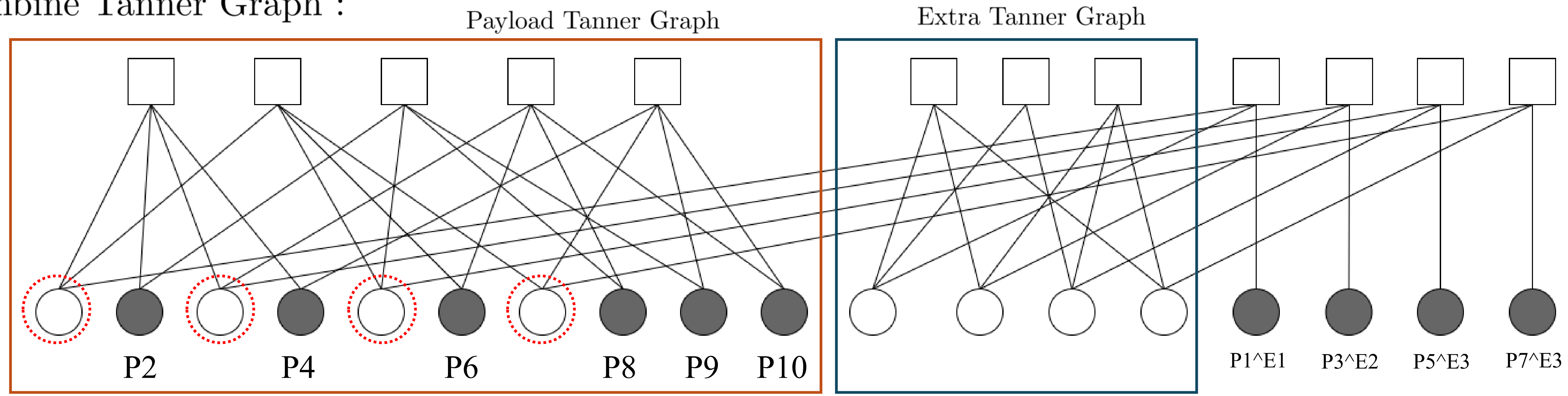
Compare different puncturing algorithm :



Outline

- **Introduction**
- **Free-Ride Concept**
  - Encoding Scheme
  - Decoding Scheme(Hard/Soft)
  - Simulation
  - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
  - One–Step Recoverable Node Scheme
  - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Combine Tanner Graph :



- filled circles: unpunctured variable nodes
- unfilled circles: punctured variable nodes

$$\boldsymbol{H}_{Combine}: \begin{bmatrix} \boldsymbol{H}_{\text{payload}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_{\text{extra}} & \mathbf{0} \\ \boldsymbol{M}_{\text{Puncpos}} & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}$$

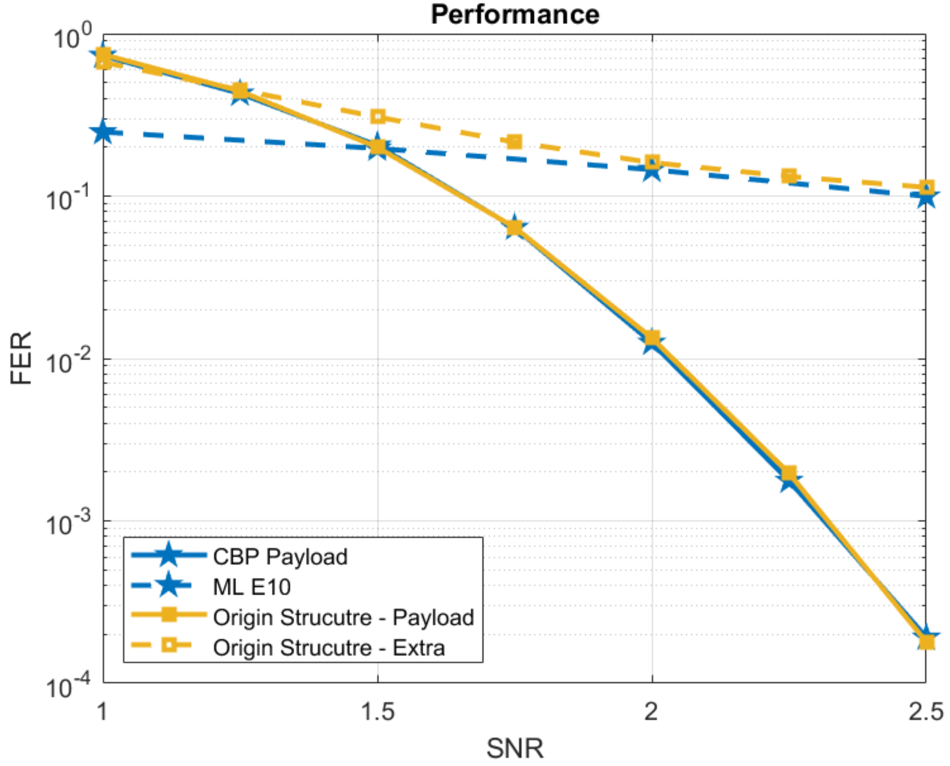
$M_{puncpos}$ each row degree is 1.

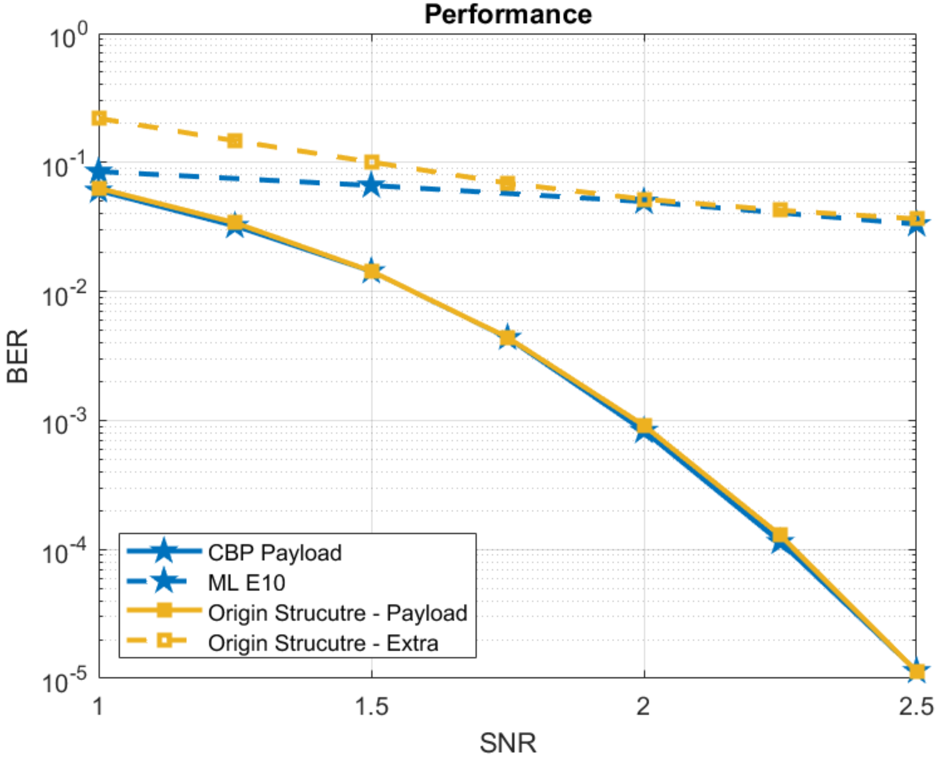
Origin Structure – Code-Rate Fix

- In our structured Tanner graph, we fix the code-rate r as

$$r = \frac{\text{length(payload info)} + \text{length(extra info)}}{\text{length(payload encode)}}$$

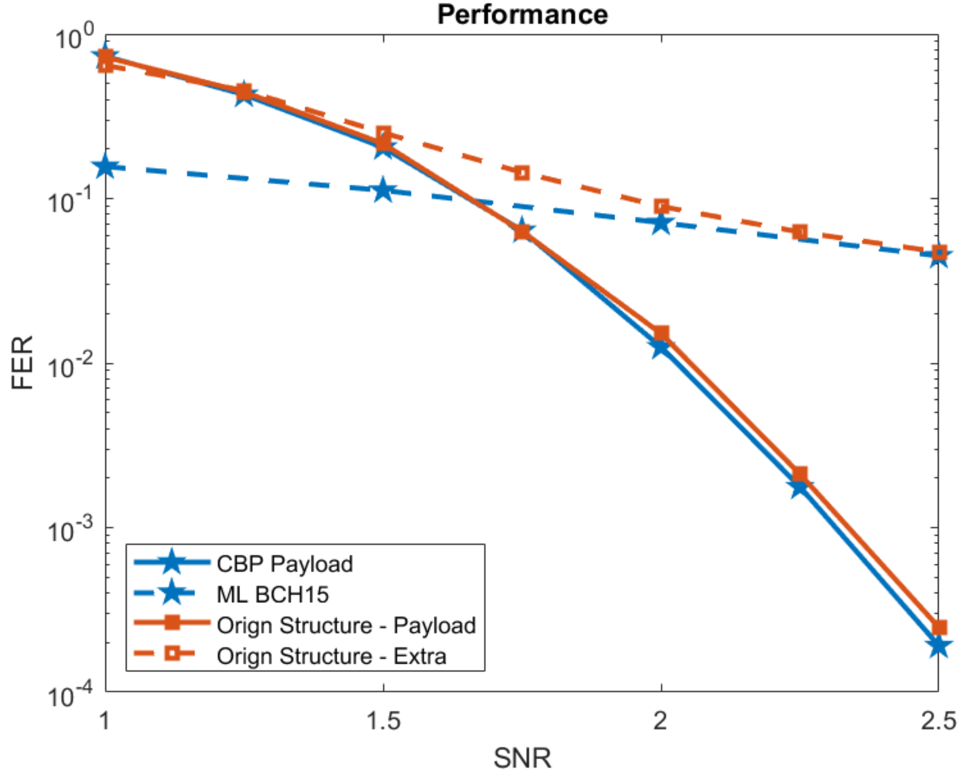
- Payload : PEG-P1
- Extra : LDPC-E1

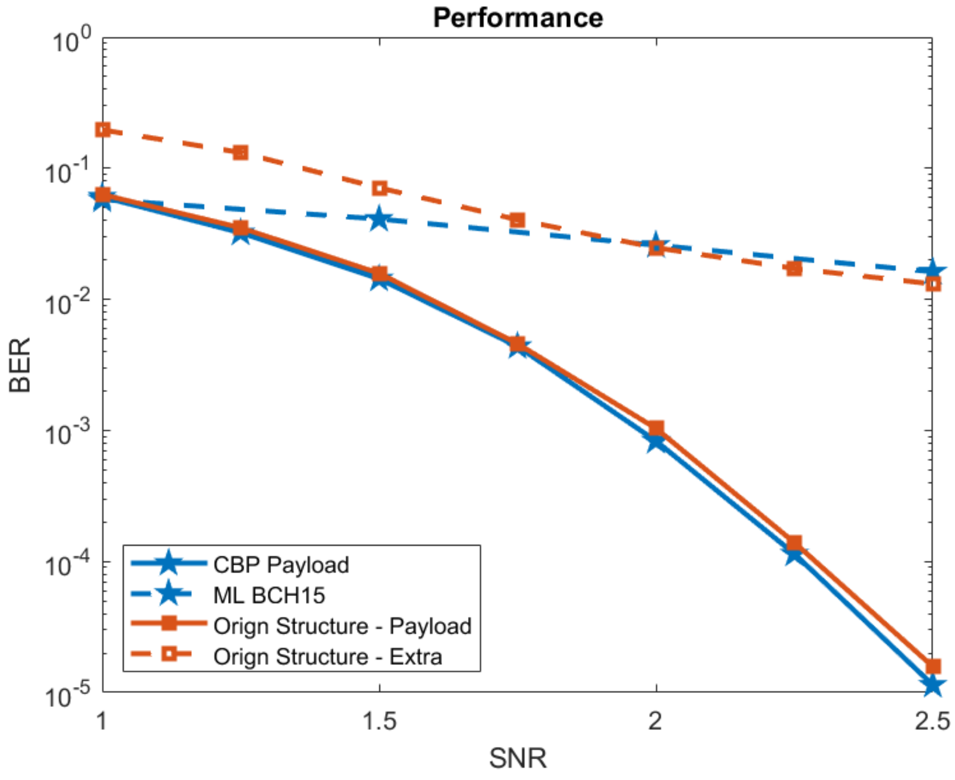




Origin Structure – Simulation

- Payload : PEG-P1
- Extra : BCH-E2





Origin Structure – Observation

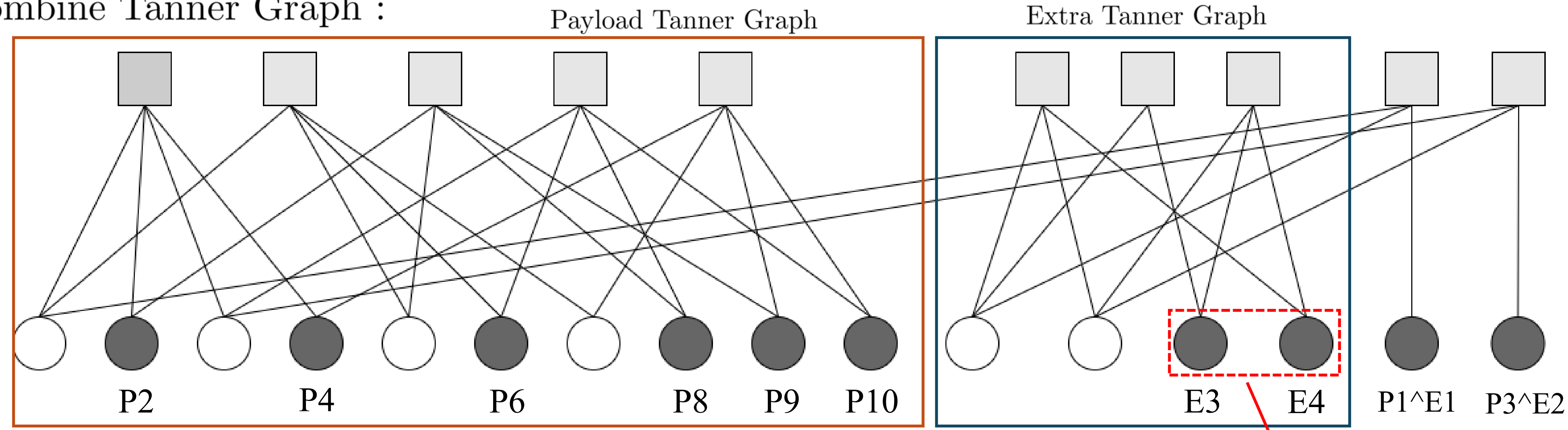
- Merging the Tanner graph allows Extra and Payload to be decoded at the same time, and the payload's performance is barely affected.
- This method of combining Tanner graphs allows Payload and Extra to be decoded in parallel.

Outline

- **Introduction**
- **Free-Ride Concept**
    - Encoding Scheme
    - Decoding Scheme(Hard/Soft)
    - Simulation
    - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
    - One–Step Recoverable Node Scheme
    - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Partial-Extra Structure

Combine Tanner Graph :



- filled circles: unpunctured variable nodes
- unfilled circles: punctured variable nodes

Transmit parital extra bits(random choose)

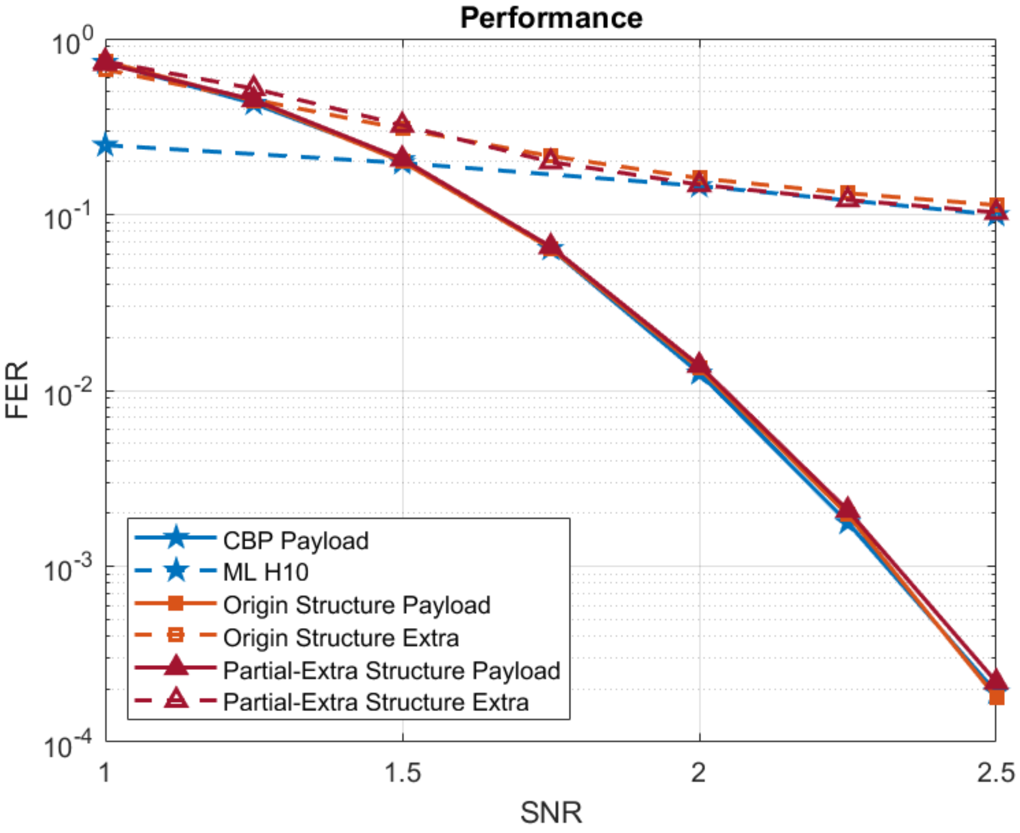
$$\boldsymbol{H}_{Combine}: \begin{bmatrix} \boldsymbol{H}_{\text{payload}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_{\text{extra}} & \mathbf{0} \\ \boldsymbol{M}_{\text{Puncpos}} & \boldsymbol{M}_{\text{extrapunc}} & \boldsymbol{I} \end{bmatrix}$$

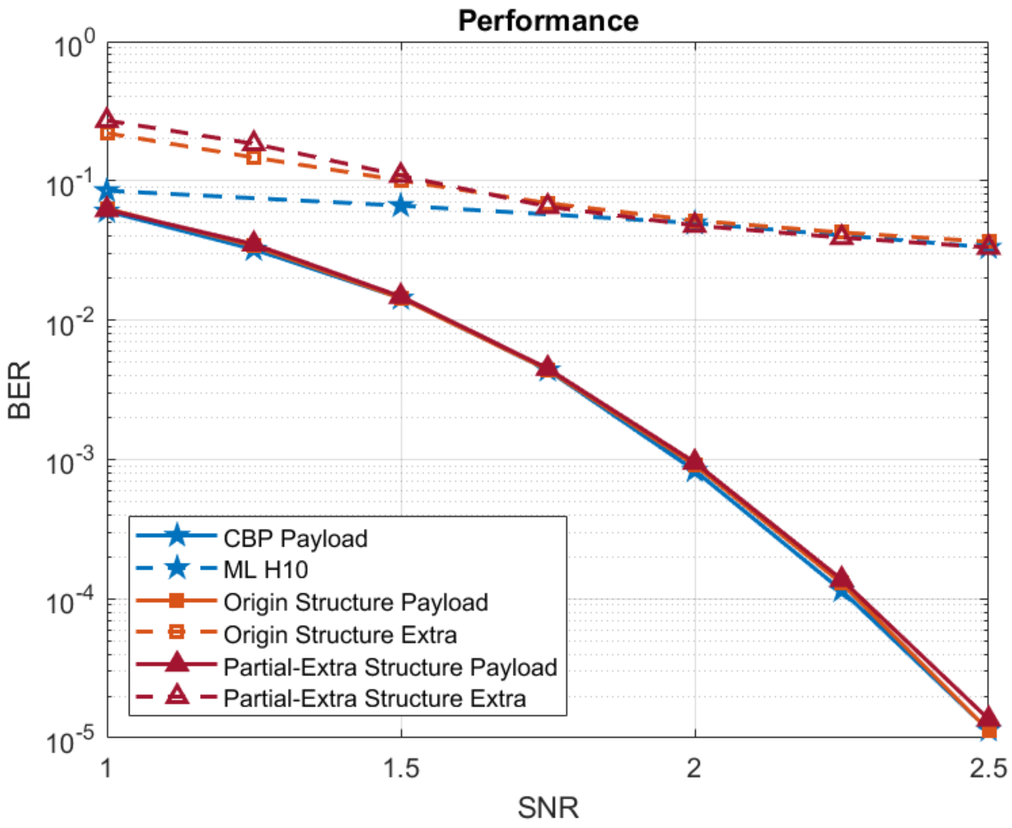
$M_{Puncpos}$ each row degree is 1.

$M_{extrapunc}$ each row degree is 1.

Partial-Extra Structure – Simulation

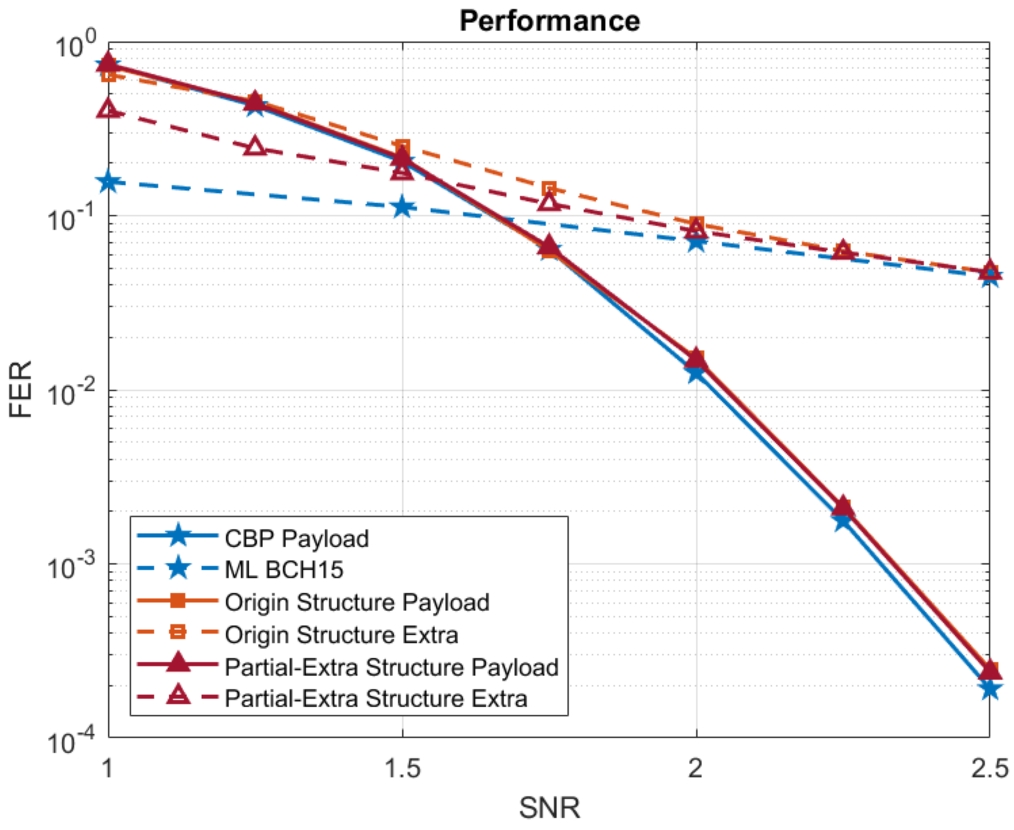
- Payload：PEG-P1
- Extra：LDPC-E1

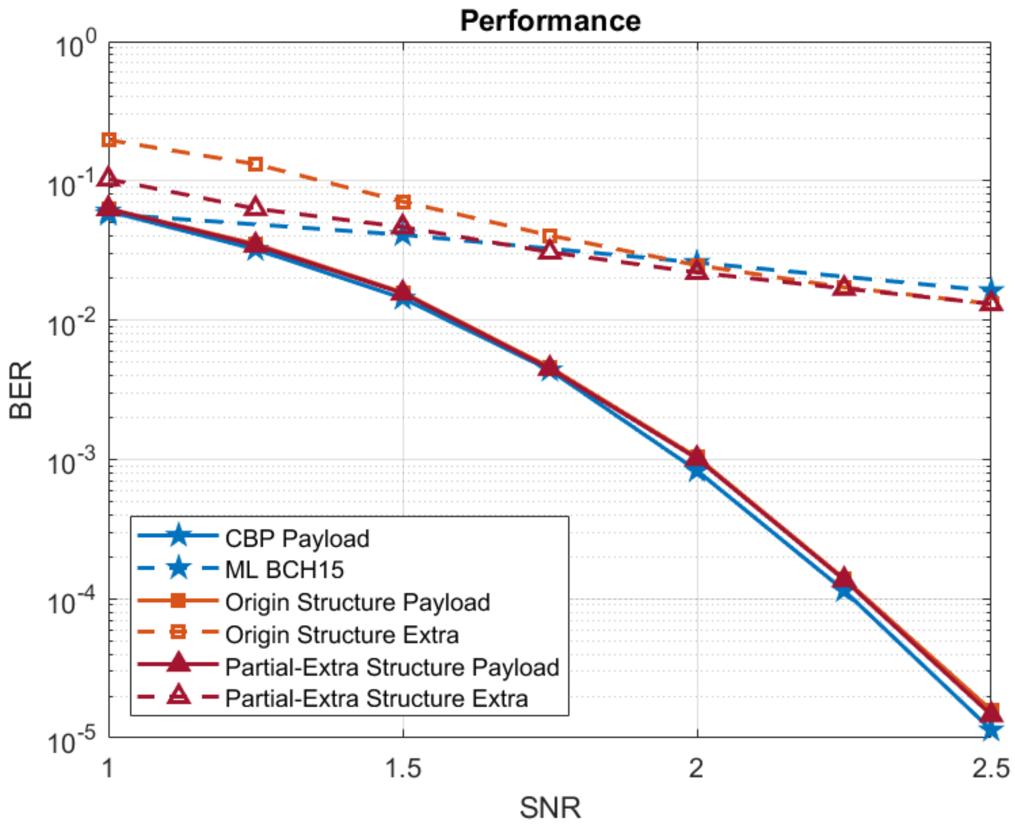




Partial-Extra Structure – Simulation

- Payload : PEG-P1
- Extra : BCH-E2





Partial-Extra Structure – Observation

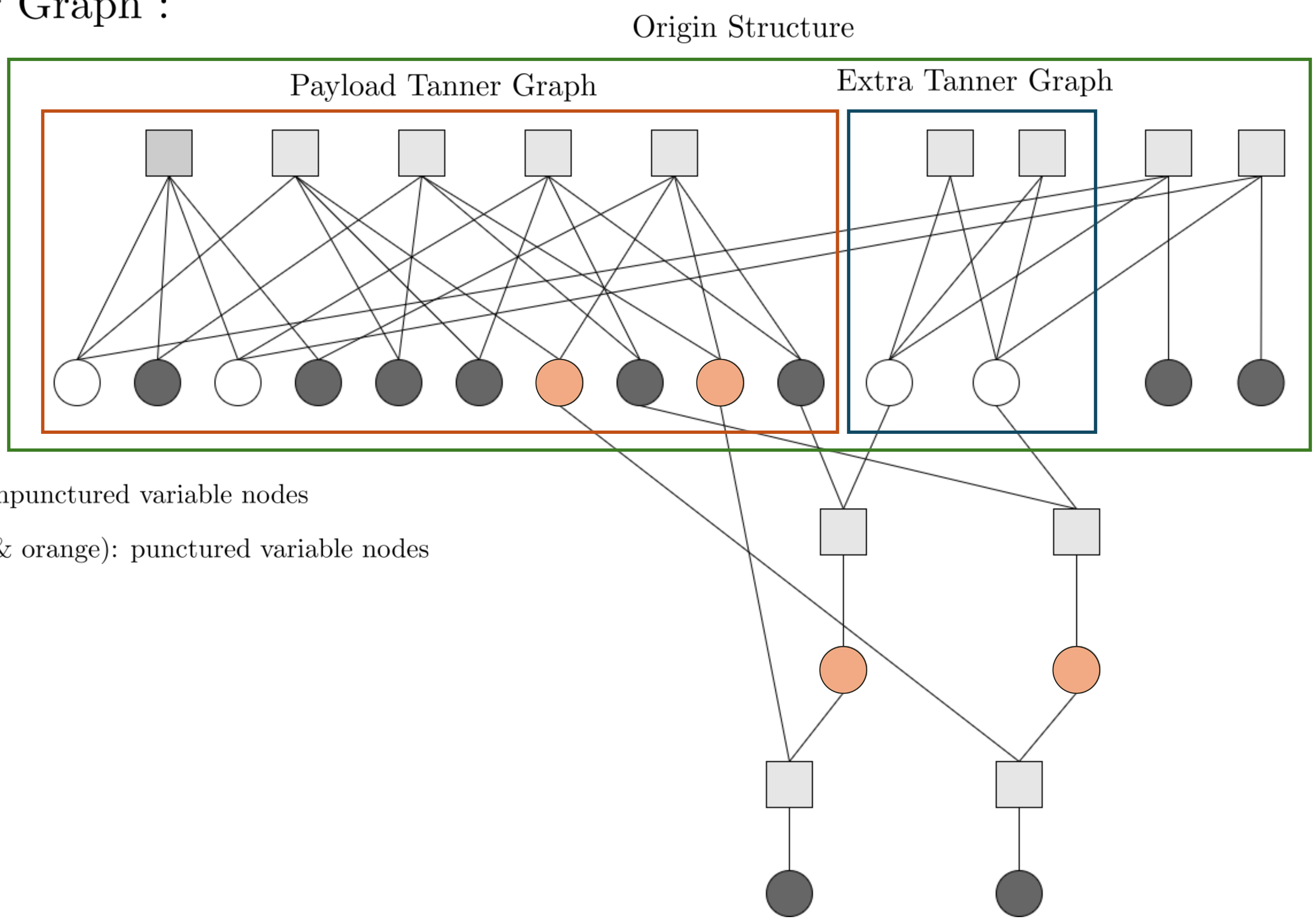
- Merging the Tanner graph allows Extra and Payload to be decoded at the same time, and the payload's performance is barely affected.
- This method of combining Tanner graphs allows Payload and Extra to be decoded in parallel.
- Sending some extra bits can improve Extra's performance at low SNR.

Outline

- **Introduction**
- **Free-Ride Concept**
  - Encoding Scheme
  - Decoding Scheme(Hard/Soft)
  - Simulation
  - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
  - One–Step Recoverable Node Scheme
  - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Enhanced Structure

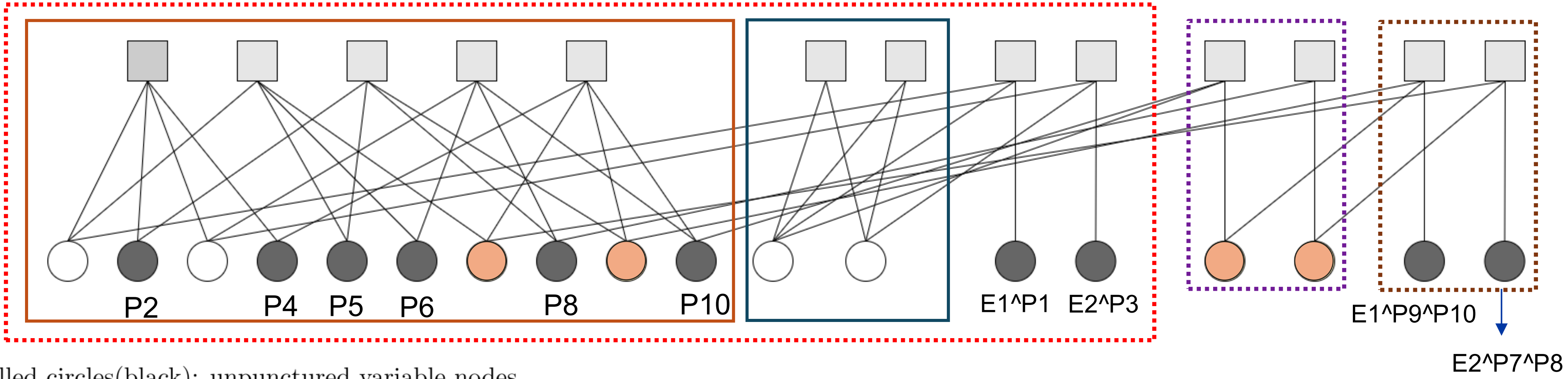
Combine Tanner Graph :



- filled circles(black): unpunctured variable nodes
- unfilled circles(white & orange): punctured variable nodes

Enhanced Structure

Combine Tanner Graph :



- filled circles(black): unpunctured variable nodes
- unfilled circles(white & orange): punctured variable nodes

$$\boldsymbol{H}_{Combine}: \begin{bmatrix} \boldsymbol{H}_{\text{payload}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_{\text{extra}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{M}_{\text{Puncpos}} & \boldsymbol{I} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{M}_{\text{Enhanced1}} & \boldsymbol{I} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \boldsymbol{M}_{\text{Enhanced2}} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}$$

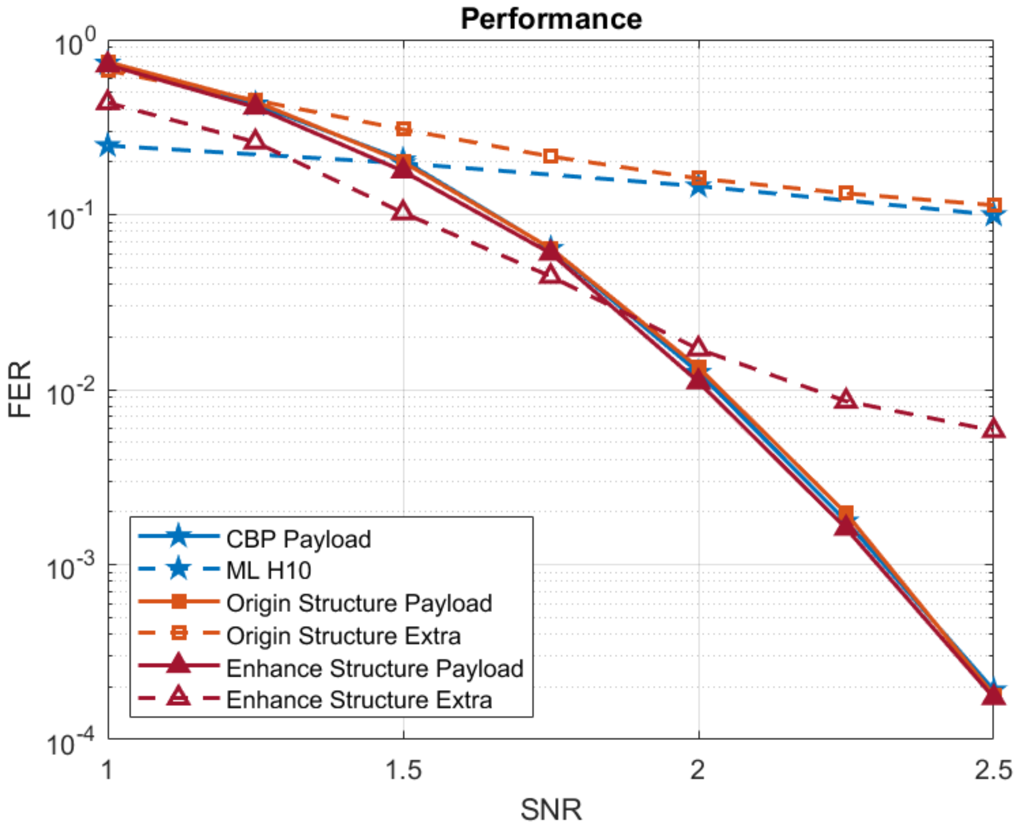
$M_{puncpos}$ each row degree is 1.

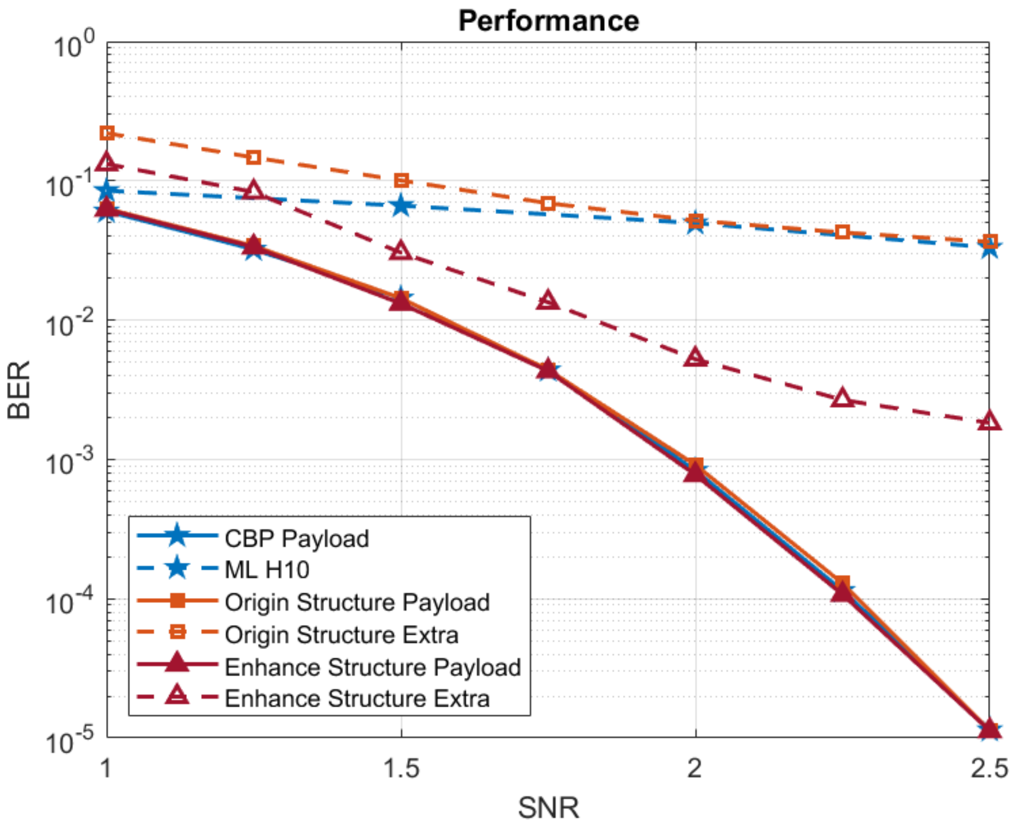
$M_{Enhanced1}$ each row degree is 1.

$M_{Enhanced2}$ each row degree is 1.

Enhanced Structure – Simulation

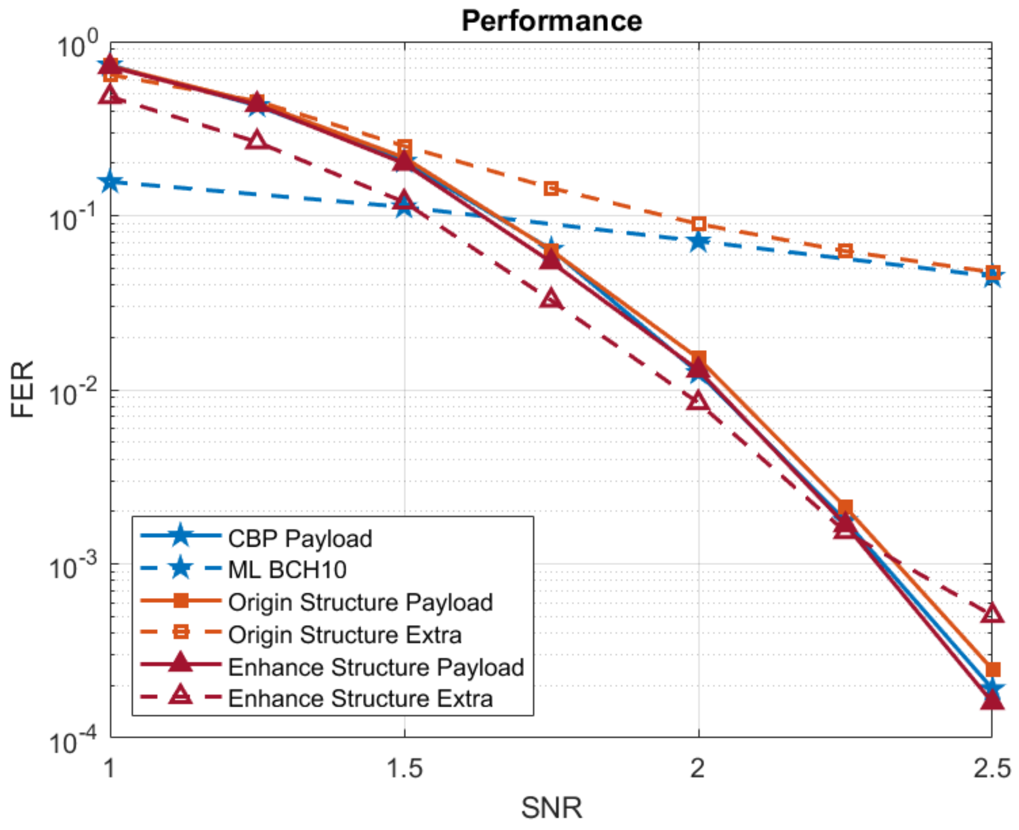
- Payload : PEG-P1
- Extra : LDPC-E1

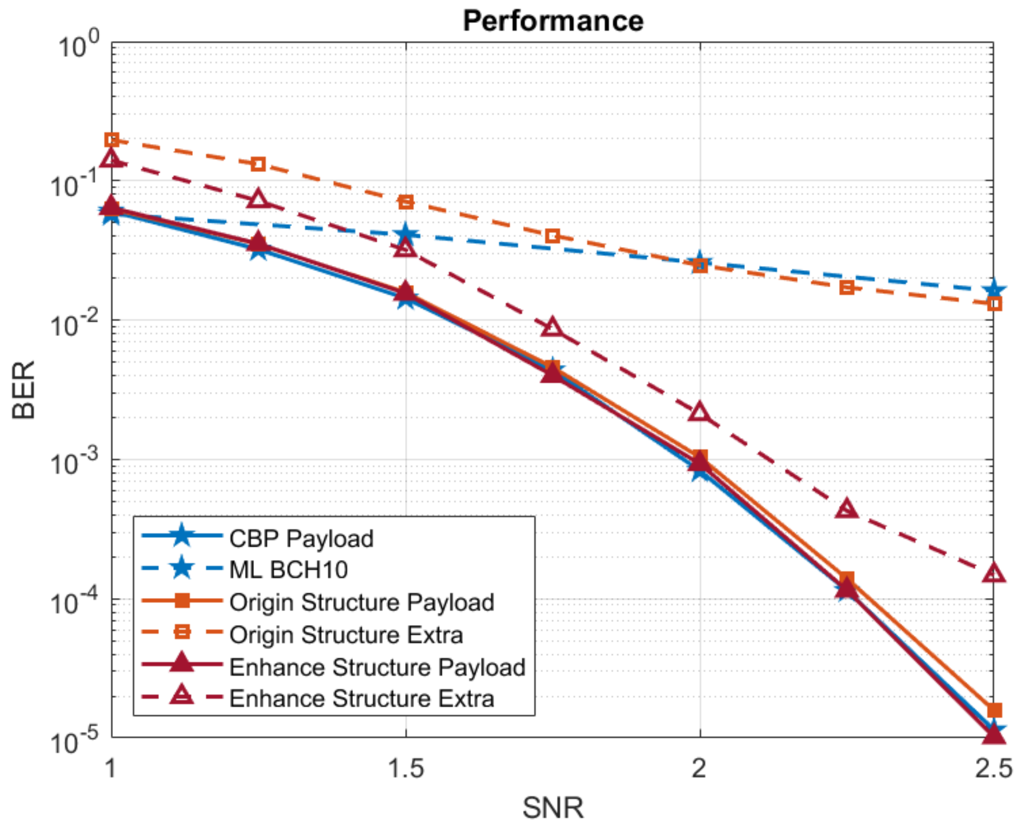




Enhanced Structure – Simulation

- Payload : PEG-P1
- Extra : BCH-E2





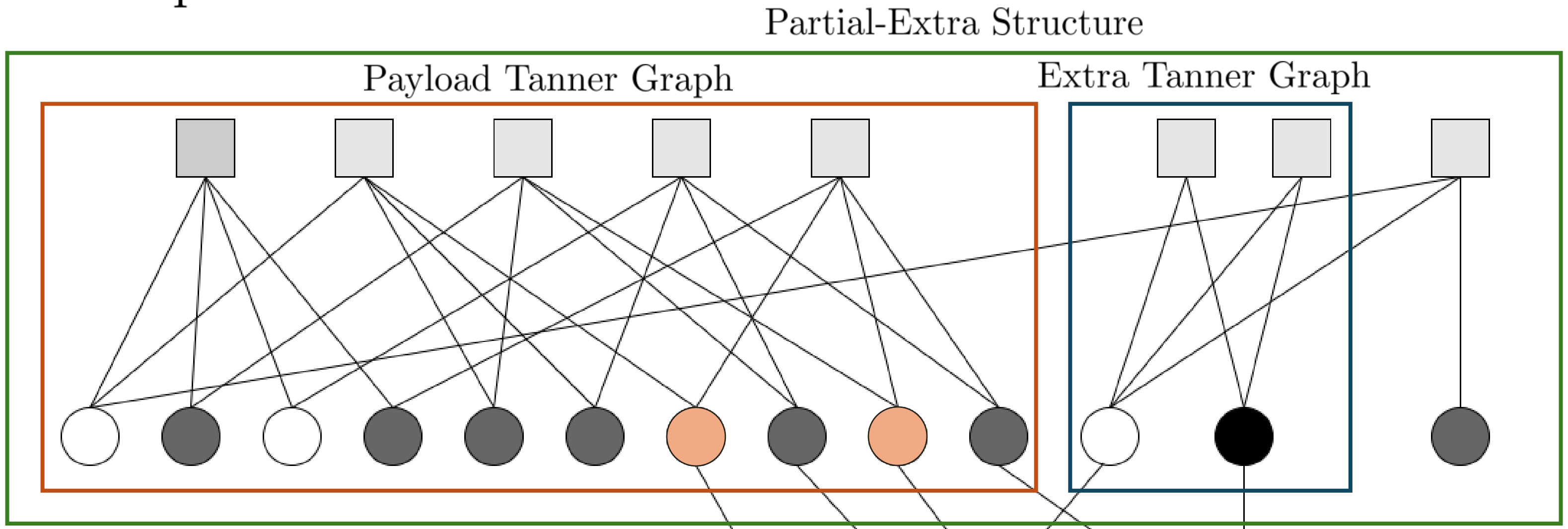
Enhanced Structure – Observation

- We observed a significant improvement in the performance of the extra bits, while the performance of the payload remained unchanged.
- The Enhanced Structure can be seen as a large parity-check matrix, where both the payload and the extra bits are part of it. This is why the extra bits can perform even better than maximum likelihood decoding.

Outline

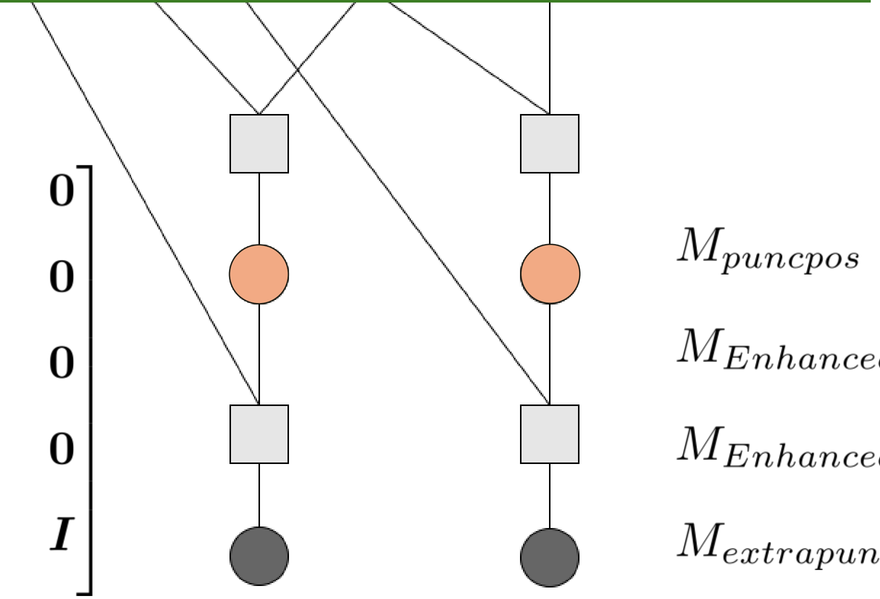
Partial-Extra & Enhanced Structure

Combine Tanner Graph :



- filled circles(black): unpunctured variable nodes
- unfilled circles(white & yellow): punctured variable nodes

$$\boldsymbol{H}_{Combine}: \begin{bmatrix} \boldsymbol{H}_{\text{payload}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_{\text{extra}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}_{\text{Puncpos}} & \boldsymbol{M}_{\text{extrapunc}} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}_{\text{Enhanced1}} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{M}_{\text{Enhanced2}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} \end{bmatrix}$$



$M_{puncpos}$ each row degree is 1.

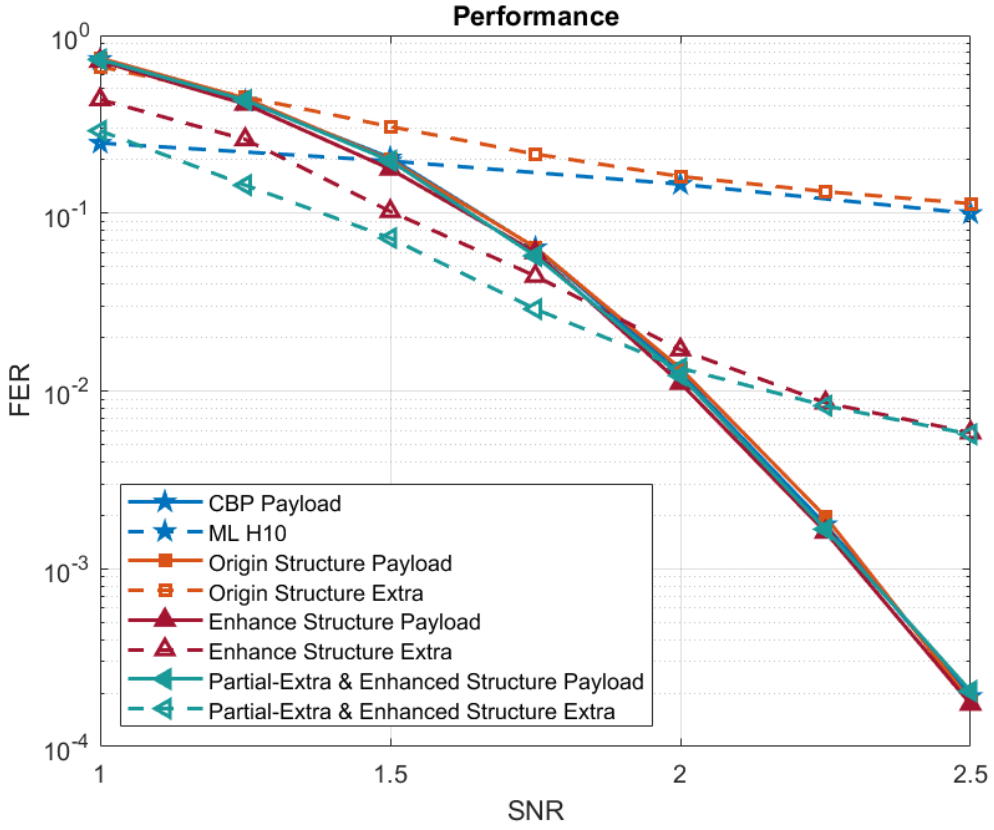
$M_{Enhanced1}$ each row degree is 1.

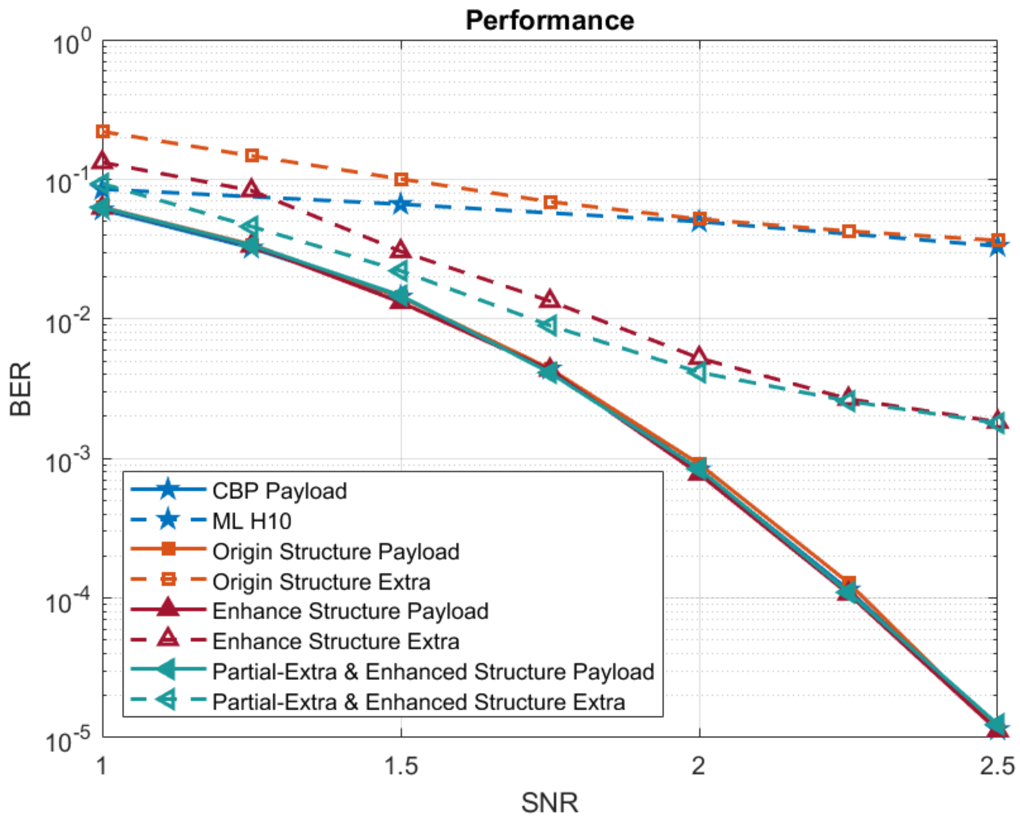
$M_{Enhanced2}$ each row degree is 1.

$M_{extrapunc}$ each row degree is 1.

Partial-Extra & Enhanced Structure – Simulation

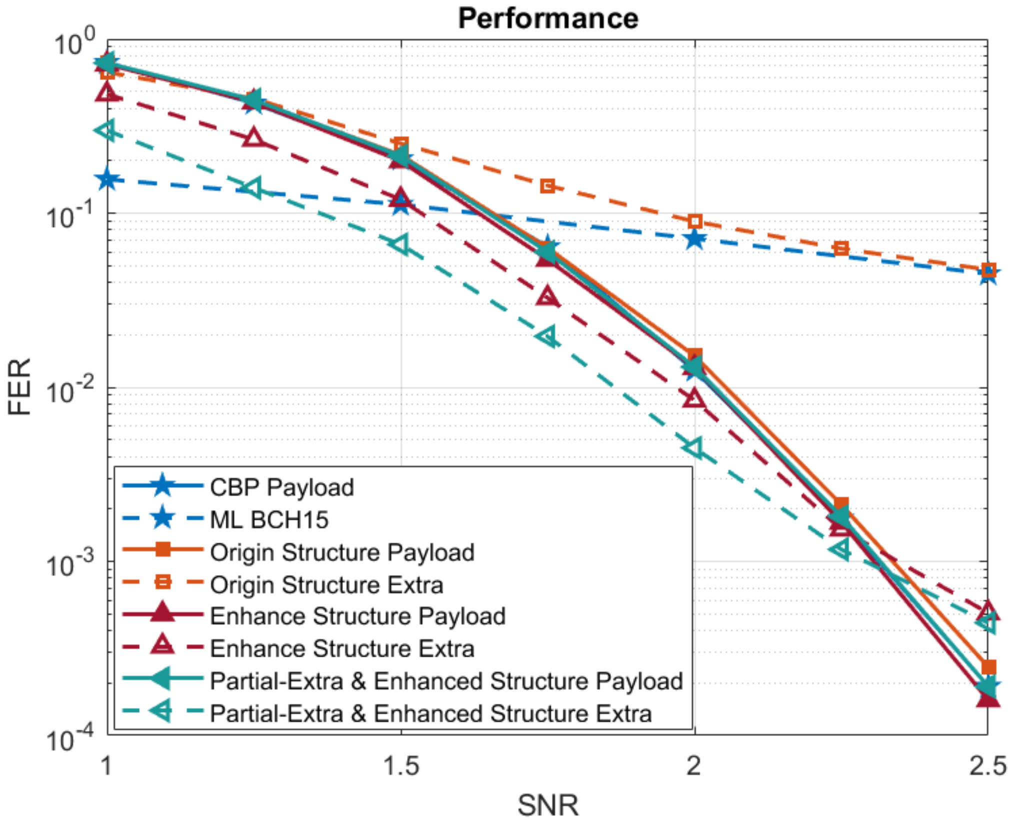
- Payload : PEG-P1
- Extra : LDPC-E1

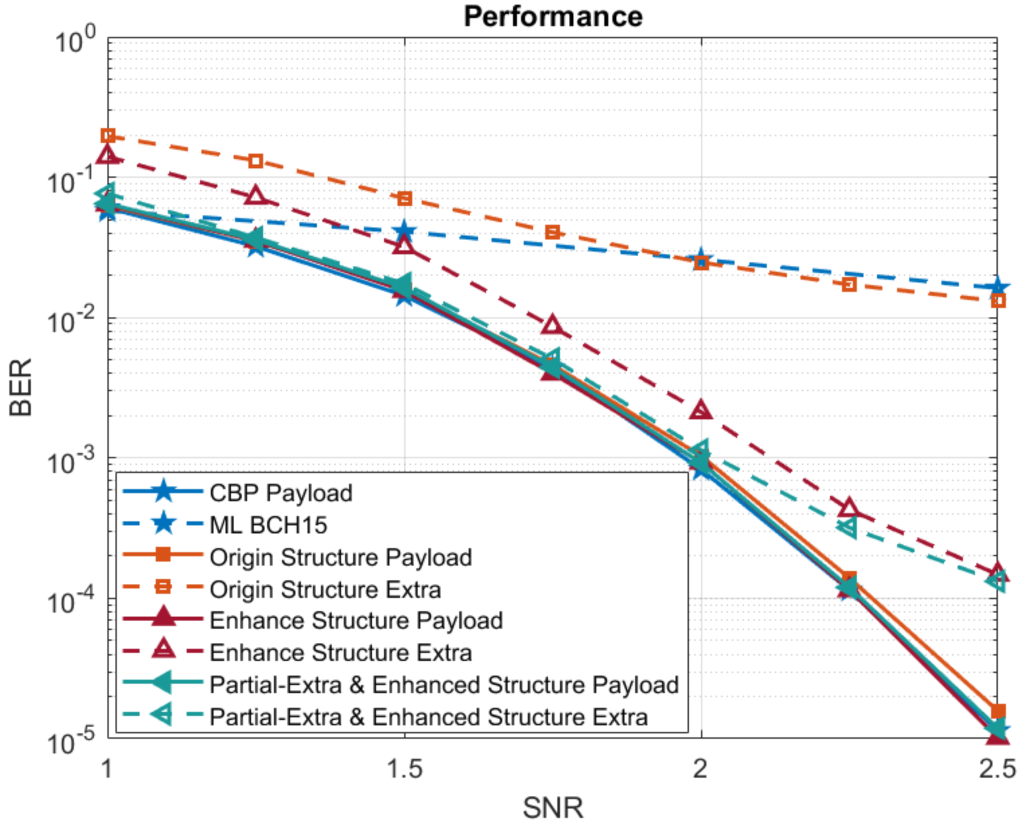




Partial-Extra & Enhanced Structure – Simulation

- Payload : PEG-P1
- Extra : BCH-E2





Partial-Extra & Enhanced Structure– Observation

- After combining the partial-extra structure and the enhanced structure, Extra's performance shows a significant improvement at low SNR.
- The Combined Structure can be seen as a large parity-check matrix, where both the payload and the extra bits are part of it. This is why the extra bits can perform even better than maximum likelihood decoding.

Outline

- **Introduction**
- **Free-Ride Concept**
  - Encoding Scheme
  - Decoding Scheme(Hard/Soft)
  - Simulation
  - Pros & Cons
- **Tanner Graph Combination**
- **Puncture Concept**
  - One–Step Recoverable Node Scheme
  - Rate–Compatible Puncturing Scheme
- **Origin Structure**
- **Partial-Extra Structure**
- **Enhanced Structure**
- **Partial-Extra & Enhanced Structure**
- **Reference**

Reference

[1] S. Cai, S. Zhao and X. Ma, "Free Ride on LDPC Coded Transmission," in IEEE Transactions on Information Theory, vol. 68, no. 1, pp. 80-92, Jan. 2022.

[2] L. Zhang, F. Ma and L. L. Cheng, "A Puncturing Scheme for Low-Density Parity-Check Codes Based on 1-SR Nodes," 2012 IEEE Vehicular Technology Conference (VTC Fall), Quebec City, QC, Canada, 2012.

[3] R. Asvadi and A. H. Banihashemi, "A Rate-Compatible Puncturing Scheme for Finite-Length LDPC Codes," in IEEE Communications Letters, vol. 17, no. 1, pp. 147-150, January 2013.

[4] Jeongseok Ha, Jaehong Kim, D. Klinc and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," in IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 728-738, Feb. 2006

[5] H. Li and L. Zheng, "Efficient Puncturing Scheme for Irregular LDPC Codes Based on Serial Schedules," in IEEE Communications Letters, vol. 19, no. 9, pp. 1508-1511, Sept. 2015

Reference

[6] Q. Wang, L. Chen and X. Ma, ”A new HARQ scheme for 5G systems via interleaved superposition retransmission,” in China Communications, vol. 20, no. 4, pp. 1-11, April 2023