A Construction of Non-binary Polar codes with 4 by 4 kernels

Sung Chul Byun, Gangsan Kim, Won Jun Kim, and Hong-Yeop Song

Yonsei University

IWSDA 2019, October 20-24

The main flow of this talk

First,

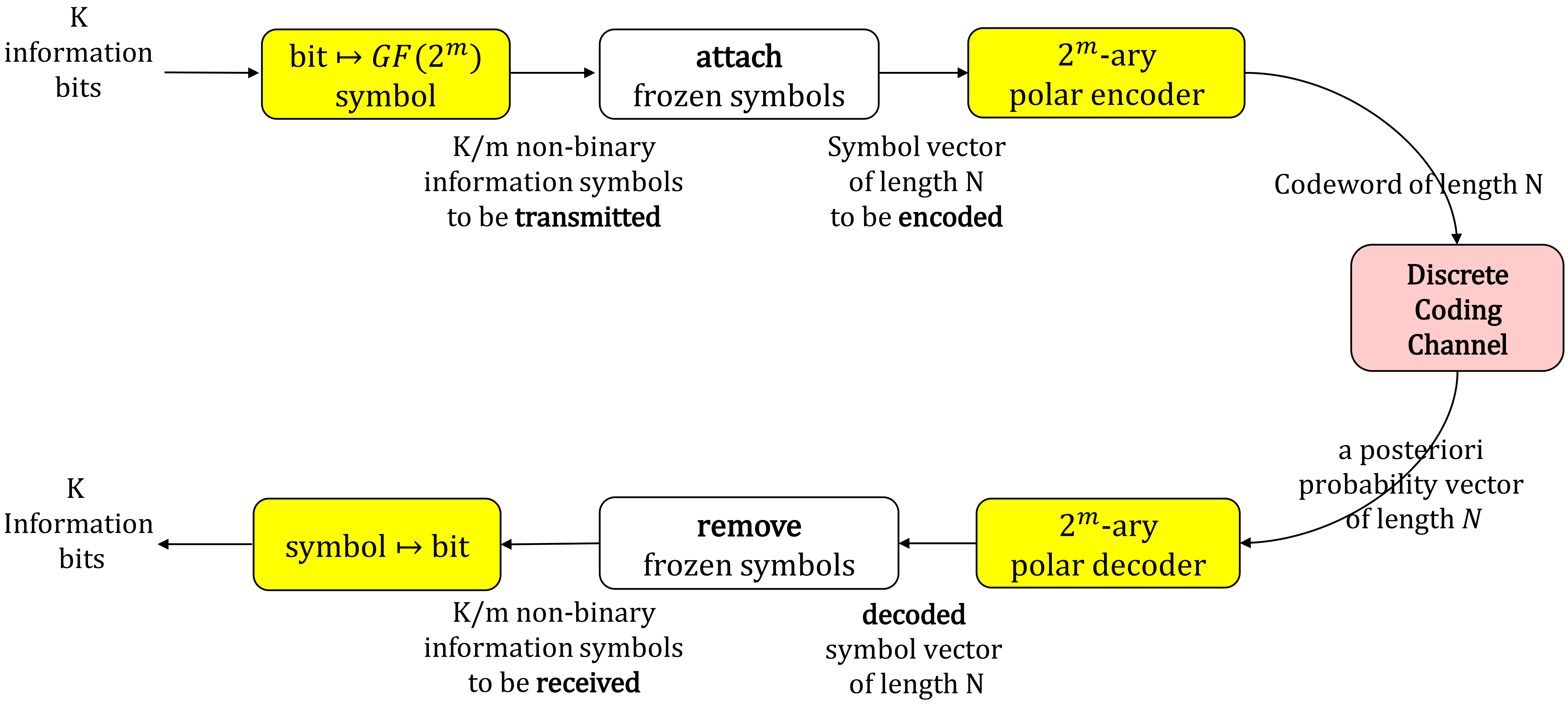
We investigate the FER of non-binary polar codes with all the non-binary 2×2 kernels of type $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$

The main flow of this talk

First, We investigate the FER of non-binary polar codes with all the non-binary 2×2 kernels of type $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$

Second, We propose non-binary 4×4 kernels with the same complexity of encoding and SC decoding but better FER performance

System model



bit $\mapsto GF(2^m)$ symbol

F_2^4

Binary 4-bits

Ex) 1 1 0 1

$b_0\ b_1\ b_2\ b_3$

4 bits

conversion

F_{2^4}

Symbols in $GF(2^4)$

$$1\cdot\alpha^0 + 1\cdot\alpha^1 + 0\cdot\alpha^2 + 1\cdot\alpha^3 = \begin{cases} 0 \\ or \\ \alpha^t \end{cases}$$

$$b_0\cdot\alpha^0 + b_1\cdot\alpha^1 + b_2\cdot\alpha^2 + b_3\cdot\alpha^3 \in GF(2^4)$$

basis of $GF(2^4)$

$\{1, \alpha, \alpha^2, \alpha^3\}$

$\{1, \alpha^2, \alpha^4, \alpha^6\}$

$\{1, \alpha^4, \alpha^8, \alpha^{12}\}$

$\{1, \alpha^8, \alpha^{16}, \alpha^{24}\}$

Polar encoding with kernel

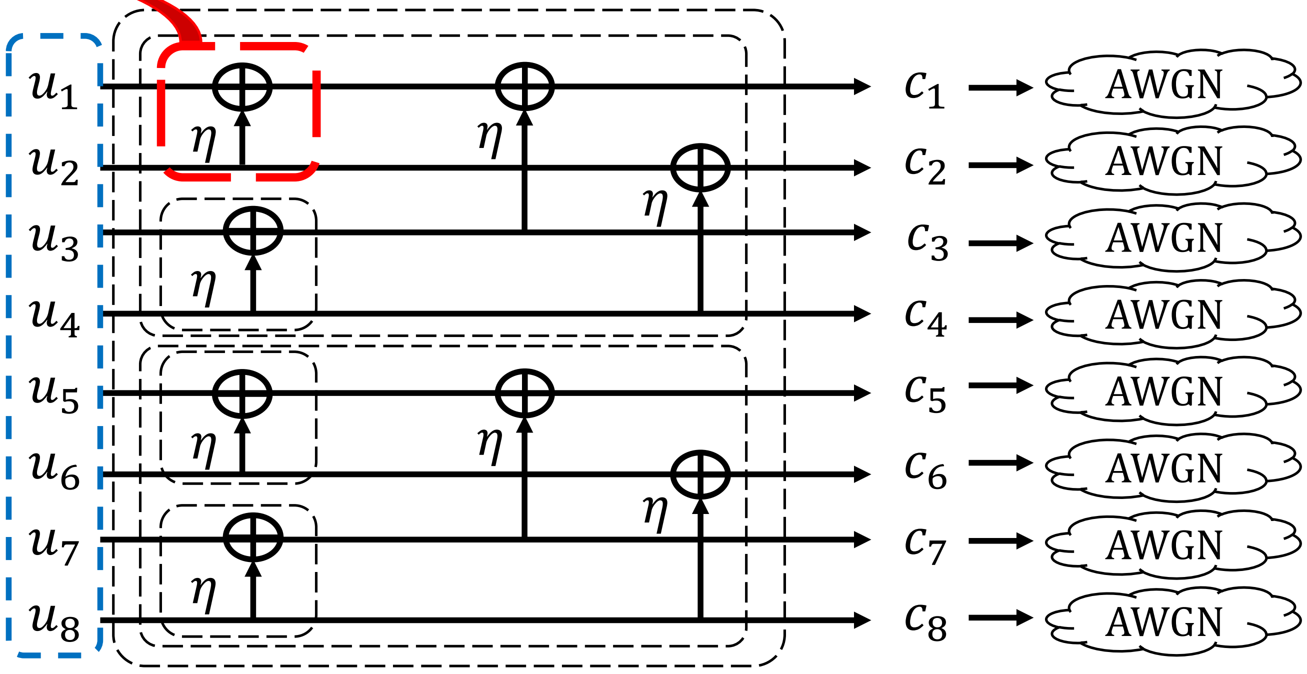
$$\begin{bmatrix} 1 & 0 \\ \alpha^z & 1 \end{bmatrix}$$

$z = 1, 3, 5, 7$

Minimal polynomial (GF (16))	Root (power of α)
$x+1$	0
x^4+x+1	1, 2, 4, 8
$x^4+x^3+x^2+x+1$	3, 6, 12, 9
x^2+x+1	5, 10
x^4+x^3+1	7, 14, 13, 11

Encoding with 2×2 kernels $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$

- 2×2 kernel : $F_2 = \begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}, \eta \in GF(2^m)$
- $\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G}$ where $G = F_2^{\otimes r} = \begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}^{\otimes r}$

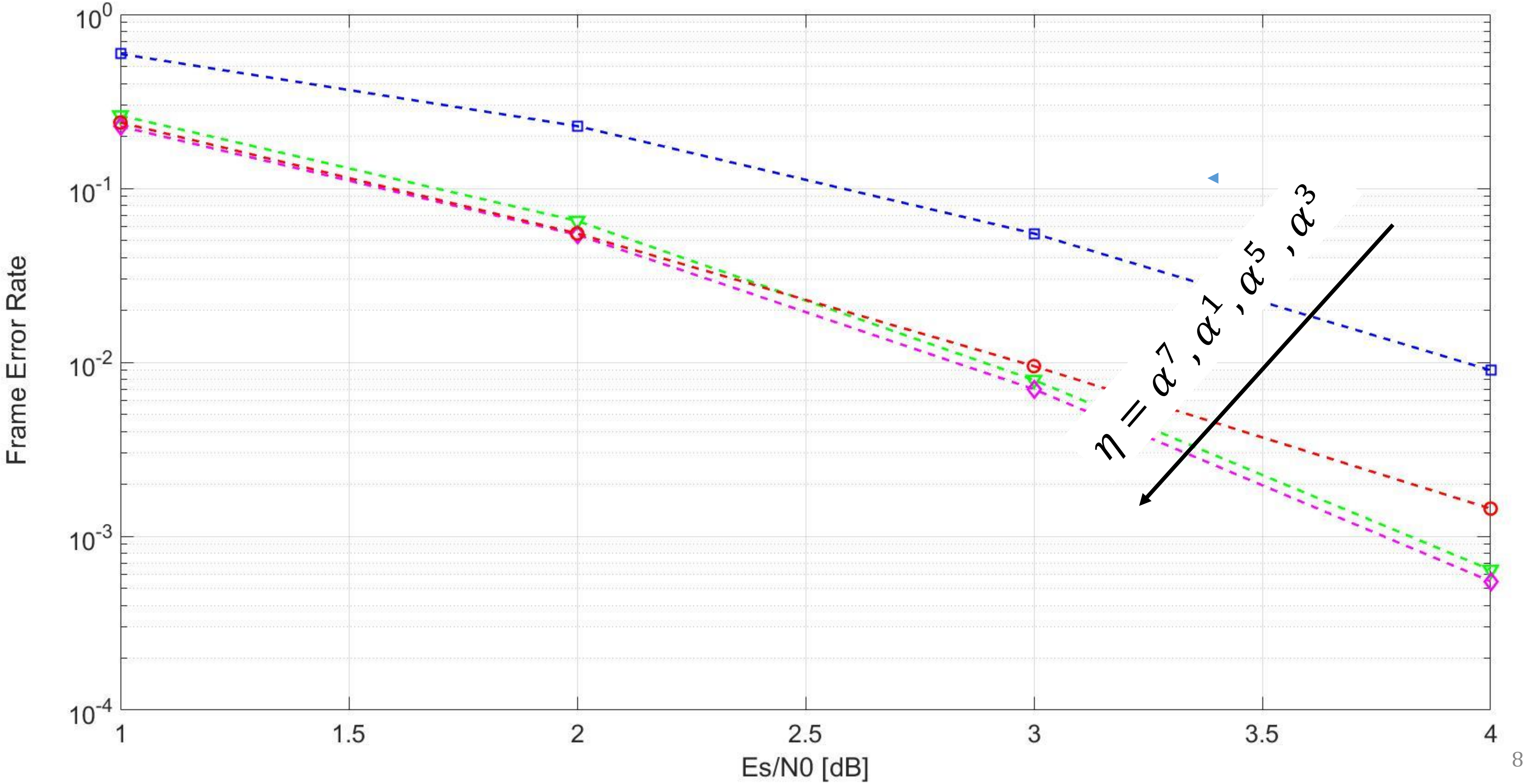


Some frozen symbols are sent thru the channel
in which the reliability is lower;
and information symbols are sent thru the channels
in which the reliability is higher

Simulation environment

- Non-binary symbols over GF(16)
- Code length : 64 symbols = 256 bits
- Code rate : ½ (32 frozen symbols and 32 data symbols)
- **2×2 kernels** $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$
- Channel : AWGN channel
- Modulation : 16 QAM (each symbol has 4-bit meaning)
- Decoding : Successive Cancelation (SC) decoding

Simulation result with 2×2 kernels $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$



Question

Can we improve the performance
using some **4×4 kernels**
instead of **2×2 kernels**,
while maintaining the complexity of
encoding/SC decoding ?

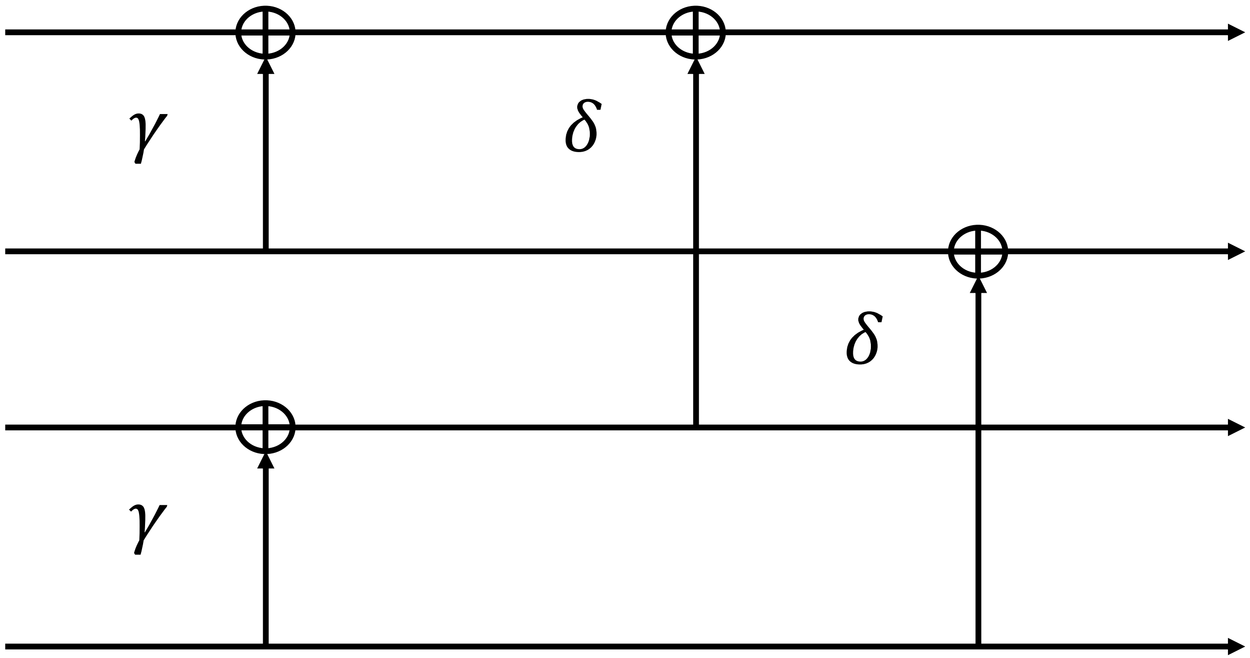
Proposed type of 4×4 kernel

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \gamma & 1 & 0 & 0 \\ \delta & 0 & 1 & 0 \\ \gamma\delta & \delta & \gamma & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix}$$

where

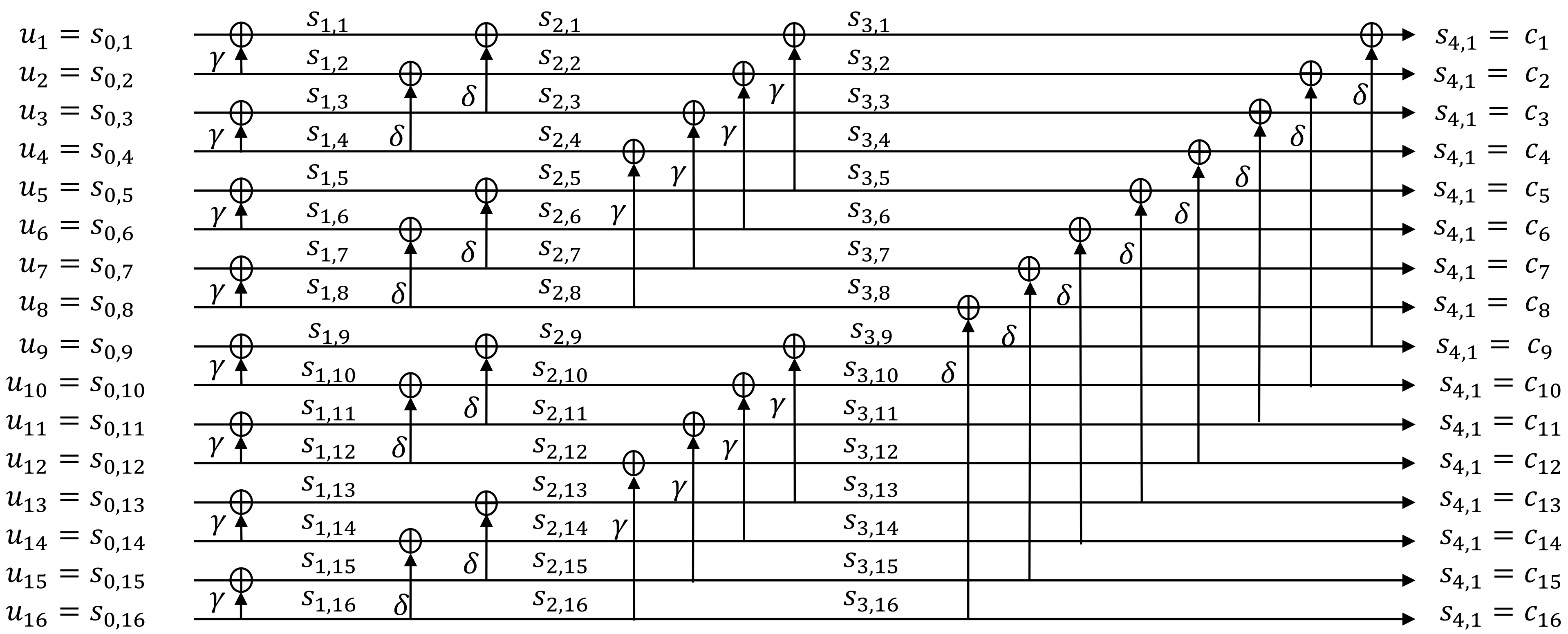
$\gamma, \delta \in GF(2^m), \gamma, \delta \neq 0$



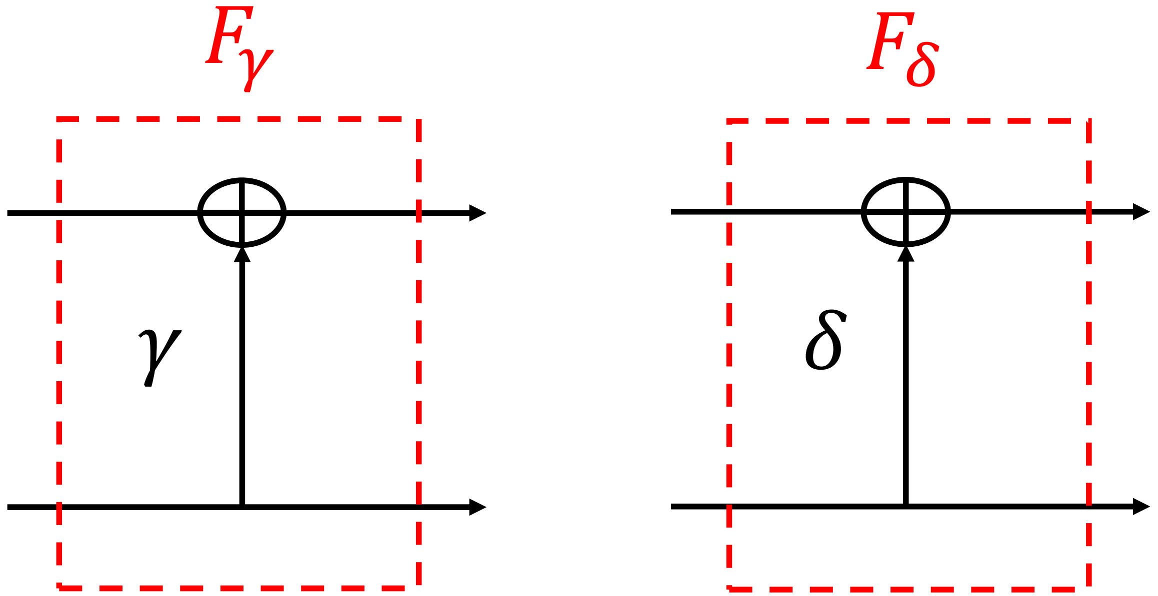
The same complexity of encoding/decoding

as those using 2×2 kernels $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$

Non-binary polar code of length 16 using the proposed 4 × 4 kernel

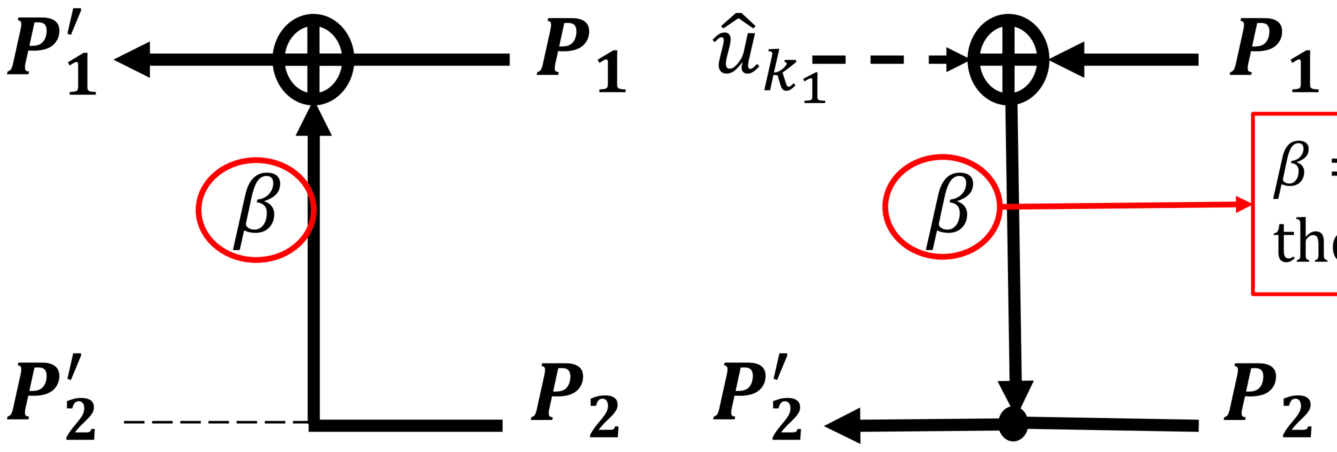


Sub-kernels F_γ and F_δ for γ and δ of the proposed 4×4 kernels



The sub-kernels F_γ and F_δ appear alternately during the encoding and decoding process

The message update rule (1/2)



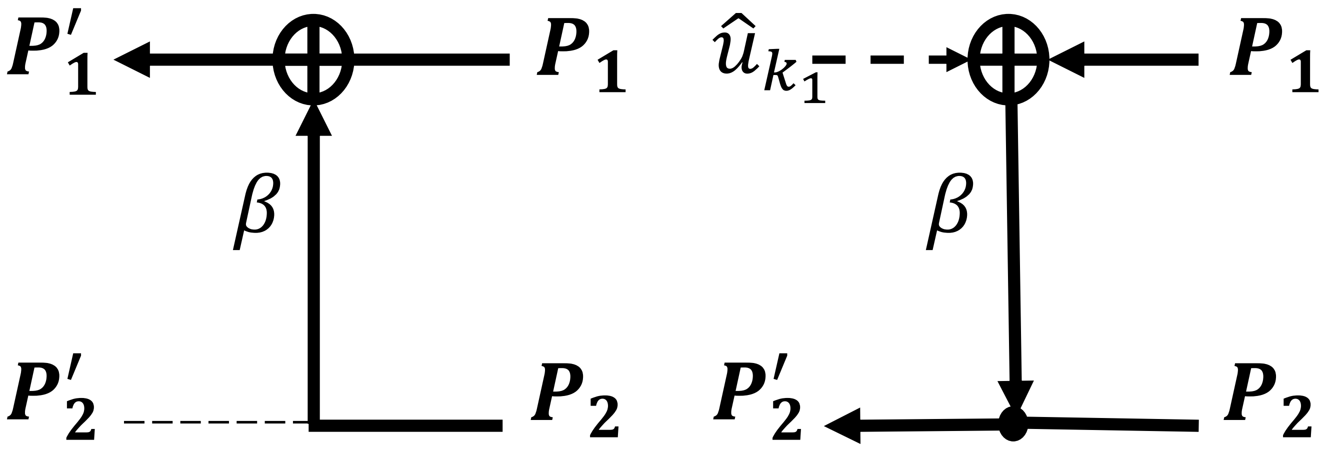
$\beta = \gamma$ or δ depending on whether the current sub-kernel is F_γ or F_δ

❖ Notation

- $\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P'_1}, \boldsymbol{P'_2}$: a posteriori probability (APP) vectors
- $\hat{u}_{k_1}$: symbol corresponding to the APP vector P'_1
- $\hat{u}_{k_2}$: symbol corresponding to the APP vector P'_2
- k_1, k_2 : decoding phase for $1 \le k \le n_c$

- $\boldsymbol{P_1} = \left[P_1(0),\ P_1(\alpha^0), \dots, P_1\left(\alpha^{2^m-2}\right)\right]$
- $\boldsymbol{P_2} = \left[P_2(0),\ P_2(\alpha^0), \dots, P_2\left(\alpha^{2^m-2}\right)\right]$
- $\boldsymbol{P'_1} = \left[P'_1(0),\ P'_1(\alpha^0), \dots, P'_1\left(\alpha^{2^m-2}\right)\right]$
- $\boldsymbol{P'_2} = \left[P'_2(0),\ P'_2(\alpha^0), \dots, P'_2\left(\alpha^{2^m-2}\right)\right]$

The message update rule (2/2)



$$P_1'(u_{k_1}) = \frac{1}{2^m} \sum_{u_{k_2} \in GF(2^m)} P_1(u_{k_1} \oplus \beta u_{k_2}) P_2(u_{k_2})$$

$$P_2'(u_{k_2}) = \frac{1}{2^m} P_1(u_{k_1} \oplus \beta u_{k_2}) P_2(u_{k_2})$$

In decoding phase k_i,

$$P_{U_{k_i}|Y^n, U^{k_i-1}} = \left[\dots, P_{U_{k_i}|Y^n, U^{k_i-1}}(u|y^n, u^{k_i-1}), \dots \right]$$, where $u \in GF(2^m)$

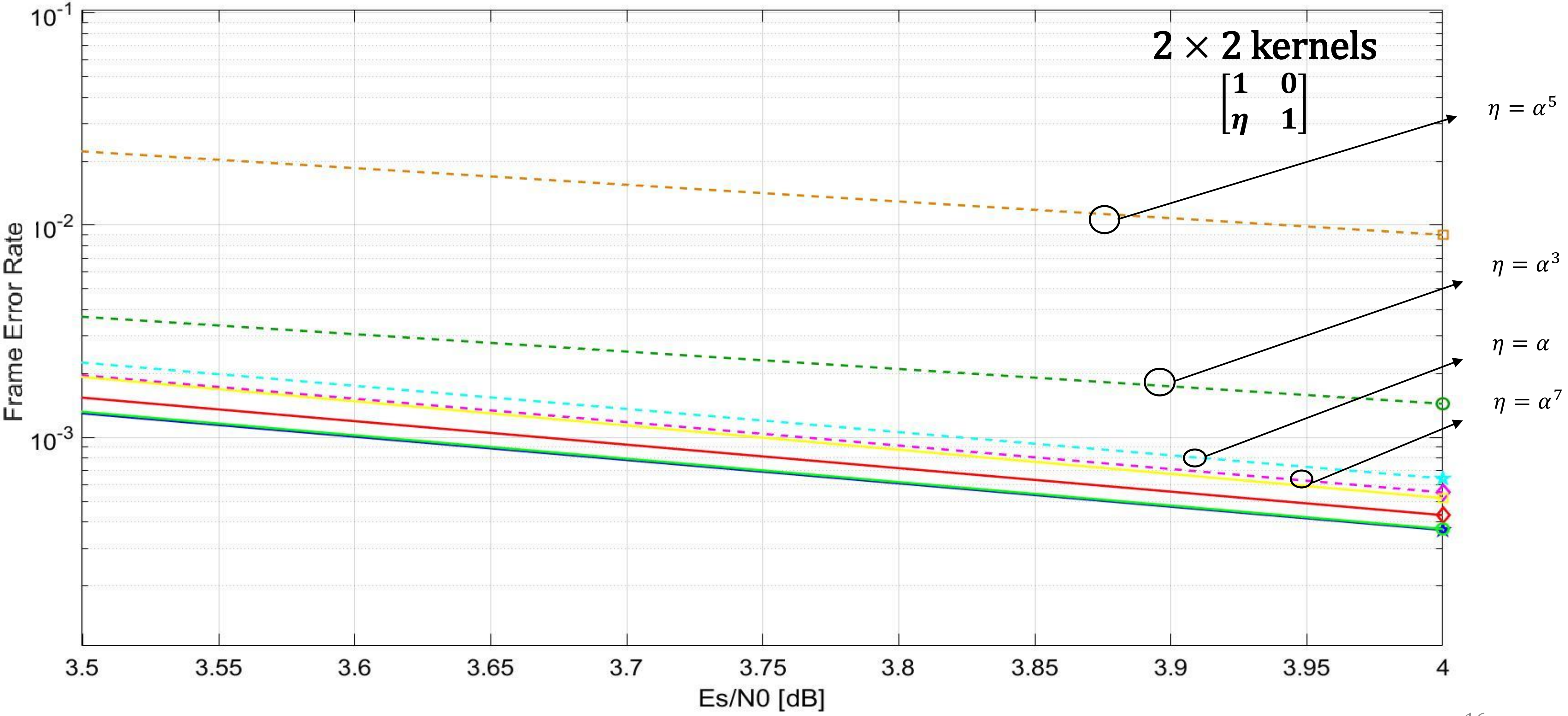
$$\hat{u}_{k_i} = \underset{u \in GF(2^m)}{\operatorname{argmax}} P_{U_{k_i}|Y^n, U^{k_i-1}}(u|y^n, u^{k_i-1})$$

by recursively updating the messages

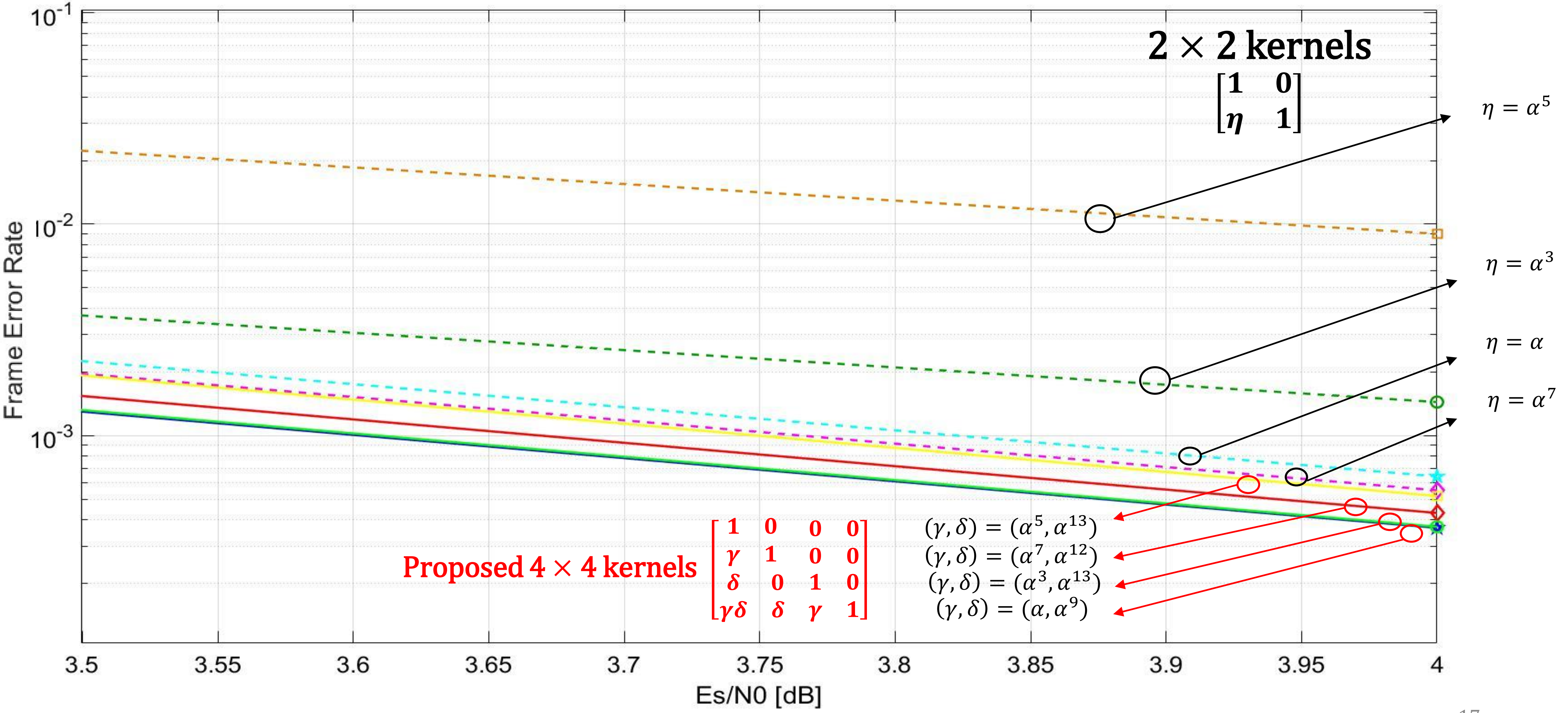
Simulation environment

- Non-binary symbols over GF(16)
- Code length : 64 symbols = 256 bits
- Code rate : ½ (32 frozen symbols and 32 data symbols)
- **4 × 4 kernels** $\begin{bmatrix} 1 & 0 & 0 & 0 \\ \gamma & 1 & 0 & 0 \\ \delta & 0 & 1 & 0 \\ \gamma\delta & \delta & \gamma & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ \delta & 1 \end{bmatrix}$
- Channel : AWGN channel
- Modulation : 16 QAM (each symbol has 4-bit meaning)
- Decoding : Successive Cancelation (SC) decoding

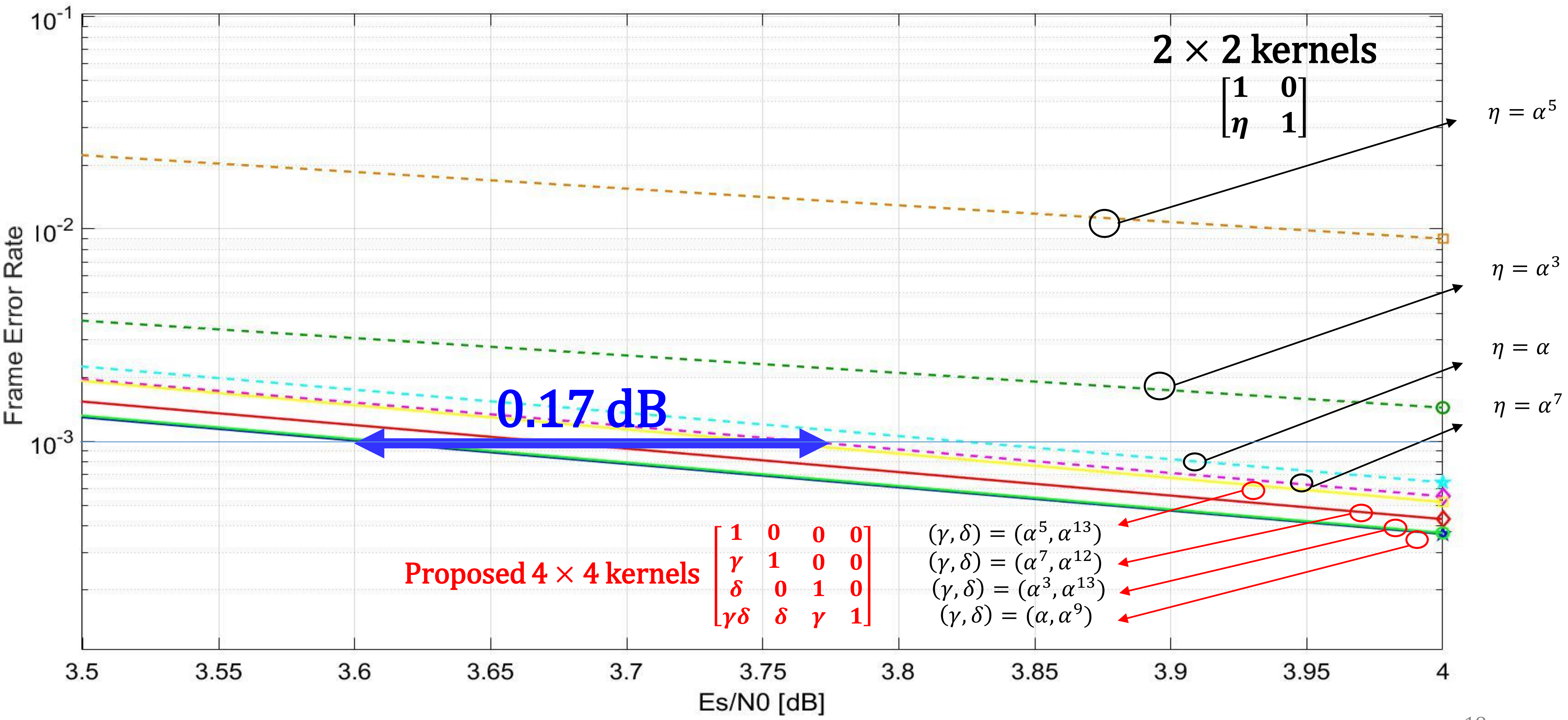
Comparison of FER performance



Comparison of FER performance



Comparison of FER performance



Conclusion

- We confirmed the existence of 4×4 kernels with better FER performance than using the best 2×2 kernel of type $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$

- This also has the advantage of having the same complexity of encoding and SC decoding

Conclusion

- We confirmed the existence of 4×4 kernels with better FER performance than using the best 2×2 kernel of type $\begin{bmatrix} 1 & 0 \\ \eta & 1 \end{bmatrix}$

- This also has the advantage of having the same complexity of encoding and SC decoding

❖ **Future research:**

1. For 2 by 2 case, do we know some fundamental reason why one is better than the other?
2. Is it true that the best 4 by 4 is always better than the best 2 by 2 ? Why?
3. Can an 8 by 8 or larger size kernels improve more?

Any Questions or Comments?