[Mimic seminar]

Introduction and construction of polar code

이형주

Communication and coding theory lab (CCL), SKKU

2025

Contents

- Why should we learn polar code & What is polar code
- Polar code
  - Encoding, Decoding(sc, scl, crc-scl)
- β-expansion A Theoretical Framework for Fast and Recursive Construction of Polar Codes (2017)
- Genetic Algorithm-based Polar Code Construction for the AWGN Channel (2019)
- Construction of Polar Codes with Reinforcement Learning (2020)

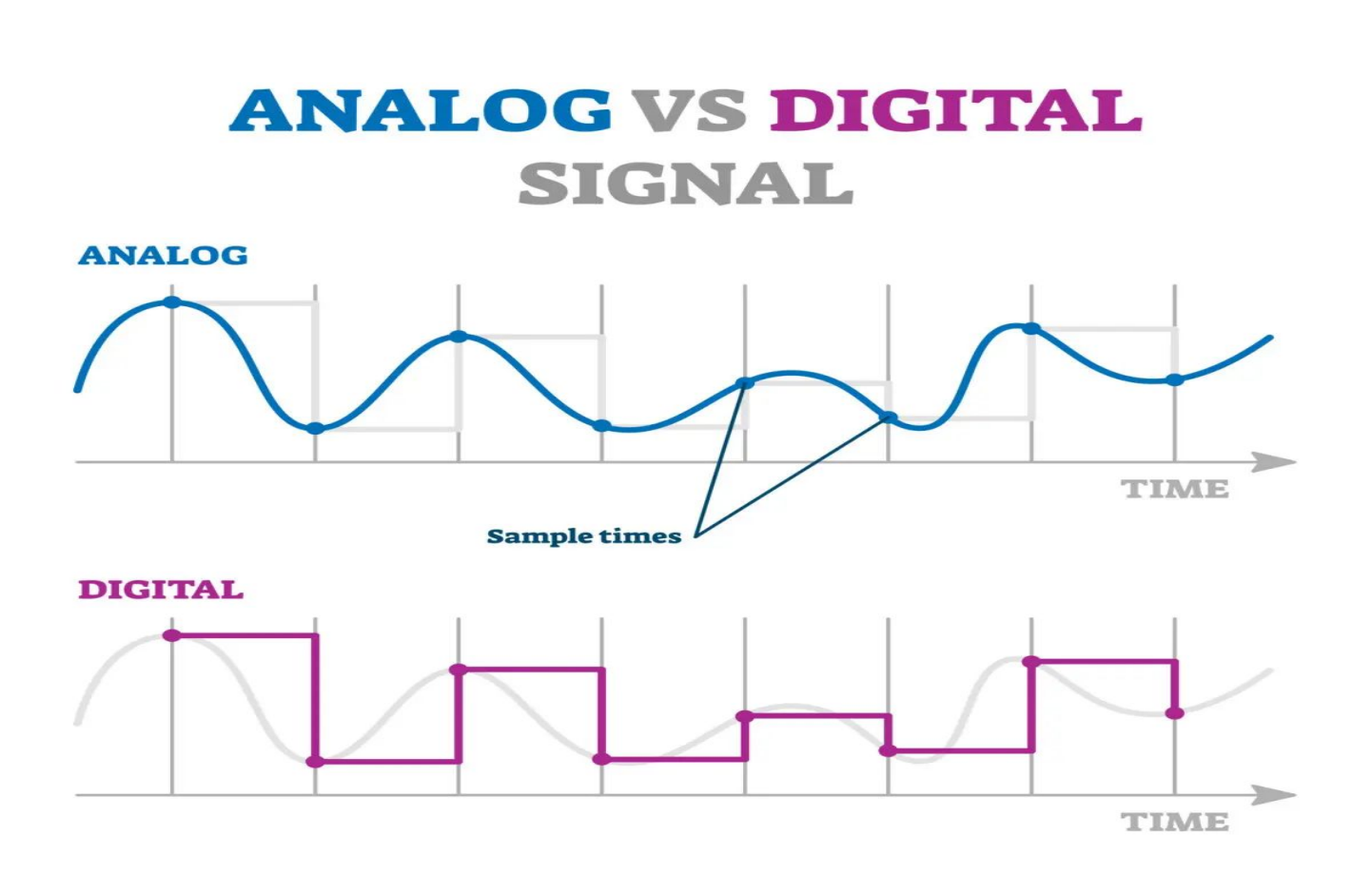
Why should we learn polar code & What is polar code

- Analog signal vs Digital signal
- Analog to Digital
- Digital signal
- Error correction code
  - Hamming code
  - Polar code
    - Encoding
    - Decoding

Analog signal vs Digital signal

Analog signal: Continuous signal that varies smoothly over time (목소리, 사진)

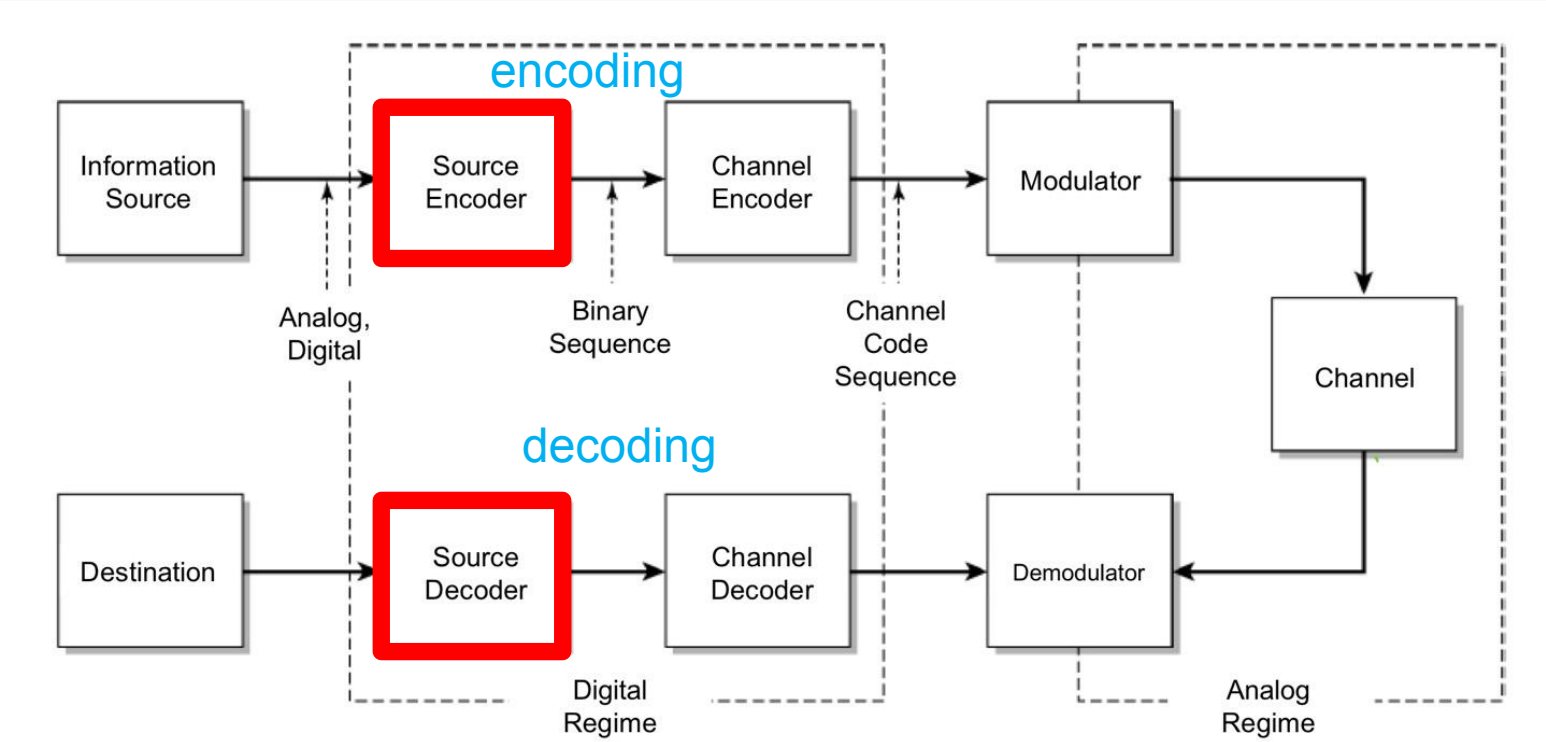
Digital signal: Discrete signal (001101010110111)



여러가지 이유 때문에Digital 이 더 좋다

추가로
Analog -> Digital

Digital signal



발표 내용

Information source(보내고 싶은 내용)에 encoding 단계에서 어떤 장난을 쳐야 channel 통과시 발생하는 오류를 잡을 수 있을까?

Error correction code(ECC)

Ex) Polar code, Hamming code

Error correction code

Hamming code (ex (7,4) 해밍 코드(4 비트 정보, 7비트 부호))

Information: 4 (1,0,1,1)
Parameter: 3 (1,2,4위치)

$$[d_1, d_2, d_3, d_4]$$

$$[p_1, p_2, d_1, p_3, d_2, d_3, d_4]$$

$$p_1 = d_1 + d_2 + d_4$$
$$p_2 = d_1 + d_3 + d_4$$
$$p_3 = d_2 + d_3 + d_4$$

$$[1,0,1,1]\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [0,1,1,0,0,1,1]$$

Mod 2

If) d2->1 (error)
[0,1,1,0,0,1,1] -> [0,1,1,0,1,1,1]
1. p_1, p_3가 이상한데
2. p_1(1번)+p_3(4번)=5번에서 오류다!

한계: 비트와 패리티 동시 오류, 최대 인식 2개, 최대 정정 1개

Polar code

Encoder(가장 간단한 case)

$$d = u \cdot G_N$$

N=4

$$[u_0, u_1, u_2, u_3]\begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0\\ 1 & 1 & 1 & 1\end{bmatrix}$$

$$= [(u_0 \oplus u_1 \oplus u_2 \oplus u_3), (u_1 \oplus u_3), (u_2 \oplus u_3), u_3]$$

Q: U값 끼리 합쳐서 보내면 뭐가 좋을까?

$a \oplus b$

a	b	XOR
0	**0**	**0**
0	1	1
1	0	1
1	1	0

u: input vector

G_N: generator matrix$(N = 2^n)$

$G_N = G_2^{\otimes n}$, $G_2 = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$

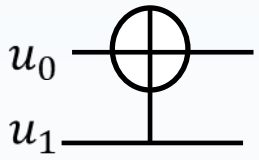
Ex) $G_4 = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0\\ 1 & 1 & 1 & 1\end{bmatrix}$

$G_8 = G_2 \otimes G_2 \otimes G_2$

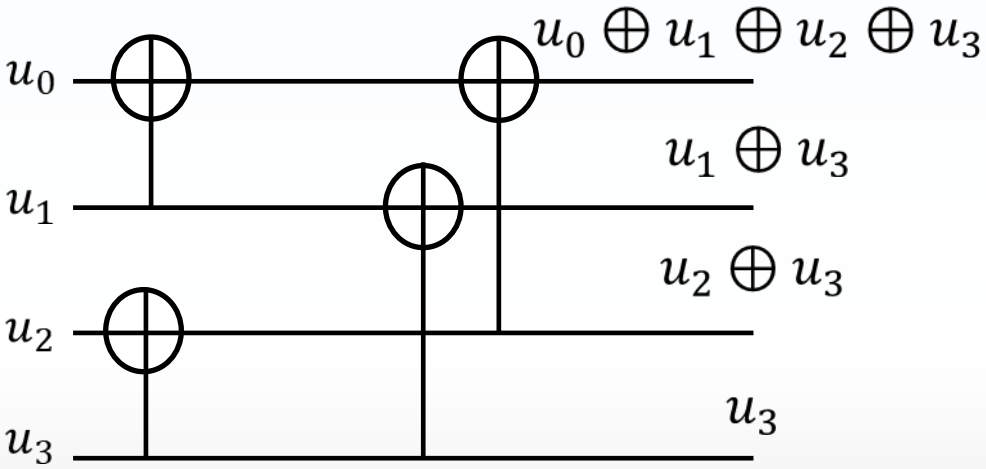
$$= \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}$$

$(\cdot)^{\otimes}$: Kronecker power

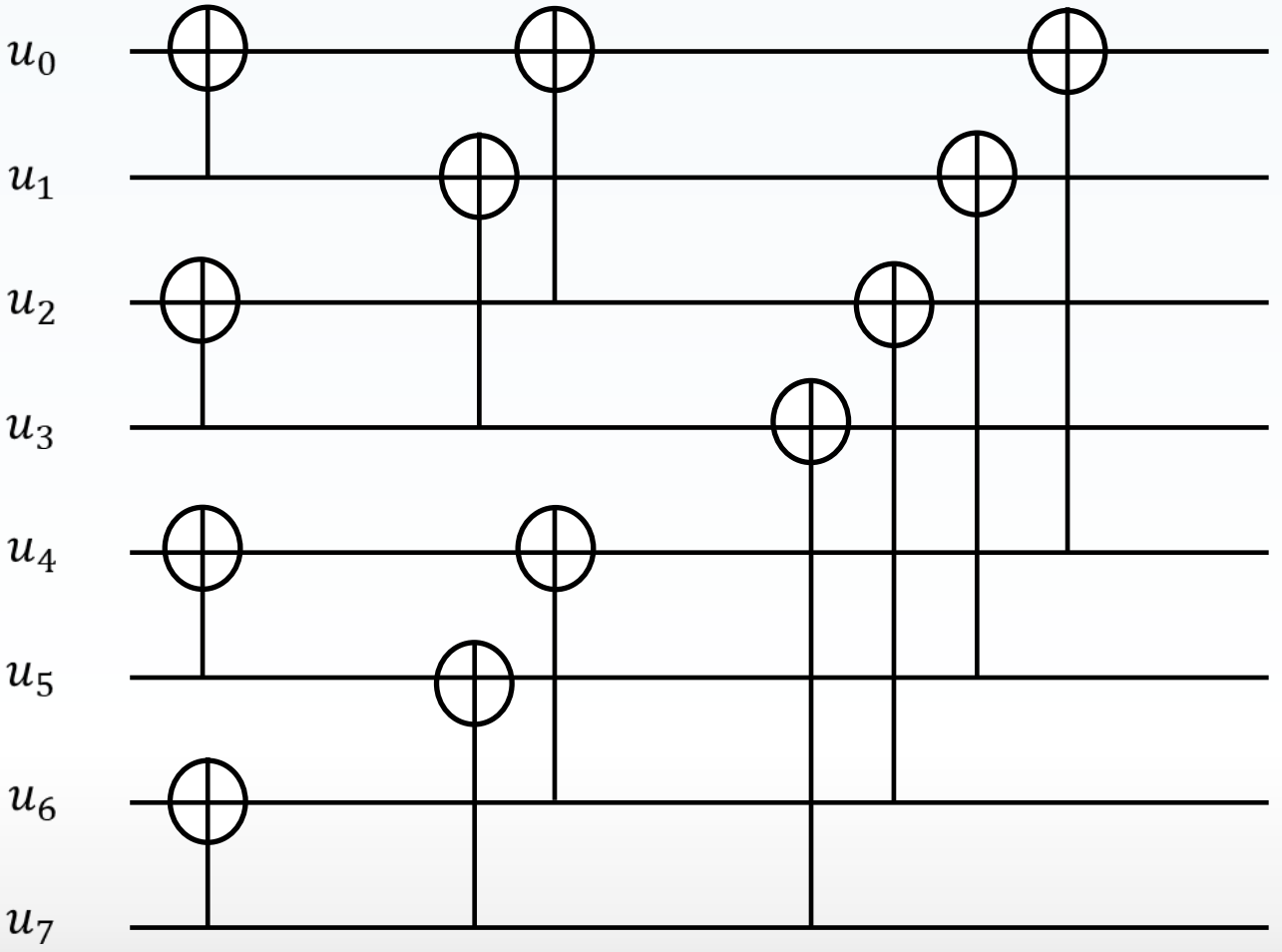
N=2


u_0
u_1

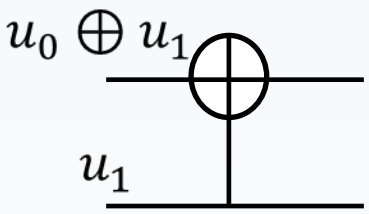
N=4


u_0
u_1
u_2
u_3
$u_0 \oplus u_1 \oplus u_2 \oplus u_3$
$u_1 \oplus u_3$
$u_2 \oplus u_3$
u_3

N=8


u_0
u_1
u_2
u_3
u_4
u_5
u_6
u_7

Polarized channel



Channel error rate: p

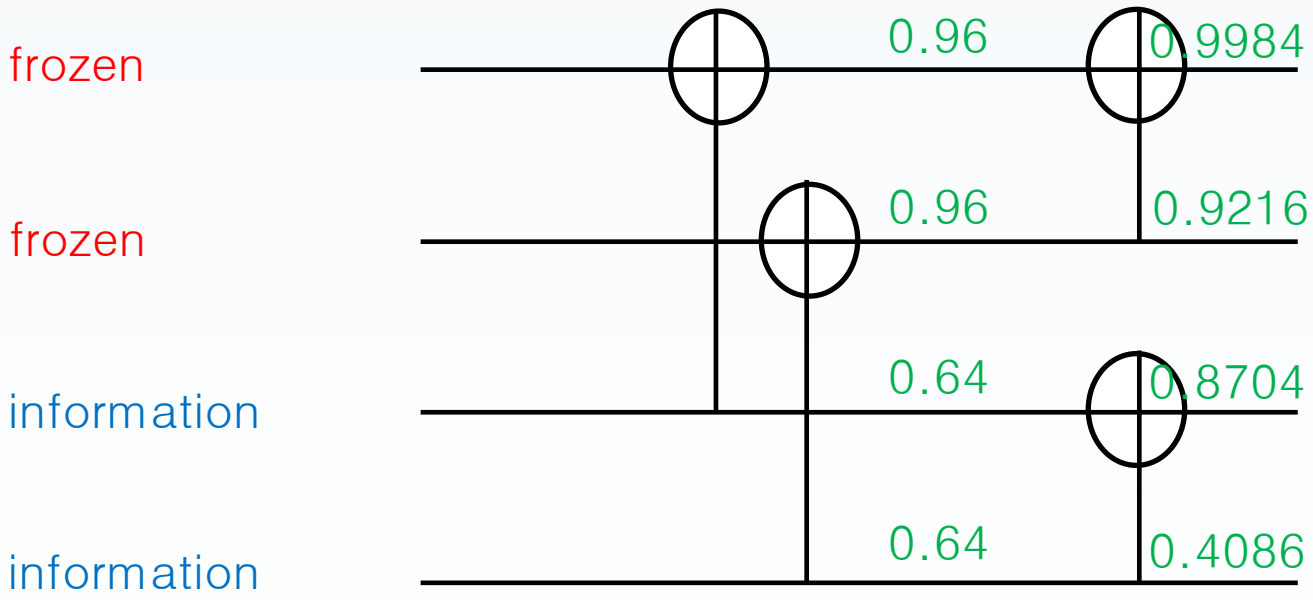
Xor channel	Second channel	u0 modulation
O	O	O
O	X	X
X	O	X
X	X	X

Xor channel	Second channel	u1 modulation
O	O	O
O	X	O
X	O	O
X	X	X

(u2 디코딩 한다는 소리는 이미 u1은 디코딩이 끝났다는 소리다)

- If) channel error rate 0.8 (BEC)

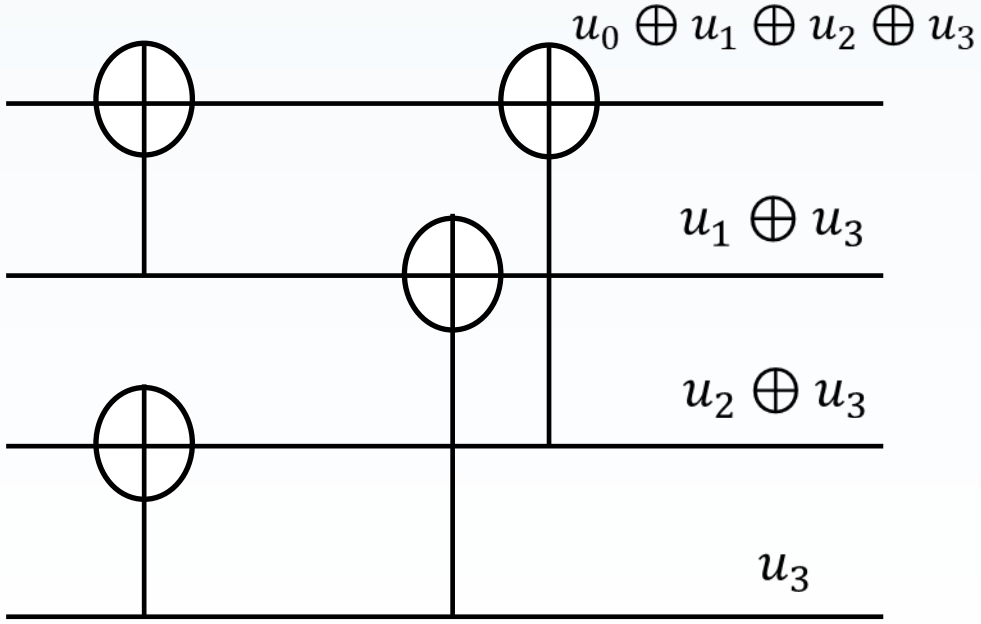
Code rate: information/N

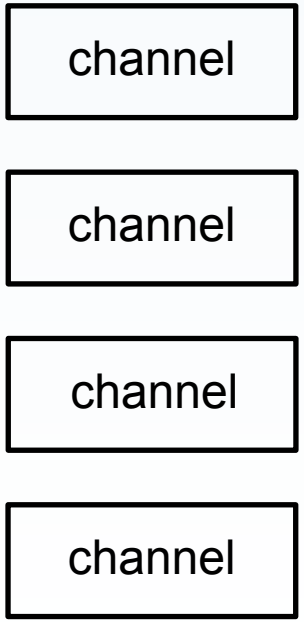


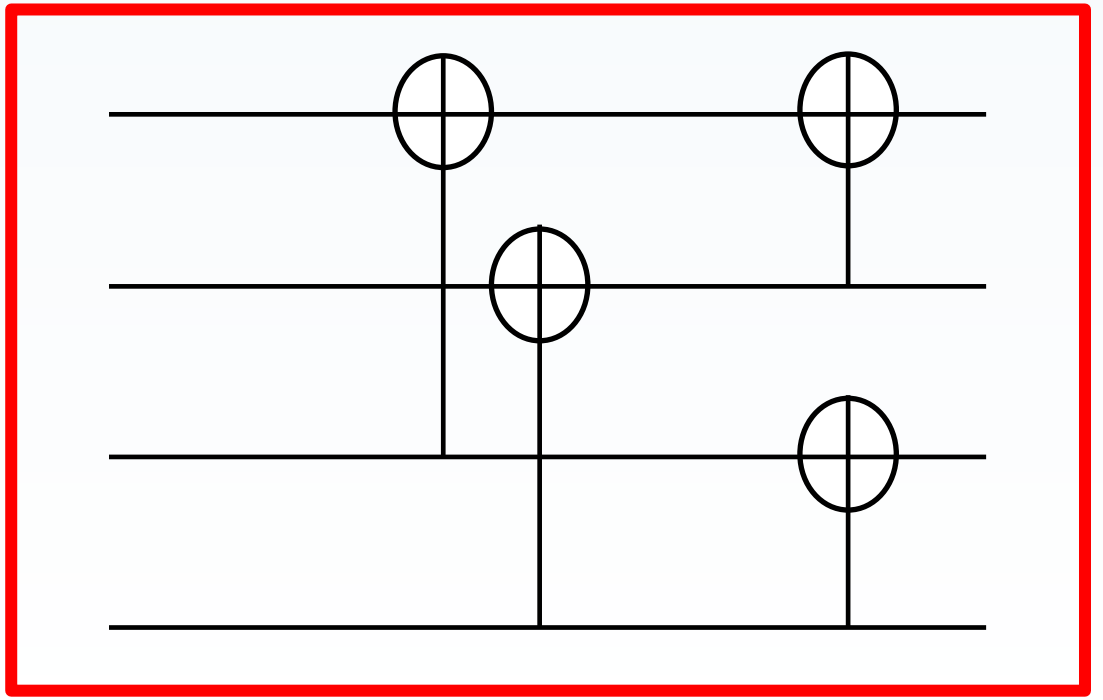
생각 해볼 수 있는 변수 2가지

Hamming code: 검산 할 수 있는 장치 추가
Polar code: 좋은 channel을 고르자

Decoder

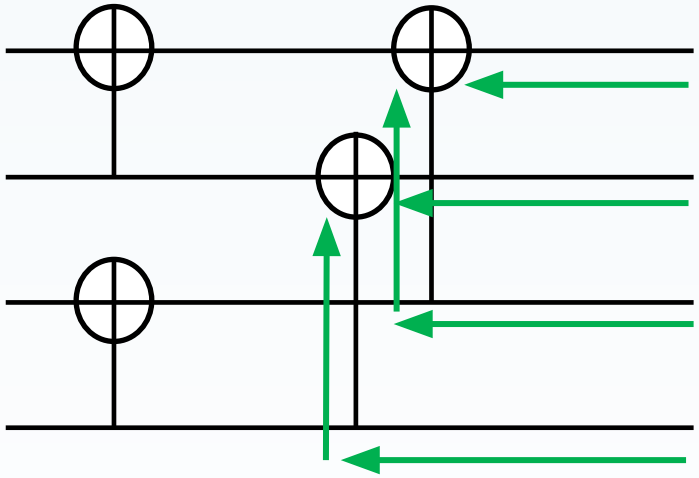

$u_0 \oplus u_1 \oplus u_2 \oplus u_3$
$u_1 \oplus u_3$
$u_2 \oplus u_3$
u_3

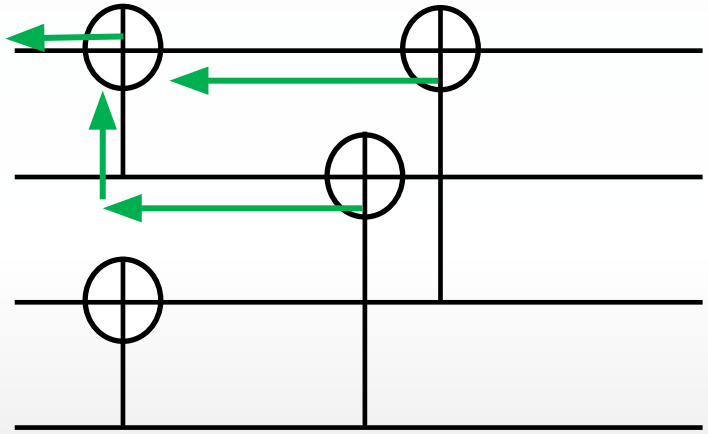

channel
channel
channel
channel

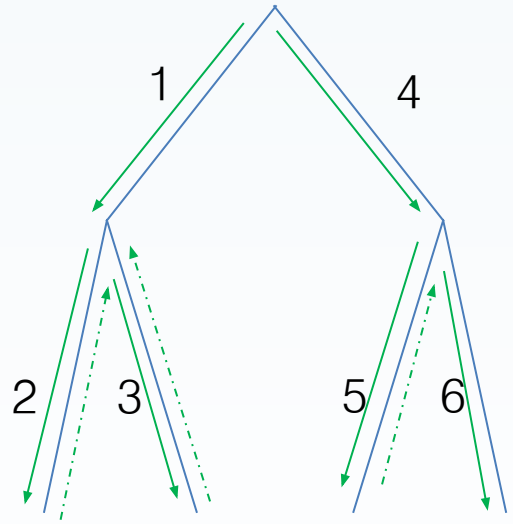


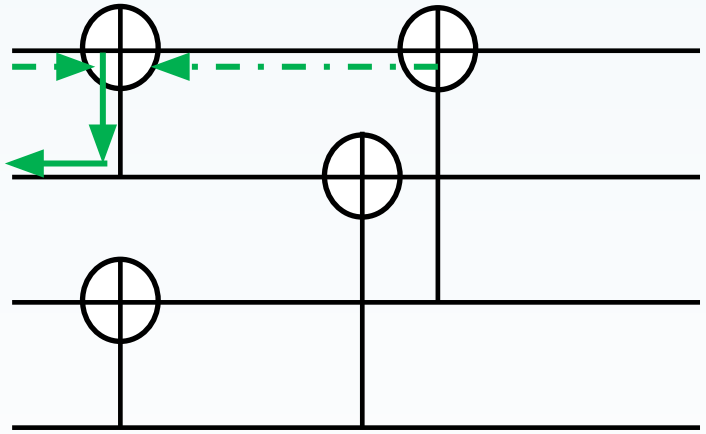
SC decoding

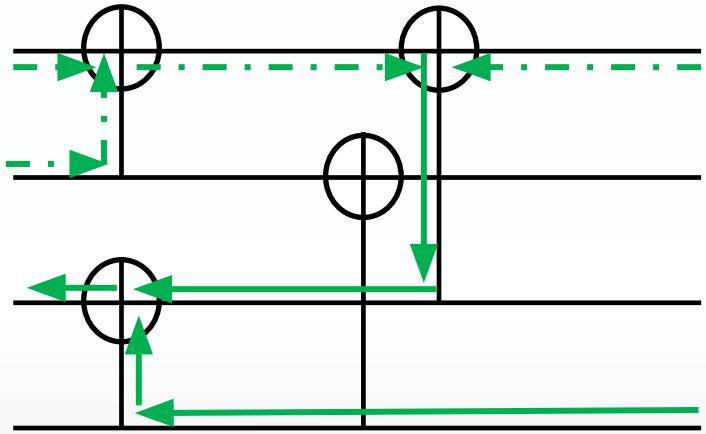
- N=4

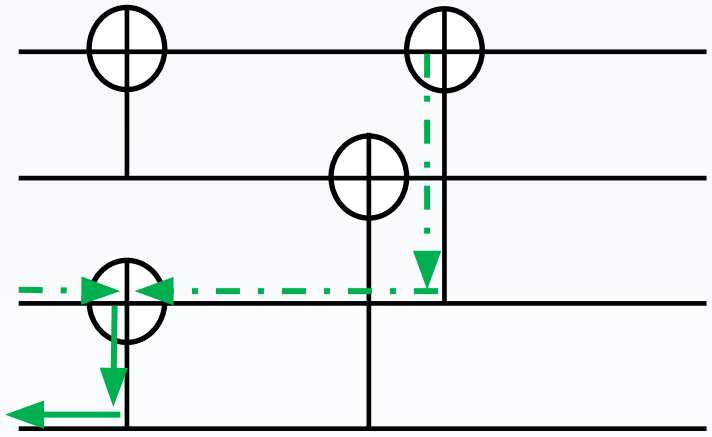












Transition probability & LLR

Transition probability

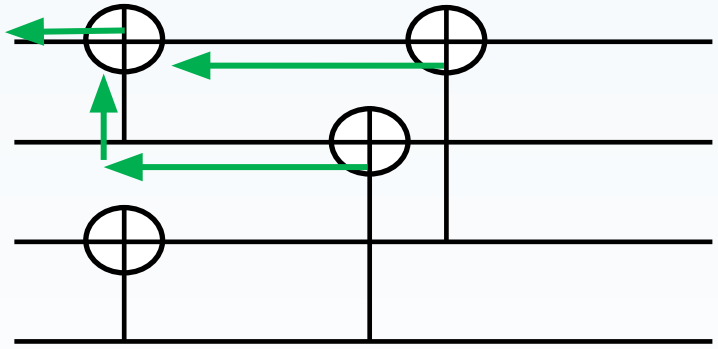
LLR

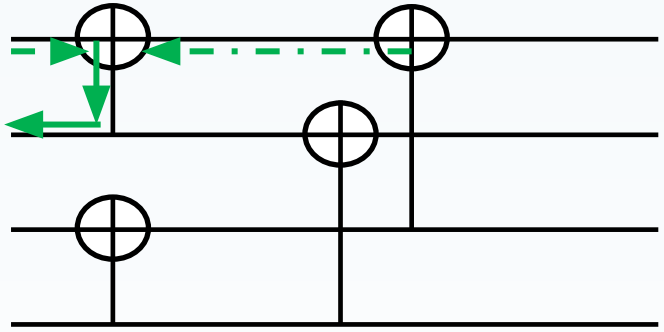
LLR (at AWGN)

- N=2

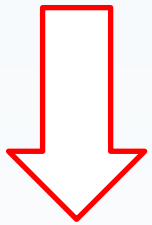
$$\begin{cases} W_2^{(0)}(y_1, y_2|u_0) = \sum_{u_1 \in \{01\}} \frac{1}{2} W(y_1|u_0 \oplus u_1) W \\ W_2^{(1)}(y_1, y_2|u_1) = \frac{1}{2} W(y_1|u_0 \oplus u_1) W(y_2 \end{cases}$$

예시)





사실 앞 그림의 화살표는 U의 값이 이동하는게 아니라 LLR이 이동하는 것이다!



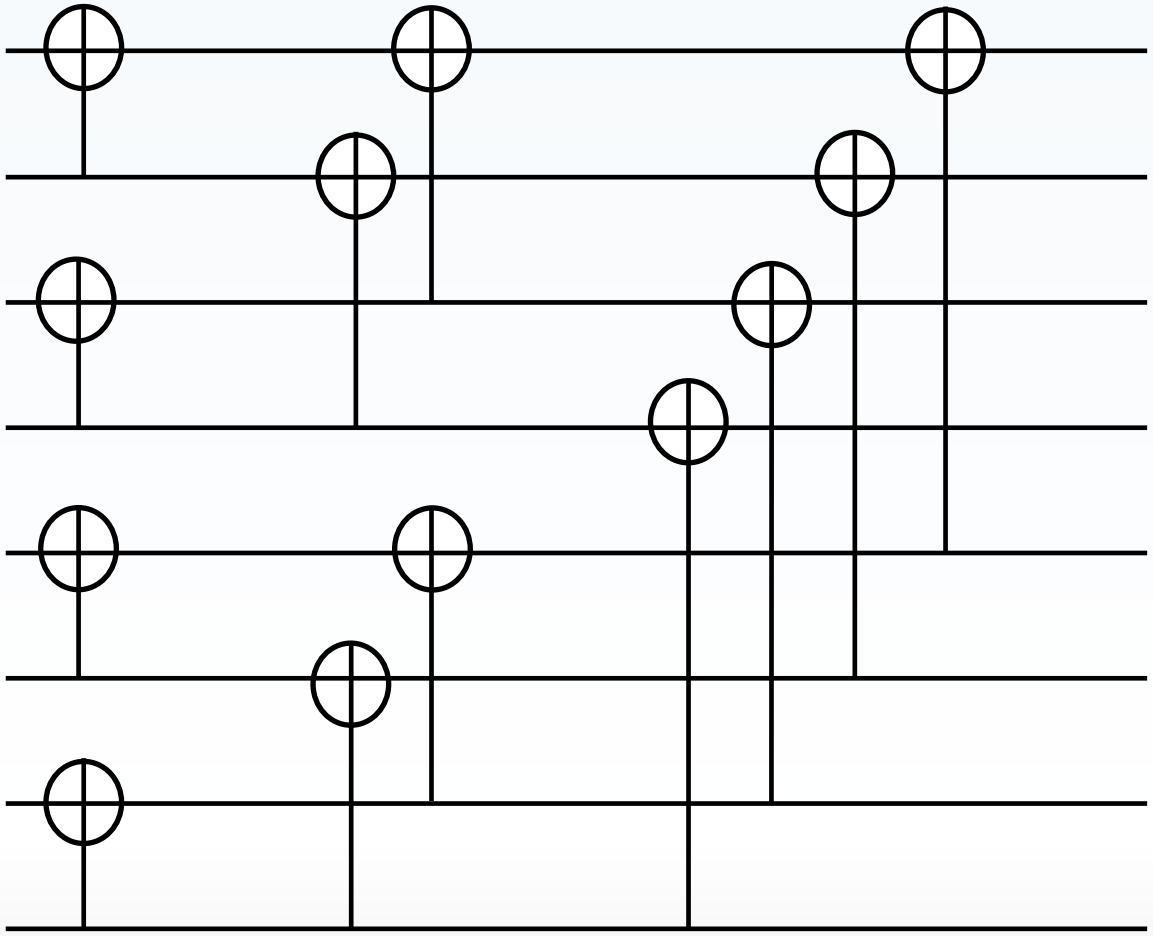
Q: SC decoding 과정과 LLR의 구하기 위한 Transition probability의 과정이 조금 다른 거 같다?

사실 polar code 최적 성능은 ML(maximum likelihood) decoding을 해야 한다.
그리고 이때 정확히 transition probability 기반 LLR을 사용한다 하지만 ML은 복잡도가
너무 커 근사시킨 모델인 SC decoding을 쓴다

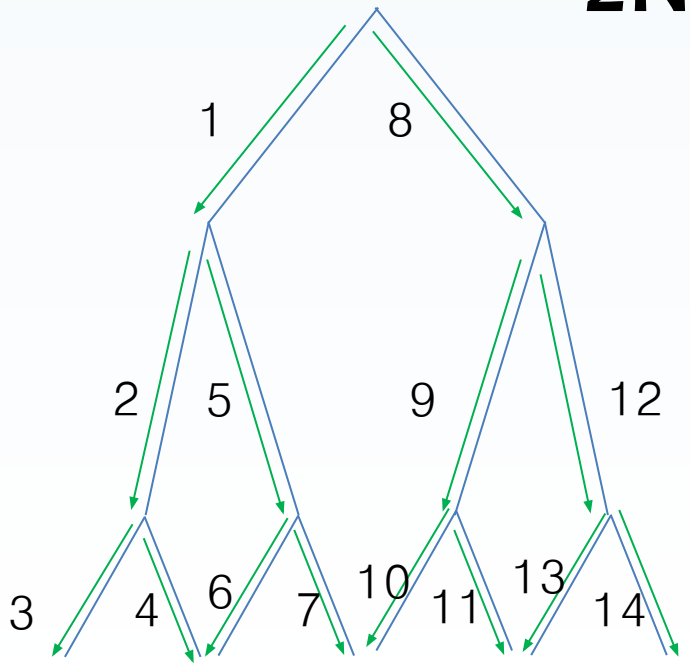
Q: SC decoding에서 LLR을 이용해도 되는 거냐?

SC decoding에서 쓰는 LLR 계산 공식은 무작정 근사한 게 아니라,
polar code의 재귀적 채널 구조(factor graph)를 이용해 transition probability를 엄밀히
유도하여 근사한 결과입니다 => 가능
수학적 증명 [1]

- N=8



2N-2



SCL decoder (Successive Cancellation List)

SC decoder problem: Limit of locally optimal search, in many cases the SC decoder cannot find the correct decoding path

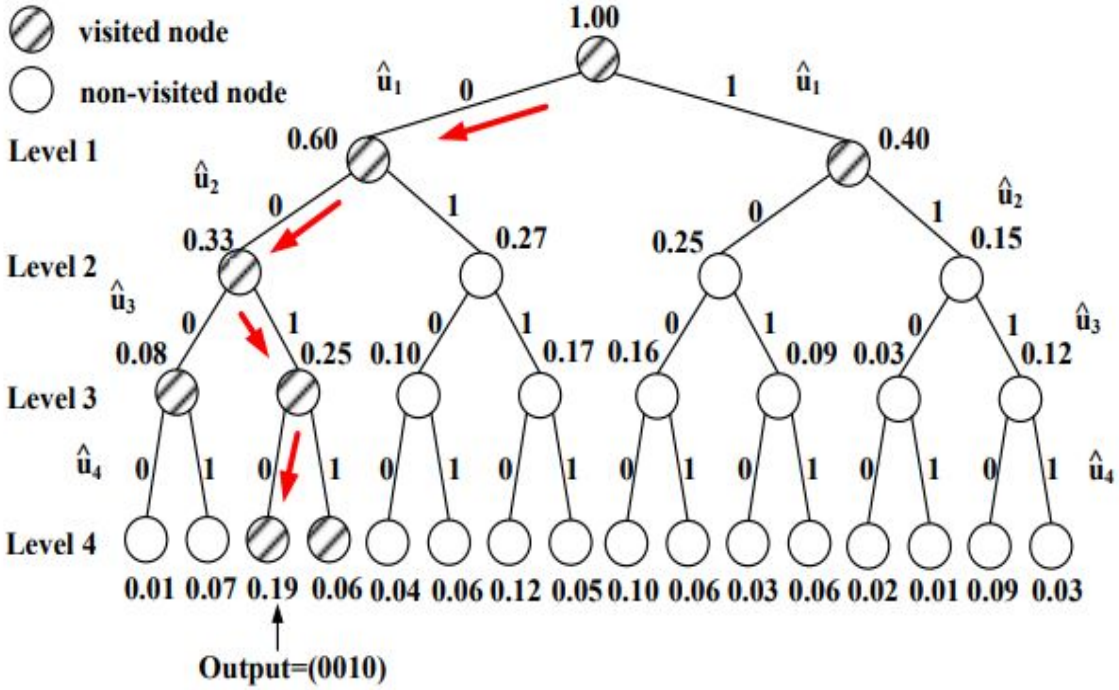


Fig. 3. Example SC searching process over code tree for n=4 polar code,

SC + List

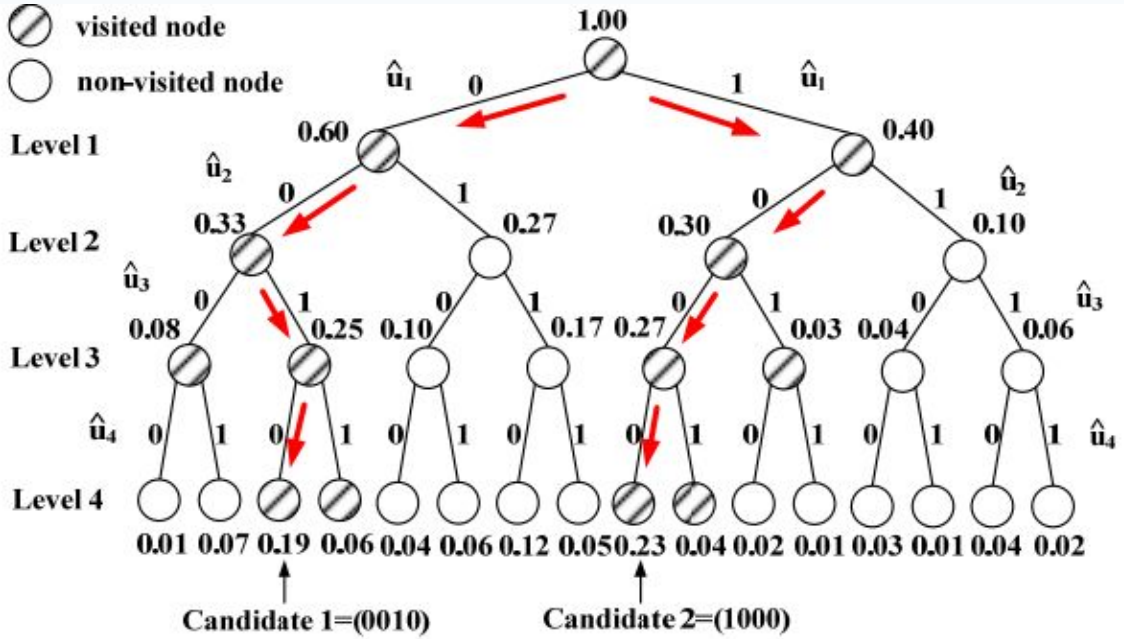
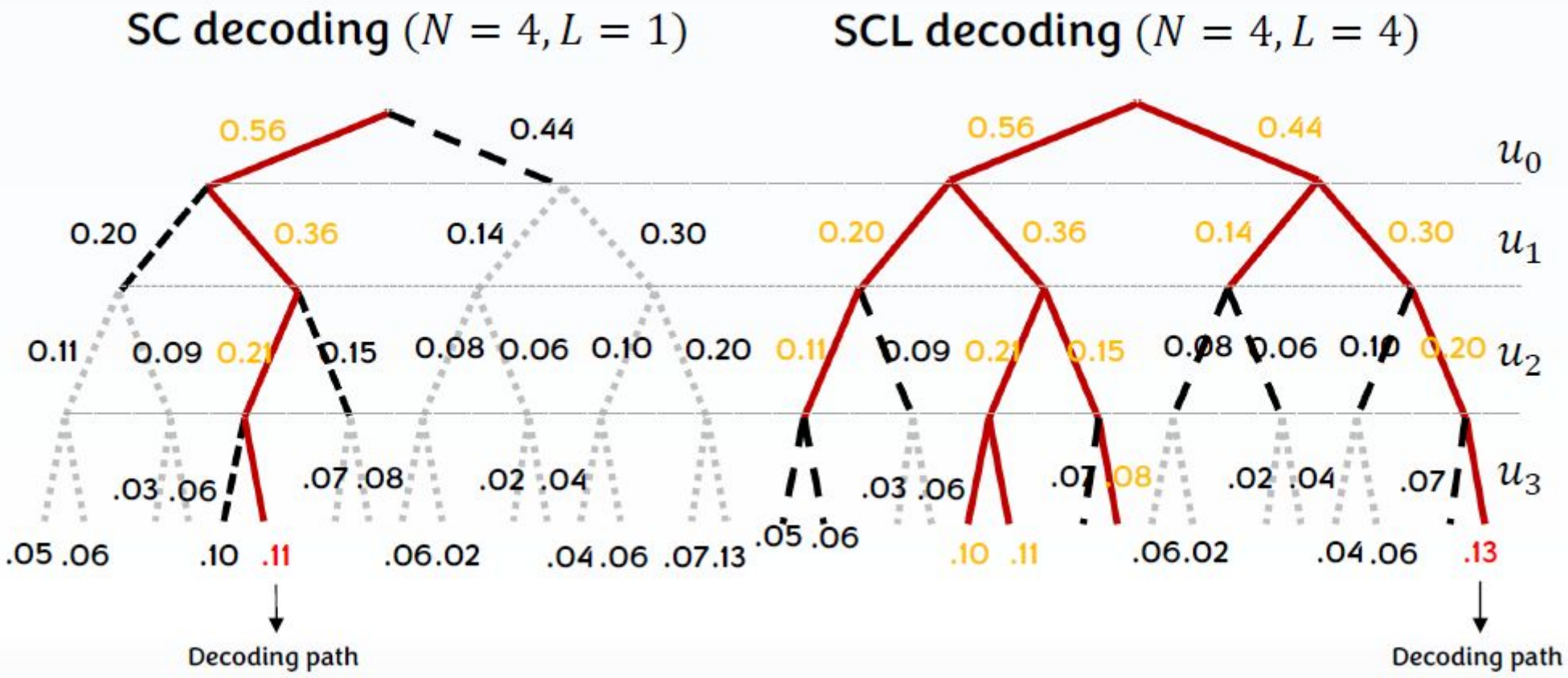


Fig. 4. Example SCL searching process over code tree for n=4 polar code with L=2,

[2
]

각 단계에서 가장 높은
확률만 살아남음!

가장 확률 높은 놈 + L-1
개만큼 추가 생존


SC decoding $(N = 4, L = 1)$
SCL decoding $(N = 4, L = 4)$
0.56
0.44
0.20
0.36
0.14
0.30
0.11
0.09
0.21
0.15
0.08
0.06
0.10
0.20
.03
.06
.07
.08
.02
.04
.05
.06
.10
.11
.06
.02
.04
.06
.07
.13
u_0
u_1
u_2
u_3
Decoding path
Decoding path

CRC-aided SCL decoder

Improve SCL

Example)
Suppose N=16, L=4, CRC bit= 001

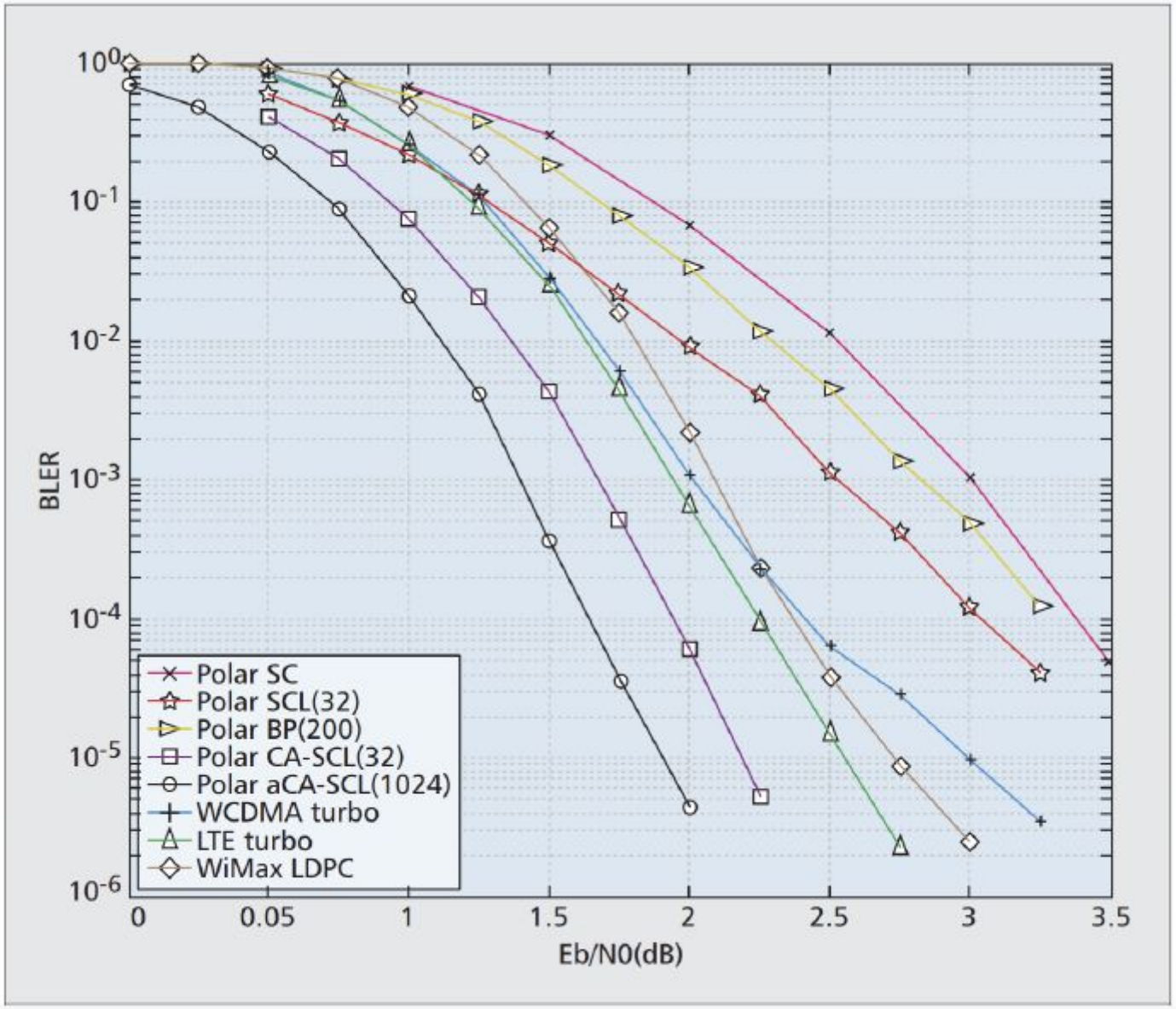
N=16 => code length is 16
L=4 => save 4 path
CRC bit=001 => 검사용 비트 3개를 넣어라(위치는 마음대로/ 아래 예시에선 앞 3개)

if decoder result
(0,0,1,1,1,0,0,0,0,1,1,1,1,0,1,1) => good
(0,0,1,0,0,0,0,0,1, 1, 1,0,1,1,0,0) => good
(1,0,1,1,1,0,0,0,1,1,1,0,0,1,0,1) => bad
(1,1,0,0,1,1,0,0,1,1,0,0,1,1,1,1) => bad

SCL + 검산 요소

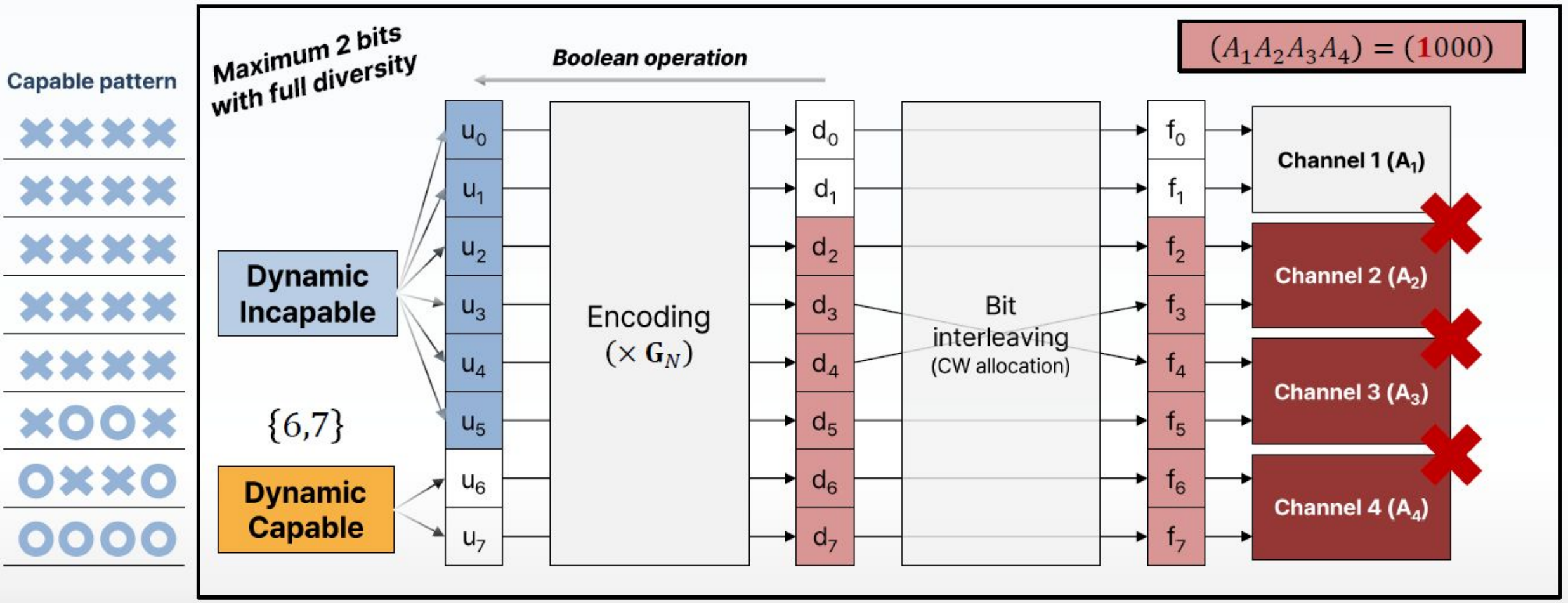
성능 비교

Polar codes are the first class of channel codes that are theoretically proven to achieve the Shannon limit for a large class of discrete memoryless channels [1]



information & frozen 배치를 어떻게 구할까?

추가 연구


Capable pattern
Maximum 2 bits with full diversity
Boolean operation
$(A_1A_2A_3A_4) = (1000)$
Dynamic Incapable
{6,7}
Dynamic Capable
u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7
Encoding ($\times \mathbf{G}_N$)
d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7
Bit interleaving (CW allocation)
f_0 f_1 f_2 f_3 f_4 f_5 f_6 f_7
Channel 1 (A_1)
Channel 2 (A_2)
Channel 3 (A_3)
Channel 4 (A_4)

2022

Reference

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] B. Yuan and K. K. Parhi, “Successive cancellation list polar decoder using log-likelihood ratios,” *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4844–4856, Oct. 2013.

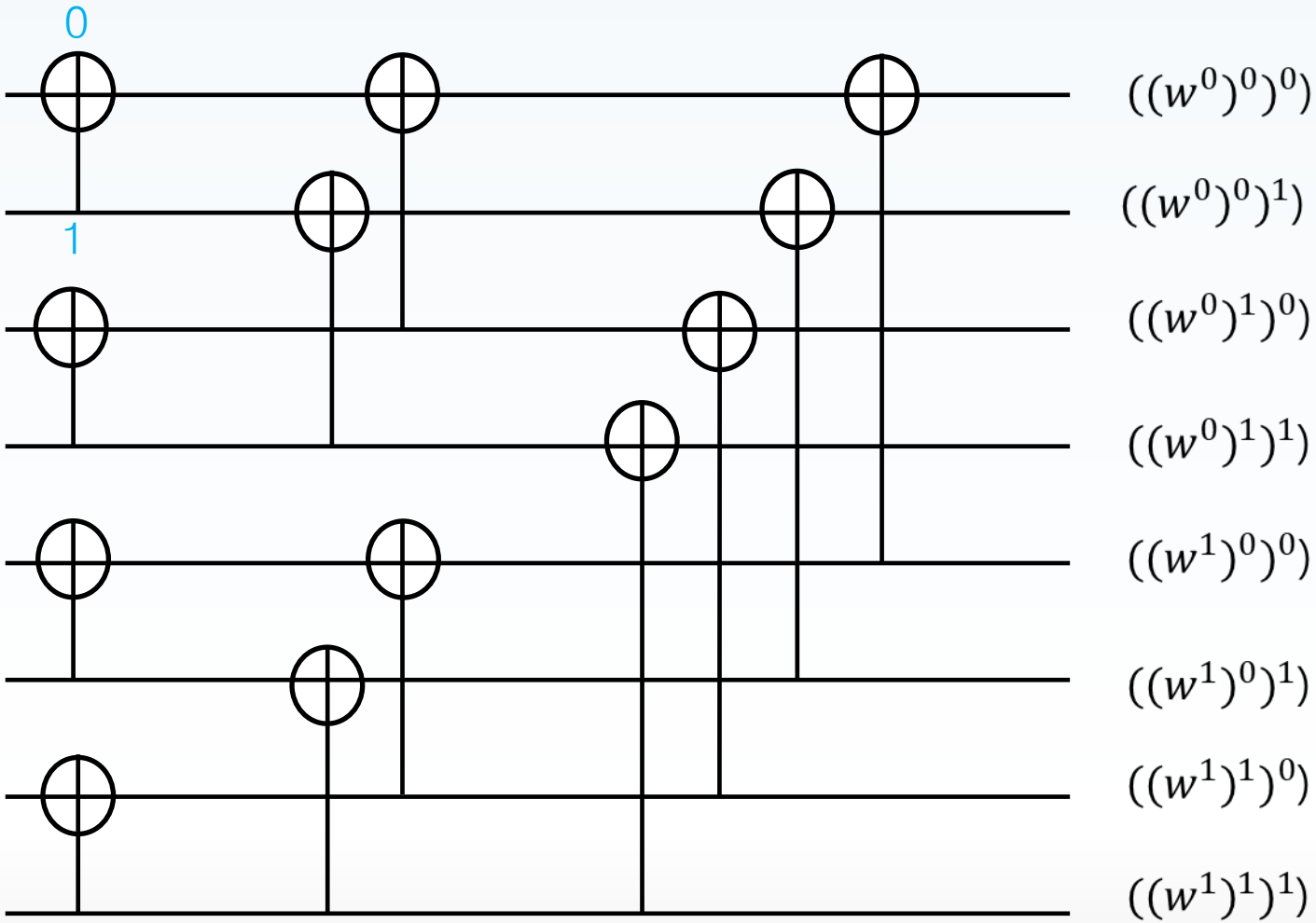
β-expansion A Theoretical Framework for Fast and Recursive Construction

- **Study purpose (X)**
- **UPO(UNIVERSAL PARTIAL ORDER)**
- **β-EXPANSION(PW algorithm)**
- **Example of fast construction(AWGN)**
- **Result(GA vs PW)**

Purpose

- Analyze the properties of universal partial order (UPO) and discover that UPO has an inherent recursive structure.
- Explore the theory of polarization weight algorithm by introducing a number theoretical concept called β-expansion.
- Find that polar codes can actually be efficiently constructed from UPO by applying β-expansion at each recursive step with a carefully chosen β.

사전지식



$$w_n^i = ((w^{b_0})^{b_1}.....)^{b_{n-1}},\ \mathrm{i} = 0, \dots, N-1$$
$$b_k \in \{0,1\}$$

UPO

- Partial order of reliability measure exists for any symmetric channels with binary inputs. [1]

- Reliability relation determined by applying the following rules. [2]

  1. Addition

  (a, b, 1, c) less reliable than (a, b, 0, c), ”1 $>$ 0”

  ex)

  2 (0, 1, 0) $<$ 3 (0, 1, 1),

  9 (1, 0, 0, 1) $<$ 15 (1, 1, 1, 1)

  2. Left-swap

  (a, 0, b, 1, c) less reliable than (a, 1, b, 0, c), ”0..1 $<$ 1..0

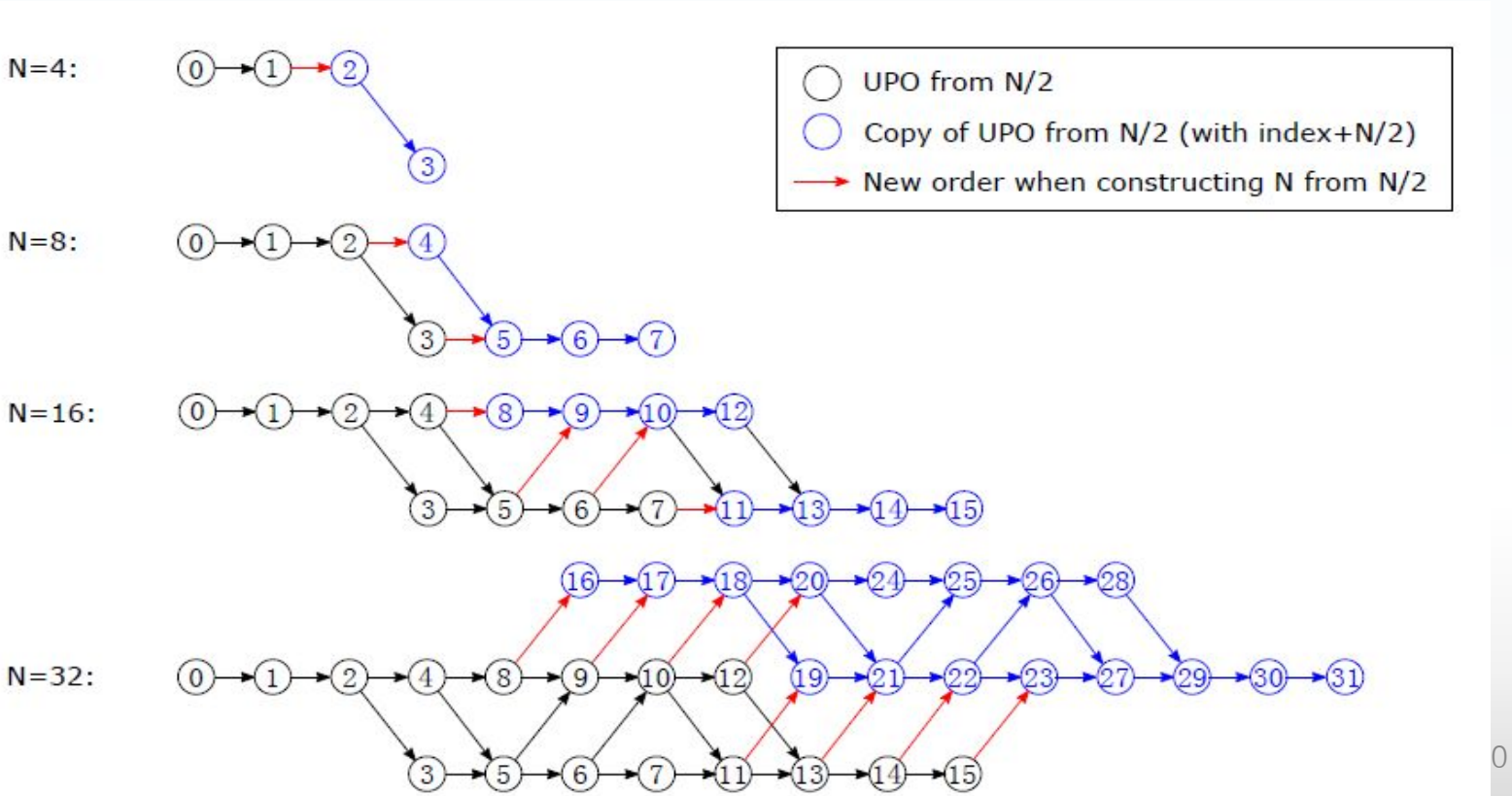
  ex)

  2 (0, 1, 0) $<$ 4 (1, 0, 0)

  12 (0, 1, 1, 0, 0) $<$ 24 (1, 1, 0, 0, 0)

- Properties of UPO
  - Nested: *Orders determined for a code of length N remain unchanged in code length of $2N$*
  - Symmetric: *Order of $x < y$ and the order of $(N-1-x) > (N-1-y)$ are twin pairs for a polar code of length N*

$$
\begin{aligned}
\mathbf{UPO}_1 &= \{\{0,1\}\},\\
\mathbf{UPO}_2 &= \{\{0,1\},\{1,2\},\{2,3\}\},\\
\mathbf{UPO}_3 &= \{\{0,1\},\{1,2\},\{2,3\},\{2,4\},\{3,5\},\\
&\quad \{4,5\},\{5,6\},\{6,7\}\},\\
\mathbf{UPO}_4 &= \{\{0,1\},\{1,2\},\{2,3\},\{2,4\},\{3,5\},\\
&\quad \{4,5\},\{5,6\},\{6,7\}\\
&\quad \{4,8\},\{5,9\},\{6,10\},\{7,11\},\\
&\quad \{8,9\},\{9,10\},\{10,11\},\{10,12\},\{11,13\},\\
&\quad \{12,13\},\{13,14\},\{14,15\}\}
\end{aligned}
$$



β-EXPANSION (PW algorithm)

- **Definition**

$$f^{pw}: x \rightarrow \sum_{i=1}^{n} b_i \beta^i$$

Ex) (β=**2**) $45 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 2^0$

- **Proposition**
  - Nested structure: *PW algorithm preserves the feature of nested code construction*
  - *Any sequence derived from PW algorithm respects the UPO of polar codes*

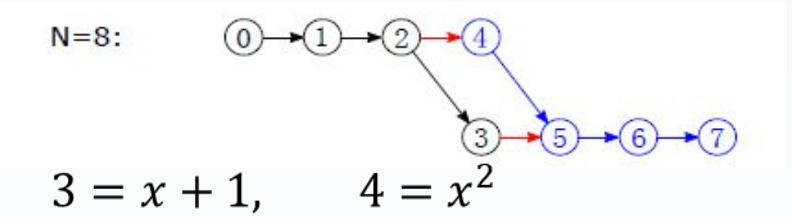
Question

Which value β should take such that PW algorithm or β-expansions can give a good sequence

Example

- n=3

$$\mathrm{UPO}_3 = \{\{0,1\},\{1,2\},\{2,3\},\{2,4\},\{3,5\},\{4,5\},\{5,6\},\{6,7\}\},$$



$3 = x + 1, \qquad 4 = x^2$

By solving $x^2 - x - 1 = 0$

$If\ \beta \in (1, 1.618) \rightarrow \beta + 1 > \beta$

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7$$

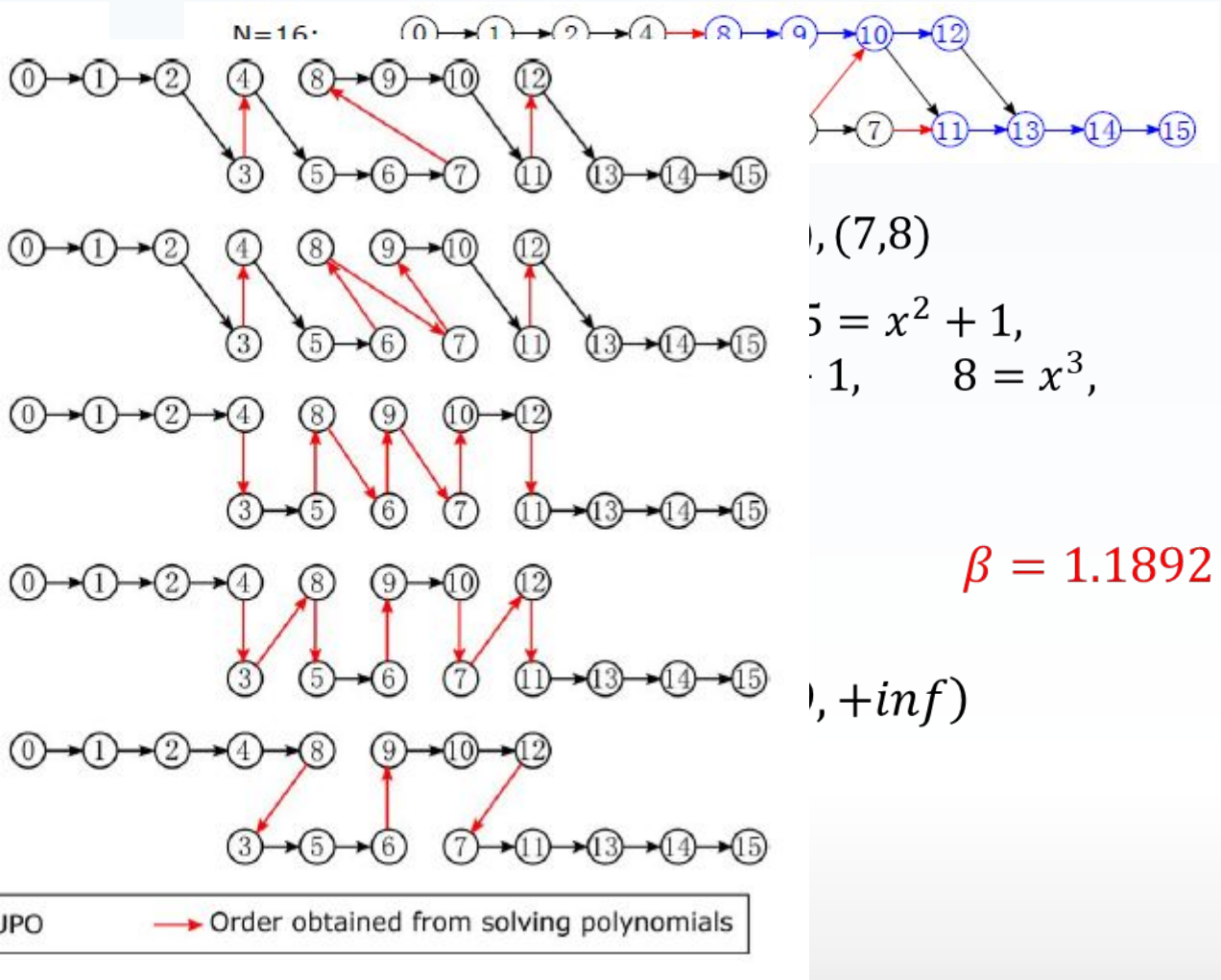
$If\ \beta \in (1.1618, inf) \rightarrow \beta + 1 < \beta$

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$$

Example

- n=4

$$\mathrm{UPO}_4 = \{\{0,1\},\{1,2\},\{2,3\},\{2,4\},\{3,5\},\{4,5\},\{5,6\},\{4,8\},\{5,9\},\{8,9\},\{9,10\}\{12,13\},\{13,$$



, (7,8)

$5 = x^2 + 1,$

$1, \quad 8 = x^3,$

By solving

$$x^3 - x^2 - x - 1 = 0$$

$$x^3 - x^2 - 1 = 0 \quad (2)$$

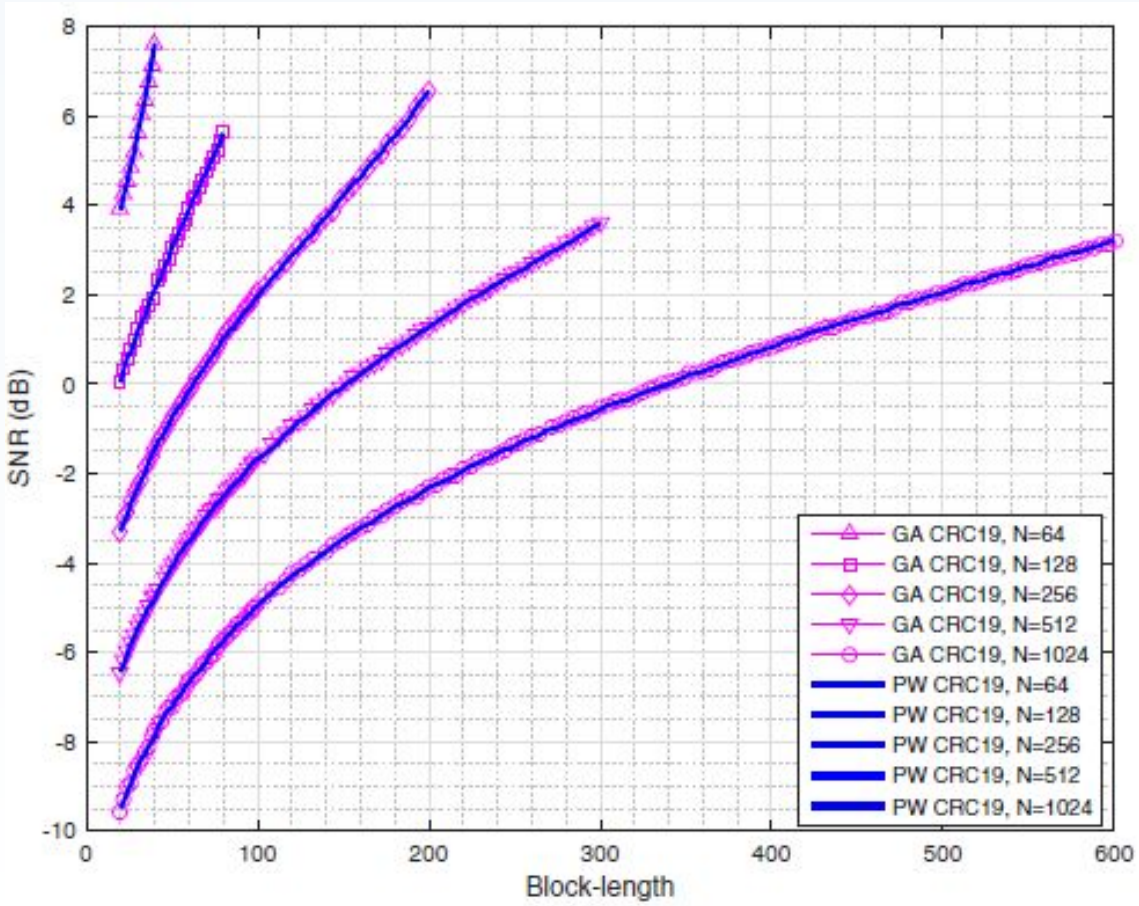
$$x^2 - x - 1 = 0 \quad (3)$$

$$x^3 - x - 1 = 0 \quad (4)$$

$$\beta = 1.1892 \approx 2^{\frac{1}{4}}$$

$, +inf)$

Result



Compare the performance of GA[3] and low-complex β-expansion construction with β = 1.1892 for the AWGN channel with QPSK and target block error rate (BLER) of 0.001

References

[1] C. Schurch, “A partial Order for the Synthesized Channels of a Polar Code, ISIT 2016.
[2] M. Mondelli, S. H. Hassani and R. Urbanke, “Construction of Polar Codeswith Sublinear Complexity,” submitted to ISIT 2017.
[3] I. Tal and A. Vardy, “How to construct polar codes,” IEEE Trans. Inf. Theory, vol. 59, no. 10, pp. 6562–6582, 2013.

Genetic Algorithm-based Polar Code Construction for the AWGN Channel

- Study purpose (x)
- Genetic algorithm-based polar code construction
  - Initialization
  - Mutation
  - Crossover
  - Selection and population update
- Simulation result over AWGN
- conclusion

Study purpose

- Design method that considers the decoder and improves the overall performance is an important step for future polar code applications
- In this work, we propose a new framework for polar code design matched to a specific decoding algorithm embedded in the well-understood Genetic Algorithm (GenAlg) context.

Final result

polar code of length 2048 with code rate 0:5, without the CRC-aid, tailored to plain successive cancellation list (SCL) decoding, achieving the same error-rate performance as the CRC-aided SCL decoding

&

Belief propagation (BP)-tailored polar code approaches the SCL error-rate performance

사전지식

- Dec
  - S
  - S
  - C
  - E
- Info
  A=[
  froz
- Cor
  - C
  - D
  - M
    i
  - C

non-

Algorithm	Channel dependent	Decoder optimized	Minimum distance improved	complexity
Arıkan (I(W), Z(W))	Yes	SC only	No	High
Density Evolution / Gaussian Approx	Yes	SC only	No	High
RM rule	No	MAP/SCL	Yes	Low
Hybrid Polar-RM	No	SCL(with out CRC)	Yes	Medium
LLR-based / Monte Carlo	Yes	BP/SCL	Yes	Medium high
Polarization Weight (PW)	No	Mainly SC	No	Very low
Partial Order (Synth. Channel Order)	Structural only	General	No	low

- The code construction can be considered as an optimization problem, where the objective is to find the optimal set of k good positions in a set of indices $\{1, \dots, N\}$

$$A_{opt} = \arg \min_{A} BER(A) \; or \; BLER(A)$$

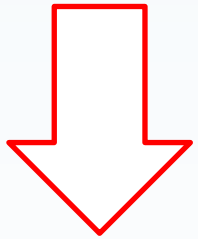
Subject to $$\left(\sum_{i=1}^{N} a_i \right) = k, \quad a_i \in \{0, 1\}$$

1. Deep Learning Based Polar Code Construction for 5G and Beyond
2. Learning to Construct Polar Codes
3. Gradient-based Optimization of Polar Codes

Genetic algorithm-based polar code construction

- **Initialization** (부모로 누굴 선택할 거냐?)

  " Population P_{init} is initialized with a sufficiently good collection of estimated A-vectors"

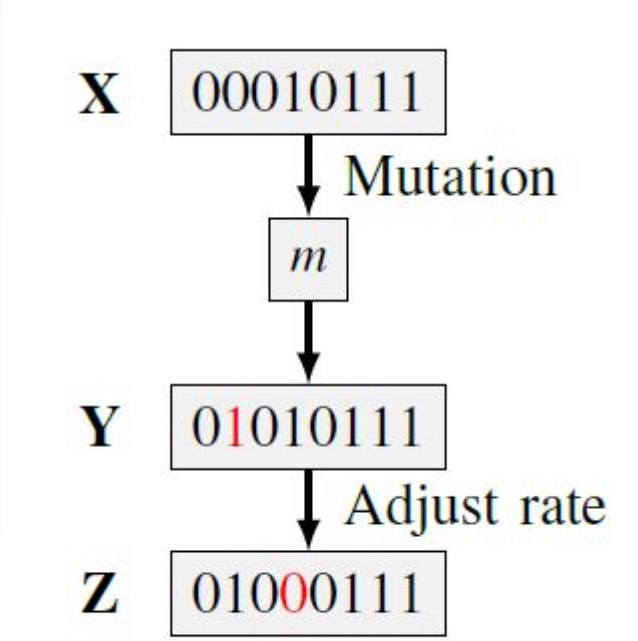


1. Based on the Bhattacharyya construction [1] obtained for BECs with various erasure probabilities.

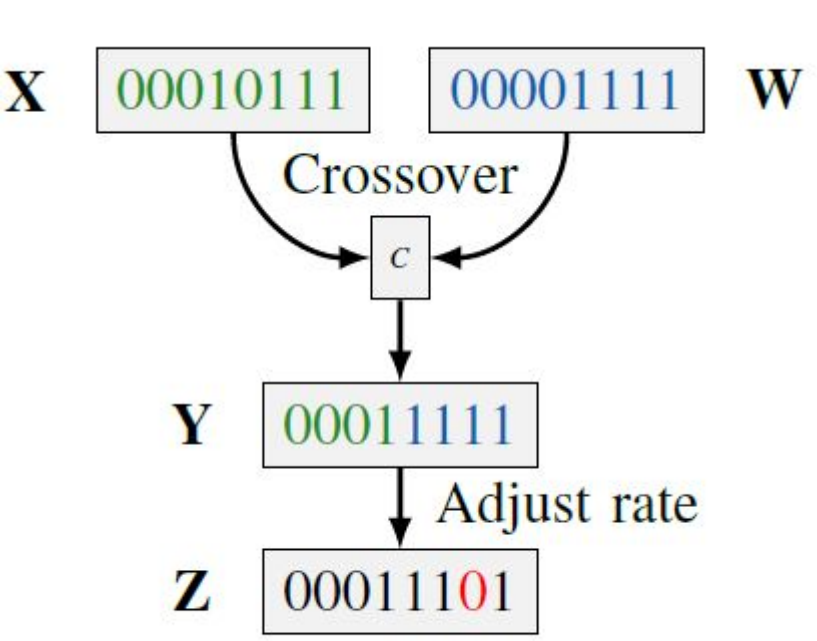
&

2. Based on hybrid RM [2] obtained in the SCL-tailored polar code construction phase

- Mutation, Crossover



(a) Mutation



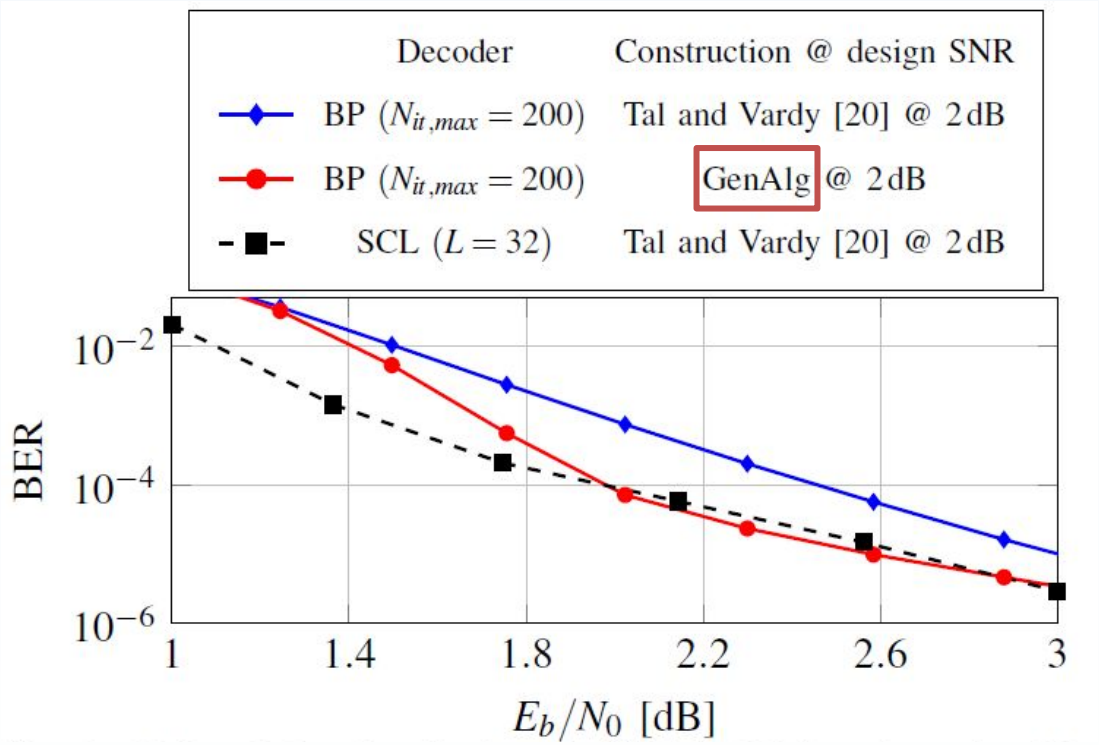
(b) Crossover

- **Selection and population update**
  - The fittest T A-vectors are always pushed forward as members of the new population
  - Crossovers are applied between each pair of the fittest T A-vectors, resulting in new $\binom{T}{2}$ offspring
  - Mutations are applied on the fittest T A-vectors, resulting in new T mutated offspring

new population S

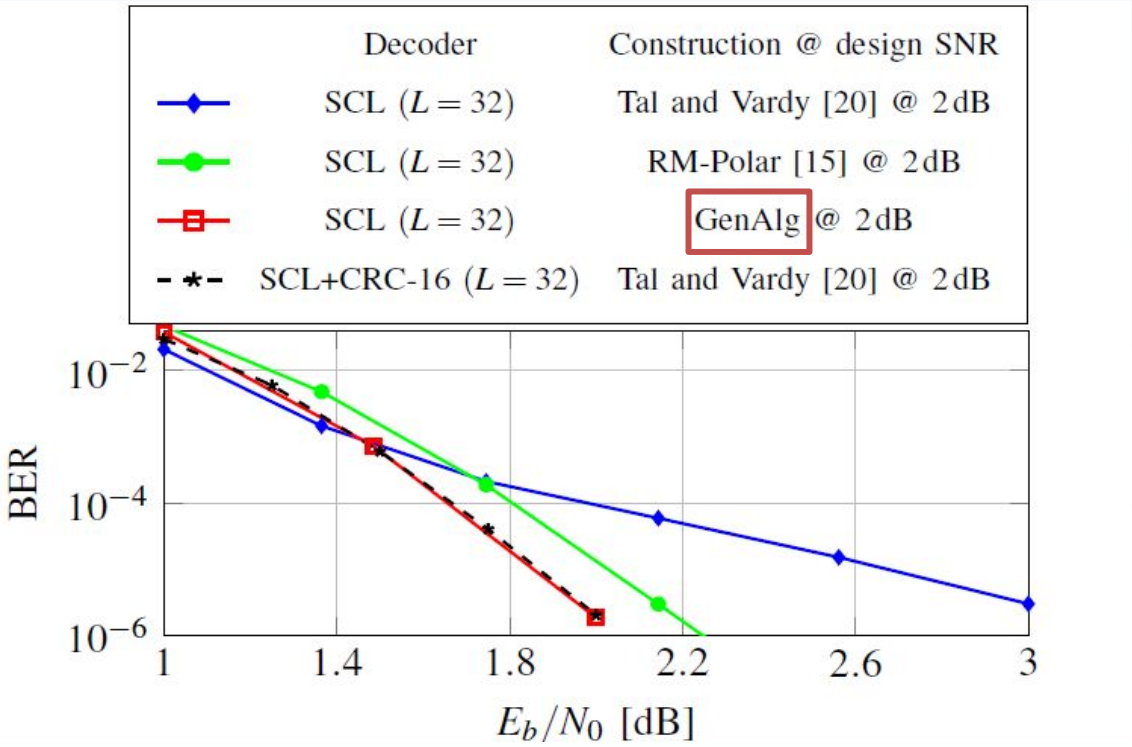
$$S = T + \binom{T}{2} + T = \frac{T^2 + 3T}{2}$$

Simulation result over AWGN



BER comparison between a code constructed via [20] and a code constructed using our proposed GenAlg under BP decoding

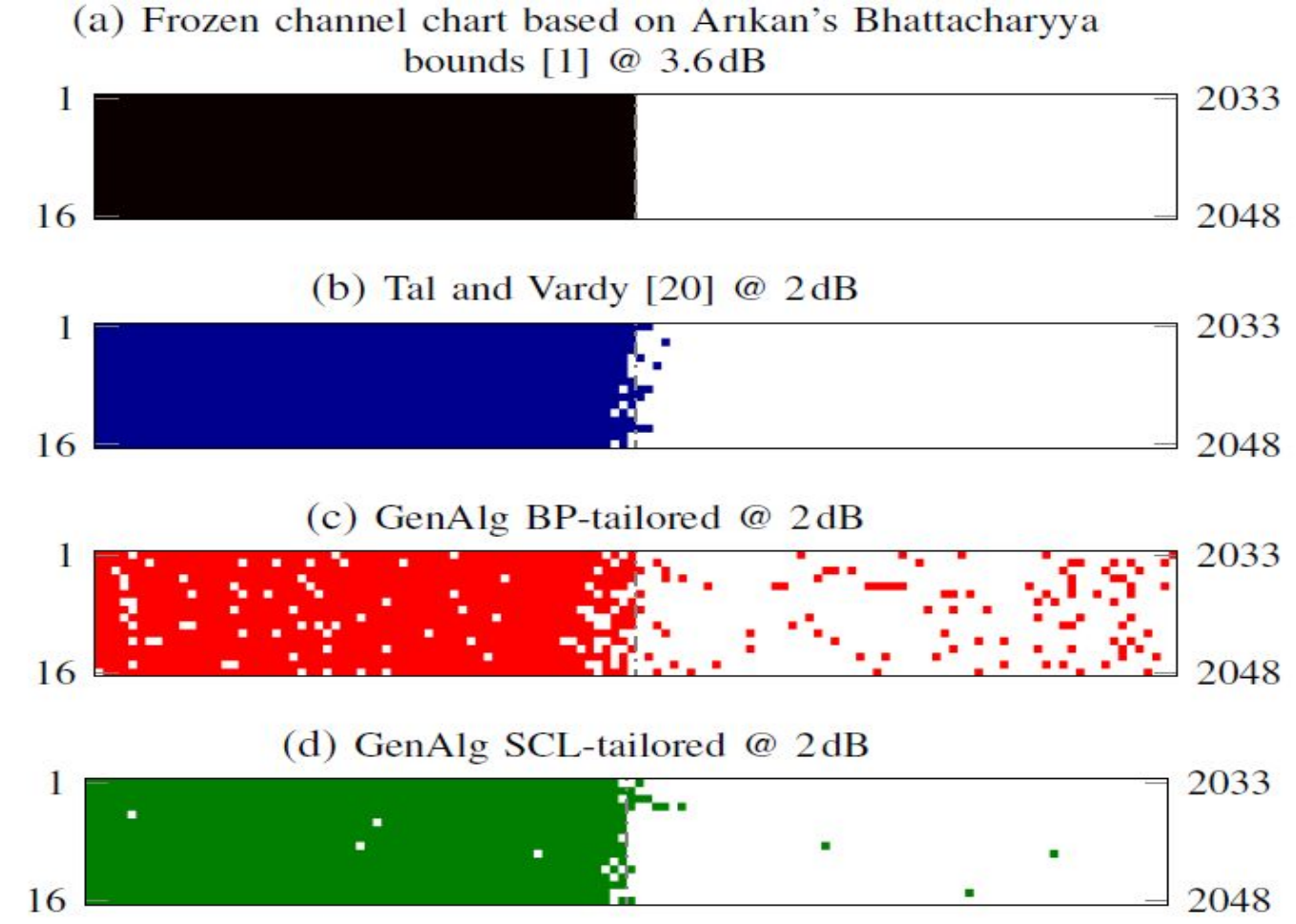
GenAlg is better than Tal and Vardy[20]



BER comparison between a code constructed [20] and a code constructed using our proposed GenAlg under SCL (L = 32) decoding.

SCL achieving the same error-rate performance as the CRC-aided SCL decoding

Distribution of frozen bit-channels over the N synthesized virtual channels



Property

1. The conventional construction techniques ([1], [20]) assume that an SC decoder is used. Thus, yield very similar A-vectors shown in Fig. 5a and 5b
2. A-vector tailored to BP decoding and SCL decoding has a much different frozen bit-channel distribution as shown in Fig. 5c

GenAlg-based polar code construction takes the decoding nature into consideration while constructing the code

추가연구

부모 노드를 Bhattacharyya 기반 + RM 기반 + 약간의 다양성으로 구성하면 GA가 매우 효율적으로 작동함 by Decoder-Tailored Polar Code Design Using the Genetic Algorithm (경험적 결론임 수학적 x)[3]

References

[1] E. Arıkan, “Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channe

[2] B. Li, H. Shen, and D. Tse, “A RM-Polar Codes,” ArXiv e-prints, July2014.

[3] Y. Takeuchi, R. Matsumoto and T. Uyematsu, "Decoder-Tailored Polar Code Design Using the Genetic Algorithm," *IEEE Access*, vol. 8, pp. 158647–158656, 2020, doi: 10.1109/ACCESS.2020.3020572.

[20] I. Tal and A. Vardy, “How to Construct Polar Codes,” IEEE Trans. Inf. Theory, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

Construction of Polar Codes with Reinforcement Learning

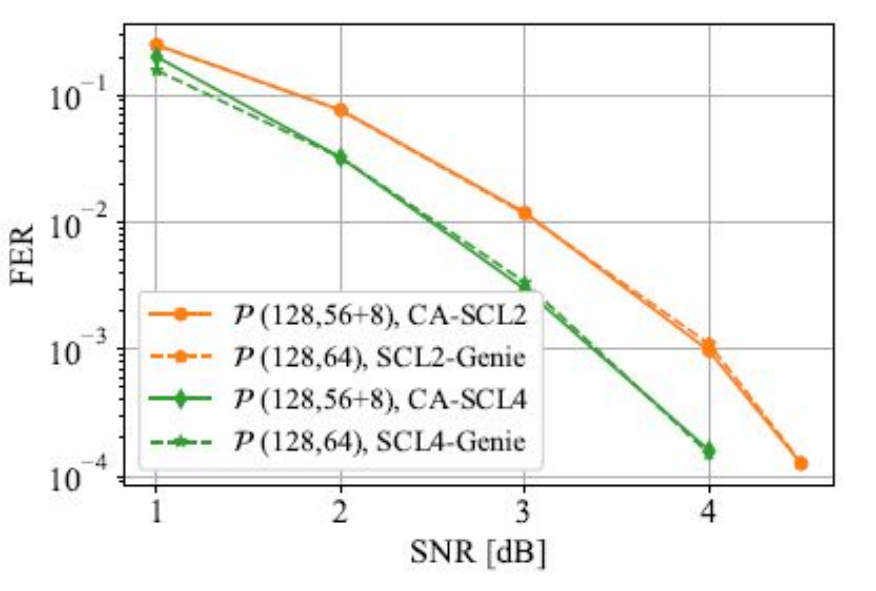
- Study purpose(x)
- Viewing polar code construction as a game
- Reinforcement learning algorithm
  - Reinforcement Learning Basics
  - SARSA(λ) with Eligibility Trace
  - Equivalence to Polar Code Construction Problem
- Simulation results

Study purpose

- Formulates the polar-code construction problem for the successive-cancellation list (SCL) decoder as a maze-traversing game(solved by reinforcement learning techniques)
- Sorting and selecting bit-channels by reliability → decides whether the input bits should be frozen in a purely sequential manner
- Simulation results show that with a moderate amount of training, the game-based polar-code constructions can match current standard constructions for SC decoding, and outperform the standard construction with SCL decoding.

사전지식

- Standard construction(x)
  - Gaussian approximation of density evolution [1], and upgrading/downgrading of channels [2]
  - machine learning were used to construct codes for specific decoders[3]
- Decoder
  - SCL without assistant bits
  - CRC-aided SCL(CA-SCL)
  - SCL-Genie: Correct codewords are known for the training samples.(이미 정답 code word를 아는 디코더)



FER of P(N,K) under the SCL-Genie decoder is almost identical to the FER of P(N,A + P) under the CA-SCL decoder when K = A + P
⇒ Learned code construction from the game by SCL-Genie is nearly optimal for P(N,A + P) under the CA-SCL decoder
(CRC-SCL 쓰는 이유)

Polar Code Construction Game

- Polar code construction game
  - Environment: A maze with height N −K+1 and width K + 1.
  - (0,0)시작 → (K, N-K)끝
  - Rule: Algorithm1
  - Reward: Associated with SCL-Genie
  - Goal: Find the best path that yields the highest expected return

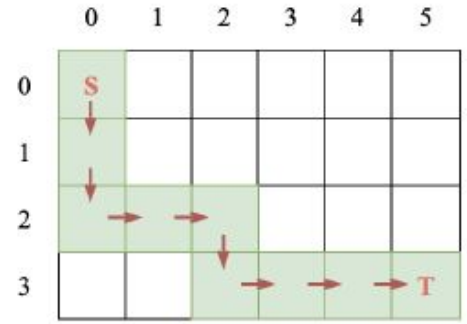


Fig. 3: An example of a polar code construction game for $\mathcal{P}(8,5)$. The action list is {down, down, right, right, down, right, right, right}. The corresponding nonfrozen bit positions are $\{2,3,5,6,7\}$.

- Reward generation via SCL-Genie decoder(X)
  - SCL-Genie
    - Decoding sequentially
    - Independent of how frozen and nonfrozen bits are distributed after it
  - Reward generation process
    - Selected actions are penalized when the SCL-Genie decoder fails to decode the correct codeword
    - Terminated after the decoder drops the correct codeword (SC-based decoders only append new bits after the already decoded bit stream, and no previous decisions will be altered

Algorithm 1: Reward generation at step k

Input: step k, action a
Global Variable: PM list, survival path list
Output: reward r, termination flag F

a
1: **if** $k = 0$ **then**
2: Transmit all-zero codeword through the channel
3: Initialize the SCL-Genie decoder with the received LLR
4: **end if**

b
5: **if** $a = 0$ **then**
6: Decode the k-th bit as if it is a frozen bit
7: Update the PM list and the survival paths
8: **else**
9: Decode the k-th bit as if it is a nonfrozen bit
10: Update the PM list and the survival paths
11: Check if the all-zero codeword survives
12: **if** all-zero codeword is dropped from the list **then**
13: Set $r = -1$, $F =$ True
14: Clear PM list, survival path list
15: **return** (r, F)
16: **end if**
17: **end if**

c
18: Set $r = 0$
19: **if** $k = N - 1$ **then**
20: Set $F =$ True
21: Clear PM list, survival path list
22: **end if**
23: Set $F =$ False
24: **return** (r, F)

Transmitting all-zero codeword

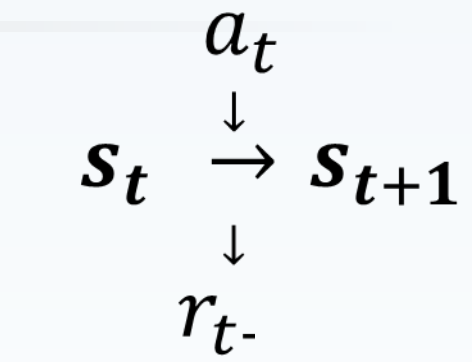
$$s_t \xrightarrow{a_t} s_{t+1}, \quad s_t \xrightarrow{} r_{t-}$$

Interpretation

A. Definition of Terms
B.
1. First step(k=0)
2. a=0(frozen)
3. a=1(nonfrozen)
4. If all-zero code word is droken → r=-1 & end
5. if all-zero code word is not droken → r=0 & continue
C. Last step (k=N-1)

Environment: State $s_t \in S$
Action: $a_t \in A$ **according to policy** $\pi : S \rightarrow A$
Return: $R = \sum_{t=0}^{T} \gamma^t r_{t+1}$

Reinforcement learning algorithm



- **Reinforcement Learning Basics**
  - **Environment: State $s_t \in S$**
  - **Action: $a_t \in A$ according to policy $\pi : S \rightarrow A$**
  - **Return: $R = \sum_{t=0}^{T} \gamma^t r_{t+1}$**
  - **r : Discount rate (how much the agent weights the future reward, T : End time**
  - **Agent policy π is derived from value function Q : S × A → R which approximates the expected return when taking action a from state s**
  - $Q^{\pi}(s,a) := E_{\pi}\left[\sum_{j=0}^{T-t-1} \gamma^t r_{t+j+1} \middle| s_t = s, a_t = a\right]$ (정의 보상의 평균값)
  - $Q^{\pi}(s,a) = E[r_{t+1} + \gamma Q^{\pi}(s_{t+1}, \pi(s_{t+1}) | s_t = s, a_t = a]$ (다른 표현) 현재 + 미래 평균값
  - **ϵ- greedy policy**

$$\pi(s) = \begin{cases} argmax Q(s,a) & 1-\epsilon \\ a \in A & \\ random\ a \in & \epsilon \end{cases}$$

Goal: Optimize its policy π to maximize the expected return E[R]

$$V_M^{\pi}(s) = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r\left(s_t, a_t\right) \mid \pi, s_0 = s\right]$$

$$Q_M^{\pi}(s,a) = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r\left(s_t, a_t\right) \mid \pi, s_0 = s, a_0 = a\right]$$

$$Q^{\pi}(s,a) = (1-\gamma)r(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[V^{\pi}\left(s'\right)\right]$$

Stochastic
by LLR

- **SARSA(λ) with Eligibility Trace**

1. SARSA

$\boldsymbol{s \rightarrow a \rightarrow r \rightarrow s' \rightarrow a'}$

- 현재: S_t
- 현재 행동: A_t
- 보상: R_{t+1}
- 다음상태: S_{t+1}
- 다음행동: A_{t+1}

과정

1. 현재 상태 s에서 행동 a 선택 by ϵ**- greedy policy**
2. 행동 a 실행 → 보상 r 받고 새로운 상태 S′이동
3. S′에서 다음 행동 a′ 선택
4. Q 업데이트 (TD update)

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \rho[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

- ρ: 학습률
- γ : 할인계수(미래 보상의 중요도)

Simulation results

- **SARSA 예시**

[0, 0, 0, 0]을 잘 보내주기 위한 polar construct [0, 0, 1, 1]

$$\begin{cases} 0: frozen, 1: information \\ r = -1(\text{탈락}), r = 3(\text{통과}) \\ \rho = 0.5, \gamma = 0.9 \\ \text{초기값}: Q(s,a) = 0 \end{cases}$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \rho[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

Step 1 [1,0,1,1]

			상태				TD update
0	0	1	정보	-1			

Step 2 [0,0,1,1]

			결과				TD update
0	0	0	고정	0	1	0	
1	1	0	고정	0	2	1	
2	2	1	정보	0	3	1	
3	3	1	정보	3			

Step 3 [0,1,1,1]

			결과				TD update
0	0	0	고정	0	1	1	
1	1	1	정보	-1			

s	a=0	a=1
0	0	-0.5
1	0	-0.5
2	0	0
3	0	1.5

Step 4 [0,0,1,1]

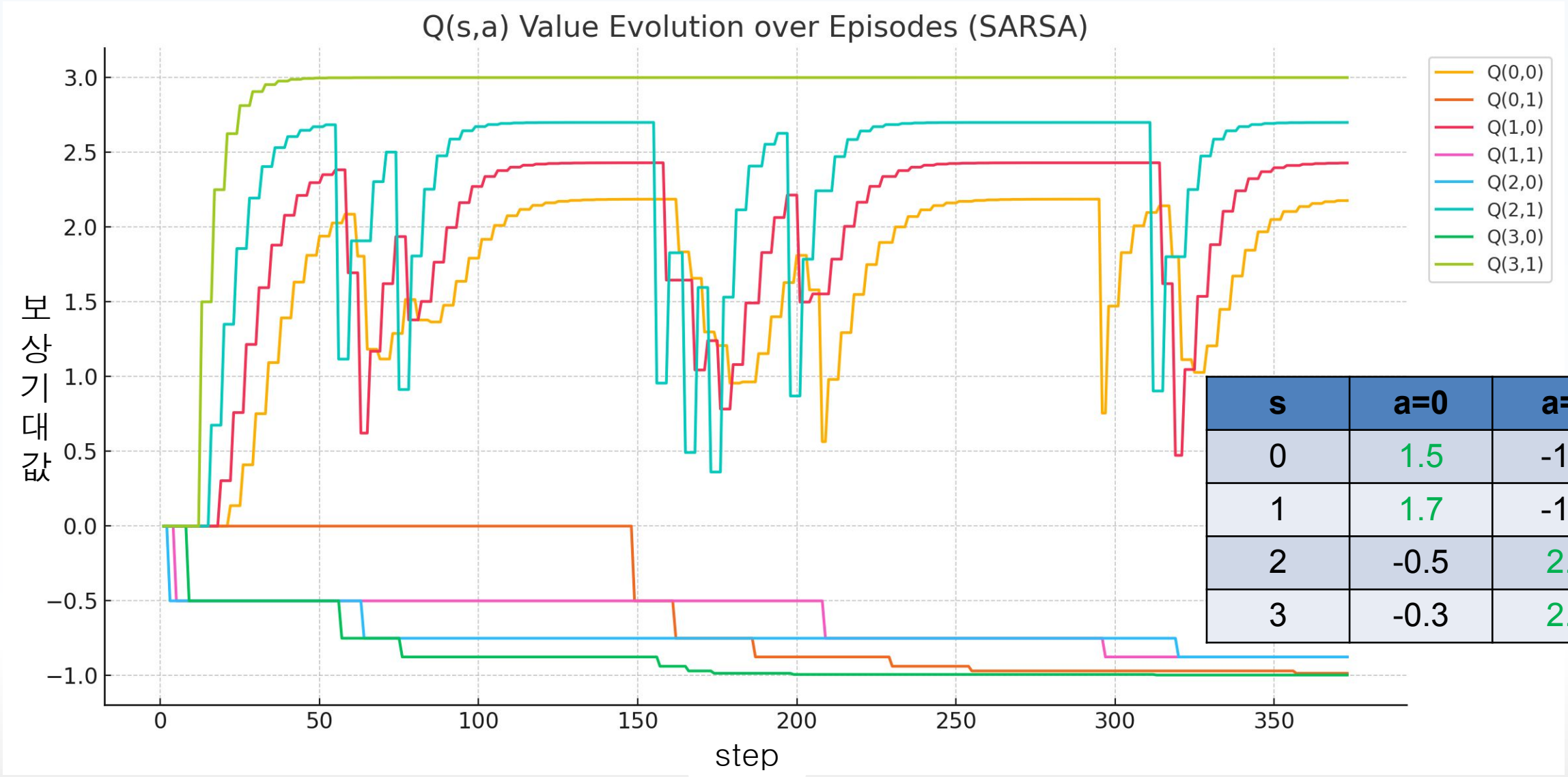
			결과				TD update
0	0	0	고정	0	1	0	
1	1	0	고정	0	2	1	
2	2	1	정보	0	3	1	
3	3	1	정보	3			

Step 5 [0,1,1,1]

			결과				TD update
0	0	0	고정	0	1	1	
1	1	1	정보	-1			

s	a=0	a=1
0	-0.225	-0.5
1	0	-0.75
2	0	0.675
3	0	2.25

If)



s	a=0	a=1
0	1.5	-1.0
1	1.7	-1.2
2	-0.5	2.0
3	-0.3	2.8

By chat gpt

2. Eligibility trace

Captures the current game's historical trace and assigns credit to every state-action pair
(과거 행동에도 학습 신호 전달)

$$E_t(s,a) = \gamma\lambda E_{t-1}(s,a) + f(s_t = s, a_t = a)$$

If)
실행 [0, 0, 1, 1], $\gamma = 1,\ \lambda = 0.8$

$$\mathrm{E}_1[0,0] = 1$$
$$\mathrm{E}_2[1,0] = 1, \mathrm{E}_2[0,0] = 0.8$$
$$\mathrm{E}_3[2,1] = 1, \mathrm{E}_3[1,0] = 0.8, \mathrm{E}_3[0,0] = 0.8^2$$
$$\mathrm{E}_4[3,1] = 1, \mathrm{E}_4[2,1] = 0.8, \mathrm{E}_4[1,0] = 0.8^2, \mathrm{E}_4[0,0] = 0.8^3$$

- 초기값
  $(E_0(s,a) = 0, \forall_s \in \mathrm{S}, \mathrm{a} \in \mathrm{A})$
- f(·) = indicator function
  $= \begin{cases} 1 \\ 0 \end{cases}$
  현재 time t에서 (s,a)를 선택했으면 1아니면 0
- λ
  How much agent would change the value function of early state

- SARSA + Eligibility Trace

$$\delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$
$$E_t(s,a) = r\lambda E_{t-1}(s,a) + f(s_t = s, a_t = a)$$

$$Q(s,a) \leftarrow Q(s,a) + \rho\delta_t E_t(s,a)$$

Step 4 [0,0,1,1]

			상태		TD update
0	0	0	고정	0	
1	1	0	고정	0	
2	2	1	정보	0	
3	3	1	정보	3	

$E_1[0,0] = 1$

$E_2[1,0] = 1, E_2[0,0] = 0.8$

$E_3[2,1] = 1, E_3[1,0] = 0.8, E_3[0,0] = 0.8^2$

$E_4[3,1] = 1, E_4[2,1] = 0.8, E_4[1,0] = 0.8^2, E_4[0,0] = 0.8^3$

If) t=3

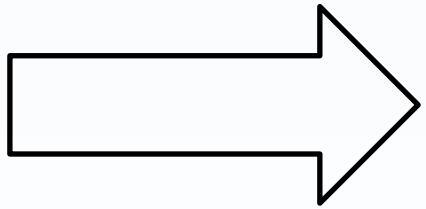
(s,a)	eligibility	TD update		Q update
(3,1)	1	2.25	0.5	0.5×2.25×1=1.125
(2,1)		2.25	0.5	0.5×2.25×0.8=0.9
(1,0)		2.25	0.5	0.5×2.25×0.64=0.72
(0,0)		2.25	0.5	0.5×2.25×0.512=0.576

step 4, t=3
(t=3에서 보상이 이뤄진다!!)

성공[0011] → (3,1)(2,1)(1,0)(0,0) : 양수
4번째 실패 [0010] → (3,0)(2,1)(1,0)(0,0): 음수
3번째 실패 [000*] → (2,0)(1,0)(0,0): 음수
2번째 실패 [01**] → (1,1)(0,0): 음수
1번째 실패 [1***] → (0,1): 음수

Sarsa

s	a=0	a=1
0	0	-0.5
1	0	-0.5
2	0	0.675
3	0	2.25

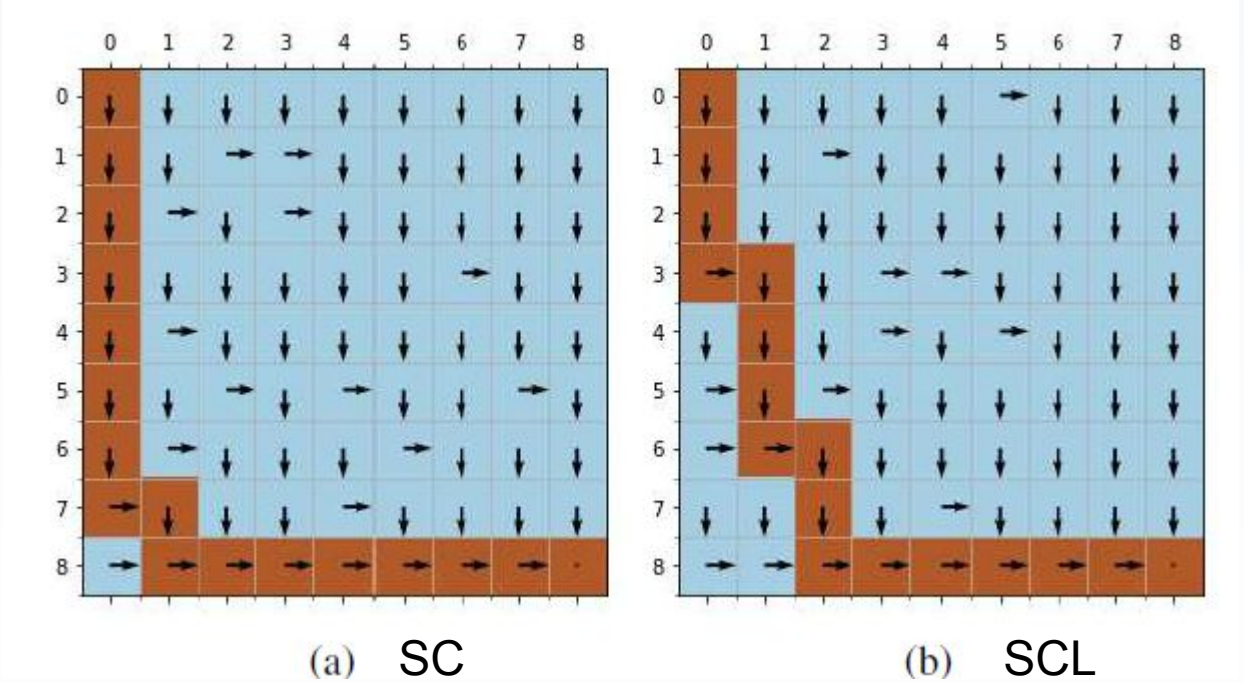


Sarsa(γ)

s	a=0	a=1
0	0+0.567	-0.5
1	0+0.72	-0.5
2	0	0.675+0.9
3	0	2.25+1.125

Step 4는 t=3에서 보상
이 값은 t=0, t=1, t=2, t=3의 값에
영향을 못 끼친다!!!

Step 4는 t=3에서 보상
이 값은 t=0, t=1, t=2, t=3의 값에
영향을 끼친다!!!



N=16(info:8, frozen: 8)
각 상태에서 LLR에 따라 a=0, 1인지 결정되는데 이에 따라 Q 값을 업데이트 된다

$$\pi(s) = \underset{a}{\operatorname{argmax}}\, Q(s, a)$$

결국 각 상태에서 들어오는 LLR 값에 따라 어떤 행동이 가장 좋은 지 나온다!

RL(N to 2N)

방식

1. $\pi(t) \rightarrow \pi(2t), \pi(2t+1)$ **2i, 2i+1**의 신뢰도를 고려 **X**
2. **β-expansion**이나 **partial order** 기반 확장 **[4]** **N→2N**

Reference

[1] P. Trifonov, “Efficient design and decoding of polar codes,” IEEETransactions on Communications, vol. 60, no. 11, pp. 3221–3227, 2012.

[2] I. Tal and A. Vardy, “How to construct polar codes,” IEEE Transactions on Information Theory, vol. 59, no. 10, pp. 6562–6582, 2013.

[3] L. Huang, H. Zhang, R. Li, Y. Ge, and J. Wang, “AI coding: Learning to construct error correction codes,” IEEE Transactions on Communica- tions, vol. 68, no. 1, pp. 26–39, 2020.

[4] S. A. Hashemi, C. Condo, and W. J. Gross, "Reinforcement learning for nested polar code construction," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6, doi: 10.1109/ICC.2018.8422650.