Code Rate Optimization via Neural Polar Decoders

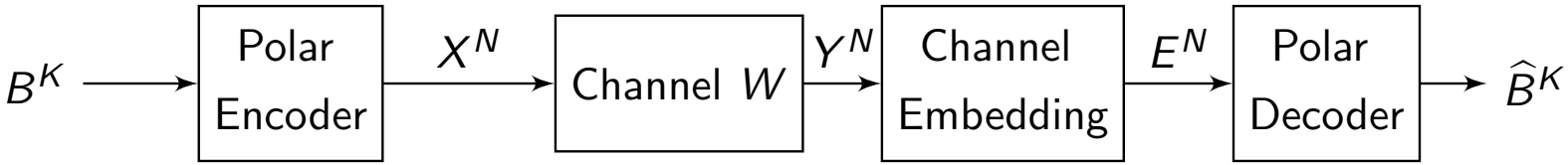
Ziv Aharoni[1], Bashar Huleihel[1], Henry D. Pfister[2], Haim H. Permuter[1]

[1]Ben-Gurion University of the Negev

[2]Duke University

International Symposium on Information Theory
December 1, 2025

Polar Coding Scheme

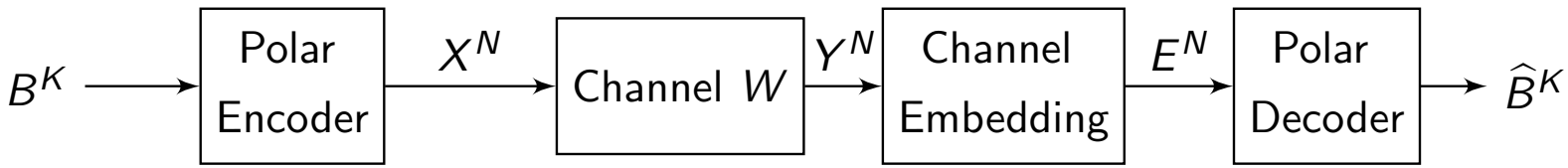


- Block length $N = 2^n$, $n \in \mathbb{N}$, K information bits
- Input alphabet $\mathcal{X} = \{0, 1\}$
- Output alphabet $\mathcal{Y}$
- Input distribution P_{X^N}, channel $W_{Y^N||X^N}$
- Polar decoder:
  - Successive cancellation (SC) decoder
  - BP is not considered
- Polar encoder: given the information set $\mathcal{A} \subset [N]$, $|\mathcal{A}| = K$

$$U_i = \begin{cases} U_i \sim \text{Ber}(0.5) & i \in \mathcal{A} \\ U_i = f\left(U^{i-1}\right) & i \in \mathcal{A}^c \end{cases}$$

$$X^N = U^N G_N$$

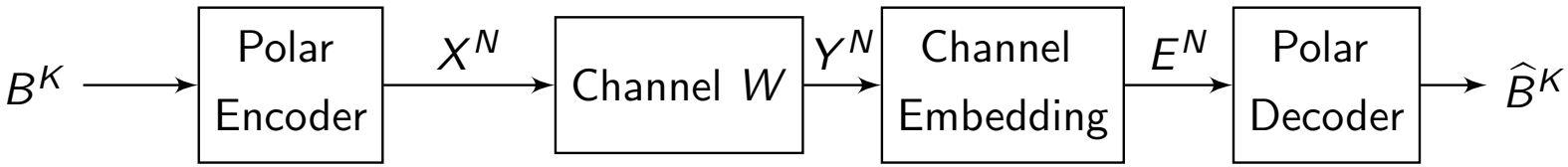
Successive Cancellation Decoder



- **Decoding requires 4 functions:**

Channel embedding:	$E : \mathcal{Y} \to \mathcal{E}$
Check-node:	$F : \mathcal{E} \times \mathcal{E} \to \mathcal{E}$
Bit-node:	$G : \mathcal{E} \times \mathcal{E} \times \mathcal{X} \to \mathcal{E}$
Embedding-to-LLR:	$H : \mathcal{E} \to \mathbb{R}$

Successive Cancellation Decoder

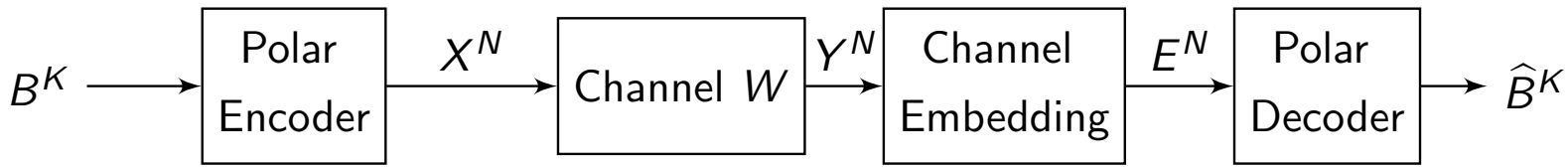


- **SC decoding for memoryless channels** [Ari09]

Channel embedding:	$E(y) = \log \frac{W(1\|y)}{W(0\|y)}$
Check-node:	$F(e_1, e_2) = -2 \tanh^{-1}\left(\tanh \frac{e_1}{2} \tanh \frac{e_2}{2}\right)$
Bit-node:	$G(e_1, e_2, u) = e_2 + (-1)^u e_1$
Embedding-to-LLR:	$H(e_1) = e_1$

  - **Computational complexity:** $O(N \log_2(N))$

Successive Cancellation Decoder



- **SC trellis decoding for finite state channels (FSCs)**
  - Channel embedding:

$$E(y) = \left\{ P_{X|S}(x|s) W(y|x,s) P_{S'|X,Y,S}(s'|x,y,s) \right\}_{x,s,s' \in \mathcal{X} \times \mathcal{S}^2}$$

  - The formulas for F, G, H are in [WHY+15]
  - The channel embedding are **vectors**.
  - **Computational complexity:** $O(|\mathcal{S}|^3 N \log_2(N))$

Neural Polar decoder

Definition(A./Huleihel/Pfister/Permuter 2023)

An neural polar decoder (NPD) is defined by $E_{\theta_E} F_{\theta_F}, G_{\theta_F}, H_{\theta_H}$ with parameters $\theta = \{\theta_E, \theta_F, \theta_G, \theta_H\}$. The functions satisfy:

- $E_\theta : \mathcal{Y} \to \mathbb{R}^d$ is the channel embedding NN.
- $F_\theta : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is the check-node NN.
- $G_\theta : \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}^d$ is the bit-node NN.
- $H_\theta : \mathbb{R}^d \to \mathbb{R}$ is the embedding-to-LLR NN.

Neural Polar decoder

Definition(A./Huleihel/Pfister/Permuter 2023)

An NPD is defined by $E_{\theta_E} F_{\theta_F}, G_{\theta_F}, H_{\theta_H}$ with parameters $\theta = \{\theta_E, \theta_F, \theta_G, \theta_H\}$. The functions satisfy:

- $E_\theta : \mathcal{Y} \to \mathbb{R}^d$ is the channel embedding NN.
- $F_\theta : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is the check-node NN.
- $G_\theta : \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}^d$ is the bit-node NN.
- $H_\theta : \mathbb{R}^d \to \mathbb{R}$ is the embedding-to-LLR NN.

- Considered in the past: [NW19, DHG18, MLJ+21]
- **In previous work:**
  - Given $\mathcal{D}_{MN} = \{x_i, y_i\}_{i=1}^{MN} \sim P_X^{MN} \otimes W_{Y^{MN}\|X^{MN}}$
  - Estimation of the NPD's parameters + consistency for FSCs.
  - Valid for a fixed input distribution P_{X^N} with the HY scheme.

Neural Polar decoder

Definition(A./Huleihel/Pfister/Permuter 2023)

An NPD is defined by $E_{\theta_E} F_{\theta_F}, G_{\theta_F}, H_{\theta_H}$ with parameters $\theta = \{\theta_E, \theta_F, \theta_G, \theta_H\}$. The functions satisfy:

- $E_\theta : \mathcal{Y} \to \mathbb{R}^d$ is the channel embedding NN.
- $F_\theta : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is the check-node NN.
- $G_\theta : \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}^d$ is the bit-node NN.
- $H_\theta : \mathbb{R}^d \to \mathbb{R}$ is the embedding-to-LLR NN.

- Considered in the past: [NW19, DHG18, MLJ+21]
- **In previous work:**
  - Given $\mathcal{D}_{MN} = \{x_i, y_i\}_{i=1}^{MN} \sim P_X^{MN} \otimes W_{Y^{MN}\|X^{MN}}$
  - Estimation of the NPD's parameters + consistency for FSCs.
  - Valid for a fixed input distribution P_{X^N} with the HY scheme.
- **Advantages:**
  - No need for an explicit channel model
  - Computational complexity doesn't depend on the channel model

Objective and Outline

Objective

- Maximize $\mathrm{I}\left(X^N; Y^N\right)$ over $P_{X^N}^{\psi}$ via NPDs
- Construct codes with optimized code rates

Outline:

- Method:
  - *Fast estimation* of the NPD's parameters for fixed $P_{X^N}^{\psi}$
  - *Improvement* of $P_{X^N}^{\psi}$ for fixed NPD
- Experiments:
  - Code design with optimized NPD and $P_{X^N}^{\psi}$

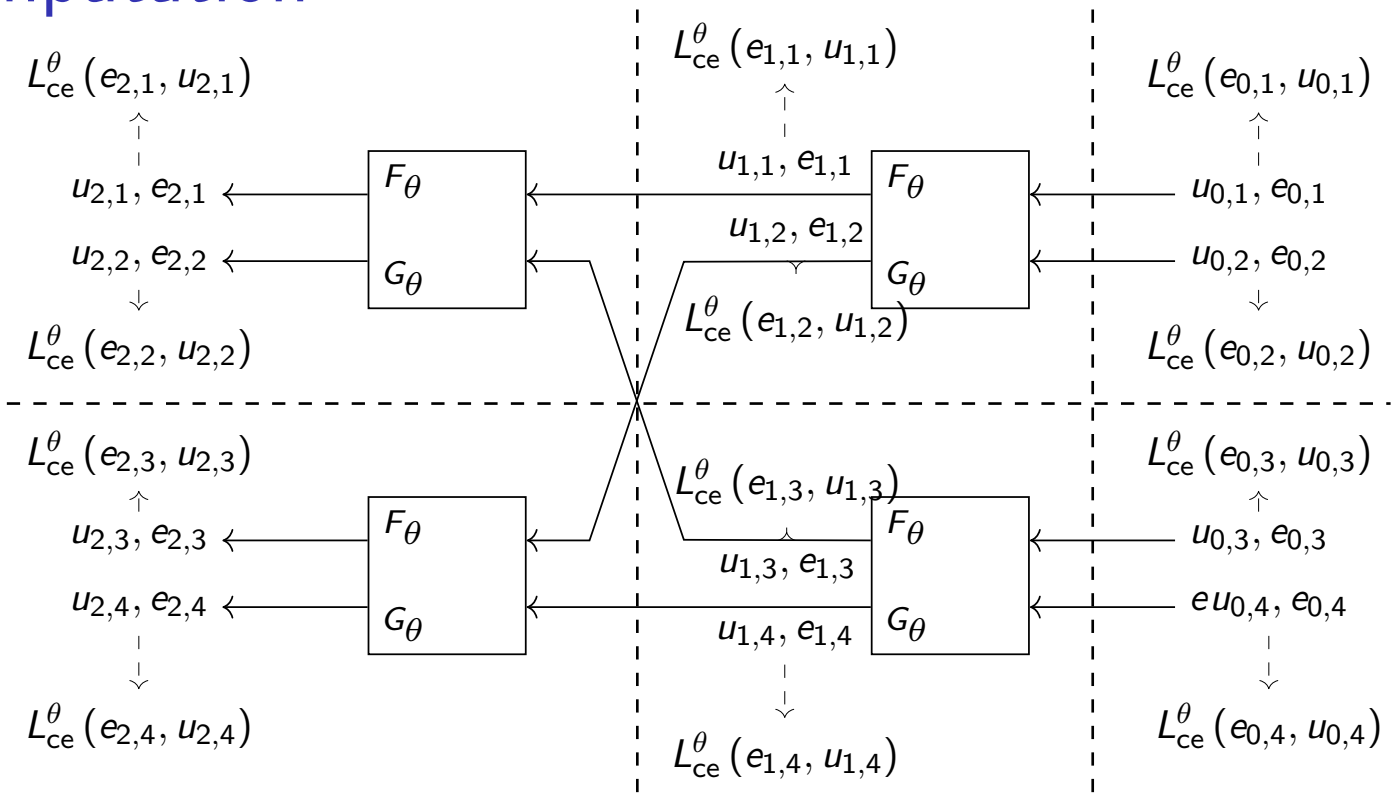
NPD Estimation

Algorithm Data-driven NPD Estimation

input: Dataset $\mathcal{D}^{\psi}_{M,N}$, #of iterations $\mathsf{N}_{\mathsf{iters}}$, learning rate γ
output: Optimized θ^*

Initiate the weights of $E_\theta, F_\theta, G_\theta, H_\theta$
for $\mathsf{N}_{\mathsf{iters}}$ iterations **do**
  Sample $x^N, y^N \sim \mathcal{D}^{\psi}_{M,N}$
  Compute $\mathcal{L}\left(x^N, y^N; \theta\right)$ ▷ In next slide
  Update $\theta := \theta - \gamma \nabla_\theta \mathcal{L}\left(x^N, y^N; \theta\right)$
end for
return θ^*

Loss Computation

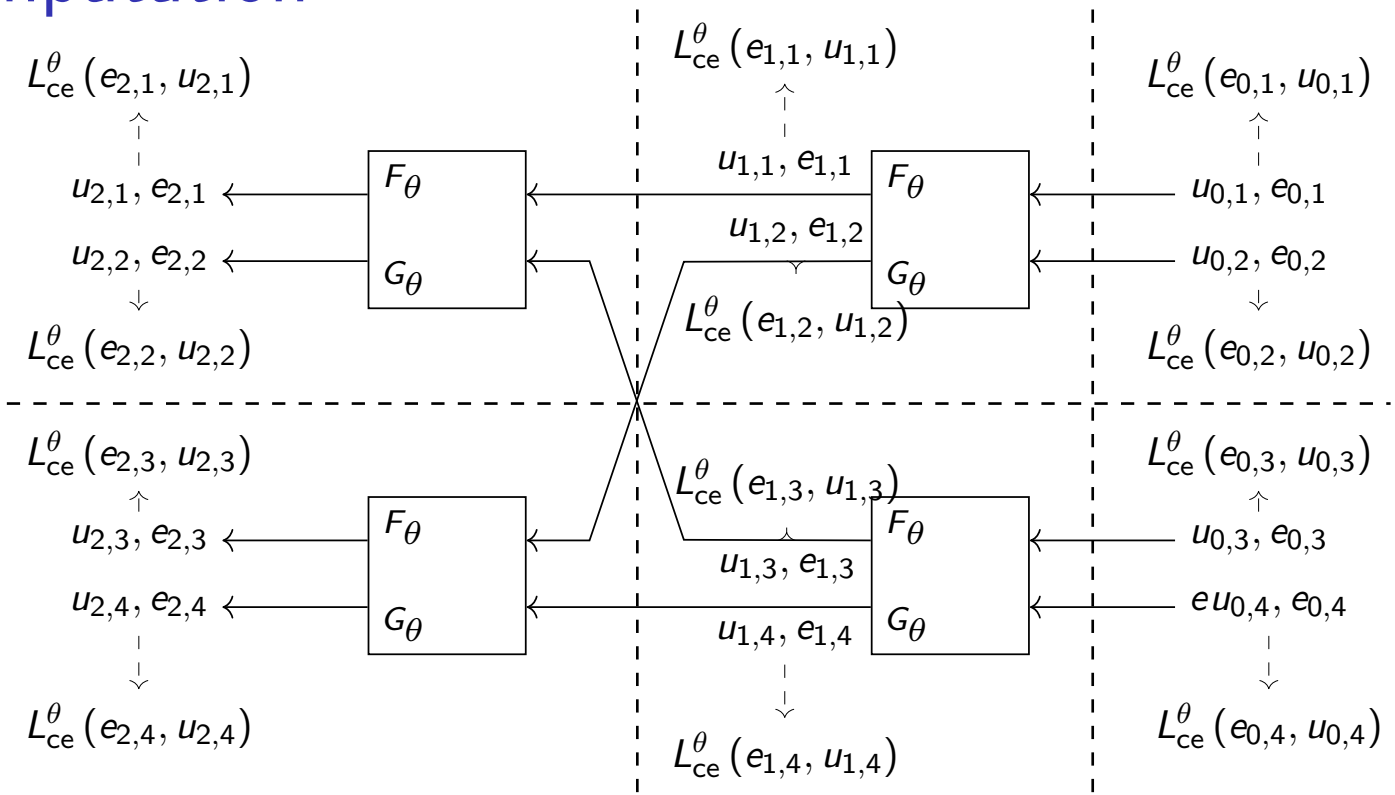


- Cross entropy loss of all loss terms:

$$L^{\theta}_{\text{ce}}(e, u) = -u \log\left(\sigma\left(H_\theta\left(e\right)\right)\right) - (1-u) \log\left(\sigma\left(1 - H_\theta\left(e\right)\right)\right),$$

where H_θ is the embedding-to-LLR NN, $\sigma(x) = \frac{1}{1+e^{-x}}$

Loss Computation

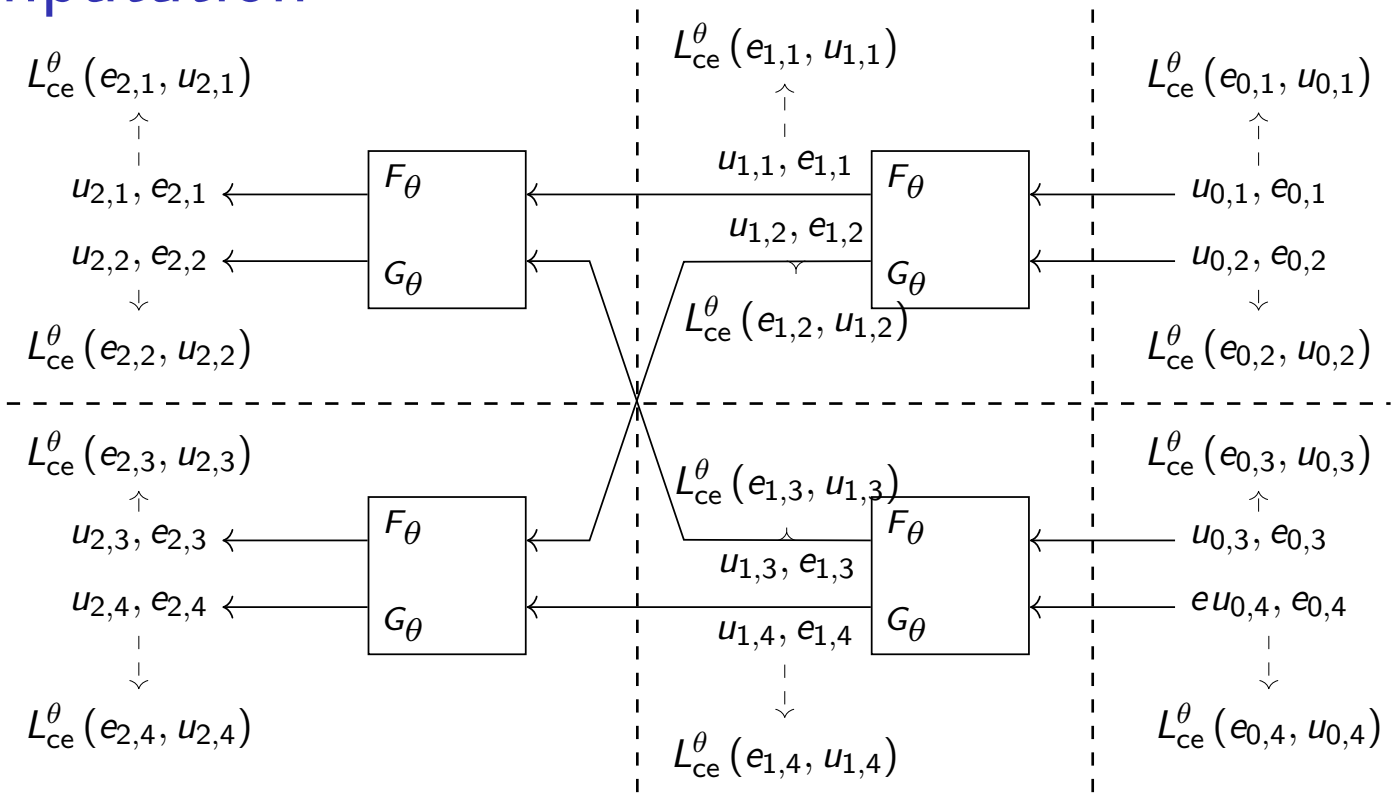


- Cross entropy loss of all loss terms:

$$L^\theta_{\text{ce}}(e, u) = -u \log\left(\sigma\left(H_\theta\left(e\right)\right)\right) - (1-u) \log\left(\sigma\left(1 - H_\theta\left(e\right)\right)\right),$$

  where H_θ is the embedding-to-LLR NN, $\sigma(x) = \frac{1}{1+e^{-x}}$
- Recursion[AHPP23]: $N \log_2(N)$ steps

Loss Computation



- Cross entropy loss of all loss terms:

$$L^\theta_{\text{ce}}(e, u) = -u \log\left(\sigma\left(H_\theta\left(e\right)\right)\right) - (1-u) \log\left(\sigma\left(1 - H_\theta\left(e\right)\right)\right),$$

  where H_θ is the embedding-to-LLR NN, $\sigma(x) = \frac{1}{1+e^{-x}}$
- Recursion[AHPP23]: $N \log_2(N)$ steps
- By stages: $\log_2(N)$ steps (only for training)
- For $N = 2^{10}$: 20X faster (in our implementation)

Estimation of Mutual Information via NPDs

- The mutual information factorizes

$$\begin{aligned} \mathrm{I}\left(X^N; Y^N\right) &= \mathrm{I}\left(U^N; Y^N\right) \\ &= \sum_{i=1}^{N} \mathrm{H}\left(U_i | U^{i-1}\right) - \sum_{i=1}^{N} \mathrm{H}\left(U_i | U^{i-1}, Y^N\right) \end{aligned}$$

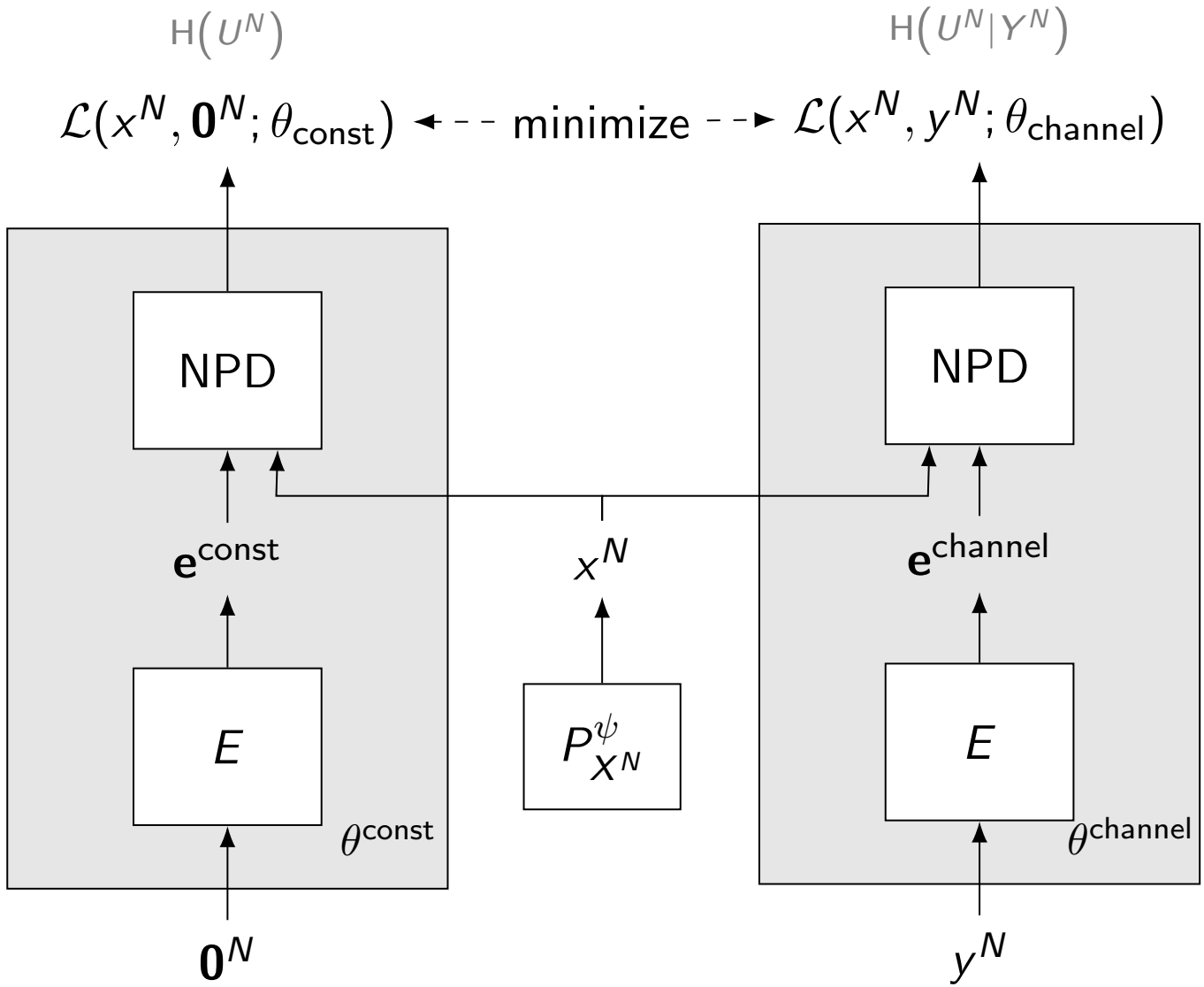
Estimation of Mutual Information via NPDs

- The mutual information factorizes

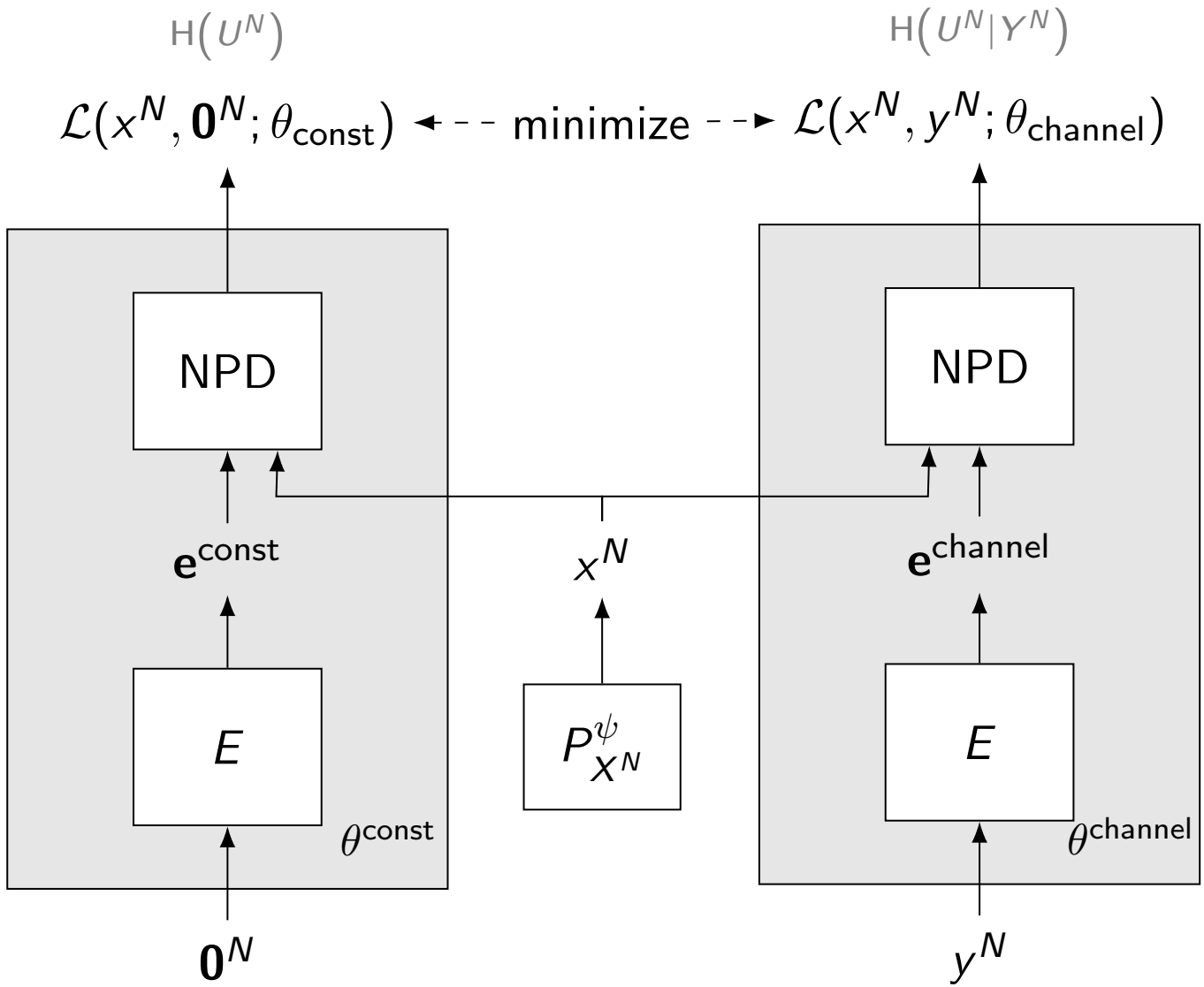
$$\mathrm{I}\left(X^N; Y^N\right) = \mathrm{I}\left(U^N; Y^N\right)$$
$$= \underbrace{\sum_{i=1}^{N} \mathrm{H}\left(U_i | U^{i-1}\right)}_{\theta^{\text{const}}} - \underbrace{\sum_{i=1}^{N} \mathrm{H}\left(U_i | U^{i-1}, Y^N\right)}_{\theta^{\text{channel}}}$$

- The NPD is applied twice with different embeddings:
  1. $\theta^{\text{const}} = \{\mathbf{E}_{\theta_{\mathbf{E}_{\text{const}}}}, F_{\theta_F}, G_{\theta_G}, H_{\theta_H}\} \Rightarrow E_{\theta_{E_{\text{const}}}}(\mathbf{0}^N)$
  2. $\theta^{\text{channel}} = \{\mathbf{E}_{\theta_{\mathbf{E}_{\text{channel}}}}, F_{\theta_F}, G_{\theta_G}, H_{\theta_H}\} \Rightarrow E_{\theta_{E_{\text{channel}}}}(y^N)$
  3. $\theta = \theta^{\text{const}} \cup \theta^{\text{channel}}$

Estimation of mutual information (MI) via NPDs



Estimation of MI via NPDs



$$\widehat{\mathrm{I}}_\theta\left(U_\psi^N; Y^N\right) = \mathcal{L}(x^N, \mathbf{0}^N; \theta_{\text{const}}) - \mathcal{L}(x^N, y^N; \theta_{\text{channel}})$$

Improvement of the Input Distribution

Theorem

The gradient of $\mathsf{I}\left(U_{\psi}^{N}; Y^{N}\right)$ with respect to (w.r.t.) ψ is given by

$$\nabla_{\psi}\mathsf{I}\left(U_{\psi}^{N}; Y^{N}\right) =$$

$$\mathbb{E}_{P_{X^N}^{\psi}\otimes W_{Y^N\|X^N}}\left[\nabla_{\psi}\log P_{X^N}^{\psi}\left(X^{N}\right)\log\frac{P_{U^N|Y^N}\left(X^{N}G_{N}|Y^{N}\right)}{P_{U^N}\left(X^{N}G_{N}\right)}\right]$$

Improvement of the Input Distribution

Theorem

The gradient of $\mathsf{I}\left(U_\psi^N; Y^N\right)$ w.r.t. ψ is given by

$$\nabla_\psi \mathsf{I}\left(U_\psi^N; Y^N\right) = \mathbb{E}_{P_{X^N}^\psi \otimes W_{Y^N \| X^N}}\left[\nabla_\psi \log P_{X^N}^\psi\left(X^N\right) \log \frac{P_{U^N|Y^N}\left(X^N G_N | Y^N\right)}{P_{U^N}\left(X^N G_N\right)}\right]$$

$$\nabla_\psi \widehat{\mathsf{I}}_\theta\left(U_\psi^N; Y^N\right) = \mathbb{E}_{P_{X^N}^\psi \otimes W_{Y^N \| X^N}}\left[\nabla_\psi \log P_{X^N}^\psi\left(X^N\right)\left[\mathcal{L}\left(X^N, \mathbf{0}^N; \theta^{*\text{const}}\right) - \mathcal{L}\left(X^N, Y^N; \theta^{*\text{channel}}\right)\right]\right]$$

Code Rate Optimization Algorithm

for $\mathsf{N}_{\mathsf{iters}}$ iterations **do**

Improvement step: (input distribution)

Sample $x^N, y^N \sim P_{X^N}^{\psi} \otimes W_{Y^N \| X^N}$

Update

$$\psi := \psi + \gamma \nabla_{\psi} \widehat{\mathsf{I}}_{\theta} \left(U_{\psi}^N; Y^N \right)$$

Estimation step: (NPD)

Sample $x^N, y^N \sim P_{X^N}^{\psi} \otimes W_{Y^N \| X^N}$

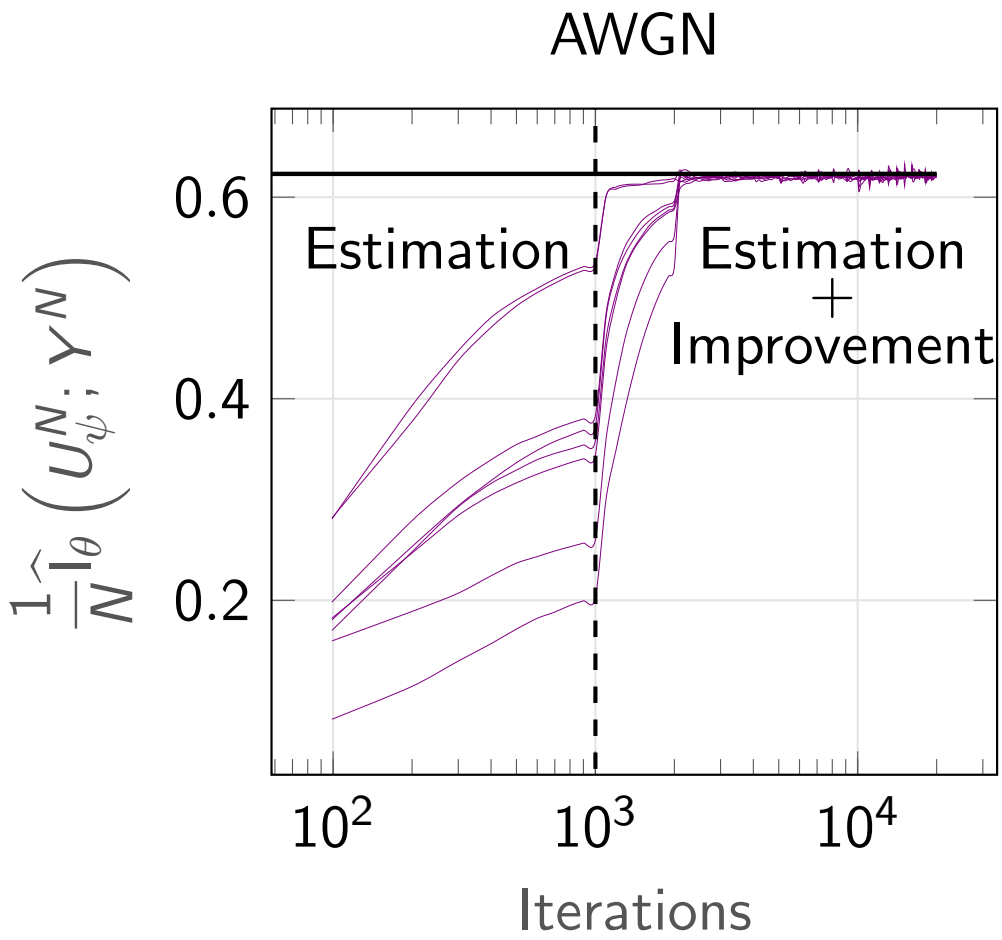
Update

$$\theta := \theta - \gamma \nabla_{\theta} \left[\mathcal{L} \left(x^N, \mathbf{0}^N; \theta^{\mathsf{const}} \right) + \mathcal{L} \left(x^N, y^N; \theta^{\mathsf{channel}} \right) \right]$$

end for

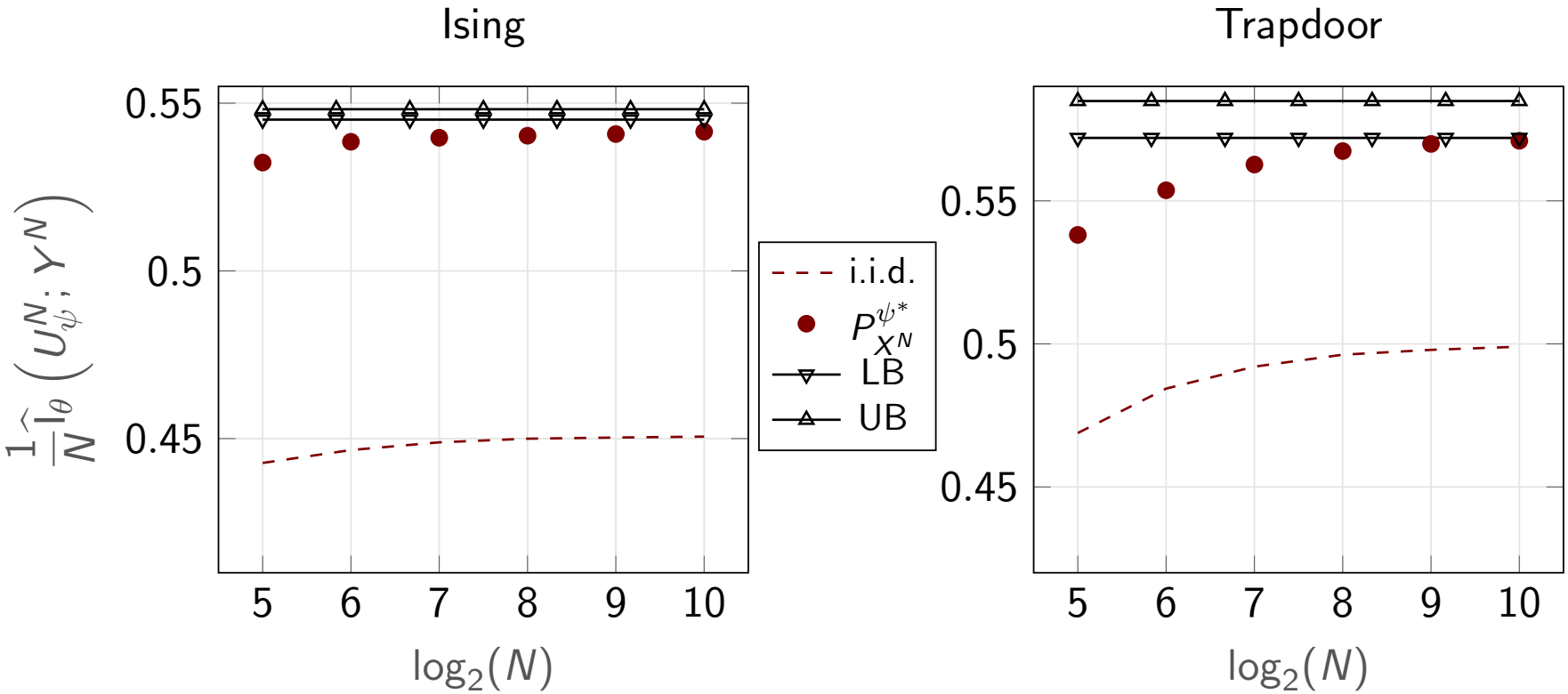
return θ, ψ

Experiments - Additive White Gaussian Noise Channel



- AWGN: $Y = X + N$, $N \sim \mathcal{N}(0, 1)$
- 10 independent runs with $\psi_0 \sim U(0, 1)$

Experiments - FSCs



Ising:

$$Y = \begin{cases} X & \text{w.p. } 0.5 \\ S & \text{w.p. } 0.5 \end{cases}$$

$$S' = X$$

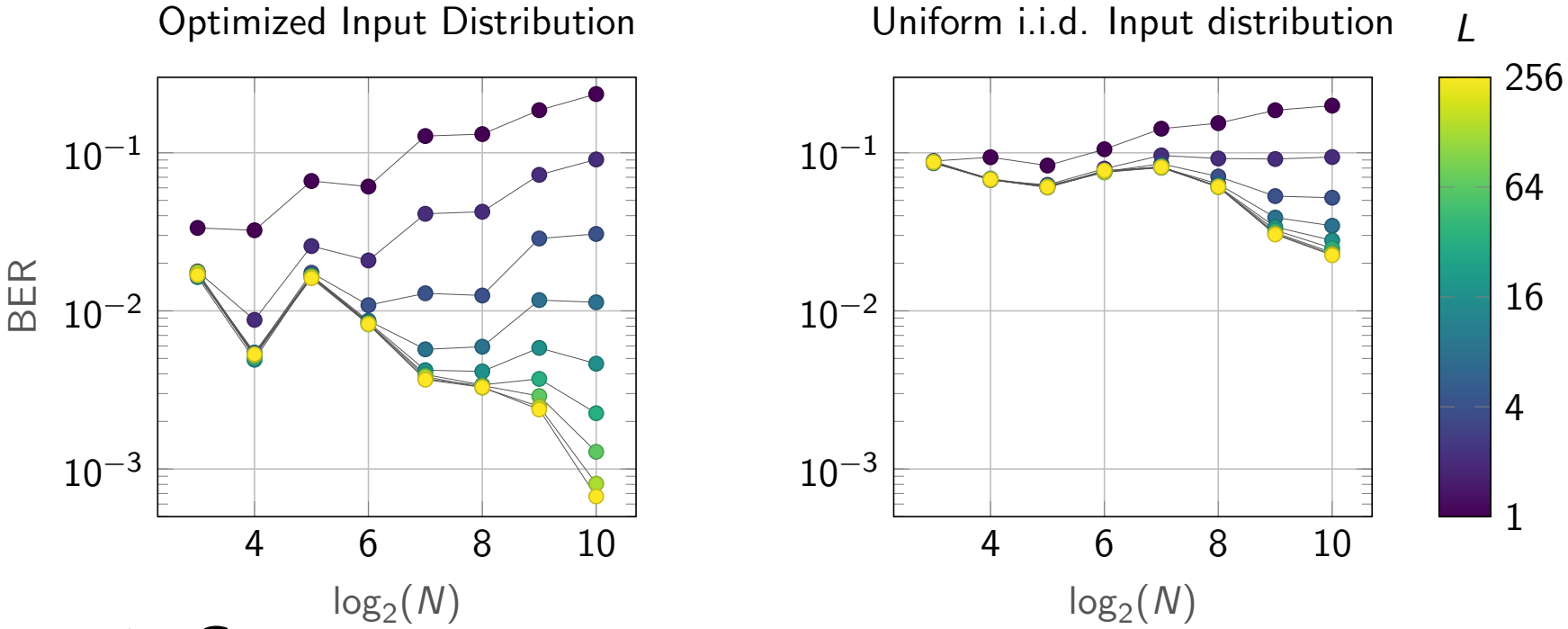
Trapdoor:

$$Y = \begin{cases} X & \text{w.p. } 0.5 \\ S & \text{w.p. } 0.5 \end{cases}$$

$$S' = X \oplus Y \oplus S$$

Bounds from [HSP+21]

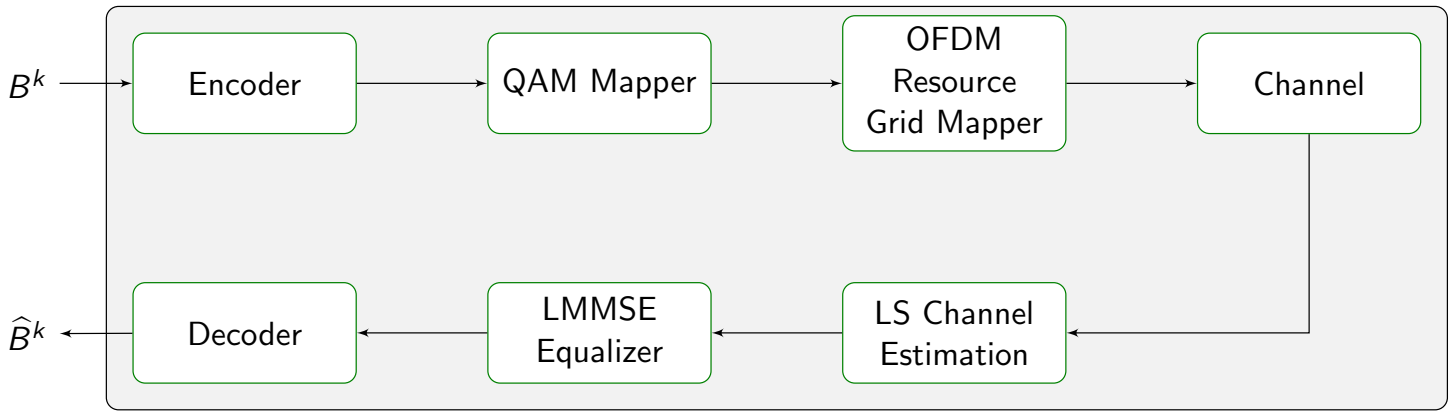
Importance of List Decoding



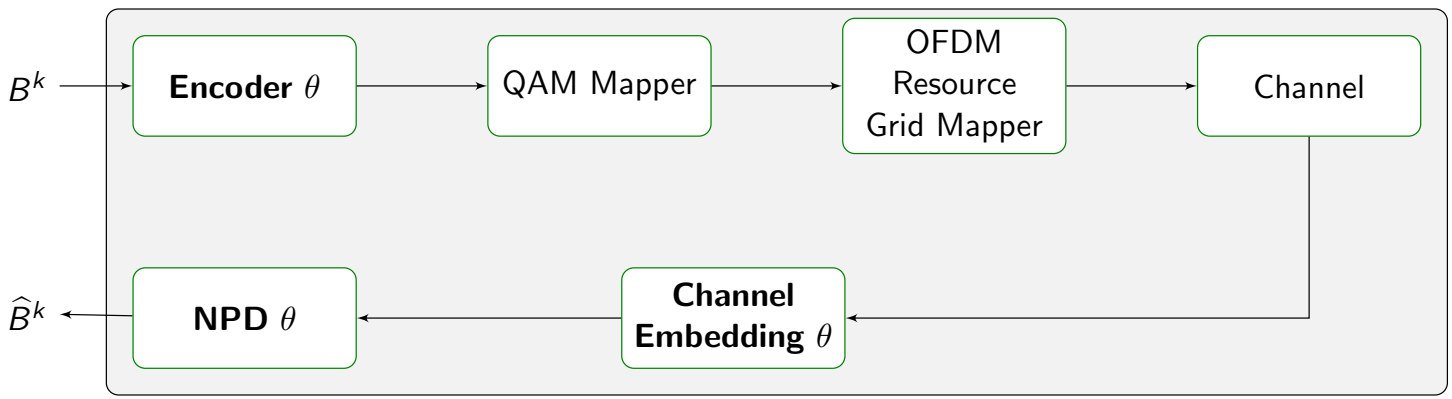
- **Common:**
  - Ising Channel
  - Code design: MC method
  - Encoding and decoding via [HY13]
  - Code rate 0.4 (below the MI for all cases)
  - SCL for $L = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ [TV15]
- **Different:**
  - Left: NPD with optimized ψ (10X faster)
  - Right: SCT with uniform i.i.d. inputs

Experiments - 5G Channel (Preliminary)

Baseline

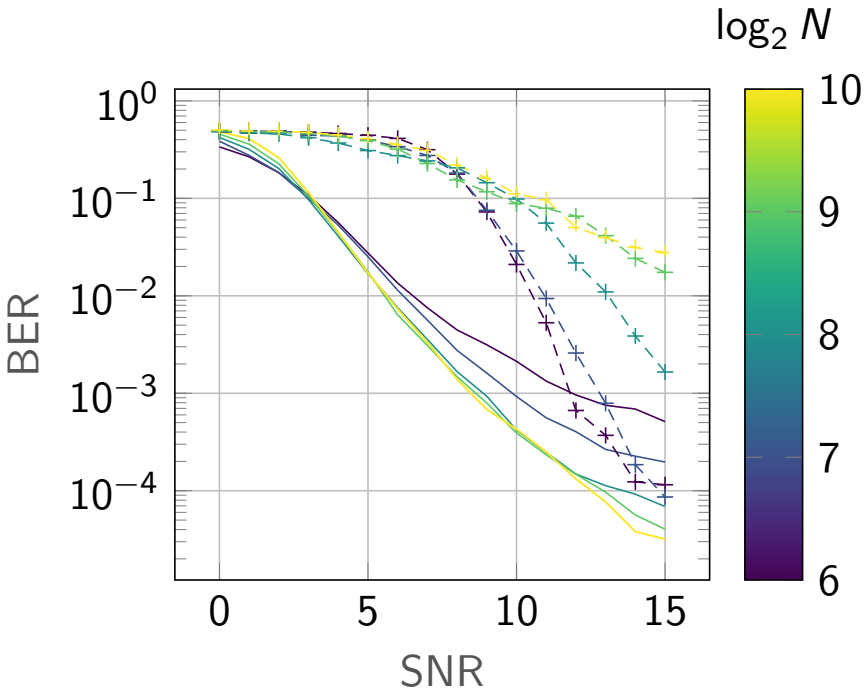


Neural Polar Decoder



- The 5G channel is implemented with Sionna

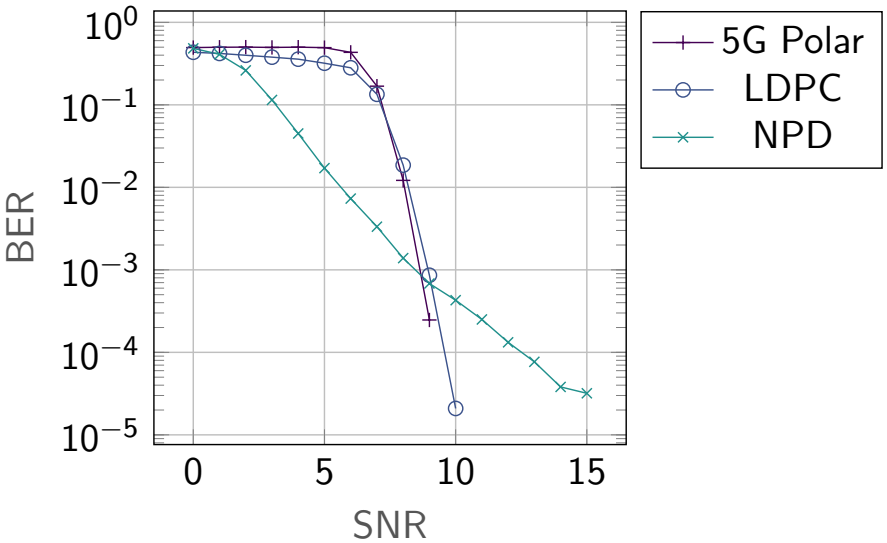
Experiments - 5G Channel (Preliminary)



Parameter	**Value**
Constellation	BPSK
Channel	**TDL–C (300ns)**
Pilots	**Kroneker 25%**
Channel estimation	LS
Equalizer	LMMSE
Carrier frequency	3.5Mhz
Sub-carrier spacing	313KHz
Sub-carriers	64
OFDM symbols	1,16

- Uniform input distribution
- Solid lines: NPD with list decoding
- Dashed with "+" marks baseline SCL (Sionna) [HCA+22]
- Both: Rate 0.3, List size 8,
- Lighter color indicates larger N
- **NPD does not use pilots!** (pilots are the training phase)

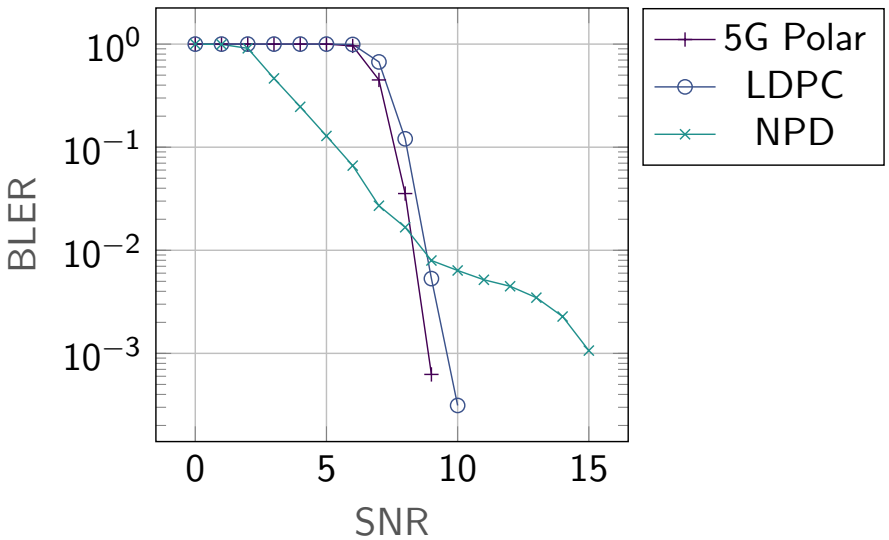
Experiments - 5G Channel (Preliminary)



Parameter	Value
Constellation	BPSK
Channel	**TDL–C (300ns)**
Pilots	**Kroneker 25%**
Channel estimation	LS
Equalizer	LMMSE
Carrier frequency	3.5Mhz
Sub-carrier spacing	313KHz
Sub-carriers	64
OFDM symbols	1,16

- $N = 2^{10}$
- NPD: List size 8. No CRC
- NPD does not use pilots! (pilots are the training phase)
- Polar 5G: List size 8 + CRC

Experiments - 5G Channel (Preliminary)



Parameter	**Value**
Constellation	BPSK
Channel	**TDL–C (300ns)**
Pilots	**Kroneker 25%**
Channel estimation	LS
Equalizer	LMMSE
Carrier frequency	3.5Mhz
Sub-carrier spacing	313KHz
Sub-carriers	64
OFDM symbols	1,16

- $N = 2^{10}$
- NPD: List size 8. No CRC
- NPD does not use pilots! (pilots are the training phase)
- Polar 5G: List size 8 + CRC

Summary

- Neural polar decoders for channels with memory
- Optimization of the input distribution – capacity estimation
- Performance on
  - FSCs (optimized input distribution)
  - 5G channels (uniform i.i.d. input distribution)

Future work

- Investigation on 5G channels:
  - High SNRs (CRC)
  - Larger modulations
  - Optimized input distribution
  - Varying Channel (statistics change)

Summary

- Neural polar decoders for channels with memory
- Optimization of the input distribution – capacity estimation
- Performance on
  - FSCs (optimized input distribution)
  - 5G channels (uniform i.i.d. input distribution)

Future work

- Investigation on 5G channels:
  - High SNRs (CRC)
  - Larger modulations
  - Optimized input distribution
  - Varying Channel (statistics change)

Thank You!

References I

[AHPP23] Ziv Aharoni, Bashar Huleihel, Henry D Pfister, and Haim H Permuter. Data-driven neural polar codes for unknown channels with and without memory. *arXiv preprint arXiv:2309.03148*, 2023.

[Ari09] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inf. Theory*, 55(7):3051–3073, 2009.

[DHG18] N. Doan, S. A. Hashemi, and W. J. Gross. Neural successive cancellation decoding of polar codes. In *2018 IEEE 19th international workshop on signal processing advances in wireless communications (SPAWC)*, pages 1–5. IEEE, 2018.

[HCA+22] Jakob Hoydis, Sebastian Cammerer, Fayçal Ait Aoudia, Avinash Vem, Nikolaus Binder, Guillermo Marcus, and Alexander Keller. Sionna: An open-source library for next-generation physical layer research. *arXiv preprint*, Mar. 2022.

[HSP+21] B. Huleihel, O. Sabag, H. H. Permuter, N. Kashyap, and S. Shamai. Computable upper bounds on the capacity of finite-state channels. *IEEE Trans. Inf. Theory*, 2021.

References II

[HY13] J. Honda and H. Yamamoto. Polar coding without alphabet extension for asymmetric models. *IEEE Trans. Inf. Theory*, 59(12):7829–7838, 2013.

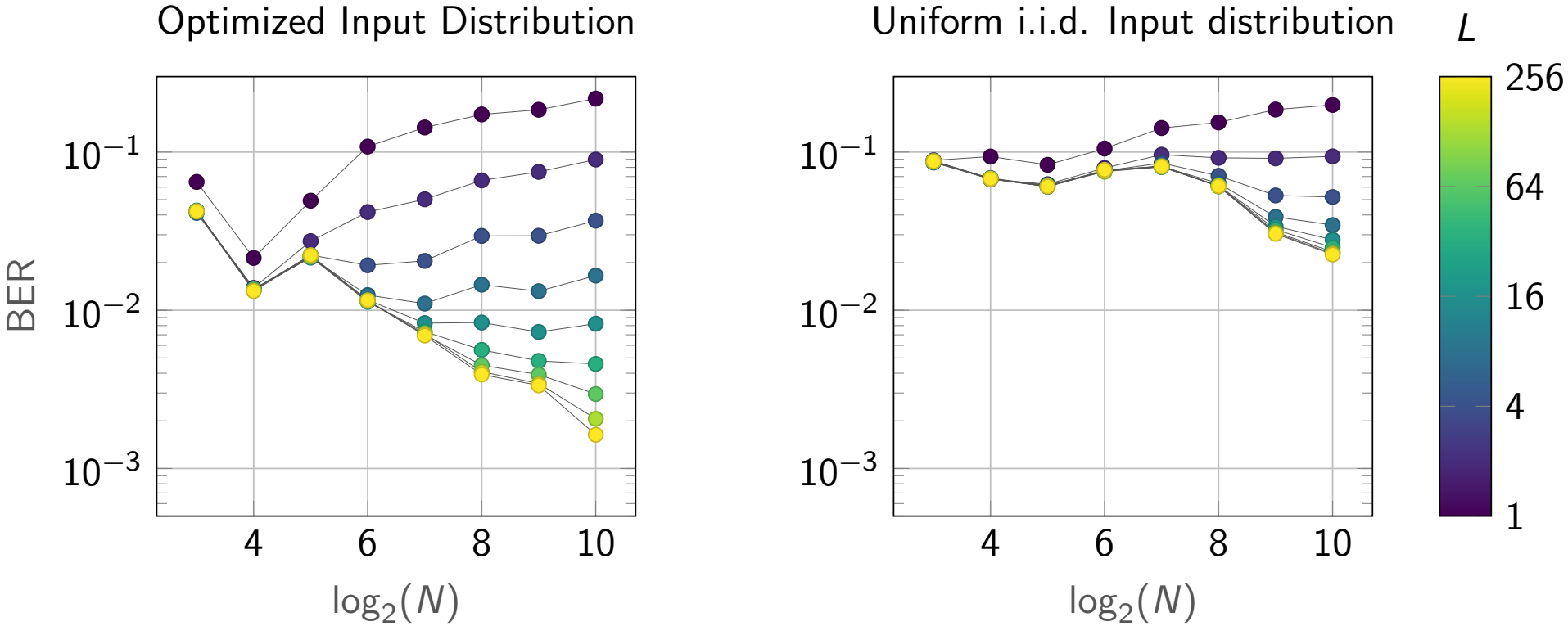
[MLJ+21] Ashok V Makkuva, Xiyang Liu, Mohammad Vahid Jamali, Hessam Mahdavifar, Sewoong Oh, and Pramod Viswanath. KO codes: inventing nonlinear encoding and decoding for reliable wireless communication via deep-learning. In *International Conference on Machine Learning*, pages 7368–7378. PMLR, 2021.

[NW19] E. Nachmani and L. Wolf. Hyper-graph-network decoders for block codes. *Advances in Neural Information Processing Systems*, 32, 2019.

[TV15] I. Tal and A. Vardy. List decoding of polar codes. *IEEE Trans. Inf. Theory*, 61(5):2213–2226, 2015.

[WHY+15] R. Wang, J. Honda, H. Yamamoto, R. Liu, and Y. Hou. Construction of polar codes for channels with memory. In *2015 IEEE Information Theory Workshop-Fall (ITW)*, pages 187–191. IEEE, 2015.

Experiments - Code design for Trapdoor Channel



- **Common:**
  - Code rate 0.4 (below the MI for all cases)
  - SC list decoding for $L = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$
- **Different:**
  - Left: NPD with optimized ψ
  - Right: SCT with uniform i.i.d. inputs