Code Rate Optimization via Neural Polar Decoders

Ziv Aharoni
Ben-Gurion University
zivah@post.bgu.ac.il

Bashar Huleihel
Ben-Gurion University
basharh@post.bgu.ac.il

Henry D. Pfister
Duke University
henry.pfister@duke.edu

Haim H. Permuter
Ben-Gurion University
haimp@bgu.ac.il

***Abstract*—In this work, we explore the enhancement of polar codes for channels with memory, focusing on achieving low decoding complexity and optimizing input distributions for maximum transmission rates. Polar codes are known for their efficient decoding, exhibiting a complexity of $O(N \log N)$ in memoryless channels, and complexity of $O(|\mathcal{S}|^3 N \log N)$ in finite state channels (FSCs), where $|\mathcal{S}|$ is the state space size. A notable recent advancement is the integration of neural networks (NNs) to create an neural polar decoder (NPD), which is adept at learning from data without the knowledge of the channel model, effectively bypassing the cubic complexity growth associated with the channel state size. In this paper, we propose a framework to optimize the input distribution for polar codes, aiming to maximize the mutual information of effective bit channels. This framework has been tested on both memoryless and FSCs, including the additive white Gaussian noise (AWGN) channel and the Ising channel, yielding promising results. The key contribution of this paper is the demonstration of the feasibility of simultaneously selecting an optimal input distribution and creating a practical decoder for various channel types, even in the absence of a channel model. This approach paves the way for new advancements in data-driven communication theory, especially for channels with memory.**

***Index Terms*—Channel capacity, channels with memory, data-driven, polar codes.**

I. Introduction

Polar codes [1] have emerged as a groundbreaking tool in the field of information theory, enabling the construction of capacity-achieving codes with remarkably low decoding complexity. For memoryless channels, polar codes achieve a decoding complexity of $O(N \log N)$, and this extends to $O(|\mathcal{S}|^3 N \log N)$ for finite state channels (FSCs) [2], where $|\mathcal{S}|$ denotes the state size of the channel. This advancement has positioned polar codes as a pivotal component in modern communication systems, offering both efficiency and scalability.

In a recent paper [3], the authors proposed the integration of neural networks (NNs) to further enhance polar codes. By developing an algorithm that inherits the structure of polar codes, they devised an neural polar decoders (NPDs) that can be constructed from data without requiring access to an explicit channel model. A significant advantage of this approach is its non-cubic growth in decoding complexity with respect to the channel state size. However, a critical gap remains in the optimal selection of an input distribution that can maximize the rate transmitted by the polar code. Addressing this gap is essential for the full realization of polar codes' potential in complex communication scenarios.

To bridge this gap, we introduce a framework specifically designed to optimize the involved input distribution. Our proposed algorithm operates by iteratively estimating the mutual information (MI) of the effective bit channels, and simultaneously adjusting the input distribution to maximize this MI. This approach is tested on various channels, including both memoryless channels and FSCs, exemplified thorough the additive white Gaussian noise (AWGN) channel and the Ising channel [4], respectively. The empirical results from these experiments are promising, demonstrating the framework's efficacy across different channel models.

The primary contribution of this work lies in its demonstration that even when a channel is treated as a black box, devoid of an explicit channel model, it is feasible to simultaneously identify an input distribution that maximizes the communication rate and develop a practical decoder tailored to this distribution. This dual capability marks a stride forward in the field of communication theory, suggesting potential implications for how we approach channel encoding and decoding in a data-driven world.

II. Notations and Preliminaries

A. Notation

Throughout this paper, we denote by $(\Omega, \mathcal{F}, \mathbb{P})$ the underlying probability space on which all random variables are defined. Sets by calligraphic letters, e.g. $\mathcal{X}$. We use the notation X^n to denote the random variable (RV) $(X_1, X_2, \dots, X_n)$ and x^n to denote its realization. The probability $\Pr[X = x]$ is denoted by $P_X(x)$. Stochastic processes are denoted by blackboard bold letters, e.g. $\mathbb{X} := (X_i)_{i \in \mathbb{N}}$. An n-coordinate projection of $\mathbb{P}$ is denoted by $P_{X^n Y^n} := \mathbb{P}|_{\sigma(X^n, Y^n)}$, where $\sigma(X^n, Y^n)$ is the σ-algebra generated by (X^n, Y^n). We denote by $[N]$ the set of integers $\{1, \dots, N\}$. The MI between two RVs X, Y is denoted by $\mathsf{I}(X; Y)$ and the binary entropy of X is denoted by $\mathsf{H}(X)$.

The tuple $\left(W_{Y|X}, \mathcal{X}, \mathcal{Y}\right)$ defines a memoryless channel with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$ and a transition kernel $W_{Y|X}$. Throughout the paper we assume that $\mathcal{X} = \{0, 1\}$. The tuple $\left(W_{Y\|X}, \mathcal{X}, \mathcal{Y}\right)$ defines a time invariant channel with memory, where $W_{Y\|X} = \left\{ W_{Y_0|Y_{-i+1}^{-1}, X_{-i+1}^{0}} \right\}_{i \in \mathbb{N}}$. The term

979-8-3503-8284-6/24/$31.00 ©2024 IEEE

$W_{Y^N\|X^N} = \prod_{i=1}^{N} W_{Y_0|Y^{-1}_{-i+1},X^{0}_{-i+1}}$ denotes the probability of observing Y^N causally conditioned on X^N [5]. We denote by $\mathcal{D}_{M,N} = \{x_{j,i}, y_{j,i}\}_{j\in[M],i\in[N]} \sim P_{X^{MN}} \otimes W_{Y^{MN}\|X^{MN}}$ a finite sample of inputs-outputs pairs of M consecutive blocks of N symbols, where $x_{j,i}, y_{j,i}$ denotes the i-th input and output of the j-th block. The term $x_{j,1}^{N}$ denotes $\{x_{j,i}\}_{i=1}^{N}$.

The class of shallow NNs with fixed input and output dimensions is defined as follows [6].

Definition 1 (NN function class)**.** For the ReLU activation function $\sigma_{\mathsf{R}}(x) = \max(x, 0)$ and $d_i, d_o \in \mathbb{N}$, define the class of NNs with $k \in \mathbb{N}$ neurons as:

$$\mathcal{G}_{\mathsf{NN}}^{(d_i,k,d_o)} := \tag{1}$$

$$\left\{ g : \mathbb{R}^{d_i} \to \mathbb{R}^{d_o} : g(x) = \sum_{j=1}^{k} \beta_j \sigma_{\mathsf{R}}(\mathrm{W}_j x + b_j),\ x \in \mathbb{R}^{d_i} \right\}, \tag{2}$$

where σ_{R} acts component-wise, $\beta_j \in \mathbb{R}, \mathrm{W}_j \in \mathbb{R}^{d_o \times d_i}$ and $b_j \in \mathbb{R}^{d_o}$ are the parameters of $g \in \mathcal{G}_{\mathsf{NN}}^{(d_i,k,d_o)}$.

B. Polar codes

Let $G_N = B_N F^{\otimes n}$ be Arikan's polar transform with the generator matrix for block length $N = 2^n$ for $n \in \mathbb{N}$. The matrix B_N is the permutation matrix associated with the bit-reversal permutation. It is defined by the recursive relation $B_N = R_N(I_2 \otimes B_{\frac{N}{2}})$ starting from $B_2 = I_2$. The term I_N denotes the identity matrix of size N and R_N denotes a permutation matrix called reverse-shuffle [1]. The term $A \otimes B$ denotes the Kronecker product of A and B when A, B are matrices, and it denotes a tensor product whenever A, B are distributions. The term $A^{\otimes N} := A \otimes A \otimes \cdots \otimes A$ denotes an application of the $\otimes$ operator N times. The symbol $:=$ denotes an assignment operator.

We define a polar code by the tuple $(\mathcal{X}, \mathcal{Y}, W, E, F, G, H)$ that contains the channel W, the channels embedding E and the core components of the successive cancellation (SC) decoder, F, G, H. We define the effective bit channels by the tuple $\left(W_N^{(i)}, \mathcal{X}, \mathcal{X}^{i-1} \times \mathcal{Y}^N\right)$ for all $i \in [N]$. The term $E : \mathcal{Y} \to \mathcal{E}$ denotes the channel embedding, where $\mathcal{E} \subset \mathbb{R}^d$. The functions $F : \mathcal{E} \times \mathcal{E} \to \mathcal{E},\ G : \mathcal{E} \times \mathcal{E} \times \mathcal{X} \to \mathcal{E}$ denote the check-node and bit-node operations, respectively. We denote by $H : \mathcal{E} \to \mathbb{R}$ a mapping of the embedding into an log likelihood ratio (LLR) value, i.e. a soft decision. For example, for a memoryless channel $W := W_{Y|X}$, a valid choice of E, F, G, H is given by

$$E(y) = \log \frac{W(y|1)}{W(y|0)} + \log \frac{P_X(1)}{P_X(0)}, \tag{3}$$

$$F(e_1, e_2) = 2\tanh^{-1}\left(\tanh\frac{e_1}{2}\tanh\frac{e_2}{2}\right), \tag{4}$$

$$G(e_1, e_2, u) = e_2 + (-1)^u e_1, \tag{5}$$

$$H(e_1) = e_1, \tag{6}$$

where $e_1, e_2 \in \mathbb{R}, u \in \mathcal{X}, y \in \mathcal{Y}$.

Applying SC decoding on the channel outputs yields an estimate of the transmitted bits and their corresponding posterior distribution [1]. In this work, we aim to use the polar coding scheme in order to estimate $\mathsf{I}\left(X^N; Y^N\right)$ and therefore we assume both u^N, y^N are known to the decoder, i.e. all u^N are frozen. Specifically, given y^N, u^N, SC decoding performs the map

$$l\left(y^N, u^i\right) = \log \frac{P_{Y^N,U^{i-1}|U_i}\left(y^N, u^{i-1}|u_i\right)}{P_{Y^N,U^{i-1}|U_i}\left(y^N, u^{i-1}|1-u_i\right)} \tag{7}$$

for $i \in [N]$. From $l\left(y^N, u^i\right)$ the kernels $W_N^{(i)}\left(y^N, u^{i-1}|u_i\right)$, $i \in [N]$, are recovered by the following mapping

$$\sigma\left(l\left(y^N, u^i\right)\right) = \begin{cases} W_N^{(i)}\left(y^N, u^{i-1}|0\right) & u_i = 0 \\ W_N^{(i)}\left(y^N, u^{i-1}|1\right) & u_i = 1 \end{cases}, \tag{8}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ is the logistic function. For more details on SC decoding, the reader may refer to [1, Section VIII].

III. Neural Polar Decoders

This section describes the integration of NNs as a core component in polar decoders. Let U^N, X^N, Y^N be the information bits, channel inputs and channel outputs, respectively. A NPD operates by decoding Y^N into $\widehat{U}^N$ via a SC procedure. The SC procedure systematically decodes the information bits in a sequential manner, utilizing the four core functions E, F, G, and H, that are repeatedly applied throughout the decoding procedure. In an NPD, these four functions are approximated by NNs.

Let E_θ, F_θ, G_θ, H_θ be the NNs components of the NPD, with $E_\theta \in \mathcal{G}_{\mathsf{NN}}^{(1,k,d)}$, $F_\theta \in \mathcal{G}_{\mathsf{NN}}^{(2d,k,d)}$, $G_\theta \in \mathcal{G}_{\mathsf{NN}}^{(2d+1,k,d)}$, and $H_\theta \in \mathcal{G}_{\mathsf{NN}}^{(d,k,1)}$. Here, k is the number of hidden units and d is the dimension of the embedding space $\mathcal{E} \subset \mathbb{R}^d$. The term $\theta = \{\theta_E, \theta_F, \theta_G, \theta_H\}$ denotes the parameter of all the NNs E_{θ_E}, F_{θ_F}, G_{θ_F}, H_{θ_H}; for simplicity, all networks are denoted by the same notation θ. Application of the recursive formulas of SC decoding [1] with E_θ, F_θ, G_θ, H_θ define the NPD. First, the channel outputs are embedded into $\mathbb{R}^d$ by employing

$$e_i^0 = E_\theta\left(y_i\right), \tag{9}$$

where e_i^j denotes the i-th bit at the j-th decoding depth. E.g., e_i^n is the embedding of U_i and e_i^0 is the embedding of X_i. Let $e^j = \left(e_i^j\right)_{i=1}^{N}$. Then, for any $j \in \{0, \ldots, n\}$, $i \in [N]$, the recursive formulas are given by

$$e_{2i-1}^{j+1} = F_\theta\left(e_i^j, e_{i+2^j}^j\right), \tag{10}$$

$$e_{2i}^{j+1} = G_\theta\left(e_i^j, e_{i+2^j}^j, u_{2i-1}^{j+1}\right). \tag{11}$$

The last function H_θ converts an embedding value into an LLR value, as defined by

$$l_i^j = H_\theta\left(e_i^j\right). \tag{12}$$

Similar to (7), we define the mappings of the NPD for $i \in [N]$, as follows

$$l_\theta\left(y^N, u^i\right) = \log \frac{P^\theta_{Y^N, U^{i-1}|U_i}\left(y^N, u^{i-1}|u_i\right)}{P^\theta_{Y^N, U^{i-1}|U_i}\left(y^N, u^{i-1}|1-u_i\right)}, \qquad (13)$$

where θ in the superscript of $P^\theta_{Y^N, U^{i-1}|U_i}$ indicates that the LLR is an estimate based on the NPD's parameters θ.

The parameters of the NPD are determined in a training phase. The goal of the training phase is to tune θ such that the performance of the NPD would match the performance of an optimal polar decoder. E.g., for memoryless channels, the NPD is trained to comply with the "vanilla" SC decoder [1]; for FSCs, the NPD is trained to comply with the successive cancellation trellis (SCT) decoder [2]. The training procedure of the NPD is composed of an iterative process in which samples of channel input-output pairs are used to compute a gradient for updating θ via stochastic gradient descent (SGD) optimization. The training algorithm is given in Algorithm 1, and it is described hereafter.

Let $P^\psi_{X^N}$ be a parametric model of the input distribution with fixed $\psi \in \Psi \subset \mathbb{R}^d$, where Ψ is a compact parameter space. We denote by $\mathcal{D}^\psi_{M,N} =\sim P^\psi_{X^{MN}} \otimes W_{Y^{MN}\|X^{MN}}$ a finite sample of inputs-outputs pairs of M consecutive blocks of N symbols. At every iteration of the algorithm, a block is drawn uniformly from $\mathcal{D}^\psi_{M,N}$. Next, the information bits are computed by $u^N = x^N G_N$ and the channel embeddings are computed by $e^0_i = E_\theta(y_i)$, $i \in [N]$. Equipped with e^0 and u^N, the NPD computes the LLRs of the effective channels $l_\theta\left(y^N, u^i\right)$, $i \in [N]$. These terms are used to compute the optimization objective of the algorithm, the negative-log-loss function, as given by

$$\mathcal{L}\left(y^N, u^N; \theta\right) = \sum_{i=1}^{N} -\log \sigma\left(l_\theta\left(y^N, u^i\right)\right). \qquad (14)$$

Finally, the gradient of the loss is computed and θ is updated via SGD. After the completion of a predetermined number of iterations N_{iter}, the algorithm ends and its output is the trained parameters of the NPD, denoted by θ^*.

Remark 1. Algorithm 1 uses both Y^N and X^N (and therefore also U^N) to tune the parameters θ. This is true only for the training phase, for the exclusive purpose of learning the NPD's parameters. After the training phase, the trained parameters θ^* are used for decoding, as described in [1], [2].

In [7], Algorithm 1 was shown to be consistent. That is, as M approaches infinity, the NPD, defined by the optimized parameters θ^*, recovers the conditional entropies of the effective channels. The consistency of the NPD [7, Theorem 4] is given herein.

Algorithm 1 Data-driven NPD Estimation

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

```
Initiate the weights of $E_\theta, F_\theta, G_\theta, H_\theta$
for $k = 1$ to $\mathsf{N}_{\text{iters}}$ do
    Sample $x^N, y^N \sim \mathcal{D}_{M,N}$
    $u^N = x^N G_N$
    Compute $e^0$ by $e^0_i = E_\theta(y_i)$
    Compute $\mathcal{L}\left(y^N, u^N; \theta\right)$ using (14)
    Update $\theta := \theta - \gamma \nabla_\theta \mathcal{L}\left(y^N, u^N; \theta\right)$
end for
return $\theta^*$
```

Theorem 1. *Let $\mathbb{X}, \mathbb{Y}$ be the inputs and outputs of an indecomposable FSC. Let $\mathcal{D}_{M,N} \sim P_{X^{MN}} \otimes W_{Y^{MN}\|X^{MN}}$, where $N = 2^n$, $M, n \in \mathbb{N}$. Let $u_{j,i} = (x^N_{j,1} G_N)_i$. Then, for every $\varepsilon > 0$ there exists $p \in \mathbb{N}$, compact $\Theta \in \mathbb{R}^p$ and $m \in \mathbb{N}$ such that for $M > m$ and $i \in [N]$, $\mathbb{P} - a.s.$*

$$\left|\mathsf{H}^M_\Theta\left(U_i|U^{i-1}, Y^N\right) - \mathsf{H}\left(U_i|U^{i-1}, Y^N\right)\right| < \varepsilon, \qquad (15)$$

where

$$\mathsf{H}^M_\Theta\left(U_i|U^{i-1}, Y^N\right) = \min_{\theta \in \Theta} \left\{ \frac{1}{M} \sum_{j=1}^{M} -\log \sigma\left(l_\theta\left(y^N, u^i\right)\right) \right\}. \qquad (16)$$

Theorem 1 states that, asymptotically, the NPD recovers the true conditional entropies $\mathsf{H}\left(U_i|U^{i-1}, Y^N\right)$, $i \in [N]$. This implies that the NPD is consistent in recovering true distributions $P_{U_i|Y^N, U^{i-1}}$, $i \in [N]$.

IV. Rate Optimization via Polar Coding

This section addresses the problem of choosing an input distribution $P^\psi_{X^N}$ that maximizes the rate $\mathsf{I}_\psi\left(X^N; Y^N\right)$. From this section onwards, the subscript ψ in the MI emphasizes the dependence of the MI on the specific input distribution parameterized by ψ. The process of determining an input distribution that maximizes $\mathsf{I}_\psi\left(X^N; Y^N\right)$ is composed of two main steps. In the first step, the input distribution is fixed. For a fixed ψ, Algorithm 1 is employed to estimate $\mathsf{I}_\psi\left(X^N; Y^N\right)$. In the second step, the parameters of the NPD are fixed. For a fixed NPD, the gradient of the input distribution ψ is computed. Together, these two steps complement each other and are applied interchangeably to form an alternated maximization procedure. This completes the overview of the rate optimization scheme that is detailed herein.

A. Step 1: MI Estimation

The first step considers a fixed input distribution $P^\psi_{X^N}$, and a time-invariant channel $W_{Y^N\|X^N}$. These distributions define the joint distribution $P_{X^N, Y^N} = P^\psi_{X^N} \otimes W_{Y^N\|X^N}$, and a corresponding MI $\mathsf{I}_\psi\left(X^N; Y^N\right)$. Since $U^N = X^N G_N$ is bijective, it follows that $\mathsf{I}_\psi\left(X^N; Y^N\right) = \mathsf{I}_\psi\left(U^N; Y^N\right)$. Also, by the factorization of the MI as a difference of conditional entropies and the chain rule, we have

$$\mathsf{I}_\psi\left(U^N;Y^N\right)=\sum_{i=1}^{N}\mathsf{H}_\psi\left(U_i|U^{i-1}\right)-\sum_{i=1}^{N}\mathsf{H}_\psi\left(U_i|U^{i-1},Y^N\right). \tag{17}$$

Equation (17) implies that by learning two NPDs the MI $\mathsf{I}_\psi\left(U^N;Y^N\right)$ is estimated. Specifically, in order to estimate the second sum in the right-hand-side (RHS) of (17), Algorithm 1 is applied with $\mathcal{D}^{\psi}_{M,N}\sim P^{\psi}_{X^{MN}}\otimes W_{Y^{MN}\|X^{MN}}$ as input; this is exactly as illustrated in Section III. Formally, employing Algorithm 1 with input $\mathcal{D}^{\psi}_{M,N}$ yields in the parameters θ^*_{XY} such that for some $\varepsilon>0$

$$\left|\frac{1}{M}\sum_{j=1}^{M}\mathcal{L}\left(y^N_{j,1},x^N_{j,1}G_N;\theta^*_{XY}\right)-\mathsf{H}\left(U_i|U^{i-1},Y^N\right)\right|<\varepsilon \tag{18}$$

For the first sum in the RHS in (17), Algorithm 1 is applied with $\widetilde{\mathcal{D}}^{\psi}_{M,N}\sim P^{\psi}_{X^{MN}}\otimes P_{Y^{MN}}$ as inputs, for some distribution on $\mathcal{Y}$, independent with $P^{\psi}_{X^{MN}}$. For simplicity and without loss of generality, we choose $Y_i=0$ with probability 1 for $i\in[N]$. Thus, the dependence on Y^N in the conditional entropy cancels out and $\mathsf{H}_\psi\left(U^N\right)$ is recovered. Formally, employing Algorithm 1 with input $\widetilde{\mathcal{D}}^{\psi}_{M,N}$ yields in the parameters θ^*_X such that for some $\varepsilon>0$

$$\left|\frac{1}{M}\sum_{j=1}^{M}\mathcal{L}\left(0^N,x^N_{j,1}G_N;\theta^*_{X}\right)-\mathsf{H}\left(U_i|U^{i-1}\right)\right|<\varepsilon. \tag{19}$$

Together, the parameters θ^*_{XY} and θ^*_X form the parameters for the estimation $\mathsf{I}_\psi\left(U^N;Y^N\right)$ as given by

$$\widehat{\mathsf{I}}_\psi\left(U^N;Y^N\right)=$$
$$\frac{1}{M}\sum_{j=1}^{M}\mathcal{L}\left(y^N_{j,1},x^N_{j,1}G_N;\theta^*_{XY}\right)-\frac{1}{M}\sum_{j=1}^{M}\mathcal{L}\left(0^N,x^N_{j,1}G_N;\theta^*_{X}\right) \tag{20}$$

B. Step 2: MI maximization

The maximization step involves holding the parameters of the NPDs, θ_{XY},θ_X, fixed and maximize $\widehat{\mathsf{I}}_\psi\left(U^N;Y^N\right)$ with respect to (w.r.t.) ψ. The following theorem presents the gradient w.r.t. ψ.

Theorem 2. *The gradient of* $\mathsf{I}_\psi\left(U^N;Y^N\right)$ *w.r.t.* ψ *is given by*

$$\nabla_\psi\mathsf{I}_\psi\left(U^N;Y^N\right)$$
$$=\mathbb{E}_{P^{\psi}_{X^N}\otimes W_{Y^N\|X^N}}\left[\nabla_\psi\log P^{\psi}_{X^N}\left(X^N\right)Q\left(X^N,Y^N\right)\right], \tag{21}$$

where

$$Q\left(X^N,Y^N\right)=\log\frac{P_{U^N|Y^N}\left(X^NG_N|Y^N\right)}{P_{U^N}\left(X^NG_N\right)}. \tag{22}$$

Theorem 2 states that the gradients of the input distribution parameters ψ are computed via a re-parameterization trick [8]; it computes the gradients of the input distribution through samples drawn from it. Practically, the algorithm exploits the consistency of the NPDs to substitute $\mathcal{L}\left(Y^N,X^NG_N;\theta^*_{XY}\right)-\mathcal{L}\left(0^N,X^NG_N;\theta^*_X\right)$ as plug-in estimator of $\log\frac{P_{U^N|Y^N}}{P_{U^N}}$. The proof of Theorem 2 is omitted due to space limitations.

C. Overall Algorithm

The overall algorithm integrates the two steps outlined in Section IV-A through an alternating maximization procedure. The algorithm initiates with a ”warm-up” phase, during which only step 1 is repeated exclusively. This initial phase focuses on estimating $\mathsf{I}_\psi\left(U^N;Y^N\right)$, as its estimate is subsequently utilized as a proxy for $\log\frac{P_{U^N|Y^N}}{P_{U^N}}$ in (21). Following the warm-up phase, step 1 is repeated for K iterations, succeeded by a single iteration of step 2. The value of K is taken as a hyper-parameter, representing the ratio between iterations on step 1 and step 2. It is tailored to maintain the accuracy of the MI estimation in step 1 before proceeding to the computation of the input distribution gradients. The complete algorithm is detailed in Algorithm 2.

Algorithm 2 Rate optimization via polar coding

input: Channel W, block length N, #of iterations $\mathsf{N}_{\mathsf{iters}}$, #of warm-up iterations $\mathsf{N}_{\mathsf{warm-up}}$, #of iterations per estimation K, learning rate γ

output: Optimized $\theta^*_{XY},\theta^*_X,\psi^*$

```
Initiate the weights of $\theta,\psi$
Generate $\mathcal{D}^{\psi}_{M,N}\sim P^{\psi}_{X^N}\otimes W_{Y^N\|X^N}$
Warm-up step:
    $\theta_{XY}=\mathsf{Alg1}\left(\mathcal{D}^{\psi}_{M,N},N,\mathsf{N}_{\mathsf{warm-up}},\gamma\right)$
    $\theta_{X}=\mathsf{Alg1}\left(\widetilde{\mathcal{D}}^{\psi}_{M,N},N,\mathsf{N}_{\mathsf{warm-up}},\gamma\right)$
for $k=1$ to $\mathsf{N}_{\mathsf{iters}}$ do
    Sample $x^N,y^N\sim\mathcal{D}^{\psi}_{M,N}$
    Maximization step: update $\psi$ according to (21) by
        $\psi:=\psi+\gamma\nabla_\psi\mathsf{I}_\psi\left(U^N;Y^N\right)$
    Generate $\mathcal{D}^{\psi}_{M,N}\sim P^{\psi}_{X^N}\otimes W_{Y^N\|X^N}$
    Estimation step:
        $\theta_{XY}=\mathsf{Alg1}\left(\mathcal{D}^{\psi}_{M,N},N,\mathsf{K},\gamma\right)$
        $\theta_{X}=\mathsf{Alg1}\left(\widetilde{\mathcal{D}}^{\psi}_{M,N},N,\mathsf{K},\gamma\right)$
end for
return $\theta_{XY},\theta_X,\psi$
```

V. Experiments

This section presents the performance of Algorithm 2 for both memoryless channels and FSCs. The AWGN is chosen as an instance of a memoryless channel, while the Ising channel

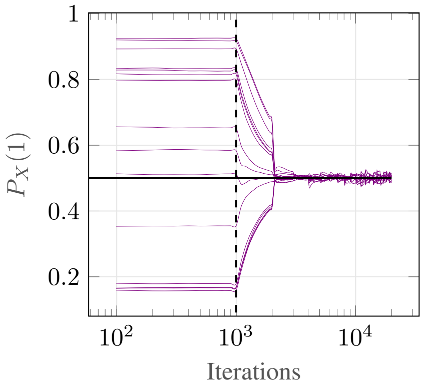

Figure 1: Evolution of $\psi \triangleq P_X(1)$ with respect to the iteration number of Algorithm 2 for the binary-input AWGN channel.

is chosen as an instance of a FSC. The AWGN channel is defined by the relation $Y = X + N$, where X is the channel input, Y is the channel output, and $N \sim \mathcal{N}(0, \sigma^2)$ is an independent and identically distributed (i.i.d.) Gaussian noise. In our experiments we took $\sigma^2 = 0.5$. The Ising channel [4] is defined by $Y = X$ or $Y = S$ with equal probability, and $S' = X$, where X is the channel input, Y is the channel output, S is the channel state at the beginning of the transmission and S' is the channel state at the end of the transmission. Algorithm 2 is applied on both channels where the estimated MI $\widehat{\mathsf{I}}_\psi\left(U^N; Y^N\right)$ is evaluated.

Figures 1 and 2 illustrate the results obtained for the binary-input AWGN channel. In Figure 1, the progression of $\psi \triangleq P_X(1)$ with the iterations of Algorithm 2 is illustrated. The figure present results from 10 independent simulations, each initialized with a randomly chosen value for ψ within the interval $[0, 1]$. Algorithm 2 is executed with $\mathsf{N}_{\text{warmup}} = 1000$, and therefore, during the initial $\mathsf{N}_{\text{warmup}}$ iterations, $P_X(1)$ remains constant. After $\mathsf{N}_{\text{warmup}}$ iterations, the algorithm starts optimizing ψ with $\mathsf{K} = 1$. Notably, after approximately 2000 iterations, the algorithm converges to the optimal value of $P_X(1) = 0.5$. Figure 2 illustrates the evolution of the estimated MI throughout the iterations of the algorithm. Similarly, the estimated MI values converge to the optimal MI for the binary-input AWGN channel.

Figure 3 illustrates the results for the Ising channel [4]. This experiment showcases the estimated value $\widehat{\mathsf{I}}_\psi\left(U^N; Y^N\right)$ obtained upon termination of Algorithm 2 for various block lengths $N = 2^n$. The benchmark for comparison in this experiment is a lower bound on the capacity of the Ising channel of 0.5451, as derived in [9]. As established in [10], this lower bound closely approximates the capacity, with $0.5451 \leq \mathsf{C}_{\text{Ising}} \leq 0.5482$. The results demonstrate that as n increases, $\frac{1}{N}\widehat{\mathsf{I}}_\psi\left(U^N; Y^N\right)$ converges toward the true value of the channel capacity.

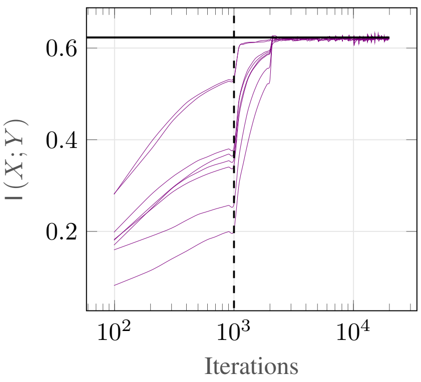

Figure 2: Evolution of $\mathsf{I}_\psi\left(U^N; Y^N\right)$ with respect to the iteration number of Algorithm 2 for the binary-input AWGN channel.

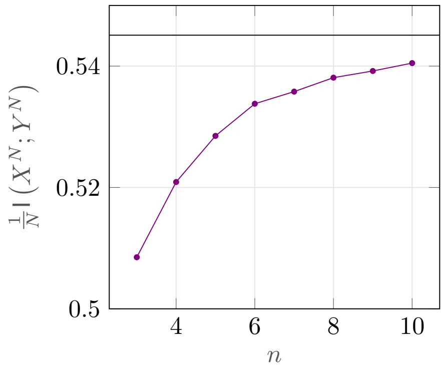

Figure 3: Estimated value of $\frac{1}{N}\mathsf{I}\left(U^N; Y^N\right)$ against the block length $N = 2^n$ for the Ising channel.

VI. Conclusions and Future Work

In this paper, we have addressed two challenges in the field of communication theory: optimizing polar codes for channels with memory and identifying the optimal input distributions for these codes. Our approach utilized NPDs, which notably circumvents the limitations of traditional polar decoders for channels with memory by avoiding cubic complexity growth with respect to the channel state size. By iteratively estimating and maximizing the MI of effective bit channels, we have demonstrated our framework across a variety of channel types, including both memoryless channels and FSCs. The empirical results from the AWGN and Ising channels underscore the efficacy and adaptability of our approach.

Looking ahead, our findings open up exciting avenues for further research, particularly in extending these methods to multi-user communication settings. The tools and methodologies developed in this work have the potential to unravel complexities in multi-user systems, much like they have done for point-to-point channels. The implications of this research offer practical solutions that could be instrumental in shaping the future of communication systems.

References

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

[2] 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)*, IEEE, 2015, pp. 187–191.

[3] Z. Aharoni, B. Huleihel, H. D. Pfister, and H. H. Permuter, "Data-driven polar codes for unknown channels with and without memory," IEEE Int. Symp. Inf. Theory (ISIT), 2023.

[4] T. Berger and F. Bonomi, "Capacity and zero-error capacity of Ising channels," *IEEE Trans. Inf. Theory*, vol. 36, pp. 173–180, 1990.

[5] G. Kramer, *Directed Information for Channels with Feedback*. 1998, vol. 11.

[6] A. M. Fer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *Proceedings of International Conference on Artificial Neural Networks*, Springer, 2006, pp. 632–640.

[7] Z. Aharoni, B. Huleihel, H. D. Pfister, and H. H. Permuter, "Data-driven neural polar codes for unknown channels with and without memory," *arXiv preprint arXiv:2309.03148*, 2023.

[8] J. Schulman, N. Heess, T. Weber, and P. Abbeel, "Gradient estimation using stochastic computation graphs," *Advances in neural information processing systems*, vol. 28, 2015.

[9] A. Sharov and R. Roth, "On the capacity of generalized ising channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2015, pp. 2256–2260.

[10] 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.