Data-Driven Polar Codes for Unknown Channels With and Without Memory

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

2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206663

***Abstract*—In this work, a novel data-driven methodology for designing polar codes is proposed. The methodology is suitable for the case where the channel is given as a "black-box" and the designer has access to the channel for generating observations of its inputs and outputs, but does not have access to the explicit channel model. The methodology consists of two components: (1) a neural estimation of the sufficient statistic of the channel outputs using recent advances in Kullback Leibler (KL) estimation, and (2) a neural successive cancellation (NSC) decoder using three neural networks that replace the core elements of the successive cancellation (SC) decoder. The parameters of the neural networks are determined during a training phase where the mutual information of the effective channels is estimated. We demonstrate the performance of the algorithm on memoryless channels and on finite state channels. Then, we compare the results with the optimal decoding given by the SC and SC trellis decoders, respectively.**

***Index Terms*—Polar codes, data-driven, channels with memory.**

I. Introduction

Polar codes allow the construction of capacity-achieving codes for symmetric binary-input memoryless channels [1]. The main idea is that, when given N independent copies of a binary discrete memoryless channel (DMC) W, the successive cancellation (SC) decoding induces a new set of N binary effective channels $W_N^{(i)}$. Channel polarization is the phenomenon whereby, for N sufficiently large, almost all of the effective bit channels $W_N^{(i)}$ have capacities close to 0 or 1. Specifically, the fraction of channels with capacity close to 1 approaches $I(W)$ and the fraction of channels with capacity close to 0 approaches $1-I(W)$, where $I(W)$ is the channel's symmetric capacity. The construction of polar codes involves choosing which rows to keep from the square generator matrix given by Arikan's transform [1, Section VII]. The encoding and decoding procedures are performed by recursive formulas whose computational complexity is $O(N \log N)$.

Polar codes can also be applied to finite state channels (FSCs) because Arikan's transform also polarizes the bit channels $W_N^{(i)}$ in the presence of memory [2]. The encoding algorithm is essentially the same as if the channel is memoryless. However, the decoding algorithm needs to be updated since the derivation of the successive cancellation (SC) decoder in [1] relies on the fact that the channel is memoryless. To account for the memory, the channel outputs are represented by a trellis, whose nodes capture the information of the channel's memory. This trellis was embedded into the SC decoding algorithm to yield the SC trellis decoding algorithm [3], [4].

However, the SC trellis decoder is only applicable when the channel model is known and when the channel's state alphabet size is finite and relatively small. The computational complexity of the SC trellis decoder is $O(M^3 N \log N)$, where M is the number of channel states. This means that for channels with large memory, the complexity of the decoder might be dominated by the operations dealing with the channel's memory rather than the block length N. For instance, deletion channels have high decoding complexity due to a large channel state space. If the state alphabet is not finite, the algorithm is not applicable without its quantization. Additionally, the algorithm cannot be used for an unknown channel with memory as it requires an explicit channel model.

We propose a novel methodology for a data-driven design of polar codes. The methodology treats the channel as a "black-box" used to generate samples of input-output pairs without access to the channel explicit model. It dissects the polar code design into two separate components. The first extracts a sufficient statistic of the channel outputs via the estimation of the channel's symmetric capacity. The channel's symmetric capacity is estimated using the algorithms in [5]–[7], that provide neural estimation of Kullback Leibler (KL) divergence between two stochastic processes with time dependencies. The main concept is to use the Donsker Vardhan (DV) variational formula of KL divergences [8] and optimize the variational formula over the space of recurrent neural networks (RNNs). The outcome of the algorithm is a RNN whose outputs are the sufficient statistics of Y^N.

The second component uses the sufficient statistic obtained by the first component as an input to a neural SC (NSC) decoder. The NSC uses three neural networks (NNs) that replace the three core elements of the SC decoder: the check-node, the bit-node and the decision operations. The parameters of these NNs are determined in a training phase, in which the mutual information (MI) of the effective channels $W_N^{(i)}$ is estimated. After the training phase, the parameters of the NSC are fixed and the set of "clean" effective channels are determined to complete the code design.

The usage of NNs with for polar codes design were con-

sidered in the past. In [9], NNs were used to decrease the decoding latency by designing a NN decoder that decodes multiple symbols at once. Other instances used NNs to aid existing algorithms, such as the work in [10]–[12]. The paper [13] presents KO codes, a family of deep-learning driven codes Reed-Muller and Polar codes on the additive white Gaussian noise (AWGN) channel. The KO codes are similar to the methods proposed here in the sense that we also leverage the structure of the Arikan's transform to design efficient decoders. However, we do not change Arikan's transform, and we consider channels with memory. To the best of our knowledge, there is no instance of a data-driven polar code design for channels with memory. This work aims to address this gap by developing the necessary algorithms for this task.

The paper is organized as follows. Sec. II defines the notations and gives the necessary background on polar codes. Sec. III-A presents the methodology for data-driven polar code design for memoryless channels. Sec. III-B extends the methodology for the case where the channel has memory. Sec. IV presented the numerical results on the binary symmetric channel (BSC) channel and on the Ising [14] channel.

II. Notations and Preliminaries

Throughout this paper, random variables will be denoted by capital letters and their realizations will be denoted by lower-case letters, e.g. X and x, respectively. Calligraphic letters denote sets, e.g. $\mathcal{X}$. We use the notation X^n to denote the random vector $(X_1, X_2, \ldots, X_n)$ and x^n to denote the realization of such a random vector. The probability $\Pr[X = x]$ is denoted by $P_X(x)$. Stochastic processes are denoted by blackboard bold letters, e.g., $\mathbb{X} := (X_i)_{i\in\mathbb{N}}$. The directed information (DI) between X^n and Y^n is defined as $\mathsf{I}(X^n \to Y^n) = \sum_{i=1}^n I(X^i; Y_i|Y^{i-1})$ [15], while the DI rate is defined as $\mathsf{I}(\mathbb{X} \to \mathbb{Y}) = \lim_{n\to\infty} \frac{1}{n}\mathsf{I}(X^n \to Y^n)$.

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 $W_{Y|X}$. The tuple $\left(W_{Y\|X}, \mathcal{X}, \mathcal{Y}\right)$ defines a time invariant channel with memory, where $W_{Y\|X} = \left\{W_{Y_i|Y^{i-1},X^i}\right\}_{i\in\mathbb{N}}$. The term $W_{Y^N\|X^N} = \prod_{i=1}^N W_{Y_i|Y^{i-1},X^i}$ denotes the probability of observing Y^N causally conditioned by X^N. Throughout the paper we assume that $\mathcal{X} = \{0, 1\}$, and $P_X(x) = 0.5$ for all $x \in \mathcal{X}$. For this choice of P_X and a channel W, the symmetric capacity of the channel is denoted by $I(W)$. We denote by $[N]$ the set $\{1, \ldots, N\}$. 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 notation $P \ll Q$ indicates that the P is absolutely continuous with respect to (w.r.t.) Q.

A. Polar codes

Let G_N be Arikan's generator matrix of block length $N = 2^n$ for $n \in \mathbb{N}$. The term I_N denotes the identity matrix of size N and R_N denotes the reverse shuffle permutation matrix [1]. We define a polar code using the tuple $(\mathcal{X}, \mathcal{Y}, W, L_W, \boxtimes, \circledast, \mathsf{SD})$ that contains the channel W and the core components of the SC decoder. The term $L_W : \mathcal{Y} \to \mathcal{L}$ denotes the channel statistics, where $\mathcal{L} \subseteq \mathbb{R}^d$. For example, for a memoryless channel $W := W_{Y|X}$, a valid choice of L_W, as used in the remainder of this paper, is given by the following:

$$L_W(y) = \log \frac{W(y|1)}{W(y|0)}. \tag{1}$$

The functions $\boxtimes : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$, $\circledast : \mathcal{L} \times \mathcal{L} \times \mathcal{X} \to \mathcal{L}$ denote the check node and bit node operations, respectively. We denote by $\mathsf{SD} : \mathcal{L} \to [0, 1]$ a mapping of the statistics into a probability value, i.e. a soft decision. With this choice, the hard decision rule $h : [0, 1] \to \{0, 1\}$ is the round function $h(l) = \mathbb{I}_{l>0.5}$, where $\mathbb{I}$ is the indicator function.

We denote by $\mathcal{D}_N = \{x_i, y_i\}_{i=1}^N$ a finite sample of inputs-outputs pairs drawn from $P_X^{\otimes N} \otimes W$, where $W = W_{Y|X}^{\otimes N}$ for memoryless channel, and $W = W_{Y\|X}$ for time invariant channel with memory. We denote by

$$\mathcal{A} = \mathsf{SC}_{\mathsf{design}}\left(\mathcal{D}_N, k, L_W, \boxtimes, \circledast, \mathsf{SD}\right)$$

the procedure of finding the set of good channels $\mathcal{A} \subset [N]$ with $|\mathcal{A}| = k$ over the sample $\mathcal{D}_N$ with a SC decoder that uses L_W, $\boxtimes$, $\circledast$, SD as its elementary operations. The dependence of the of $\mathsf{SC}_{\mathsf{design}}$ on $\mathcal{D}_N$ stems that the MI of the effective bit channels is estimated by a Monte-Carlo averaging of polar decoder output to yield and estimate of $H\left(U_i|U^{i-1}, Y^N\right)$.

For memoryless channels with L_W as defined in Eq. (1) we have

$$\begin{aligned}
&\boxtimes(l_1, l_2) = 2\tanh^{-1}\left(\tanh\frac{l_1}{2}\tanh\frac{l_2}{2}\right),\\
&\circledast(l_1, l_2, u) = l_2 + (-1)^u l_1,\\
&\mathsf{SD}(l_1) = \sigma(l_1),
\end{aligned} \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic function and $l_1, l_2 \in \mathcal{L}$. We define the effective bit channels by the tuple $\left(W_N^{(i)}, \mathcal{X}, \mathcal{X}^{i-1} \times \mathcal{Y}^N\right)$.

III. Polar codes for Unknown Channels

This section addresses polar code design for the case where an explicit channel model is not available. For memoryless channels, the problem of constructing data-driven polar codes boils down to estimating the channels statistics via the mutual information neural estimator (MINE) algorithm [16]. This is presented in Sec. III-A. To address channels with memory, we extend the algorithm in Sec. III-A. It involves computing the channels statistics via the directed information neural estimator (DINE) algorithm [5], and devising a SC decoder that is tailored to these statistics. This approach is applicable for both memoryless channels and channels with memory and it is presented in Sec. III-B.

A. Data-Driven Polar codes for Memoryless Channels

Let $W := W_{Y|X}$ be a binary-input memoryless channel and let $\mathcal{D}_N = \{x_i, y_i\}_{i=1}^N \sim (P_X \otimes W)^{\otimes N}$ be a finite sample of its inputs-outputs pairs. The SC decoding algorithm converts the channel statistics $\{L_W(y_i)\}_{i=1}^N$, as defined in (1), into the effective bit channels statistics $\left\{L_{W_N^{(i)}}(y^N, u^{i-1})\right\}_{i=1}^N$ using the recursive formulas given in [1, Prop. 3]. Accordingly, the SC decoder requires explicit channel's statistics; however, in the data-driven scenarios, the transition kernel is unknown.

In order to address this issue, we employ the MINE algorithm [16] to approximate the channel statistics and its highest achievable rate. Given $\mathcal{D}_N$, the MINE algorithm estimates $I(X;Y)$ via the DV variational formula of KL divergences. This yields an estimation of the symmetric capacity by

$$\widehat{I}(X;Y) = \max_{T \in \mathcal{G}_{\mathsf{NN}}} \frac{1}{N} \sum_{i=1}^{N} T(x_i, y_i) - \log \frac{1}{N} \sum_{i=1}^{N} e^{T(x_i, \tilde{y}_i)}, \tag{3}$$

where $\tilde{y}^N$ is a random shuffle of y^N, $\widehat{T}$ denotes the estimated maximizer of the DV formula and $\mathcal{G}_{\mathsf{NN}}$ is the space of NNs. We denote the MINE algorithm with $\widehat{T} = \mathsf{MINE}(\mathcal{D}_N)$.

The optimal solution of the DV formula is given by $T^*(x,y) = \log \frac{W(y|x)}{\frac{1}{2}W(y|0)+\frac{1}{2}W(y|1)} + c$ for $c \in \mathbb{R}$. This equation connects T^* and L_W through the relation

$$L_W(y) = T^*(1,y) - T^*(0,y). \tag{4}$$

Therefore, when the statistics of the channel are not known, the MINE algorithm's output is used as a proxy for $L_W(y)$ via Eq. (4), i.e. $\widehat{L}_W(y) = \widehat{T}(1,y) - \widehat{T}(0,y)$. The estimate of the symmetric capacity $\widehat{I}(X;Y)$ is utilized to determine the number of information bits, k, of the polar code. This process is outlined in Algorithm 1.

Algorithm 1 Data-driven polar code for memoryless channels

input: Dataset $\mathcal{D}_N$, #of info. bits k
output: Clean set $\mathcal{A}$

$\widehat{T} = \mathsf{MINE}(\mathcal{D}_N)$
$\widehat{L}_W = \widehat{T}(1,\cdot) - \widehat{T}(0,\cdot)$
$\mathcal{A} = \mathsf{SC}_{\mathsf{design}}\left(\mathcal{D}_N, k, \widehat{L}_W, \boxtimes, \circledast, \sigma\right)$

Remark 1. The estimate $\widehat{T}$ is a consistent estimator T^* [16]; however, it is estimated with a finite sample which yields an estimation error. Empirically, these errors have a mild effect on the decoding error of the SC decoder. This is in correspondence to other approximations of the optimal decision rule, such as the min-sum-density [17].

B. Data-Driven Polar codes for Channels with Memory

This section presents the channel statistics computation via the DINE algorithm [6], as described in Sec. III-B1. The adapted SC decoder is described in Sec. III-B2.

1) Extracting Sufficient Statistics: This section describes the process of obtaining sufficient statistics from the channel outputs using the DINE algorithm. Let $W := W_{Y\|X}$ be a binary-input channel with memory and let $\mathcal{D}_N = \{x_i, y_i\}_{i=1}^N \sim \left(P_X^{\otimes N} \otimes W_{Y^N\|X^N}\right)$ be a finite sample of its inputs-outputs pairs. The DINE algorithm estimates the DI rate from X^N to Y^N using the following formula:

$$\begin{aligned}
&\widehat{I}(\mathbb{X} \to \mathbb{Y}) \\
&= \max_{T_{XY} \in \mathcal{G}_{\mathsf{RNN}}} \left\{ \frac{1}{N} \sum_{i=1}^{N} T_{XY}\left(x_i, y_i | x^{i-1}, y^{i-1}\right) \right. \\
&\left. - \log \frac{1}{N} \sum_{i=1}^{N} e^{T_{XY}\left(x_i, z_i | x^{i-1}, y^{i-1}\right)} \right\} \\
&- \max_{T_Y \in \mathcal{G}_{\mathsf{RNN}}} \left\{ \frac{1}{N} \sum_{i=1}^{N} T_Y\left(y_i | y^{i-1}\right) - \log \frac{1}{N} \sum_{i=1}^{N} e^{T_Y\left(z_i | y^{i-1}\right)} \right\},
\end{aligned} \tag{5}$$

where $\mathcal{G}_{\mathsf{RNN}}$ is the space of RNNs and $\widehat{T}_{XY}$ and $\widehat{T}_Y$ are the estimated maximizers of the first and second term in Equation 5, respectively. The random variables (RVs) Z^N are independently identically distributed (i.i.d.) auxiliary RVs, uniformly distributed on $\mathcal{Y}$ and independent of X^N, Y^N. They are used for the estimation of the DI as presented in [5]. We denote the DINE algorithm with $\widehat{T}_{XY}, \widehat{T}_Y = \mathsf{DINE}(\mathcal{D}_N)$. The estimation of the DI yields both an estimate of the channel's symmetric capacity and its outputs' sufficient statistics.

The optimal maximizers of the first term in Eq. (5) are given by $T_i^* = \log \frac{P_{Y_i|Y^{i-1},X^i}}{P_Z} + c$ for $c \in \mathbb{R}$ and $i \in \mathbb{N}$. For fixed y^N, we define a new RV $T_{y^i}^* : \mathcal{X}^i \to \mathbb{R}$ by

$$T_{y^i}^*\left(x^i\right) = \log \frac{P_{Y_i|Y^{i-1},X^i}\left(y_i | y^{i-1}, x^i\right)}{P_Z(y_i)}. \tag{6}$$

The following theorem states that $T^N \triangleq \left\{T_{Y^i}^*\right\}_{i=1}^N$ is a sufficient statistic of Y^N for the estimation U^N.

Theorem 1. *Let $X^N, Y^N \sim P_X^{\otimes N} \otimes W_{Y^N\|X^N}$ and P_Z such that $P_Y \ll P_Z$. Then T^N, as defined in Eq.* (6)*, satisfies*

$$U^N - Y^N - T^N, \tag{7}$$

$$U^N - T^N - Y^N. \tag{8}$$

Proof of Theorem 1. The first Markov relation is straightforward as T^N is a function of Y^N and $U^N = X^N G_N$. The second Markov relation is derived by showing that T^N is a sufficient statistic of Y^N for the estimation of X^N, or equivalently, $I\left(X^N; Y^N\right) = I\left(X^N; T^N\right)$. For $x^N, y^N \in \mathcal{X}^N \times \mathcal{Y}^N$, consider the following chain of equalities:

$$\begin{aligned}
P_{X^N,Y^N}\left(x^n, y^n\right) &= \prod_{i=1}^{N} P_{X_i,Y_i|X^{i-1},Y^{i-1}}\left(x_i, y_i | x^{i-1}, y^{i-1}\right) \\
&\overset{(a)}{=} \prod_{i=1}^{N} P_{X_i|X^{i-1}}\left(x_i | x^{i-1}\right) P_Z(y_i) \frac{P_{Y_i|X^i,Y^{i-1}}\left(y_i | x^i, y^{i-1}\right)}{P_Z(y_i)}
\end{aligned}$$

$$\stackrel{(b)}{=} \exp\left(\log P_Z^{\otimes N}\left(y^N\right)\right) \exp\left(\sum_{i=1}^{N} \log P_{X_i|X^{i-1}}\left(x_i|x^{i-1}\right)\right)$$
$$\exp\left(\sum_{i=1}^{N} \log \frac{P_{Y_i|X^i,Y^{i-1}}\left(y_i|x^i,y^{i-1}\right)}{P_Z\left(y_i\right)}\right),$$

where (a) follows from the chain rule, the absence of outputs feedback, and $P_{Y_i|X^i,Y^{i-1}} \ll P_Z$; and (b) is a result of rearranging the terms into exponents. Next, we identify that $P_{X^N,Y^N}\left(x^n,y^n\right) = h\left(y^N\right) g\left(t^N\left(y^N\right),x^N\right)$, where

$$h\left(y^N\right) \triangleq \exp\left(\log P_Z^{\otimes N}\left(y^N\right)\right),$$
$$g\left(t^N,x^N\right) \triangleq \exp\left(\sum_{i=1}^{N} \log P_{X_i|X^{i-1}}\left(x_i|x^{i-1}\right) + t_{y^i}\left(x^i\right)\right).$$

This is exactly the factorization in the well-known Fisher–Neyman factorization theorem [18], [19], and thus T^N is a sufficient statistic of Y^N for the estimation of X^N. Since $U^N = X^N G_N$ is bijective, we conclude the theorem. □

The following remark indicates the relationship between T^N and the sufficient statistics of a memoryless channel W.

Remark 2 (Relation to SC decoder)**.** For the case where $W := W_{Y|X}$ it follows that $W_{Y^N\|X^N} = W_{Y|X}^{\otimes N}$. Accordingly, the sufficient statistics satisfy $T_{y^i}^*\left(x^i\right) = \log \frac{P_{Y|X}(y_i|x_i)}{P_Z(y_i)}$ that is connected to L_W through Eq. (4).

Theorem 1 suggests that DI estimation is an appropriate objective for the construction of the sufficient statistics of Y^N needed for the SC decoder. However, the evaluation of T^N for all $x^N \in \mathcal{X}^N$ involves an exponential number of computations. To overcome this, recall that according to Eq. (5), $\widehat{T}_{XY}$ is approximated by a RNN that contains a sequence of layers. We design $\widehat{T}_{XY}$ to process y^N and x^N separately before combining them into the output of $\widehat{T}_{XY}$. We denote this construction by $\widehat{T}_{XY}\left(x^i,y^i\right) = \widetilde{T}_{XY}\left(x^i,e^i\right)$, where $e_i = E\left(y_i\right)$, $E : \mathcal{Y} \to \mathbb{R}^d$ is an embedding of Y_i. Since $\widehat{T}_{XY}$ is composed of sequential layers, any intermediate layer of $\widehat{T}_{XY}$ must preserve the information that flows to its outputs. Therefore, we choose E^N as the sufficient statistics required for the SC decoder. The next section describes the algorithm for the design of a SC decoder that is tailored for E^N.

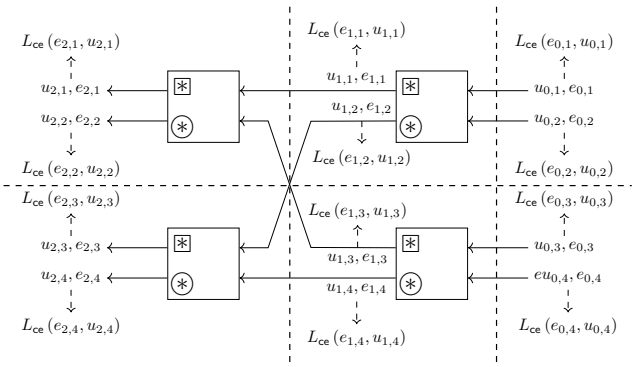

Figure 1: A depiction of Algorithm 2 for $N = 4$. $L_{\mathsf{ce}}(e,u)$ denotes a cross-entropy loss, and the overall training loss L is calculated as the sum of all losses shown in the figure.

2) Neural Successive Cancellation Decoder: This section describes the construction of a NSC decoder that is tailored for E^N. The NSC uses the same structure of the original SC decoder, except its elementary operations are replaced with NNs. Specifically, instead of using $\boxtimes, \circledast, \mathsf{SD}(\cdot)$ as defined in Eq. (2), we use $\boxtimes_{\mathsf{NN}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, $\circledast_{\mathsf{NN}} : \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}^d$ and $\mathsf{SD}_{\mathsf{NN}} : \mathbb{R}^d \to \{0,1\}$, all are parameterized with NNs.

Training the NSC amounts into optimizing the parameters of $\circledast_{\mathsf{NN}}, \boxtimes_{\mathsf{NN}}$ and $\mathsf{SD}_{\mathsf{NN}}$ such that the symmetric capacities of $I\left(W_N^{(i)}\right)$ are computed. It follows that

$$I\left(W_N^{(i)}\right) = 1 - H\left(U_i|U^{i-1},Y^N\right), \tag{9}$$

where $H\left(U_i|U^{i-1}\right) = 1$ since $U_i \overset{iid}{\sim} \mathsf{Ber}\left(0.5\right)$. Hence, choos-

Algorithm 2 $\mathsf{NSCTrain}(\mathbf{e},\mathbf{u},\mathsf{L})$

$N = \mathsf{dim}\left(\mathbf{u}\right)$
if $N = 1$ **then**
  $\mathsf{L} = \mathsf{L} + L_{\mathsf{ce}}(e_1,u_1)$
  return $\mathsf{L},\mathbf{u}$
end if
Split $\mathbf{e}$ into even and odd indices $\mathbf{e}_e,\mathbf{e}_o$
$\mathbf{e}_{\mathsf{C}} = \boxtimes_{\mathsf{NN}}\left(\mathbf{e}_e,\mathbf{e}_o\right)$
$\mathsf{L},\mathbf{v}_1 = \mathsf{NSCTrain}\left(\mathbf{e}_{\mathsf{C}},\mathbf{u}_1^{N/2},\mathsf{L}\right)$
$\mathbf{e}_{\mathsf{B}} = \circledast_{\mathsf{NN}}\left(\mathbf{e}_e,\mathbf{e}_o,\mathbf{v}_1\right)$
$\mathsf{L},\mathbf{v}_2 = \mathsf{NSCTrain}\left(\mathbf{e}_{\mathsf{B}},\mathbf{u}_{N/2+1}^{N},\mathsf{L}\right)$
$\mathbf{v} = [\mathbf{v}_1,\mathbf{v}_2]$
$\mathbf{v} = \mathbf{v}\left(I_{N/2} \otimes G_2\right) R_N$
$\mathsf{L} = \mathsf{L} + \sum_{i=1}^{N} L_{\mathsf{ce}}(e_i,v_i)$
return $\mathsf{L},\mathbf{v}$

Algorithm 3 Data-driven polar code design for channels with memory

input: Channel $W_{Y\|X}$, blocklength n_{t}, #of info. bits k
output: Clean set $\mathcal{A}$

Compute E by applying $\mathsf{DINE}\left(\mathcal{D}_N\right)$
Initiate the weights of $\circledast_{\mathsf{NN}}, \boxtimes_{\mathsf{NN}}, \mathsf{SD}_{\mathsf{NN}}$
$N = 2^{n_{\mathsf{t}}}$
for $l = 1$ to $\mathsf{N}_{\mathsf{iters}}$ **do**
  Sample x^N,y^N
  $u^N = x^N G_N$
  Compute e^N by $e_i = E\left(y_i\right)$
  Compute L by applying $\mathsf{NSCTrain}\left(e^N,u^N,0\right)$
  Minimize L w.r.t. $\circledast_{\mathsf{NN}}, \boxtimes_{\mathsf{NN}}, \mathsf{SD}_{\mathsf{NN}}$
end for
$\mathcal{A} = \mathsf{SC}_{\mathsf{design}}\left(\mathcal{D}_N,k,E,\circledast_{\mathsf{NN}},\boxtimes_{\mathsf{NN}},\mathsf{SD}_{\mathsf{NN}}\right)$

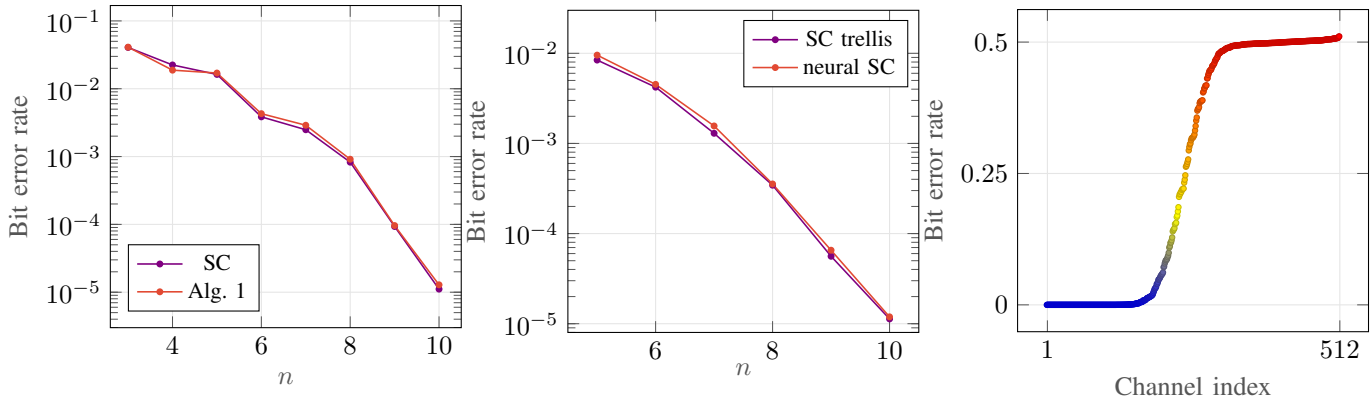

Figure 2: The left figure compares the BERs incurred by Algorithm 1 and the SC decoder on a BSC. The middle figure compares the BERs incurred by Algorithm 3 on the Ising channel; its polarization is illustrated in the right figure.

ing the i-s with the highest value of $I\left(W_N^{(i)}\right)$ is equivalent to choosing the i-s with lowest value of $H\left(U_i|U^{i-1},Y^N\right)$. Hence, we set the goal of estimating $P_{U_i|U^{i-1},Y^N}$ as the goal needed to identify the clean effective bit channels. Accordingly, the cross-entropy loss is used as the loss function for training $\circledast_{\mathsf{NN}}, \boxtimes_{\mathsf{NN}}$ and $\mathsf{SD}_{\mathsf{NN}}$.

In the training phase, the optimization procedure admits the following steps. As an input, the NSC observes the channel's outputs y^N, and the channel's inputs x^N. First, the sufficient statistics are computed by $e^N = E\left(y^N\right)$. We denote by e_i^N the embedding vector at the i-th depth of the decoding recursion and by $e_{i,j}$ the j-th bit at i-th depth. Further, the channel inputs are used to compute the labels by $u^N = x^N G_N$.

In the next step, we initiate the training loss to be $\mathsf{L} = 0$. Then, the NSC decoder starts the recursive computation of the effective bit channels in which the loss is accumulated until the recursion ends. The first loss term is accumulated when the recursion reaches the first effective bit channel. At this point, a loss term $L_{\mathsf{ce}}(e_{\log N,1}, u_1)$ is computed via $L_{\mathsf{ce}}(e,u) = -u\log\left(\mathsf{SD}_{\mathsf{NN}}\left(e\right)\right) - (1-u)\log\left(1-\mathsf{SD}_{\mathsf{NN}}\left(e\right)\right)$. Then, at each leaf of the recursion, such loss term is computed and added to the entire training loss L. That is, each time reaching a leaf, L is updated according to the following rule $\mathsf{L} = \mathsf{L} + L_{\mathsf{ce}}(e_{\log N,i}, u_i),\ \ i \in [N]$. In addition, we make the algorithm more robust by accumulating the loss incurred by bits with intermediate recurrence depth of $0, 1, \ldots, \log N - 1$. The loss L is minimized using stochastic gradient descent.

This procedure is valid for any value of $N = 2^n$, for $n \in \mathbb{N}$. We denote by $2^{n_{\mathsf{t}}}$ the block length that is used in the training phase. This procedure is illustrated in Fig. 1 and described in Algorithm 2. The complete algorithm is given in Algorithm 3.

IV. Experiments

This section presents the experiments on memoryless channels and channels with memory. For the case of memoryless channels, using Algorithm 1, we design a data-driven polar code for the BSC with parameter $p = 0.1$ and compare its bit error rate (BER) with the "vanilla" SC decoder. We choose the code rate to be $R = 0.3$, and accordingly, $|\mathcal{A}| = \lfloor RN \rfloor$, where $\lfloor\cdot\rfloor$ is the floor function. The results are in Fig. 2.

For channels with memory, we design a data-driven polar code for the Ising channel and compare its BER with the SC trellis decoder. We apply Algorithm 3 with a code rate of $R = 0.3$, and training block length $n_{\mathsf{t}} = 8$. The NNs, $\circledast_{\mathsf{NN}}, \boxtimes_{\mathsf{NN}}$ and $\mathsf{SD}_{\mathsf{NN}}$, are parameterized with 3 layers of 50 neurons each. Fig. 2, illustrates the decoding BER of Algorithm 3 in comparison to the SC trellis decoder [4] on the Ising channel, and the incurred BER of the effective bit channels $W_N^{(i)}$ for the Ising channel for $N = 512$.

V. Conclusions and Future Work

This paper presents a data-driven algorithm for the design of polar codes. It demonstrated that estimating the symmetric capacity of the channel via the MINE or the DINE, for memoryless channels or channels with memory, respectively, yields sufficient statistics for the SC decoder. For channels with memory, we presented an algorithm for the design of an adapted SC decoder for the sufficient statistics extracted by the DINE, called NSC. We demonstrated our approach on the BSC, as an instance of a memoryless channel, and on the Ising channel, an instance of a channel with memory.

Our next steps would be to extend our methodology, e.g. for list decoding with cyclic redundancy check [20]. We also plan to extend our result to design capacity achieving codes via the Honda and Yamamoto scheme [21].

VI. Acknowledgements

The work of Henry Pfister was supported in part by NSF Grant #2212437.

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] E. Şaşoğlu and I. Tal, "Polar coding for processes with memory," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 1994–2003, 2019.

[3] R. Wang, R. Liu, and Y. Hou, "Joint successive cancellation decoding of polar codes over intersymbol interference channels," *arXiv preprint arXiv:1404.3001*, 2014.

[4] 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.

[5] D. Tsur, Z. Aharoni, Z. Goldfeld, and H. H. Permuter, "Neural estimation and optimization of directed information over continuous spaces," *submitted to IEEE Trans. Inf. Theory*,

[6] D. Tsur and Z. Aharoni and Z. Goldfeld and H. H. Permuter, "Optimizing estimated directed information over discrete alphabets," in *2022 IEEE Int. Symp. Inf. Theory (ISIT)*, IEEE, 2022, pp. 2898–2903.

[7] Z. Aharoni, D. Tsur, and H. H. Permuter, "Density estimation of processes with memory via donsker vardhan," in *2022 IEEE Int. Symp. Inf. Theory (ISIT)*, IEEE, 2022, pp. 330–335.

[8] M. D. Donsker and S. S. Varadhan, "Asymptotic evaluation of certain Markov process expectations for large time. iv," *Communications on Pure and Applied Mathematics*, vol. 36, no. 2, pp. 183–212, 1983.

[9] 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)*, IEEE, 2018, pp. 1–5.

[10] W. Xu, X. Tan, Y. Be'ery, *et al.*, "Deep learning-aided belief propagation decoder for polar codes," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 2, pp. 189–203, 2020.

[11] M. Ebada, S. Cammerer, A. Elkelesh, and S. ten Brink, "Deep learning-based polar code design," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2019, pp. 177–183.

[12] W. Xu, Z. Wu, Y. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *2017 IEEE International workshop on signal processing systems (SiPS)*, IEEE, 2017, pp. 1–6.

[13] A. V. Makkuva, X. Liu, M. V. Jamali, H. Mahdavifar, S. Oh, and P. Viswanath, "Ko codes: Inventing nonlinear encoding and decoding for reliable wireless communication via deep-learning," in *International Conference on Machine Learning*, PMLR, 2021, pp. 7368–7378.

[14] E. Ising, "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik*, vol. 31, no. 1, pp. 253–258, 1925.

[15] J. Massey, "Causality, feedback and directed information," *Proc. Int. Symp. Inf. Theory Applic. (ISITA-90)*, pp. 303–305, 1990.

[16] M. I. Belghazi, A. Baratin, S. Rajeswar, *et al.*, "MINE: Mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[17] D. Kern, S. Vorköper, and V. Kühn, "A new code construction for polar codes using min-sum density," in *2014 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, IEEE, 2014, pp. 228–232.

[18] R. A. Fisher, "The logic of inductive inference," *Journal of the royal statistical society*, vol. 98, no. 1, pp. 39–82, 1935.

[19] J. Neyman and K. Iwaszkiewicz, "Statistical problems in agricultural experimentation," *Supplement to the Journal of the Royal Statistical Society*, vol. 2, no. 2, pp. 107–180, 1935.

[20] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[21] J. Honda and H. Yamamoto, "Polar coding without alphabet extension for asymmetric models," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7829–7838, 2013.