Data-Driven List Polar Decoder for Symmetric and Asymmetric Input Distributions

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*—This paper introduces extensions to data-driven polar decoders, enabling list decoding and accommodating asymmetric input distributions. These are crucial steps to develop data-driven codes that 1) achieve capacity and 2) are competitive in moderate block lengths. We commence by integrating list decoding into the data-driven polar codes, which significantly alleviates the inherent error propagation issues associated with successive cancellation decoding. Secondly, we expand the applicability of these codes to channels with stationary, non-uniform input distributions by incorporating the Honda-Yamamoto scheme. Both modifications are computationally efficient and do not require an explicit channel model. Numerical results validate the efficacy of our contributions, which offer a robust and versatile coding mechanism for various channel conditions.**

I. Introduction

Polar codes allow the construction of capacity-achieving codes for symmetric binary-input memoryless channels [1]. When given N independent copies of a binary discrete memoryless channel (DMC) W, 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 $\mathsf{I}(W)$ and the fraction of channels with capacity close to 0 approaches $1-\mathsf{I}(W)$, where $\mathsf{I}(W)$ 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). Arikan's transform also polarizes the bit channels $W_N^{(i)}$ in the presence of memory [2], and thus the encoding algorithm is the same as if the channel is memoryless. However, the decoding algorithm needs to be updated since the derivation of the SC decoder in [1] relies on the memoryless property. To account for the channel 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 successive cancellation trellis (SCT) decoding algorithm [3], [4].

However, the SCT decoder is only applicable when the channel model is known and when the channel's state alphabet size is finite and relatively small. For FSCs, the computational complexity of the SCT decoder is $O(|\mathcal{S}|^3 N \log N)$, where $|\mathcal{S}|$ is the number of channel states. For Markov channels where the set of channel states is not finite, the SCT decoder is not applicable without quantization of its states. With quantization, there may be a strong tension between the computational complexity and the error introduced by quantization. Additionally, the SCT decoder cannot be used for an unknown channel with memory without first estimating the channel as it requires an explicit channel model.

The authors of [5] proposed a novel methodology for data-driven polar decoders. The methodology uses a neural SC (NSC) decoder, which uses four distinct neural networks (NNs) instead of the elementary operations of the SC decoder. Specifically, the NNs approximate the channel's output statistics, the check-node, the bit-node, and the soft decision operations, denoted by E, F, G, H, respectively. The parameters of E, F, G, H 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 set of "clean" effective channels are determined by a Monte Carlo (MC) evaluation of the MI of the effective bit channels to complete the code design. The main advantage of this scheme is 1) its computational complexity does not grow cubicly with the channel's state alphabet size, and 2) it does not require an explicit channel model.

However, despite the fact that polar codes are capacity achieving, their performance under SC decoding are inferior to low density parity check (LDPC) and turbo codes at moderate block lengths. One of the reasons for that, as identified in [6], is that in SC decoding, decoding errors at early stages of the decoding procedure propagate to the succeeding bits, which yields in sub-optimal performance. Hence, the authors of [6] design a successive cancellation list (SCL) decoder for polar codes that instead of decoding a single codeword, as in the SC decoder, it decodes L codewords. Then, the decoder chooses one codeword from the list with the highest likelihood[1]. The performance of the SCL decoder improved dramatically towards the performance of the maximum likelihood (ML) decoder, and accordingly it is now part of the 5G standard. Therefore, it is of great interest to examine the performance of data-driven polar codes with list decoding, which is the first goal of this paper.

An additional issue to be addressed when designing capacity achieving codes is to accommodate data-driven polar codes

[1]The authors of [6] also showed the cyclic redundancy check (CRC) bits can be used as side information shared between the decoder and the encoder that allows to choose the correct codeword by checking which word in the list passes the CRC.

with asymmetric input distributions, as the capacity achieving input distribution is not necessarily uniform independently identically distributed (i.i.d.). In that regard, this paper provides an extension of data driven polar codes for stationary input distributions by incorporating the Honda-Yamamoto scheme [7] into the methodology of data-driven polar codes. This is the second goal of the paper.

The paper is organized as follows. Section II defines the notation and gives the necessary background on polar codes. Specifically, it presents polar codes as given in [1], and data-driven polar codes, as given in [5]. Section III extends data-driven polar codes to stationary input distributions. Section IV presents the idea of list decoding and its application to data-driven polar codes. Section V presents the numerical experiments.

II. Notations and Preliminaries

Throughout this paper, we denote by $(\Omega, \mathcal{F}, \mathbb{P})$ the underlying probability space on which all random variables are defined, with $\mathbb{E}$ denoting expectation. Random variables (RVs) are denoted by capital letters and their realizations are denoted by lower-case letters. Calligraphic letters denote sets, e.g. $\mathcal{X}$. We use the notation X^n to denote the RV $(X_1, X_2, \ldots, X_n)$ 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 $P_{X^nY^n} := \mathbb{P}\big|_{\sigma(X^n,Y^n)}$, where $\sigma(X^n, Y^n)$ is the σ-algebra generated by (X^n, Y^n). We denote by $[N]$ the set of integers $\{1, \ldots, N\}$. The MI between two RVs X, Y is denoted by $\mathsf{I}(X;Y)$.For two distributions P, Q, the cross entropy (CE) is denoted by $h_{\mathsf{CE}}(P, Q)$, the entropy is denoted by $\mathsf{H}(P)$ and the Kullback Leibler (KL) divergence is denoted by $\mathsf{D}_{\mathsf{KL}}(P\|Q)$. The notation $P \ll Q$ indicates that P is absolutely continuous with respect to Q.

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 $\mathcal{X} = \{0, 1\}$. For a memoryless channel, we denote its input distribution by $P_X = P_{X_i}$ for all $i \in \mathbb{Z}$. 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 $W_{Y^N\|X^N} = \prod_{i=1}^{N} W_{Y_0|Y_{-i+1}^{-1}, X_{-i+1}^{0}}$ denotes the probability of observing Y^N causally conditioned on X^N [8]. The symmetric capacity of a channel is denoted by $\mathsf{I}(W)$. 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 pairs of input-output vectors for 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.

A. Polar Codes for Symmetric Channels

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 $A^{\otimes N} := A \otimes \cdots \otimes A$ denotes an application of the $\otimes$ operator N times.

We define a polar code by the tuple $\left(\mathcal{X}, \mathcal{Y}, W, E^W, F, G, H\right)$ that contains the channel W, the channels embedding E^W and the core components of the 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 $E^W : \mathcal{Y} \to \mathcal{E}$ denotes the channel embedding, where $\mathcal{E} \subset \mathbb{R}^d$. For example, for a memoryless channel $W := W_{Y|X}$, a valid choice of E^W, as used in the remainder of this paper, is given by the following:

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

where the second term in the right-hand-side (RHS) cancels out in the case where P_X is uniform.

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 $H : \mathcal{E} \to [0, 1]$ a mapping of the embedding into a probability value, i.e. a soft decision. For the choice of E^W in (1), F, G, H are given by

$$\begin{aligned} F(e_1, e_2) &= 2\tanh^{-1}\left(\tanh\frac{e_1}{2}\tanh\frac{e_2}{2}\right), \\ G(e_1, e_2, u) &= e_2 + (-1)^u e_1, \\ H(e_1) &= \sigma(e_1), \end{aligned} \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the logistic function and $e_1, e_2 \in \mathcal{E}, u \in \mathcal{X}$. For this choice, the hard decision rule $h : [0, 1] \to \mathcal{X}$ is the round function $h(l) = \mathbb{I}_{l>0.5}$, where $\mathbb{I}$ is the indicator function. Applying SC decoding on the channel outputs yields an estimate of the transmitted bits and their corresponding posterior distribution [1]. Specifically, after observing y^N, SC decoding performs the map $(y^N, f^N) \mapsto \left\{\hat{u}_i, P_{U_i|U^{i-1},Y^N}\left(1|\hat{u}^{i-1}, y^N\right)\right\}_{i\in[N]}$, where f^N are the frozen bits that are shared between the encoder and the decoder. That is, $f_i \in \{0, 1\}$ if $i \in [N]$ is frozen, and $f_i = 0.5$[2] if i is an information bit. This mapping is denoted by

$$\left\{\hat{u}_i, P_{U_i|U^{i-1},Y^N}\left(1|\hat{u}^{i-1}, y^N\right)\right\}_{i\in[N]} = \mathsf{SC}_{\mathsf{decode}}\left(y^N, f^N\right). \tag{3}$$

For more details on SC decoding, the reader may refer to [1, Section VIII].

B. Neural Successive Cancellation Decoder

A NSC decoder [5] is defined by the tuple $\left(\mathcal{X}, \mathcal{Y}, W, E_{\theta_1}^W, F_{\theta_2}, G_{\theta_3}, H_{\theta_4}\right)$, where $E_{\theta_1}^W, F_{\theta_2}, G_{\theta_3}, H_{\theta_4}$ are NNs with parameters $\theta_i \in \Theta \subset \mathbb{R}^p$ in a compact space Θ. For simplicity, we denote $\theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$. The parameters θ are estimated in a training phase, in which the MI of the effective bit channels is estimated. The training phase includes the following steps. First, draw $x^N, y^N \sim \mathcal{D}_{M,N}$ and

[2]The value 0.5 is chosen arbitrarily to indicate that the bit needs to be decoded.

compute $u^N = x^N G_N$. Next, the functions $E_\theta^W, F_\theta, G_\theta, H_\theta$, are used to decode u^N with the SC decoding scheme, i.e. by applying $\mathsf{SC}_{\mathsf{decode}}\left(y^N, f^N\right)$ with $f^N = u^N$. This yields in an estimate of $\left\{P_{U_i|U^{i-1},Y^N}\left(1|u^{i-1}, y^N\right)\right\}_{i\in[N]}$ denoted by $\left\{P^\theta_{U_i|U^{i-1},Y^N}\left(1|u^{i-1}, y^N\right)\right\}_{i\in[N]}$. Finally, θ is optimized by the negative-log-loss (NLL) via stochastic gradient descent (SGD), as given by:

$$\min_{\theta\in\Theta} -\frac{1}{M}\sum_{i=1}^{M} \log P^\theta_{U_i|U^{i-1},Y^N}\left(u_i|u^{i-1}, y^N\right). \tag{4}$$

In [5, Algorithm 2], the authors showed a recursive formula for the computation of the NLL. Let

$$\mathsf{L} = \mathsf{NSCTrain}\left(e^N, u^N\right), \tag{5}$$

where $e_i = E_\theta^W(y_i)$, denote the computation of the NSC loss. Also, the authors of [5] showed that the NSC decoder is a consistent estimator of the theoretical polar decoder whenever W is a FSC.

III. Data-Driven Polar codes for Asymmetric Sources

This section describes how to extend the data-driven polar decoder in Section II-B to the case where the input distribution is not necessarily symmetric. Specifically, it starts with a brief description of the Honda-Yamamoto scheme [7]. Then, it extends the data-driven polar decoder to accommodate asymmetric input distributions by incorporating this scheme in Section III-B.

A. Honda-Yamamoto Scheme for Asymmetric Channels

The Honda-Yamamoto scheme [7] generalizes polar coding for asymmetric input distributions. Here, the polar decoder is applied twice: first, before observing the channel outputs and second, after observing the channel outputs. An equivalent interpretation is that the first application of SC decoding is done on a different channel whose outputs are independent of its inputs. Indeed, in this case, as given in (1), the first term of the RHS cancels out, and it follows that the channel embedding are constant for all $y \in \mathcal{Y}$. Thus, for the first application of SC decoding, we denote the constant *input embedding* by E^X (rather than E^W). The second application of SC decoding follows the same procedure as in the case of symmetric channels.

Accordingly, a polar decoder with non symmetric input distribution is defined by the tuple $\left(\mathcal{X}, \mathcal{Y}, W, E^X, E^W, F, G, H\right)$. Here, we add the input embedding E^X to the definition, where $E_X(y)$ is constant for all $y \in \mathcal{Y}$. An important observation is that the functions F, G, H are independent of the channel, i.e. both applications of SC decoding (before and after observing the channel outputs) share the same functions F, G, H.

B. Honda-Yamamoto Scheme for Data-Driven Polar Decoders

This section considers two issues. The first is the choice of an input distribution. This is addressed by employing algorithms for capacity estimation [9], [10]. The second issue addresses the construction of a NSC decoder that is tailored for stationary input distributions.

Algorithm 1 Data-driven polar code design for channels with memory and non-i.i.d. input distribution

input: Input distribution P_{X^N}, Channel $W_{Y^N\|X^N}$, block length n_{t}, #of info. bits k
output: Optimized $e_\theta^X, E_\theta^W, G_\theta, F_\theta, H_\theta$

Initiate the weights of $e_\theta^X, E_\theta^W, G_\theta, F_\theta, H_\theta$
$N = 2^{n_{\mathsf{t}}}$
for $k = 1$ to $\mathsf{N}_{\mathsf{iters}}$ **do**
  Sample $x^N, y^N \sim P_{X^N} \otimes W_{Y^N\|X^N}$
  $u^N = x^N G_N$
  Duplicate e_θ^X to e_X^N
  Compute e_Y^N by $e_i = E_\theta^W(y_i)$
  Compute L_X by applying $\mathsf{NSCTrain}\left(e_X^N, u^N, 0\right)$
  Compute L_Y by applying $\mathsf{NSCTrain}\left(e_Y^N, u^N, 0\right)$
  Minimize $\mathsf{L}_X + \mathsf{L}_Y$ w.r.t. θ.
end for
return Optimized θ

For the choice of the input distribution, we employ a recent method for the optimization of the directed information neural estimator (DINE), as presented in [10]. Therein, the authors provide an reinforcement learning (RL) algorithm that uses DINE to estimate capacity achieving input distributions. The input distribution is approximated with an recurrent neural network (RNN) with parameter space denoted by Π. Let P_X^π be the estimated capacity achieving input distribution. Thus, by application of [10, Algorithm 1], we obtain a model of P_X^π from which we are able to sample the channel inputs.

Extension of the NSC decoder to P_{X^N} (that is not uniform and i.i.d.) involves introducing additional parameters, that we denote by $\theta_5 \in \Theta$. Accordingly, we denote the set of the channel embedding by θ_1, θ_5, where θ_5 denotes the parameters of E^X and θ_1 are the parameters of E^W. We define $E_\theta^X : \mathcal{Y} \to \mathbb{R}^d$ as a constant RV that satisfies $E_\theta^X(y) = e_X \in \mathbb{R}^d$ for all $y \in \mathcal{Y}$. Accordingly, the NSC in this case is defined by $E_\theta^X, E_\theta^W, F_\theta, G_\theta, H_\theta$. Thus, the NSC decoder needs to be updated in order to optimize E_θ^X as well. This is addressed by first applying the NSC with inputs e_X^N to compute $P^\theta_{U_i|U^{i-1}}$, where $e_X^N \in \mathbb{R}^{d\times N}$ is a matrix whose columns are duplicates of e_X. Second, the NSC is applied with e_Y^N to compute $P^\theta_{U_i|U^{i-1},Y^N}$, where $e_Y^N \in \mathbb{R}^{d\times N}$ is a matrix whose i-th column is $E_\theta^W(y_i)$.

The training procedure admits the following steps. First, the channel inputs and outputs are sampled by $x^N, y^N \sim P_{X^N}^\pi \otimes W_{Y^N\|X^N}$. Then, the values of $u^N = x^N G_N$ are computed, and form the labels of the algorithm. Next, the channel statistics e_Y^N are computed and the input statistics are duplicated to obtain e_X^N. The next step is to apply the NSC-Train procedure twice, i.e.

$$\mathsf{L}_X = \mathsf{NSCTrain}(e_X^N, u^N) \tag{6}$$

$$\mathsf{L}_Y = \mathsf{NSCTrain}(e_Y^N, u^N), \tag{7}$$

which are minimized via SGD, as shown in Algorithm 1.

IV. List Decoding of Data-Driven Polar Codes

In this section, we delve into the concept of list decoding for polar codes and discuss its integration into our data-driven polar codes. To this end, the NSC is benchmarked against two ground truth decoding methods: the SC decoder and the SCT decoder, depending on the presence or absence of channel memory. Notably, contemporary algorithms predominantly utilize the list decoding technique, known for its improved performance compared to the conventional SC algorithm. Consequently, to enable the NSC decoder to compete with state-of-the-art algorithms, this section incorporates list decoding with the NSC decoder.

A. SC List Decoder

To enhance the error correction performance of polar codes, especially with codes of moderate lengths, the SCL decoding algorithm was introduced in [6]. The fundamental concept behind list decoding lies in leveraging the structured nature of the polar transformation. Instead of relying solely on a single SC decoder, the SCL decoder concurrently decodes multiple codeword candidates. This is achieved by applying multiple SC decoders over the same channel's outputs, with the number of these decoders denoted as the list size L.

The SCL decoder generates a list of potential codewords, each ranked by its likelihood of being the transmitted message. Subsequently, this list undergoes a refining process to identify the most likely original message. To achieve this, the SCL algorithm estimates each bit's value (0 or 1) while considering both possibilities. At each estimation step, the number of codeword candidates (also referred to as paths) doubles. To manage the algorithm's complexity, it employs a memory-saving strategy by retaining only a limited set of L codeword candidates at any given time. Consequently, after each estimation, half of the paths are discarded. To determine which paths to retain, a path metric (PM) is associated with each path. This metric is continuously updated with each new estimation and is computed via the log-likelihood ratios (LLRs). The algorithm maintains the L paths with the lowest path metrics, allowing them to persist and continue the decoding process.

B. NSC List Decoder

Here we highlight that the concept of list decoding can be integrated into our data-driven polar codes. Recall that the NSC decoder uses the same structure as the SC decoder and the SCT decoder, with the only distinction being the replacement of elementary operations with NN. Accordingly, we can seamlessly incorporate the list decoding concept into the NSC decoder. Specifically, since the NSC decoding algorithm can estimate the LLRs at the decision points, we can leverage them to compute the PM and follow the same SCL decoding procedure.

C. Computational Complexity

The standard SC algorithm has a computational complexity of $O\left(N\log(N)\right)$, whereas the SCT algorithm's computational complexity is $O\left(|\mathcal{S}|^3 N\log(N)\right)$. In the context of list decoding, a technique based on leveraging the memory sharing structure among the candidate paths was introduced in [6]. This technique demonstrates that the SCL decoder can be implemented with a computational complexity of $O\left(LN\log(N)\right)$. When applying the same technique to the SCT algorithm with list decoding, it follows directly that the computational complexity increases to $O\left(L|\mathcal{S}|^3 N\log(N)\right)$.

The following theorem examines the computational complexity of the NSC list decoder for the case where $E_\theta, F_\theta, G_\theta, H_\theta$ are NNs with k hidden units and the embedding space satisfies $\mathcal{E}\subset\mathbb{R}^d$. Due to space limitation, the proof is omitted.

Theorem 1. *Let $E_\theta, F_\theta, G_\theta, H_\theta$ be NNs with k hidden units and let $\mathcal{E}\subset\mathbb{R}^d$. Then, the computational complexity of NSC list decoding is $O\left(LkdN\log_2 N\right)$.*

The main purpose of Theorem 1 is to facilitate a comparison between the NSC list decoder and SCT list decoder. Note that the computational complexity of the SCT list decoder, as previously mentioned, scales with the memory size of the channel $O\left(L|\mathcal{S}|^3 N\log N\right)$. This highlights a key advantage of the NSC list decoder since its computational complexity remains independent of the channel's memory size.

V. Experiments

This section presents experiments designed to evaluate the performance of our proposed algorithms. It begins with asymmetric channels in Section V-A and continues with list decoding in Section V-B. In all experiments, the NNs, $F_\theta, G_\theta, H_\theta, E_\theta^X, E_\theta^W$, are implemented by two layered fully-connected NNs with 50 hidden units per layer.

A. Asymmetric Channels

In this section, we conduct experiments to evaluate our methodology for designing polar codes tailored to asymmetric channels. As an example of a memoryless channel, we consider the non-symmetric BEC, as defined in [11]. This channel is defined by two erasures probabilities, ϵ_0, ϵ_1,

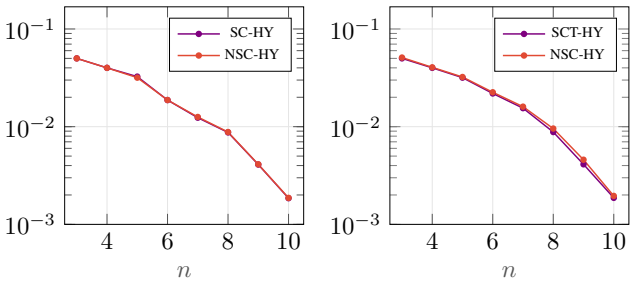

Figure 1: These figures compare the bit error rates (BERs) attained by the Honda-Yamamoto scheme (SC-HY) and its extension to the NSC decoder (NSC-HY). The left and right figures show the results on an asymmetric binary errasure channel (BEC), and the Ising channel, respectively.

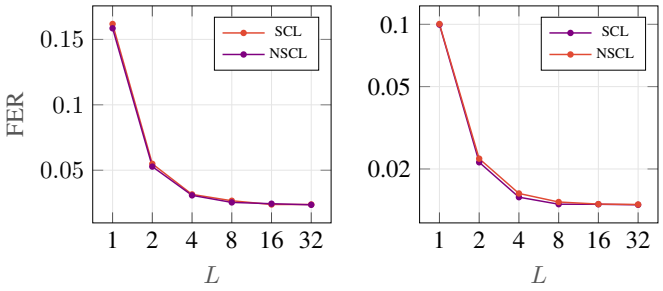

Figure 2: These figures compare the FERs attained by the the SCL decoder and its extension the NSC decoder (NSCL). The left figure shows the results on an binary-input AWGN channel with signal-to-noise ratio (SNR) of 1.5, and the right figure shows the results on the Ising channel.

namely the probabilities for an erasure of the "0" symbol and the "1" symbol, respectively. Accordingly, $W(x|x) = 1 - \epsilon_x$, $W(?|x) = \epsilon_x$ for $x \in \{0, 1\}$. Similar to [11], we choose $\epsilon_0 = 0.4$ and $\epsilon_1 = 0.8159$.

As an instance of a channel with memory, we choose the Ising channel that was introduced in [12], which belongs to the family of FSCs, and therefore, its optimal decoding rule is given by the SCT decoder. This channel 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 states at the beginning of the transmission and S' is the channel's state at the end of the transmission.

Figure 1 compares the BERs obtained via the extension of the Honda-Yamamoto scheme, as described in Section III, and by the optimal decoding rule of the Honda-Yamamoto scheme. The left figure compares the result on the asymmetric BEC, a memoryless channel, and the right figure compares the results on the Ising channel, a FSC.

B. List Decoding

In this Section, we demonstrate the performance of NSC list decoder compared to the SCL decoder. As an example of a memoryless channel, we consider the additive white Gaussian noise (AWGN) channel. The AWGN channel is defined by the following 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 i.i.d. Gaussian noise. In our experiments $\sigma^2 = 1.5$. Figure 2 illustrates the frame error rates (FERs) obtained via the SCL decoder with the FERs obtained via the NSC list decoder as a function of the list size L. The left figure demonstrates the results for the AWGN channel while the right figure compares the results for the Ising channel. As can be seen in the figures, the NSC list decoder indeed converges to the ground truth SCL decoder for both channels.

VI. Conclusions

This paper presents pivotal extensions to data-driven polar decoder, addressing two critical applications: list decoding and adaptation to asymmetric input distributions. These enhancements are essential steps towards realizing data-driven codes that achieve channel capacity and excel at moderate block lengths. By seamlessly integrating list decoding, we effectively mitigate error propagation issues inherent to SC decoding, improving the practical performance of polar codes. Simultaneously, our incorporation of the Honda-Yamamoto scheme enables these codes to adapt to non-uniform input distributions in a computationally efficient manner, without the need for explicit channel model. Our numerical results validate the effectiveness of these contributions, establishing data-driven polar codes as robust and versatile coding solutions adaptable to diverse channel conditions.

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

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

[7] 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.

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

[9] 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*,

[10] 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.

[11] J. Honda and H. Yamamoto, "Polar coding without alphabet extension for asymmetric channels," in *2012 IEEE International Symposium on Information Theory Proceedings*, IEEE, 2012, pp. 2147–2151.

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