Source Polarization-Adjusted Convolutional Codes

Tyler Kann[1], Shrinivas Kudekar, Matthieu Bloch[1]
[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
tkann3@gatech.edu, kudekar@gmail.com, matthieu.bloch@ece.gatech.edu

2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206454

***Abstract*—Motivated by applications to low-latency secret key generation in physical-layer security, we study Polarization-Adjusted Convolutional (PAC) codes for source coding with side information. Source PAC codes operate in a dual manner to channel PAC codes by introducing a rate-one convolutional code *after* the polarization transform. The decoding of source PAC codes requires a careful scheduling of the successive cancellation decoder and a careful optimization of the rate profiling. Our empirical results demonstrate the improved performance of source PAC codes over regular polar codes using Successive Cancellation List (SCL) decoding. We illustrate the performance in terms for key generation rate in a secret-key generation setup over an Additive White Gaussian Noise (AWGN) channel, suggesting that PAC codes could improve the performance of physical-layer security schemes at short blocklength.**

I. Introduction

Driven in part by emerging applications such as augmented reality and remote control, existing and emerging communication standards have pushed the development of low-latency and reliable coding schemes, as exemplified by the Ultra-Reliable Low-Latency Communication (URLLC) requirements in the 5G standard. These new stringent requirements have called for the development of new and improved error-control coding at very short blocklength of $N = 256$ and below [1]. Candidate coding schemes that have shown promising performance include extended Bose-Chaudhuri-Hocquengham (eBCH) [2], Reed-Muller (RM) codes [3], polar codes [4] and Polarization-Adjusted Convolutional (PAC) codes [5]–[7], especially when used in combination with decoding techniques such as Successive Cancellation List (SCL) [7], [8], Ordered-Statistics Decoding (OSD) [9], or Guessing Random Additive Noise Decoding (GRAND) [10].

While much of the focus has been on channel coding, source coding with side information [11] is another coding mechanism with useful applications. In particular, source coding with side information plays a central role in physical-layer security [12] for the reconciliation phase of secret-key generation protocols [13], [14], including Quantum Key Distribution (QKD) [15]. Despite the duality between source coding with side information and channel coding [16], which allows one to transform a good channel in a good source code with side information for symmetric channels and sources, extending the results to asymmetric sources presents some challenges [17]. Hence, finding explicit and direct code constructions for source coding with side information remains of interest. Polar codes are of particular interest for source coding with side information, not only because of their asymptotic optimality [18] and finite length performance [14], but also because polar codes for source coding with side information have proven a useful building block to construct codes for a host of multi-user information theory problem in an almost systematic way [19]–[21]. The main contribution of the present work is i) to propose encoding and decoding algorithms for source PAC codes; and, ii) to demonstrate numerically their excellent performance.

The remaining of the paper is organized as follows. We review necessary notation and concepts related to PAC codes in Section II-B. We then introduce the source coding with side information problem and our proposed source PAC codes in Section III, and present numerical results in Section IV. We conclude the paper with a discussion of the usefulness of our proposed approach for low-latency key generation over wireless channels in Section V.

II. Polarization-Adjusted Convolutional Codes

A. Polar Codes

A polar code for channel coding is characterized by its blocklength $N \triangleq 2^n$, the number of information bits K, and an information set $\mathcal{A} \subset [1; N]$ that specifies how to encode information bits. Specifically, a vector of K information bits m is encoded into a length N vector u such that $u_{\mathcal{A}} \triangleq m$ and $u_{\mathcal{A}^c} \triangleq 0$, the all-zero vector. The set $\mathcal{A}^c$ is called the frozen set. Upon setting $G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, the base polarization matrix, and $G^{\otimes n}$ the n^{th} order Kronecker product of G, a codeword x is created through the operation $x \triangleq uG^{\otimes n}$, a process known as the polar transform. The structure of the matrix $G^{\otimes n}$ allows for an encoding complexity of $O(N \log N)$. The codeword x is then sent over the channel to a receiver that observes a noisy version y, e.g., $y_i = x_i + n_i, n_i \sim \mathcal{N}(0, \sigma)$ if the channel is an Additive White Gaussian Noise (AWGN).

The standard decoding algorithm for polar codes is the Successive Cancellation (SC) decoder, by which bits from the information set $\mathcal{A}$ are successively decoded based on their Log-Likelihood Ratio (LLR) given the past decoded bits according to the maximum-likelihood rule:[1]

$$\forall i \in \mathcal{A} \quad \hat{u}_i = \begin{cases} 0 & \text{if } \lambda_i^0 = \ln \frac{P(y, \hat{u}^{0:i-1} | \hat{u}_i = 0)}{P(y, \hat{u}^{0:i-1} | \hat{u}_i = 1)} > 0 \\ 1 & \text{else.} \end{cases}, \quad (1)$$

This works was supported in part by the National Science Foundation (NSF) under grant 2148400 as part of the Resilient & Intelligent NextG Systems (RINGS) program.

[1]It is sometimes convenient to consider a randomized decoding rule to analyze polar codes [22].

$$\forall i \in \mathcal{A}^c \quad \hat{u}_i = 0, \tag{2}$$

where $\hat{u}^{0:i-1} = \{\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{i-1}\}$ denotes the vector of past decisions. The LLRs can be efficiently computed recursively, resulting in a decoding complexity of $O(N \log N)$. The LLRs can also be viewed as decisions made at the output of individual *bit channels*, corresponding to channels with input bit u_i and output $(y, \hat{u}^{0:i-1})$. The choice of the set $\mathcal{A}$, called *rate profiling*, plays a crucial role in determining the performance of polar codes, and several criteria have been proposed based on the capacity of the bit channels [23] or the RM profile [24], which consists in enumerating the Hamming weight of RM codewords and selecting those with highest weight.

Given that each decision in the SC decoder relies on previously decoded bits, one bad decision can propagate and affect future ones. SCL list decoding [25] mitigates this problem and be implemented relatively efficiently. In SCL, the decoding process is viewed as following branches of a tree and tracks up to L branches in a list. The tree splits at every non-frozen bit, creating two branches $u_i = 0$ and $u_i = 1$. A path metric is updated based on the decision of each path, and when the number of paths is greater than L, the list is pruned to track L paths, keeping the paths with the lowest path metrics.

B. PAC Codes

Polarization-Adjusted Convolutional (PAC) codes have been introduced as a means to improve the finite-length performance of polar codes. In polar codes, the rate profiling consists in identifying the bit-channels that are either almost noiseless or completely noisy, corresponding to a rate assignment of 0 or 1 to each bit-channel. While this approach is asymptotically optimal [4], there is a loss in performance at small and medium blocklengths because polarization takes place relatively slowly. The main idea behind PAC codes is to augment the polar code with an outer *rate one* convolutional code *before* the polar transform. Upon denoting v and u the input and the output of the convolutional code, respectively, the encoding process remains similar to that of polar codes. A rate-profile associated to an information set $\mathcal{A}$ determines the placement of information bits, so that $v_{\mathcal{A}}$ contains the information bits and $v_{\mathcal{A}^c} = 0$. The use of an RM profile was shown to yield significant performance improvements [5] over polar codes, while recent results suggest that adapting the rate-profile to the specific structure of PAC codes yields further improvements [26]. Despite attempts to analyze PAC codes, e.g., from the perspective of the number of minimum weight codewords [8], [27], a theoretical justification for the performance of PAC codes and the identification of optimal rate profiles remains elusive. Nevertheless, several follow-up works have provided insight into the complexity of PAC decoding [7], [28]. PAC codes can be viewed as a specific case of polar codes with dynamically frozen sets [29], [30].

III. Source PAC codes

A. Source Coding with Side information

We are interested here in the problem of source coding with side information [11], in which the encoder observes N independent and identically distributed (i.i.d.) realizations $x \triangleq \{x_i\}_{i=1}^N$, $x_i \in \{\pm 1\}$, of a Bern(p) distribution and compresses them into a message m. The receiver obtains m and has access to N observations $y \triangleq \{y_i\}_{i=1}^N$ where $y_i = x_i + n_i$ and n_i are i.i.d. realizations of an $\mathcal{N}(0, \sigma^2)$ distribution to form an estimate $\hat{x} \triangleq \{\hat{x}_i\}_{i=1}^N$ of x. This special setup models the type of source models that is common in secret-key generation over wireless channels [14]. The objective is to achieve a low frame error rate $\mathbb{P}(\hat{x} \neq x)$ while reducing the size of m as much as possible.

Polar source coding operates in a dual manner to polar channel coding. Polarized bits are computed from the sequence of observations x as $u \triangleq xG^{\otimes n}$ and a rate-profile is used to select the polarized bits forming the message m. Intuitively, the bits revealed are high entropy bits that are difficult to reconstruct, while the non-revealed bits with low-entropy are reconstructed by the decoder using SC decoding and the side information y. The asymptotic optimality of source polarization [18] holds under slightly more general assumptions than channel polarization [4] because it does not require any symmetry in the source distributions. This observation was leveraged to extend polar constructions to multi-user channel coding problems, see e.g., [20] for one of the first such applications.

B. Source PAC Codes

The encoding and decoding process for source PAC codes is illustrated in Fig. 1. As in traditional source polarization, the N source bits x are mapped to u through the polarization transform $G^{\otimes n}$ as $u \triangleq xG^{\otimes n}$. Following polarization, the bits of u are sent through the $N \times N$ bit-reversal operation matrix P and we let $u^P \triangleq uP$. The rationale for this permutation matrix will be further discussed in Section III-C, suffice here to say that it ensures that the decoder is able to decode bits in the correct order during successive cancellation. The permuted polarized bits u^P are then convolved using a polynomial $\mathbf{g} = [g_0, g_1 \ldots, g_m]$ to generate v, where $v_i = \sum_{j=0}^m g_j u^P_{i-j}$, $g_k \in \{0, 1\}$ and $g_0 = g_m = 1$ by convention.[2] The convolutional

[2]This order of encoding constitutes the main difference with the PAC scheme for channel coding in which the convolution is performed *before* polarization encoding.

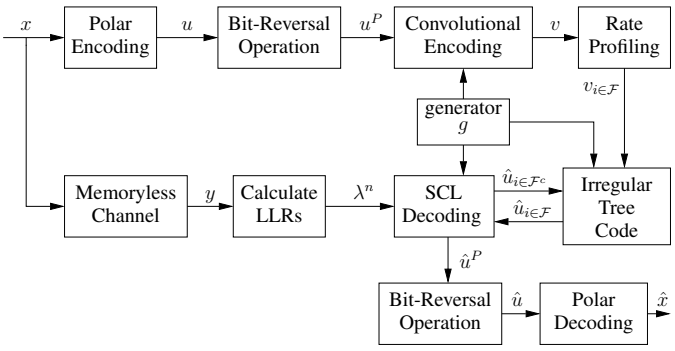

Fig. 1: Block diagram of source coding with side information using PAC methodology.

coding is succinctly described by the matrix operation $v \triangleq u^P T$, with T an upper triangular Toeplitz matrix, filled out by the coefficients of $\mathbf{g}$ as in (3) below.

$$T = \begin{bmatrix} g_0 & \dots & g_m & 0 & \dots & 0 \\ 0 & g_0 & \dots & g_m & & \vdots \\ \vdots & 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & 0 & \vdots \\ \vdots & & 0 & 0 & g_0 & g_1 \\ 0 & \dots & \dots & 0 & 0 & g_0 \end{bmatrix} \tag{3}$$

Once v is generated, rate profiling is performed to select a set of positions $\mathcal{F} \subset [1; N]$ of bits to reveal. Thus, only the bits $v_{\mathcal{F}}$ are revealed to the decoder. The decoder's task is to then identify the remaining bits in the set $\mathcal{F}^c$ with the help of the side information. Once u has been obtained, the polar transform is inverted to get back the original source bits x. As for PAC codes, the choice of the rate-profiling significantly impacts the overall performance and is discussed in Section III-D.

C. Decoding of Source PAC codes

The complete algorithm for SCL decoding of source PAC codes is given in Algorithm 1. Note that the algorithm relies on routines updateLLRs (to update LLRs), UpdatePM (to update path metrics), updatePartialSums (to reconstruct the bits at each stage of polarization), duplicatePath (to duplicate a path in the decoding tree), prunePaths (to remove branches in the tree), and pruneToOnePath (to reduce the list of paths to a single one) that are identical to those in [25] and not reproduced here for brevity. List source decoding of PAC codes with side information operates similar to the non-PAC case. For bits that are not revealed, $\hat{u}_i$ is estimated just as source coding is traditionally done. For $i \in \mathcal{F}$, v_i is revealed and u_i^P is deconvolved from v_i and the current memory (lines 31-35 in Algorithm 1). If the recent bits of u^P (more specifically the vector of bits $u_{i-m:i-1}^P$ in memory) have all been decoded correctly, v_i, which was a linear combination of u_i^P and the memory, will map back to u_i^P correctly and will behave just as if u_i^P were revealed. However, if the local memory contains erroneous bits, decoding even the frozen bit correctly bit is not guaranteed. This means that, while good paths are unaffected by PAC, bad paths become even worse and generate a much larger path metric.

We now discuss the importance of the shuffling, using Figure 2 as an illustration. The first two bits to be decoded in polar codes are u_0 and u_2, hence if $\mathcal{F} = \{0, 1\}$, one would reveal $u_0^P = u_0, u_1^P = u_2$. However, since v is already created from the shuffled u, for the same $\mathcal{F}$, one would reveal v_0, v_1. This emphasizes the need for the shuffling, for without it, the second v_i to be decoded would be a function of u_0 and u_1, a bit that has not been decoded yet. Furthermore, even if v_i were to be revealed, it would not give us information on u_2, which is necessary to continue the decoding. *Thus, the shuffling ensures*

Algorithm 1: SCL decoding of source PAC codes

input : $v_{i\in\mathcal{F}}$, received LLRs λ^n, $\mathcal{F}$, list size L, convolutional generator $\mathbf{g}$
output: message estimate x

1 $\mathcal{L} \leftarrow 1$
2 LLRs = zeros([N, n+1, $2L$]) `// Contains all` λ `at all stages and lists`
3 β = zeros([N, n+1, $2L$]) `// Contains all bits at all stages and lists`
4 PM = zeros([1:$2L$])
5 **for** $i \leftarrow 0$ **to** $N-1$ **do**
6 **if** $i \in \mathcal{F}$ **then**
7 **for** $l \leftarrow 1$ **to** $\mathcal{L}$ **do**
8 j = BitPerm(i)
9 LLRs[l] $\leftarrow$ updateLLRs(LLRs[l], $\beta[l]$, j) `// Same as SCL`
10 $v_i \leftarrow v_i$ `// Convolved bit` v_i `revealed`
11 mem $\leftarrow \beta[0, i-m : i-1, l]$ `// If` $i - m < 0$`, append with zeroes`
12 $\hat{u}_j^P \leftarrow$ convUndo(v_i, mem, g) `//` u^P `is recreated`
13 $\beta[0, i, l] \leftarrow \hat{u}_j^P$
14 PM[l] = UpdatePM(LLRs[0,j,l],$\hat{u}_j^P$)
15 $\beta[l] \leftarrow$ updatePartialSums($\beta[l]$) `// same as SCL`
16 **end**
17 **else**
18 **for** $l \leftarrow 1$ **to** $\mathcal{L}$ **do**
19 LLRs[$\mathcal{L}+l$], $\beta[\mathcal{L}+l]$, PM[$\mathcal{L}+l$] $\leftarrow$ duplicatePath(LLRs[l], $\beta[l]$, PM[l]) `// same as SCL`
20 $\mathcal{L} \leftarrow 2\mathcal{L}$
21 **end**
22 **if** $\mathcal{L} > L$ **then**
23 LLRs, β, PMs $\leftarrow$ prunePaths(LLRs, β, PMs) `// same as SCL`
24 $\mathcal{L} = L$
25 **end**
26 **end**
27 $\beta^* \leftarrow$ pruneToOnePath(β, PM) `// Keep the 1 path with lowest metric`
28 $\hat{u}^P = \beta^*[0, :]$ `// This row is are the bits on the far right`
29 **return** $\hat{x} \leftarrow \hat{u}^P PG$
30 **end**
31 **subroutine** undoConv(v, mem, g)
32 $u_i^P \leftarrow v_i$
33 **for** $i \leftarrow 1$ **to** m **do**
34 $u_i^P \leftarrow u_i^P \oplus$ mem$[m-i]$g$[i]$
35 **end**
36 **return** u_i^P

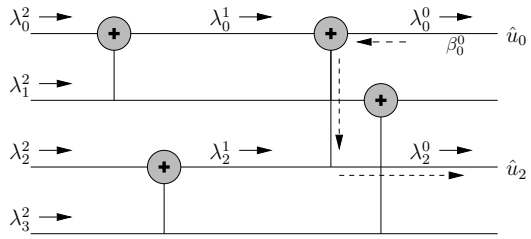

Fig. 2: The first two bits decoded in an $N = 4$ length polar code diagram. The side information comes from x (the left hand side), and the λs flow left to right.

that decoding actions are causal and that one obtains the bit that would be expected in standard polar coding. Additionally, the LLR matrix is a matrix filled with all LLRs λs at all stages, and is filled in at indices $j = i^P$ following the decoding process (e.g. LLR[1] is filled and then LLR[3] from Figure 2). Our β is a matrix of the same size but is filled with all the bits of each stage, and it is filled such that $\beta[i] = u_i^P$. This allows the memory to be easily grabbed, but results in $\hat{u}^P$ being the output, instead of $\hat{u}$. The process can be seen in more detail in Algorithm 1, and using Figure 2 we shall discuss a simple example. We start with $\lambda^2 = (\lambda_0^2, \lambda_1^2, \lambda_2^2, \lambda_3^2)$, which corresponds to the vector of received LLRs from the channel, and fill the first row of our LLR matrix. We make our calculations left to right to generate λ_0^1 and λ_2^1, the LLRs after the first stage of polarization, and fill the second row of the matrix at positions 0 and 2. Suppose for the sake of illustration, we reveal only bits at position 0 ($v_0, i = 0$) and 1 ($v_1, i = 2$) to the decoder ($\mathcal{F} = \{0, 2\}$). The decoder's task is then to determine bits at the positions 2 and 3 to recover the whole vector u and then reconstruct x. Since bit at position 0 is known, we know $v_0 = u_0$ and propagate it back to calculate λ_2^0 from λ_0^1, λ_2^1 and value of the bit at position 2. More precisely, we add the two LLRs with λ_0^1 adjusted for sign by multiplying with $1 - 2\hat{u}_0$. That would then allow us to decode $\hat{u}_2$. We can now update our LLR matrix at the last row in indices 0 and 2, generate β at indices 0 and 1 at the last row , as well as β at the row above in indices 0 and 1, since we can calculate $\{u_0^P + u_1^P, u_1^P\}$. We then obtain λ_3^1 and λ_3^0 (and update the matrix accordingly). At $i = 2, BitPerm(i) = 1$, we are given $v_2 = u_2^P + u_1^P = u_1 + u_2$ (note that due the shuffling we are using previously decoded bits and only have one unknown u_1). If we have calculated previous bits correctly, we can properly estimate $\beta[2] = u_1$, and ultimately use it to decode u_3. At the end, $\hat{u}_P$ will have been created and $\hat{x}$ can be recovered as $\hat{u}_P P G^{\otimes n}$.

D. Weighted Sum for Source Coding with Side Information

The choice of the rate profiling is key to achieve finite-length performance with PAC codes. While the RM profile offers good performance, we propose a modification based on [26] to further improve performance. The key idea behind the rate-profile of [26] is to identify the bit-channels to transmit information by accounting for the entire transform consisting of both the convolution and the polarization matrices. [26] considers each bit before convolution and accounts for all the polarization channels W it goes through. Briefly, τ represents the number of information bits being combined by the convolutional code, to be transmitted over a particular polarized bit-channel whose reliability is given by the metric ω. Then, the metric θ, for the information bit before the convolution, is the sum over all the bit-channels that the information bit is sent over, but weighted by $\omega/(\tau + 1)$ for each underlying bit-channel. The more information bits being linearly combined from a generator g in a channel, the more τ is increased, meaning an error here would be more disastrous and the metric θ is lowered. However it is increased if the channels that are being combined have strong reliability metrics themselves, as ω is proportional to the reliability of any one channel. We impose the same metric found in equation (12) of [26], yet we account for the fact that the choice of revealed bits comes at the end, as opposed to channel codes when the selection of frozen bits is at the beginning. This means any bit v is a sum of previous (shuffled) bits. While we use the same overall algorithm as [26], we modify equation (12) to be:

$$\theta_i \triangleq \sum_{j=0}^{m} \frac{g_j \omega_{[i-j]^P}}{1 + \tau_{[i-j]^P}},$$

with $\frac{\omega_{[i-j]^P}}{1+\tau_{[i-j]^P}} = 0$ when $i - j < 0$. We denote the resulting profiling from [26] and our modified equation as Source Weighted Sum (SWS).

IV. Numerical Results

We now investigate different setups of PAC source coding with side information and compare the error correcting performance of PAC codes versus compared the original polar codes. To evaluate performance, we explore different rates, block lengths, and rate profiles. Our simulations consider a Binary Phase-Shift Keying (BPSK) source x whose side information y is obtained by transmitting the symbols over an AWGN channel. For the case of a symmetric source x, we obtain as expected results very similar to [7] because of the duality of channel and source coding. In addition, we also show how the benefits of PAC codes extend to asymmetric sources. Results are summarized in Figures 3-5.

For the convolutional transform we use 0o133 as the generator coefficients, with a constraint length $m = 6$ (generator length = 7). This generator is commonly used throughout the literature and yields good performance, although there is currently no known optimal generator.

We compare different list sizes in Figure 3. We see that using PAC, a list size of 32 is able to compete with list size of 256. Using list size 256 and PAC, we are able to see nearly a third of a dB gain in performance. As seen in Fig. 4, we also show that PAC codes maintain the performance gain when the source x is asymmetric. Therein, PAC with a list size of 32 when the source is Bern(.3) offers benefits similar to when it is a Bern(.5) source. Lastly, in Figure 5, the codes considered have a rate different than .5. This figure, alongside Figure

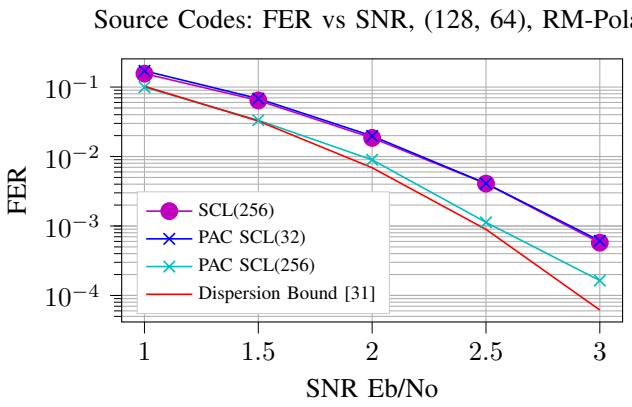

Fig. 3: PAC performance for $(128, 64)$ code and symmetric source $X \sim \text{Bern}(0.5)$. PAC source codes achieve polar code performance with a much smaller list size and outperform them for the same list size. Note that for (128,64) SWS and RM are equivalent. We are able to achieve similar results as [7].

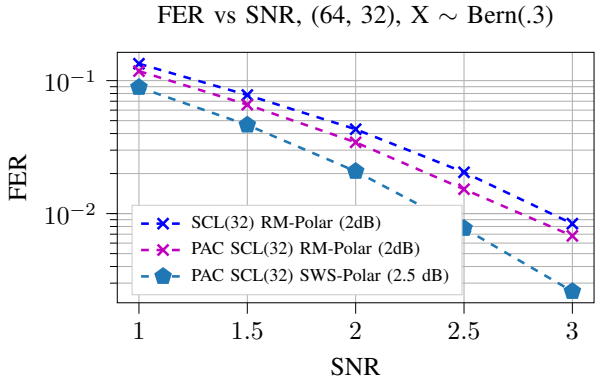

Fig. 4: PAC performance for $N = (64, 32)$ code from an asymmetric source $X \sim \text{Bern}(0.3)$.

4, demonstrates the importance of choosing the frozen set correctly, as our SWS metric can be seen to vastly outperform the metric of RM-Polar from [24].

V. Secret Key Generation

The benefits of source PAC codes over standard source polarization yield direct benefits to any application that uses source polarization in the short block length regime. Notably, [32], [33] exploit polar codes for secret key generation. In [14], Alice and Bob have access to X and Y respectively, where Y is a noisy version of X. Alice and Bob then perform information reconciliation, with the goal of Bob reliably reconstructing a quantized version of X with as few bits as possible. Eve also has access to the bits revealed to Bob. Ultimately the key is generated through source polarization with side information, meaning any improvement in the source coding rate yields an equal improvement in the secret key rate. In Figure 6, analogous to showing that one can use a higher rate with PAC codes, we show the FER vs Rate under a fixed noise

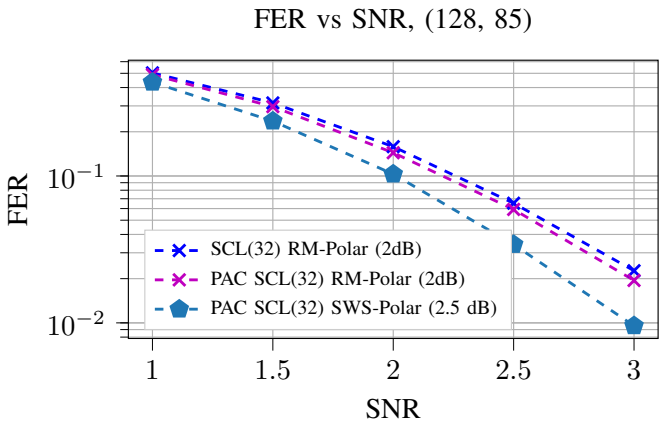

Fig. 5: PAC at a rate smaller than $\frac{1}{2}$. Here we use the generator found in [26], and are able to achieve similar performance.

constraint for $N = 64$ length codes. It is clear that for any rate, our source PAC codes achieve improved performance. At such a small blocklength, this increase is non-trivial. For example, if one wants to transmit with probability of error $\epsilon \leqslant .01$, one needs to reveal only 36 bits with PAC and 38 with Polar, a 5% decrease in rate.

Since the PAC benefits in our algorithm only result from the frozen bits, it is no surprise that the benefits become more apparent when $|\mathcal{F}|$ gets larger. Even further, if the last revealed bit comes before the first information bit, as is common in very low compression rates, our codes will behave exactly as polar codes. We also notice that due to bits relying more on the correct decoding of previous bits, as well as the fact that revealed bits are no longer guaranteed to be correct, even though the overall FER sees improvement, the BER is similar when comparing Polar and PAC. More specifically this means that the BER of PAC messages that were not decoded correctly are much higher than incorrect Polar messages, giving weight to our interpolation in III-C that incorrect paths are made even worse.

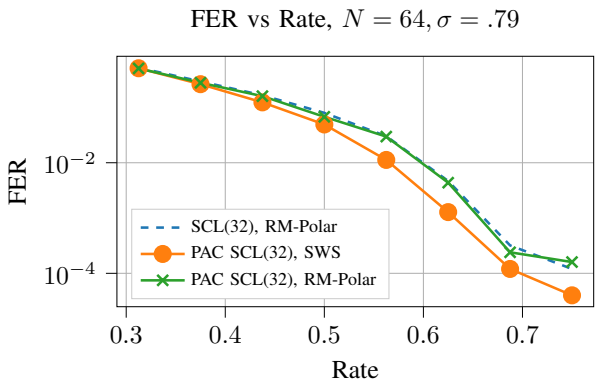

Fig. 6: Given a fixed noise of $\sigma = .79$ for the side information, the FER at various rates, under list size L = 32.

REFERENCES

[1] C. Yue, V. Miloslavskaya, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, "Efficient decoders for short block length codes in 6G URLLC," Dec. 2022.

[2] M. C. Coskun, G. Durisi, T. Jerkovits, G. Liva, W. E. Ryan, B. Stein, and F. Steiner, "Efficient error-correcting codes in the short blocklength regime," *CoRR*, vol. abs/1812.08562, 2018. [Online]. Available: http://arxiv.org/abs/1812.08562

[3] M. C. Coskun, J. Neu, and H. D. Pfister, "Successive cancellation inactivation decoding for modified reed-muller and eBCH codes," in *Proc. of IEEE International Symposium on Information Theory*, Los Angeles, California, USA, Jun. 2020.

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

[5] E. Arıkan, "From sequential decoding to channel polarization and back again," Aug. 2019.

[6] M. Rowshan and E. Viterbo, "List viterbi decoding of PAC codes," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2428–2435, Mar. 2021.

[7] M. Rowshan, A. Burg, and E. Viterbo, "Polarization-adjusted convolutional (PAC) codes: Sequential decoding vs list decoding," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1434–1447, Feb. 2021.

[8] H. Yao, A. Fazeli, and A. Vardy, "List decoding of arikan's PAC codes," *CoRR*, vol. abs/2005.13711, 2020. [Online]. Available: https://arxiv.org/abs/2005.13711

[9] M. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[10] K. R. Duffy, J. Li, and M. Medard, "Capacity-achieving guessing random additive noise decoding," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4023–4040, Jul. 2019.

[11] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[12] M. Bloch and J. Barros, *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge University Press, Oct. 2011.

[13] G. Brassard and L. Salvail, "Secret-key reconciliation by public discussion," in *Advances in Cryptology-Eurocrypt'93*, T. Helleseth, Ed. Springer-Verlag, 1993, pp. 411–423.

[14] H. Hentila, Y. Y. Shkel, and V. Koivunen, "Secret key generation using short blocklength polar coding over wireless channels," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 1, pp. 144–157, Jan. 2022.

[15] J. Martinez-Mateo, D. Elkouss, and V. Martin, "Key reconciliation for high performance quantum key distribution," *Scientif Reports*, vol. 3, p. 1576, Apr. 2013.

[16] A. Wyner, "Recent results in the shannon theory," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.

[17] N. Ghaddar, S. Ganguly, L. Wang, and Y.-H. Kim, "A lego-brick approach to coding for asymmetric channels and channels with state," in *Proc. of IEEE International Symposium on Information Theory*, Melbourne, Australia, Jul. 2021.

[18] E. Arikan, "Source polarization," in *Proc. of IEEE International Symposium on Information Theory*, Austin, TX, Jun. 2010, pp. 899–903.

[19] M. Yassaee, M. Aref, and A. Gohari, "Achievability proof via output statistics of random binning," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 6760–6786, Nov. 2014.

[20] M. Mondelli, S. H. Hassani, I. Sason, and R. L. Urbanke, "Achieving marton's region for broadcast channels using polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 783–800, Feb. 2015.

[21] R. A. Chou and M. R. Bloch, "Polar coding for the broadcast channel with confidential messages: A random binning analogy," *IEEE Transactions on Information Theory*, vol. 62, no. 5, pp. 2410–2429, May 2016.

[22] S. B. Korada and R. L. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1751–1768, 2010.

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

[24] B. Li, H. Shen, and D. Tse, "A RM-polar codes," *CoRR*, vol. abs/1407.5483, 2014. [Online]. Available: http://arxiv.org/abs/1407.5483

[25] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[26] W. Liu, L. Chen, and X. Liu, "A weighted sum based construction of PAC codes," *IEEE Communications Letters*, vol. 27, no. 1, pp. 28–31, 2023.

[27] M. Rowshan and E. Viterbo, "On convolutional precoding in PAC codes," in *2021 IEEE Globecom Workshops*, 2021, pp. 1–6.

[28] M. Moradi, "On sequential decoding metric function of polarization-adjusted convolutional (PAC) codes," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 7913–7922, Dec. 2021.

[29] P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *2013 IEEE Information Theory Workshop*, Seville, Spain, Sep. 2013.

[30] P. Trifonov, "Randomized polar subcodes with optimized error coefficient," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6714–6722, Nov. 2020.

[31] T. Erseghe, "Coding in the finite-blocklength regime: Bounds based on laplace integrals and their asymptotic approximations," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 6854–6883, 2016.

[32] R. A. Chou, M. R. Bloch, and E. Abbe, "Polar coding for secret-key generation," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 6213–6237, Nov. 2015.

[33] H. Hentila, Y. Y. Shkel, V. Koivunen, and H. V. Poor, "On polar coding for finite blocklength secret key generation over wireless channels," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Barcelona, Spain, May 2020.