Polar Codes for Channel Simulation

Sharang M. Sriramu
School of ECE
Cornell University
Ithaca, NY 14853
sms579@cornell.edu

Rochelle Barsz
School of ECE
Cornell University
Ithaca, NY 14853
rsb359@cornell.edu

Elizabeth Polito
School of ECE
Cornell University
Ithaca, NY 14853
emp234@cornell.edu

Aaron B. Wagner
School of ECE
Cornell University
Ithaca, NY 14853
wagner@cornell.edu

Abstract

We consider the design of practically-implementable schemes for the task of channel simulation. Existing methods do not scale with the number of simultaneous uses of the channel and are therefore unable to harness the amortization gains associated with simulating many uses of the channel at once. We propose a new scheme that uses polar codes to efficiently simulate i.i.d. copies of a class of binary-output channels.

1 Introduction

Channel simulation refers to the task in which Alice observes a realization x of a random variable X and sends a bit string to Bob. Bob, who shares common randomness with Alice, outputs a random variable Y using both the message and the common randomness. The goal is to minimize the length of the bit string subject to the constraint that Y should have a specified distribution $P_{Y|X}(\cdot|x)$. This problem can be viewed as a "soft" or "stochastic" generalization of quantization. As with quantization, Y has a digital representation (through Alice's message), and it can be viewed as a degraded version of X. The difference is that here the degrading process is stochastic in general. In fact, the quantization problem is subsumed by taking the channel $P_{Y|X}$ to be deterministic.

This stochastic generalization of quantization arises in lossy compression of various types of sources including images [Flamich et al., 2020, Ballé et al., 2020], models [Havasi et al., 2018], and gradients [Shah et al., 2022]. In these applications, X often represents a vector of latent variables, model weights, or even an image consisting of millions of pixels [Theis et al., 2022]. The channel of interest is therefore high dimensional, and it is usually independent across the dimensions. For conventional quantization, it has long been recognized that the optimum rate-distortion tradeoff is more favorable in higher dimensions [Cover and Thomas, 2006], a trend that one expects to generalize to channel simulation. Indeed, let $n \cdot R_n$ be the minimum number of bits required to generate $Y^n = (Y_1, \ldots, Y_n)$, when X^n is i.i.d. and Y^n is conditionally i.i.d. given X^n. Thus R_n is the minimum number of bits per dimension when simulating the channel n times. The sequence nR_n can be shown to be *subadditive* and therefore satisfies (e.g., Liggett [1999, Thm. B22])

$$\lim_{n\to\infty} R_n = \inf_n R_n. \tag{1}$$

It is known that R_1 satisfies [Li and El Gamal, 2018]

$$I(X;Y) \leq R_1 \leq I(X;Y) + \log(I(X;Y)+1) + 5, \tag{2}$$

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

where $I(X;Y)$ refers to conventional Shannon mutual information. Applying this to i.i.d. (X^n, Y^n) and using the fact that $I(X^n;Y^n) = nI(X;Y)$, we have

$$nI(X;Y) \leq n \cdot R_n \leq nI(X;Y) + \log(nI(X;Y) + 1) + 5, \tag{3}$$

which shows that as $n \to \infty$, R_n approaches the lower bound $I(X;Y)$. The challenge, for both quantization and channel simulation, is that the complexity of schemes tends to grow exponentially in n. In fact, although many channel simulation schemes have been proposed [Harsha et al., 2007], [Li and El Gamal, 2018], [Flamich et al., 2022], [Flamich and Theis, 2023], [Flamich et al., 2024], none have complexity that scales subexponentially in n, ignoring isolated examples for which R_1 happens to equal $I(X;Y)$ [Zamir and Feder, 1992], [Agustsson and Theis, 2020].

Vector quantization has long been recognized to be the dual, in a precise sense, of channel coding [Pradhan et al., 2003]. In channel coding, the decoder maps an arbitrary point to an element of a finite set that is "close" in some channel-dependent sense. This is analogous to the role of the encoder in quantization. Likewise the encoder in channel coding is analogous to the decoder in quantization: both map bit strings to elements of said discrete set. Thus new techniques for channel coding can often be applied to vector quantization [Goblick, 1963], [Viterbi and Omura, 1974] and vice versa [Laroia et al., 1994].

The goal of this paper is to demonstrate how ideas from coding theory can likewise be applied to the channel simulation problem. We shall see that by adopting these techniques, we can develop schemes that significantly outperform state-of-the art methods, both in terms of their scalability and their rate performance. Specifically, we show how *polar codes* [Arikan, 2009] can be applied to the simulation problem using a method called `PolarSim`. Polar codes make for a good exemplar of this general proposal for five reasons. First, they have excellent channel coding performance, both theoretically [Mondelli et al., 2016] and practically [Egilmez et al., 2019]. Second, their complexity scales as $n \log n$. Third, they require no manual tuning. Fourth, they are simple to describe, requiring minimal background in coding theory. Finally, there exist highly optimized implementations of the encoding and decoding algorithms (e.g., [Pfister, 2023]). Their limitation is that, in their basic form, they can only be applied to symmetric binary-input channels. As we shall see, this means that `PolarSim` can only simulate symmetric binary-output channels. This class includes, for example, the binary symmetric channel, the (reverse) binary erasure channel, and channels of the form $X \to \mathrm{signum}(X + Z)$, where X and Z are real-valued, independent random variables with symmetric distributions. Note that the input to the channel need not be binary and may even be continuous.

For these channels, we show both theoretically and experimentally that, by scaling up the dimension, the rate of `PolarSim` can be made to approach the mutual information lower bound $I(X;Y) \leq R_n$ from (3). The superior scalability of `PolarSim` thus translates to a significant rate improvement over the state-of-the-art, since those schemes are not able to harness the amortization gain associated with letting n grow. Although `PolarSim` is restricted to binary-output channels, it is worth noting that there are currently no known schemes that simulate any nontrivial class of channels with even subexponential complexity in n. Also, for compression applications, a binary output alphabet is not unreasonable. It should be emphasized that the binary-output restriction is particular to the basic form of polar codes, not error-correction methods in general. In the supplementary materials we discuss how a different coding technique, trellis coded modulation, can be applied to closely simulate a Gaussian channel. See also the discussion in the Concluding Remarks section on non-binary polar codes.

One lesson from the coding theory literature, especially with the advent of modern coding theory in the 1990s [Richardson and Urbanke, 2008], is that it is advantageous to prioritize scalability with the dimension n over achieving optimal performance for particular n. The reason is that performance naturally improves with increasing n (as in (1)-(3) above), and this improvement can overcome suboptimality at any given value of n. For state-of-the-art channel codes, the decoder typically does not implement the optimal (maximum likelihood) decision rule. Instead, it implements a scalable approximation to it. This design strategy of favoring scalability over optimality is now well established in coding theory and vector quantization, and our goal here is to show how it can be profitably applied to channel simulation.

1.1 Terminology and Notation

We follow the standard convention of denoting the dimensionality of vectors by their superscript. We will also denote compound i.i.d. channels by superscripts: $p^{\times n}$ denotes n i.i.d. copies of a distribution p. The number of copies here is referred to as the *block length* or *dimension* of the channel. All logarithms mentioned in this paper are base 2. The *binary entropy function* $h_B : [0, \frac{1}{2}] \mapsto [0, 1]$ is defined as $h_B(p) = -p \log p - (1-p) \log(1-p)$. We will also refer to its inverse $h_B^{-1} : [0, 1] \mapsto [0, \frac{1}{2}]$ defined such that $h_B^{-1}(h_B(p)) = p$.

1.2 The Channel Simulation Problem

Consider a joint probability measure p_{XY} on the set $\mathcal{X} \times \mathcal{Y}$. Alice receives a sequence of n symbols from the input alphabet $\mathcal{X}$ drawn according to $p_X^{\times n}$ and encodes it into a binary string that she transmits to Bob. Upon receiving the message from Alice, Bob then decodes it to generate a sample from the channel $p_{Y|X}^{\times n}$. The objective is to find coding schemes that minimize the average *rate*— i.e., the average amortized length of the bit string transmitted by Alice. We refer to n as the *block length* or the *dimension.* We require that the set of strings that Alice can transmit to Bob to form a *prefix-free* set, meaning that no string in the set is a prefix of any other. Alice's message is thus self-terminating, and schemes for block length m and n can be combined to obtain a scheme for block length $m + n$ by concatenation. If R_n denotes the minimum average rate, i.e., the minimum average length of Alice's string over all schemes, normalized by n, then nR_n is subadditive in n, as noted earlier.

Both Alice and Bob are permitted to use randomized strategies and are assumed to share a source of common randomness. Under this assumption, Li and El Gamal [2018] prove the performance bounds (2) and (3) above (see also Harsha et al. [2007]). For large n, Sriramu and Wagner [2024] improve upon this result for i.i.d. discrete memoryless channels, showing that the logarithmic redundancy term can be halved for some channels and eliminated for all others. While these schemes are nearly rate-optimal, their complexity scales exponentially in n. Other practical schemes have been proposed, although none have even subexponential scaling in n outside the small class of channels for which the lower bound in (3) is tight for all n.

1.3 Related Work

The achievability proof in Li and El Gamal [2018] inspired several practical schemes that exploit properties of the Poisson process and perform well at short block lengths [Flamich et al., 2022, Flamich, 2024]. Similarly, the rejection sampler proposed by Harsha et al. [2007] has been generalized by Flamich et al. [2024] to work for arbitrary probability spaces. None of these schemes exhibit subexpoential complexity with n when applied to product channels, however. If one restricts attention to $n = 1$ and unimodal distributions, then improved schemes are possible [Flamich et al., 2024, Hegazy and Li, 2022], although by their nature such schemes cannot harness the amortization gain associated with increasing n.

Chou et al. [2018] addresses the problem of total variation approximate channel simulation and proposes a fixed rate scheme based on a soft covering argument that uses polar codes.

As noted in the introduction, the primary application of the channel simulation task is learned compression. Lei et al. [2024] and Li et al. [2020] consider how vector quantization methods can be directly applied to the compression task without explicitly simulating a channel. Although their methods do not simulate an i.i.d. channel, their use of ideas from error-correcting codes and vector quantization makes them the closest prior works to the present paper along with Chou et al. [2018].

2 Proposed Scheme

We begin by describing a "toy scheme" for simulating binary output channels. This is not a rate-efficient scheme in its own right, but it serves as a foundation for one.

2.1 Toy Scheme for Binary Output Channel Simulation

Consider a joint distribution p_{XY} where p_Y is $\text{Bernoulli}\left(\frac{1}{2}\right)$. Then, the following algorithm simulates $p_{Y|X}$ exactly:

1. Use the common randomness to generate $Z \sim \mathrm{Unif}(0,1)$ and $V = \mathbf{1}\left(Z > \frac{1}{2}\right)$ at both the encoder and decoder.
2. At the encoder, having observed an input realization x, compute the output bit $Y = \mathbf{1}\left(Z > p_{Y|X}(0|x)\right)$ and the correction bit $\Delta = Y \oplus V$.
3. Transmit Δ to the decoder after lossless compression.
4. Recover $Y = \Delta \oplus V$ at the decoder.

In Appendix A, we show that the rate associated with repeated application of this scheme is upper bounded by $h_B\left(\frac{1}{2} - h_B^{-1}(1 - I(X;Y))\right)$. As we can see in Figure 3, this is highly suboptimal in general. However, we note that for the special case in which the mutual information is *polarized*, i.e., where $I(X;Y) \approx 0$ or $I(X;Y) \approx 1$, the toy scheme is close to optimal.

This observation is crucial as it suggests a path forward: If we can transform a given channel simulation problem to the problem of simulating polarized channels, it can be solved rate-efficiently using the toy scheme. Polar codes provide us with the means to achieve such a transformation.

2.2 Channel Simulation Using Polar Codes

Let us examine the problem of simulating two independent realizations of a symmetric[1] binary output channel $p_{Y|X}$. Consider the following bijection applied to the output $Y^2 = (Y_1, Y_2)$:

$$U_1 = Y_1 \oplus Y_2 \tag{4}$$
$$U_2 = Y_2. \tag{5}$$

It is clear that simulating the original pair of i.i.d. channels $p_{Y_1|X_1}$ and $p_{Y_2|X_2}$ is equivalent to simulating the transformed pair of channels $p_{U_1|X_1,X_2}$ and $p_{U_2|X_1,X_2,U_1}$ sequentially. The mutual information of each of the original i.i.d. channels are equal to $I(X;Y)$. However, the two transformed channels differ in terms of mutual information: $I(X_1, X_2, U_1; U_2) > I(X;Y)$ because U_2 is observed through two different channels. This necessitates that the other channel has lower mutual information: $I(X_1, X_2; U_1) < I(X;Y)$.

Therefore, the linear transformation we applied to the output had the effect of *polarizing* the target channel. For larger blocklengths, we can apply this transform recursively: for larger block lengths , divide the channels into pairs and apply the two-dimensional polar transform we saw above. Then, collect pairs of transformed channels that have the same mutual information and repeat the procedure. This can be succinctly summarized by the following equation [Arikan, 2009]:

$$U^n = Y^n G_n^{-1}, \text{where} \tag{6}$$
$$G_n = B_n \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{\otimes n}, \text{ with } B_n \text{ being the } \textit{bit-reversal} \text{ permutation.} \tag{7}$$

As we saw in the two-dimensional problem, applying this transform leaves us with the problem of simulating the *subchannels* $p_{U_1|X^n}, p_{U_2|X^n,U_1}, \ldots, p_{U_n|X^n,U^{n-1}}$. Arikan [2009] shows that these subchannels are highly polarized for large n. For each i, $I(U_i; X^n, U^{i-1}) \approx 0$ or $I(U_i; X^n, U^{i-1}) \approx 1$. This allows us to simulate them using the toy algorithm described in the previous section.

Algorithms 1 and 2 describe the complete scheme. The encoder input $\bar{p}_i$ refers to the subchannel parameter

$$\bar{p}_i = h_B^{-1}(H(U_i|U^{i-1}, X^n)). \tag{8}$$

These can be calculated offline using the techniques used to compute subchannel quality for communication (See Tal and Vardy [2013], Zhang et al. [2014]) The `SoftPolarDec`(u^{i-1}, x^n) subroutine outputs

$$\Pr(U_i = 0 | U^{i-1} = u^{i-1}, X^n = x^n), \tag{9}$$

which can be calculated with $O(n \log n)$ complexity using a recursion given by Arikan [2009]. The `PolarEnc`(u^n) subroutine simply multiplies by the generator matrix in: $y^n = u^n G_n$. This can

[1] Specifically, we focus on joint distributions p_{XY} such that p_Y is Bernoulli $\left(\frac{1}{2}\right)$, and that there exists a self-inverting bijection $A : \mathcal{X} \mapsto \mathcal{X}$ with $p_{Y|X}(0|x) = p_{Y|X}(1|A(x))$ for all x.

be implemented in $O(n \log n)$ by exploiting the recursive structure. The $\texttt{Compress}(\Delta^n, \bar{p}^n)$ and $\texttt{Decompress}(b, \bar{p}^n)$ routines can be any lossless compressor/decompressor pair that uses at most

$$c + \sum_{i=1}^{n} \left[\mathbf{1}(\Delta_i = 1) \log \frac{1}{1/2 - \bar{p}_i} + \mathbf{1}(\Delta_i = 0) \log \frac{1}{1/2 + \bar{p}_i} \right] \tag{10}$$

bits to send Δ^n, where c is some constant independent of n and Δ^n. Arithmetic coding (Rissanen [1976]) is a widely-used scheme that achieves this guarantee with $c = 2$.

Algorithm 1: Encoder for simulating a channel using polar codes.

Input : Block length $n = 2^k$, $k > 0$
Random bit string $z^n \sim \text{Unif}([0,1]^n)$
Probability table $\bar{p}^n \in [0,1]^n$
Source string $x^n \in \mathcal{X}^n$
Output : string $b \in \{0,1\}^*$
for $i = 1, \ldots, n$ **do**
  if $z_i > \texttt{SoftPolarDec}(x^n, u^{i-1})$ **then** $u_i \leftarrow 1$ **else** $u_i \leftarrow 0$
  if $z_i > 1/2$ **then** $v_i \leftarrow 1$ **else** $v_i \leftarrow 0$
  $\Delta_i \leftarrow u_i + v_i$
$b \leftarrow \texttt{Compress}(\Delta^n, \bar{p}^n)$
return b

Algorithm 2: Decoder for simulating a channel using polar codes.

Input : Block length $n = 2^k$, $k > 0$
Random bit string $z^n \sim \text{Unif}([0,1]^n)$
Probability table $\bar{p}^n \in [0,1]^n$
Compressed offset string b
Output : Simulated channel output $y^n \in \{0,1\}^n$
$\Delta^n \leftarrow \texttt{Decompress}(b, \bar{p}^n)$
for $i = 1, \ldots, n$ **do**
  if $z_i > 1/2$ **then** $v_i \leftarrow 1$ **else** $v_i \leftarrow 0$
  $u_i \leftarrow \Delta_i + v_i$
$y^n \leftarrow \texttt{PolarEnc}(u^n)$
return y^n

If one uses a linear complexity algorithm for `Compress` and `Decompress`, then the overall complexity of `PolarSim` is $O(n \log n)$ for both encoding an decoding. Using the fact that polar codes achieve capacity for symmetric channels, we can show that `PolarSim` is rate optimal in the large-n limit, making it currently the only scheme with subexponential complexity in n with a comparable guarantee. Theorem 1 in the appendix formalizes these guarantees.

2.3 Experimental Results

We run `PolarSim` on the following three channels: (1) the *binary symmetric channel*, (BSC), defined by $X = Y \oplus Z$, where $Z \sim \text{Bernoulli}(p)$ for $p \in [0,1]$ BSC with a uniform input (2) the binary erasure channel, $X = Z \cdot Y$, where Y is uniform over $\{-1, 1\}$ and Z is Bernoulli(ϵ), and (3) the binary Gaussian channel $X = Y + Z$, where Y is again uniform over $\{-1, 1\}$ and Z is $\mathcal{N}(0, \sigma^2)$. Note that the reverse of the BSC with a uniform input is the BSC itself.

Fig. 1 shows the rate performance of these simulations. Even at a block length of 2^{12}, the performance is already close to the mutual information lower bound across all channels and rates. Performance improves with n as expected, with both the average rate and the variance in the rate decreasing.

We also compare our scheme to the state-of-the-art scheme for channel simulation, Greedy Poisson Rejection Sampling (GPRS) [Flamich, 2024] (see Appendix C for the implementation details). Fig. 2 shows that `PolarSim` significantly outperforms GPRS in terms of the communication rate, even when the latter is optimized for the channel at hand. Table 1 in Appendix D shows that its computational

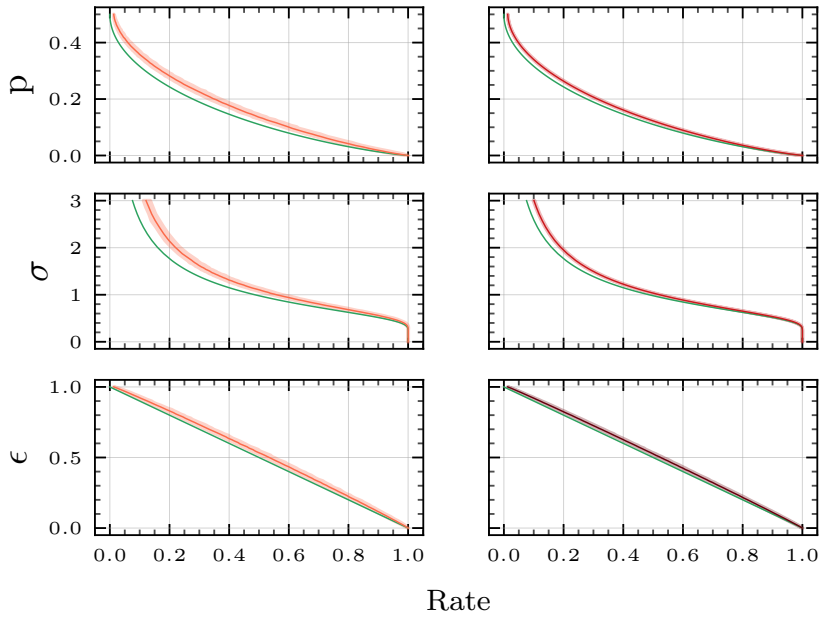

Figure 1: Rates achieved by `PolarSim` at different block lengths — ■ $n = 2^{12}$ (left), ■ $n = 2^{17}$ (top-right and middle-right), ■ $n = 2^{14}$ (bottom-right) for different noise levels across different channels, compared against the theoretical lower bound ■ $I(X;Y)$. **Top:** BSC_p for $p \in (0, \frac{1}{2})$, **Middle:** Gaussian for $\sigma^2 \in (0, 3)$, **Bottom:** Erasure for $\epsilon \in (0, 1)$. The lines represent the median values, and the boundaries of shaded regions represent the 5^{th} to 95^{th} percentile rates over 200 simulation runs.

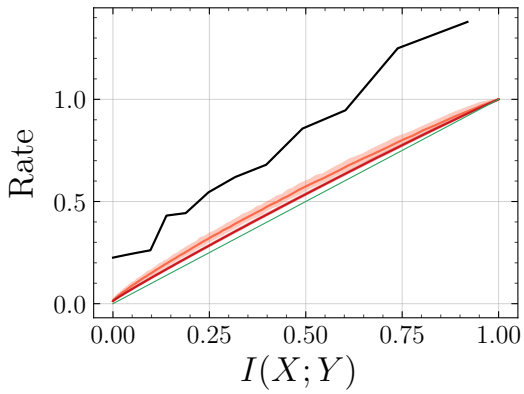

Figure 2: Comparison of schemes for BSC simulation: Average rates for `PolarSim` with ■ $n = 2^{12}$ and ■ $n = 2^{17}$ compared against ■ GPRS with $n = 8$ and the theoretical lower bound ■ $I(X;Y)$ over 1000 simulation runs.

efficiency is also significantly better, by several orders of magnitude. This is due to the exponential computational complexity of GPRS in n compared to the pseudolinear complexity of `PolarSim`.

References

Gergely Flamich, Marton Havasi, and José Miguel Hernández-Lobato. Compressing images by encoding their latent representations with relative entropy coding. *Advances in Neural Information Processing Systems*, 33:16131–16141, 2020.

Johannes Ballé, Philip A Chou, David Minnen, Saurabh Singh, Nick Johnston, Eirikur Agustsson, Sung Jin Hwang, and George Toderici. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020.

Marton Havasi, Robert Peharz, and José Miguel Hernández-Lobato. Minimal random code learning: Getting bits back from compressed model parameters. *International Conference on Learning Representations*, 2018.

Abhin Shah, Wei-Ning Chen, Johannes Ballé, Peter Kairouz, and Lucas Theis. Optimal compression of locally differentially private mechanisms. *arXiv preprint arXiv:2111.00092*, 2022.

Lucas Theis, Tim Salimans, Matthew D. Hoffman, and Fabian Mentzer. Lossy compression with Gaussian diffusion. *arXiv preprint arXiv:2206.08889*, 2022.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken, 2nd edition, 2006.

Thomas M. Liggett. *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes*. Springer-Verlag, New York, 1999.

Cheuk Ting Li and Abbas El Gamal. Strong functional representation lemma and applications to coding theorems. *IEEE Transactions on Information Theory*, 64(11):6967–6978, 2018.

Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 10–23. IEEE, 2007.

Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Fast relative entropy coding with A* coding. In *International Conference on Machine Learning*, pages 6548–6577. PMLR, 2022.

Gergely Flamich and Lucas Theis. Adaptive greedy rejection sampling. *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 454–459, 2023.

Gergely Flamich, Stratis Markou, and José Miguel Hernández-Lobato. Faster relative entropy coding with greedy rejection coding. *Advances in Neural Information Processing Systems*, 36, 2024.

Ram Zamir and Meir Feder. On universal quantization by randomized uniform/lattice quantizers. *IEEE Transactions on Information Theory*, 38(2):428–436, 1992.

Eirikur Agustsson and Lucas Theis. Universally quantized neural compression. *Advances in Neural Information Processing Systems*, 33:12367–12376, 2020.

S.S. Pradhan, J. Chou, and K. Ramchandran. Duality between source coding and channel coding and its extension to the side information case. *IEEE Transactions on Information Theory*, 49(5): 1181–1203, 2003. doi: 10.1109/TIT.2003.810622.

Thomas John Goblick. *Coding for a discrete information source with a distortion measure*. PhD thesis, Massachusetts Institute of Technology, 1963.

A Viterbi and J Omura. Trellis encoding of memoryless discrete-time sources with a fidelity criterion. *IEEE Transactions on Information Theory*, 20(3):325–332, 1974.

Rajiv Laroia, Nariman Farvardin, and Steven A Tretter. On optimal shaping of multidimensional constellations. *IEEE Transactions on Information Theory*, 40(4):1044–1056, 1994.

Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7): 3051–3073, 2009.

Marco Mondelli, S Hamed Hassani, and Rüdiger L Urbanke. Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors. *IEEE Transactions on Information Theory*, 62(12):6698–6712, 2016.

Zeynep B Kaykac Egilmez, Luping Xiang, Robert G Maunder, and Lajos Hanzo. The development, operation and performance of the 5g polar codes. *IEEE Communications Surveys & Tutorials*, 22 (1):96–122, 2019.

Henry Pfister. polar_intro. `https://github.com/henrypfister/polar_intro`, 2023.

Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

Sharang M Sriramu and Aaron B Wagner. Optimal redundancy in exact channel synthesis. *arXiv preprint arXiv:2401.16707*, 2024.

Gergely Flamich. Greedy Poisson rejection sampling. *Advances in Neural Information Processing Systems*, 36, 2024.

Mahmoud Hegazy and Cheuk Ting Li. Randomized quantization with exact error distribution. In *IEEE Information Theory Workshop (ITW)*, pages 350–355, 2022.

Rémi A Chou, Matthieu R Bloch, and Jörg Kliewer. Empirical and strong coordination via soft covering with polar codes. *IEEE Transactions on Information Theory*, 64(7):5087–5100, 2018.

Eric Lei, Hamed Hassani, and Shirin Saeedi Bidokhti. Approaching rate-distortion limits in neural compression with lattice transform coding. *arXiv preprint arXiv:2403.07320*, 2024.

Binglin Li, Mohammad Akbari, Jie Liang, and Yang Wang. Deep learning-based image compression with trellis coded quantization. In *Data Compression Conference (DCC)*, pages 13–22, 2020.

Ido Tal and Alexander Vardy. How to construct polar codes. *IEEE Transactions on Information Theory*, 59(10):6562–6582, 2013. doi: 10.1109/TIT.2013.2272694.

Yingxian Zhang, Aijun Liu, Kegang Pan, Chao Gong, and Sixiang Yang. A practical construction method for polar codes. *IEEE Communications Letters*, 18(11):1871–1874, 2014.

Jorma J Rissanen. Generalized Kraft inequality and arithmetic coding. *IBM Journal of research and development*, 20(3):198–203, 1976.

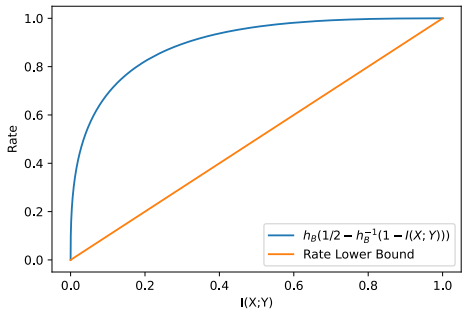

Figure 3: The upper bound on the rate of the toy scheme described in section 2.1 is plotted against the mutual information lower bound.

A Rate of the Toy Scheme

The rate of this scheme is governed by the cost of compressing the correction bit Δ. We have

$$\Pr(\Delta = 1) = E\left[\Pr(\Delta = 1|X)\right] \tag{11}$$

$$= E\left[\frac{1}{2} - \min_{j\in\{0,1\}} p_{Y|X}(j|X)\right] \tag{12}$$

$$= \frac{1}{2} - E\left[\min_{j\in\{0,1\}} p_{Y|X}(j|X)\right] \tag{13}$$

$$\implies H(\Delta) = h_B\left(\frac{1}{2} - E\left[\min_{j\in\{0,1\}} p_{Y|X}(j|X)\right]\right). \tag{14}$$

Rearranging this, we can obtain

$$E\left[\min_{j\in\{0,1\}} p_{Y|X}(j|X)\right] = \frac{1}{2} - h_B^{-1}(H(\Delta)). \tag{15}$$

Next, consider

$$I(X;Y) = 1 - H(Y|X) \tag{16}$$

$$= 1 - E\left[h_B\left(\min_{j\in\{0,1\}} p_{Y|X}(j|X)\right)\right] \tag{17}$$

$$\geq 1 - h_B\left(E\left[\min_{j\in\{0,1\}} p_{Y|X}(j|X)\right]\right) \tag{18}$$

$$= 1 - h_B\left(\frac{1}{2} - h_B^{-1}(H(\Delta))\right). \tag{19}$$

Rearranging this, we finally obtain the required upper bound:

$$H(\Delta) \leq h_B\left(\frac{1}{2} - h_B^{-1}(1 - I(X;Y))\right). \tag{20}$$

B Theoretical Guarantees

Theorem 1. *Consider a symmetric binary output joint distribution P_{XY}. Suppose* `Compress` *and* `Decompress` *achieve the guarantee in (10).*

1. *(*Correctness:*) Algorithms 1 and 2 simulate the channel $P^{\times n}(Y|X)$ exactly: If Z^n is i.i.d.* $\mathrm{Unif}[0,1]$*, and $\bar{p}_i = h_B^{-1}(H(U_i|U^{i-1}, X^n))$, then the conditional probability that Algorithm 2 outputs y^n given that x^n is the input to Algorithm 1 is*

$$\prod_{i=1}^{n} P_{Y|X}(y_i|x_i). \tag{21}$$

2. *(*Optimality:*) Algorithms 1 and 2 are asymptotically rate optimal:*

$$\lim_{n\to\infty} \frac{1}{n} E[\ell(b)] \to I(X;Y), \tag{22}$$

where b is the output of the encoder.

Proof. Suppose n is a power of two, and consider the following joint distribution between (U^n, X^n, Y^n):

$$\Pr(U^n = u^n, Y^n = y^n, X^n = x^n) = \prod_{i=1}^{n} P_X(x_i) \prod_{i=1}^{n} P_{Y|X}(y_i|x_i) 1(u^n = y^n G_n^{-1}) \tag{23}$$

$$= \frac{1}{2^n} 1(y^n = u^n G_n) \prod_{i=1}^{n} P_{X|Y}(x_i|y_i). \tag{24}$$

Given the input X^n, the U^n string generated by the encoder has distribution $P_{U|X}^{\times n}$ by construction, since `SoftPolarDec`$(x^n, u^{i-1}, P_{X|Y})$ computes $\Pr(U_i = 0|X^n = x^n, U^{i-1} = u^{i-1})$ under the distribution in (23)-(24). Due to the lossless nature of the compression, Δ^n is recovered exactly at the decoder. Since V^n is common to the two terminals, U^n is thus recovered exactly by the decoder. Then setting $Y^n = U^n \cdot G_n$ results in (X^n, Y^n) having the joint distribution $P_{XY}^{\times n}$ as desired. This establishes the correctness of the algorithm for n that is a power of two. Correctness of the algorithm for any n immediately follows.

Turning to optimality, again suppose n is a power of two. For the sequence Δ^n we have

$$\Pr(\Delta_i = 1) = \frac{1}{2} - \sum_{u^{i-1}, x^n} P_{U^{i-1}, X^n}(u^{i-1}, x^n) \min\Big(\Pr(U_i = 1|U^{i-1} = u^{i-1}, X^n = x^n), \tag{25}$$

$$\Pr(U_i = 0|U^{i-1} = u^{i-1}, X^n = x^n)\Big) \tag{26}$$

$$= \frac{1}{2} - \sum_{u^{i-1}, x^n} P_{U^{i-1}, X^n}(u^{i-1}, x^n) h_B^{-1}(H(U_i|U^{i-1} = u^{i-1}, X^n = x^n)) \tag{27}$$

$$\leq \frac{1}{2} - h_B^{-1}(H(U_i|U^{i-1}, X^n)) \tag{28}$$

$$= \frac{1}{2} - \bar{p}_i, \tag{29}$$

where the inequality follows by the convexity of $h_B^{-1}(\cdot)$. Thus the average rate may be bounded as

$$\frac{1}{n} \mathrm{E}[\ell(b)] \leq \frac{1}{n} \sum_{i=1}^{n} \left[\Pr(\Delta_i = 1) \log \frac{1}{1/2 - \bar{p}_i} + \Pr(\Delta_i = 0) \log \frac{1}{1/2 + \bar{p}_i} \right] + \frac{c}{n} \tag{30}$$

$$\leq \frac{1}{n} \sum_{i=1}^{n} h_B\left(\frac{1}{2} - \bar{p}_i\right) + \frac{c}{n}. \tag{31}$$

We bound this quantity using the polarization property. Fix $\delta > 0$. By Arikan [2009], for all sufficiently large n there is a set $\mathcal{N}_n$ of "noisy" indices such that

$$\frac{|\mathcal{N}_n|}{n} \geq 1 - I(X;Y) - \delta \tag{32}$$

$$h_B(1/2 - \bar{p}_i) \leq \delta \quad \text{for all } i \in \mathcal{N}_n. \tag{33}$$

The rate is thus upper bounded by

$$\frac{\mathrm{E}[\ell(b)]}{n} \leq \frac{n - |\mathcal{N}_n|}{n} + \frac{1}{n} \sum_{i \in \mathcal{N}_n} h_B(1/2 - \bar{p}_i) + \frac{c}{n} \tag{34}$$

$$\leq \frac{n - |\mathcal{N}_n|}{n} + \frac{\delta \cdot |\mathcal{N}_n|}{n} + \frac{c}{n} \tag{35}$$

$$\leq I(X;Y) + 2\delta + \frac{c}{n}. \tag{36}$$

If we let $\bar{R}_n = \mathrm{E}[\ell(b)]/n$ denote the minimum rate of `PolarSim` at block length n (minimized over all partitions if it is not a power of two), then $n\bar{R}_n$ is itself subadditive. As n increases through powers of two, from (36) we have

$$\bar{R}_n \to I(X;Y). \tag{37}$$

and thus

$$\inf_n \bar{R}_n = I(X;Y). \tag{38}$$

The result then follows by subadditivity (cf. 1). □

C GPRS Implementation Details

We implement Algorithm 3 in Flamich [2024]. The proposal distribution P is chosen to be i.i.d. Bernoulli$(1/2)$ with $n = 8$. Given the input $X^n = x^n$, the target distribution $Q(y^n)$ is chosen to be $\prod_{i=1}^n \mathsf{BSC}_p(y_i|x_i)$, where p ranges over $(0, 1/2)$. The stretch function σ was derived using the definitions provided in [Flamich 2023]:

$$\begin{aligned}
w_P(h) &= F\left(\frac{\log h - n(\log(1-p)+1)}{\log p - \log(1-p)}, n, \frac{1}{2}\right), \\
w_Q(h) &= F\left(\frac{\log h - n(\log(1-p)+1)}{\log p - \log(1-p)}, n, p\right), \text{ and} \\
\sigma(h) &= \int_0^h \frac{1}{w_Q(\eta) - \eta w_P(\eta)} d\eta,
\end{aligned}$$

where $F(\cdot, n, p)$ is the CDF of a Binomial(n, p) random variable. This stretch function was evaluated numerically for each input. The algorithm outputs a positive integer n, which is entropy coded using the Zeta distribution in [Flamich 2023, (151)]. The number of bits is divided by $n = 8$ to obtain the rate.

We also make use of the fact that the selection rule employed by the algorithm needs to be evaluated for each point in the output sample space only once — if the first occurrence of an output sequence in the randomly generated codebook is rejected, all its subsequent occurrences are also rejected. This improvement helps speed up the execution significantly and also reduces the rate as repetitions need not be indexed.

D Execution Time Comparison Table

p	`PolarSim`			**GPRS**			λ
	Median	**p5**	**p95**	**Median**	**p5**	**p95**	
0.00	0.009	0.008	0.01	192.0	161.4	205.8	21333
0.25	0.009	0.008	0.01	246.1	180.7	415.7	27347
0.49	0.009	0.008	0.01	227.7	176.1	318.7	25305

Table 1: Execution time comparison between `PolarSim` and GPRS, for simulating BSC's with block length $n = 2^{12}$. The reported statistics are computed over 1000 trials for each value of the crossover probability p. GPRS cannot directly simulate such large block lengths. Therefore, the GPRS runtimes are obtained by scaling up the runtime for $n = 8$ blocks. This is justified by subadditivity (see (1)). The column λ computes the ratio between the medians of the two schemes. For our chosen block length, `PolarSim` performs over four orders of magnitude faster than GPRS.