Algorithmic Polarization for Hidden Markov Models

Venkatesan Guruswami[1]
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
venkatg@cs.cmu.edu

Preetum Nakkiran[2]
Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA
preetum@cs.harvard.edu

Madhu Sudan[3]
Harvard John A. Paulson School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA
madhu@cs.harvard.edu

Abstract

Using a mild variant of polar codes we design linear compression schemes compressing Hidden Markov sources (where the source is a Markov chain, but whose state is not necessarily observable from its output), and to decode from Hidden Markov channels (where the channel has a state and the error introduced depends on the state). We give the first polynomial time algorithms that manage to compress and decompress (or encode and decode) at input lengths that are polynomial *both* in the gap to capacity and the mixing time of the Markov chain. Prior work achieved capacity only asymptotically in the limit of large lengths, and polynomial bounds were not available with respect to either the gap to capacity or mixing time. Our results operate in the setting where the source (or the channel) is *known*. If the source is *unknown* then compression at such short lengths would lead to effective algorithms for learning parity with noise – thus our results are the first to suggest a separation between the complexity of the problem when the source is known versus when it is unknown.

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases polar codes, error-correcting codes, compression, hidden markov model

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.39

1 Introduction

We study the problem of designing coding schemes, specifically encoding and decoding algorithms, that overcome errors caused by stochastic, but not memoryless, channels. Specifically we consider the class of "(hidden) Markov channels" that are stateful, with the states evolving according to some Markov process, and where the distribution of error depends on

[1] Most of this work was done when the author was visiting the Center for Mathematical Sciences and Applications, Harvard University, Cambridge, MA. Research supported in part by NSF grants CCF-1422045 and CCF-1814603.
[2] Work supported in part by the NSF Graduate Research Fellowship Grant No. DGE1144152, and Madhu Sudan's Simons Investigator Award and NSF Award CCF 1715187.
[3] Work supported in part by a Simons Investigator Award and NSF Award CCF 1715187.

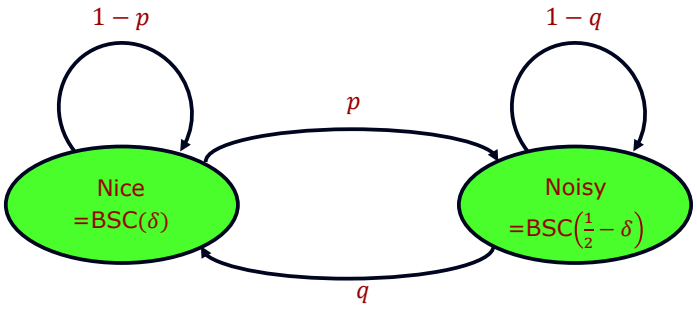

Figure 1 A Markovian Channel: The Nice state flips bits with probability δ whereas the Noisy state flips with probability $1/2 - \delta$. The stationary probability of the Nice state is q/p times that of the Noisy state.

the state.[4] Such Markovian models capture many natural settings of error, such as bursty error models. (See for example, Figure 1.) Yet they are often less understood than their memoryless counterparts (or even "explicit Markov models" where the state is completely determined by the actions of the channel). For instance (though this is not relevant to our work) even the capacity of such channels is not known to have a closed form expression in terms of channel parameters. (In particular the exact capacity of the channel in Figure 1 is not known as a function of δ, p and q!)

In this work we aim to design coding schemes that achieve rates arbitrarily close to capacity. Specifically given a channel of capacity C and gap parameter $\varepsilon > 0$, we would like to design codes that achieve a rate of at least $C - \varepsilon$, that admit polynomial time algorithms even at small block lengths $n \geq \text{poly}(1/\varepsilon)$. Even for the memoryless case such coding schemes were not known till recently. In 2008, Arikan [1] invented a completely novel approach to constructing codes based on "channel polarization" for communication on binary-input memoryless channels, and proved that they enable achieving capacity in the limit of large code lengths with near-linear complexity encoding and decoding. In 2013, independent works by Guruswami and Xia [5] and Hassani et al. [6] gave a finite-length analysis of Arikan's polar codes, proving that they approach capacity fast, at block lengths bounded by $\text{poly}(1/\varepsilon)$ where $\varepsilon > 0$ is the difference between the channel capacity and code rate.

The success of polar codes on the memoryless channels might lead to the hope that maybe these codes, or some variants, might lead to similar coding schemes for channels with memory. But such a hope is not easily justified: the analysis of polar codes relies heavily on the fact that errors introduced by the channel are independent and this is exactly what is not true for channels with memory. Despite this seemingly insurmountable barrier, Şaşoğlu [4] and later Şaşoğlu and Tal [9] showed, quite surprisingly, that the analysis of polar codes can be carried out even with Markovian channels (and potentially even broader classes of channels). Specifically they show that these codes converge to capacity and even the probability of decoding error, under maximum likelihood decoding, drops exponentially fast in the block length (specifically as $2^{-n^{\Omega(1)}}$ on codes of length n; see also [10], where exponentially fast polarization was also shown at the high entropy end). An extension of Arikan's successive cancellation decoder from the memoryless case was also given by [12], building on an earlier version [13] specific to intersymbol interference channels, leading to efficient decoding algorithms.

[4] We use the term *hidden* to emphasize the fact that the state itself is not directly observable from the actions of the channel, though in the interest of succinctness we will omit this term for most of the rest of this section.

However, none of the works above give small bounds on the block length of the codes as a function of the gap to capacity, and more centrally to this work, on the mixing time of the Markov chain. The latter issue gains importance when we turn to the issue of "compressing Markov sources" which turns out to be an intimately related task to that of error-correction for Markov channels as we elaborate below and which is also the central task we turn to in this paper. We start by describing Markov source and the (linear) compression problem.

A (hidden) Markov source over alphabet Σ is given by a Markov chain on some finite state space where each state s has an associated distribution D_s over Σ. The source produces information by performing a walk on the chain and at each time step t, outputting a letter of Σ drawn according to the distribution associated with the state at time t (independent of all previous choices, and previous states).[5] In the special case of additive Markovian channels where the output of the channel is the sum of the transmitted word with an error vector produced by a Markov source, a well-known correspondence shows that error-correction for the additive Markov channel reduces to the task of designing a compression and decompression algorithm for Markovian sources, with the compression being *linear*. Indeed in this paper we only focus on this task: our goal turns into that of compressing n bits generated by the source to its entropy upto an additive factor of εn, while n is only polynomially large in $1/\varepsilon$.

A central issue in the task of compressing a source is whether the source is *known* to the compression algorithm or not. While ostensibly the problem should be easier in the "known" setting than in the "unknown" one, we are not aware of any formal results suggesting a difference in complexity. It turns out that compression in the setting where the source is unknown is at least as hard as "learning parity with noise" (we argue this in Appendix B), *if* the compression works at lengths polynomial in the mixing time and gap to capacity. This suggests that the unknown source setting is hard (under some current beliefs). No corresponding hardness was known for the task of compressing sources when they are known, but no easiness result seems to have been known either (and certainly no linear compression algorithm was known). This leads to the main question addressed (positively) in this work.

Our Results

Our main result is a construction of codes for *additive Markov channels* that gets ε close to capacity at block lengths polynomial in $1/\varepsilon$ and the mixing time of the Markov chain, with polynomial (in fact near-linear) encoding and decoding time. Informally additive channels are those that map inputs from some alphabet Σ to outputs over Σ with an abelian group defined on Σ and the channel generates an error sequence independent of the input sequence, and the output of the channel is just the coordinatewise sum of the input sequence with the error sequence. (In our case the alphabet Σ is a finite field of prime cardinality.) The exact class of channels is described in Definition 4, and Theorem 10 states our result formally. We stress that we work with additive channels only for conceptual simplicity and that our results should extend to more general symmetric channels though we don't do so here. Prior to this work no non-trivial Markov channel was known to achieve efficient encoding and decoding at block lengths polynomial in either parameter (gap to capacity or mixing time).

Our construction and analyses turn out to be relatively simple given the works of Şaşoğlu and Tal [4, 9] and the work of Blasiok et al. [2]. The former provides insights on how to work with channels with memory, whereas the latter provides tools needed to get short block

[5] The phrase "hidden" emphasizes the fact that the output produced by the source does not necessarily reveal the sequence of states visited.

length and cleaner abstractions of the efficient decoding algorithm that enable us to apply it in our setting. Our codes are a slight variant of polar codes, where we apply the polar transforms independently to blocks of inputs. This enables us to apply the analysis of [2] in an essentially black box manner, benefiting both from its polynomially fast convergence guarantee to capacity as well as its generality covering all polarizing matrices over any prime alphabet (and not just the basic Boolean 2×2 transform covered in [9]).

We give a more detailed summary of how our codes are obtained and how we analyze them in Section 3 after stating our results and main theorem formally.

2 Definitions and Main Results

2.1 Notation and Definitions

We will use $\mathbb{F}_q$ to denote the finite field with q elements. Throughout the paper, we will deal only with the case when q is a prime. (This restriction in turn comes from the work of [2] whose results we use here.)

We use several notations to index matrices. For a matrix $M \in \mathbb{F}_q^{m \times n}$, the entry in the ith row, jth column is denoted $M_{i,j}$ or $M_{(i,j)}$. Columns are denoted by superscripts, i.e., $M^j \in \mathbb{F}_q^m$ denotes the jth column of M. Note that $M_i^j = M_{(i,j)}$. We also use the indices as sets in the natural way. For example $M^{\leq j} \in \mathbb{F}_q^{m \times j}$ denotes the first j columns of M. $M_{\leq i}^{\leq j}$ denotes the submatrix of elements in the first j columns and first i rows. $M_{\prec(i,j)}$ denotes the set of elements of M indexed by lexicographically smaller indices than (i, j). Multiplication of a matrix $M \in \mathbb{F}_q^{m \times n}$ with a vector $v \in \mathbb{F}_q^n$ is denoted Mv.

For a finite set S, let $\Delta(S)$ denote the set of probability distributions over S. For a random variable X and event E, we write $X|E$ to denote the conditional distribution of X, conditioned on E. For example, we may write $X|\{X_1 = 0\}$.

The *total-variation distance* between two distributions $p, q \in \Delta(U)$ is

$$||p - q||_1 := \sum_i |p(i) - q(i)|$$

We consider compression schemes, as a map $\mathbb{F}_q^n \to \mathbb{F}_q^m$. The *rate* of a compression scheme $\mathbb{F}_q^n \to \mathbb{F}_q^m$ is the ratio m/n.

For a random variable $X \in [q]$, the *(non-normalized) entropy* is denoted $H(X)$, and is

$$H(X) := -\sum_i \Pr[X = i] \log(\Pr[X = i])$$

and the *normalized entropy* is denoted $\overline{H}(X)$, and is

$$\overline{H}(X) := \frac{1}{\log(q)} H(X)$$

▶ **Definition 1.** A *Markov chain* $\mathcal{M} = (\ell, \Pi, \pi_0)$ is given by an ℓ representing the state space $[\ell]$, a transition matrix $\Pi \in \mathbb{R}^{\ell \times \ell}$, and a distribution on initial state $\pi_0 \in \Delta([\ell])$. The rows of Π, denoted $\Pi_1, \ldots, \Pi_\ell$ are thus elements of $\Delta([\ell])$. A Markov chain generates a random sequence of states $X_0, X_1, X_2, \ldots$ determined by letting $X_0 \sim \pi_0$, and $X_t \sim \Pi_{X_{t-1}}$ for $t > 0$ given $X_0, \ldots, X_{t-1}$. The stationary distribution $\pi \in \Delta([\ell])$ is the distribution such that if $X_0 \sim \pi$, then all X_t's are marginally identically distributed as π.

We consider only Markov chains which are irreducible and aperiodic, and hence have a stationary distribution to which they converge in the limit. The rate of convergence is measured by the mixing time, defined below.

▶ **Definition 2.** The *mixing time* of a Markov chain is the constant $\tau > 0$ such that for every initial state s_0 of the Markov chain, the distribution of state s_ℓ is $\exp(-\ell/\tau)$-close in total variation distance to the stationary distribution π.

▶ **Definition 3.** A *(stationary, hidden) Markov source* $\mathcal{H} = (\Sigma, \mathcal{M}, \{\mathcal{S}_1, \ldots, \mathcal{S}_\ell\})$ is specified by an alphabet Σ, a Markov chain $\mathcal{M}$ on ℓ states and distributions $\{\mathcal{S}_i \in \Delta(\Sigma)\}_{i\in[\ell]}$. The output of the source is a sequence $Z_1, Z_2, \ldots$, of random variables obtained by first sampling a sequence $X_0, X_1, X_2, \ldots$ according to $\mathcal{M}$ and then sampling $Z_i \sim \mathcal{S}_{X_i}$ independently for each i. We let $\mathcal{H}_t$ the distribution of output sequences of length t, and $\mathcal{H}_t^{\otimes s}$ denote the distribution of s i.i.d. samples from $\mathcal{H}_t$.

Similarly, we define an *additive Markov channel* as a channel which adds noise from a Markov source.

▶ **Definition 4.** An *additive Markov channel* $\mathcal{C}_\mathcal{H}$, specified by a Markov source $\mathcal{H}$ over alphabet $\mathbb{F}_q$, is a randomized map $\mathcal{C}_\mathcal{H} : \mathbb{F}_q^* \to \mathbb{F}_q^*$ obtained as follows: On channel input $X_1, \ldots, X_n$, the channel outputs $Y_1, \ldots, Y_n$ where $Y_i = X_i + Z_i$ where $Z = (Z_1, \ldots, Z_n) \sim \mathcal{H}_n$.

▶ **Definition 5.** A *linear code* is a linear map $C : \mathbb{F}_q^k \to \mathbb{F}_q^n$. The *rate* of a code is the ratio k/n.

▶ **Definition 6.** For all sets A, B, a *constructive source over* $(A|B)$ *samplable in time* T is a distribution $\mathcal{D} \in \Delta(A \times B)$ such that $(a, b) \sim \mathcal{D}$ can be sampled efficiently in time at most T, and for every fixed $b \in B$, the conditional distribution $A|\{B = b\}$ can be sampled efficiently in time at most T.

▶ **Proposition 7.** *Every Markov source with state space* $[\ell]$ *is a constructive source samplable in time* $\mathcal{O}(n\ell^2)$*. That is, for every* n*, let* $Y_1, \ldots Y_n$ *be the random variables generated by the Markov source. Then, the sequence* $Y_1, \ldots Y_n$ *can be sampled in time at most* $\mathcal{O}(n\ell^2)$*, and moreover for every setting of* $Y_{<n} = y_{<n}$*, the distribution* $(Y_n|Y_{<n} = y_{<n})$ *can be sampled in time* $\mathcal{O}(n\ell^2)$*.*

Proof. Sampling $Y_1, \ldots, Y_n$ can clearly be done by simulating the Markov chain, and sampling from the conditional distribution $(Y_n|Y_{<n} = y_{<n})$ is possible using the standard *Forward Algorithm* for inference in Hidden Markov Models, which we describe for completeness in Appendix A. ◀

Finally, we will use the following notion of *mixing matrices* from [7, 2], characterizing which matrices lead to good polar codes. In the study of polarization it is well-known that lower-triangular matrices do not polarize at all, and the polarization characteristics of matrices are invariant under column permutations. Mixing matrices are defined to be those that avoid the above cases.

▶ **Definition 8.** For prime q and $M \in \mathbb{F}_q^{k\times k}$, M is said to be a *mixing matrix* if M is invertible and for every permutation of the columns of M, the resulting matrix is not lower-triangular.

2.2 Main Theorems

We are now ready to state the main results of this work formally. We begin with the statement for compressing the output of a hidden Markov model.

▶ **Theorem 9.** *For every prime* q *and mixing matrix* $M \in \mathbb{F}_q^{k\times k}$ *there exists a preprocessing algorithm (*Polar-Preprocess*, Algorithm 6.3), a compression algorithm (*Polar-Compress*, Algorithm 4.1), a decompression algorithm (*Polar-Decompress*, Algorithm 4.2) and a polynomial* $p(\cdot)$ *such that for every* $\varepsilon > 0$*, the following properties hold:*

1. Polar-Preprocess *is a randomized algorithm that takes as input a Markov source $\mathcal{H}$ with ℓ states, and $t \in \mathbb{N}$, and runs in time* $\mathrm{poly}(n, \ell, 1/\varepsilon, q)$ *where* $n = k^{2t}$ *and outputs auxiliary information for the compressor and decompressor (for $\mathcal{H}_n$).*
2. Polar-Compress *takes as input a sequence $Z \in \mathbb{F}_q^n$ as well as the auxiliary information output by the preprocessor, runs in time $\mathcal{O}(n \log n)$, and outputs a compressed string $\tilde{U} \in \mathbb{F}_q^{\overline{H}(Z)+\varepsilon n}$. Further, for every auxiliary input, the map $Z \to \tilde{U}$ is a linear map.*
3. Polar-Decompress *takes as input a Markov source $\mathcal{H}$ a compressed string $\tilde{U} \in \mathbb{F}_q^{\overline{H}(Z)+\varepsilon n}$ and the auxiliary information output by the preprocessor, runs in time $\mathcal{O}(n^{3/2}\ell^2 + n \log n)$ and outputs $\hat{Z} \in \mathbb{F}_q^n$.* [6]

The guarantee provided by the above algorithms is that with probability at least $1 - \exp(-\Omega(n))$*, the Preprocessing Algorithm outputs auxiliary information S such that*

$$\Pr_{Z \sim \mathcal{H}_n}[\textsc{Polar-Decompress}(\mathcal{H}, S; \textsc{Polar-Compress}(Z; S)) \neq Z] \leq \mathcal{O}(\frac{1}{n^2}),$$

provided $n > p(\tau/\varepsilon)$ where τ is the mixing time of $\mathcal{H}$.

(In the above $\mathcal{O}(\cdot)$ hides constants depending k and q, but not on ℓ or n.)

The above linear compression directly yields channel coding for additive Markov channels, via a standard reduction (the details of which are in Section 7.)

▶ **Theorem 10.** *For every prime q and mixing matrix $M \in \mathbb{F}_q^{k \times k}$ there exists a randomized preprocessing algorithm* Preprocess*, an encoding algorithm* Enc*, a decoding algorithm* Dec*, and a polynomial $p(\cdot)$ such that for every $\varepsilon > 0$, the following properties hold:*

1. Preprocess *is a randomized algorithm that takes as input an additive Markov channel $\mathcal{C}_\mathcal{H}$ described by Markov source $\mathcal{H}$ with ℓ states, and $t \in \mathbb{N}$, and runs in time* $\mathrm{poly}(n, \ell, 1/\varepsilon)$ *where $n = k^{2t}$, and outputs auxiliary information for $\mathcal{H}_n$.*
2. Enc *takes as input a message $x \in \mathbb{F}_q^r$, where $r \geq n(1 - \frac{\overline{H}(Z)}{n} - \varepsilon)$, as well as auxiliary information from the preprocessor and outputs and computes $\textsc{Enc}(x) \in \mathbb{F}_q^n$ in $\mathcal{O}(n \log n)$ time.*
3. Dec *takes as input the Markov source $\mathcal{H}$, auxiliary information from the preprocessor and a string $z \in \mathbb{F}_q^n$, runs in time $\mathcal{O}_q(n^{3/2}\ell^2 + n \log n)$, and outputs an estimate $\hat{x} \in \mathbb{F}_q^r$ of the message x.* [7]

The guarantee provided by the above algorithms is that with probability at least $1 - \exp(-\Omega(n))$*, the Preprocessing algorithm outputs S such that for all $x \in \mathbb{F}_q^r$ we have*

$$\Pr_{\mathcal{C}_\mathcal{H}}[\textsc{Dec}(\mathcal{H}; \mathcal{C}_\mathcal{H}(\textsc{Enc}(C; x))) \neq x] \leq \mathcal{O}(\frac{1}{n^2}),$$

provided $n > p(\tau/\varepsilon)$ where τ is the mixing time of $\mathcal{H}$.

(In the above $\mathcal{O}(\cdot)$ hides constants that may depend on k and q but not on ℓ or n.)

Theorem 10 follows relatively easily from Theorem 9 and so in the next section we focus on the overview of the proof of the latter.

---

[6] The runtime of the decompression algorithm can be improved to a runtime of $\mathcal{O}(n^{1+\delta}\ell^2 + n \log n)$ by a simple modification. In particular, by taking the input matrix Z to be $n^{1-\delta} \times n^\delta$ instead of $n^{1/2} \times n^{1/2}$. In fact we believe the decoding algorithm can be improved to an $O(n \log n)$ time algorithm with some extra bookkeeping though we don't do so here.

[7] This can similarly be improved to a runtime of $\mathcal{O}_q(n^{1+\delta}\ell^2 + n \log n)$.

3 Overview of our construction

Basics of polarization. We start with the basics of polarization in the setting of compressing samples from an i.i.d. source. To compress a sequence $Z \in \mathbb{F}_2^n$ drawn from some source, the idea is to build an invertible linear function P such that for all but ε fraction of the output coordinates $i \in [n]$, the conditional entropy $H(P(Z)_i|P(Z)_{<i})$ is close to 0 and or close to 1. (Such an effect is called *polarization*, as the entropies are driven to polarize toward the two extreme values.) Since a deterministic invertible transformation preserves the total entropy, it follows that roughly $H(Z)$ output coordinates can have entropy close to 1 and $n - H(Z)$ coordinates have (conditional) entropy close to 0. Letting S denote the coordinates whose conditional entropies that are not close to zero, the compression function is simply $Z \mapsto P(Z)_S$, the projection of the output $P(Z)$ onto the coordinates in S.

Picking a random linear function P would satisfy the properties above with high probability, but this is not known (and unlikely) to be accompanied by efficient algorithms. To get the algorithmics (how to compute P efficiently, to determine S efficiently, and to decompress efficiently) one uses a recursive construction of P. For our purposes the following explanation works best: Let $n = m^2$ and view $Z = (Z_{11}, Z_{12}, \ldots, Z_{mm})$ and as an $m \times m$ matrix over $\mathbb{F}_2$, where the elements of Z arrive one row at a time. Let $P_m^{\text{row}}(\cdot)$ denote the operation mapping $\mathbb{F}_2^{m\times m}$ to $\mathbb{F}_2^{m\times m}$ that applies P_m to each row of separately. Let $P_m^{\text{column}}(\cdot)$ denote the operation that applies P_m to each column separately. Then $P_n(Z) = P_m^{\text{column}}(P_m^{\text{row}}(Z))^T$. The base case is given by $P_2(U, V) = (U + V, V)$.

Intuitively, when the elements of Z are *independent* and identical, the operation P_m already polarizes the outputs somewhat and so a moderate fraction of the outputs of $P_m^{\text{row}}(Z)$ have conditional entropies moderately close to 0 or 1. The further application of $P_m^{\text{column}}(\cdot)$ further polarizes the output bringing a larger fraction of he conditional entropies of the output even closer to 0 or 1.

Polarization for Markovian Sources. When applied to source Z with memory, roughly the analysis in [9], reinterpreted to facilitate our subsequent modification of the above polar constructuion, goes as follows: Since the elements of the row Z_i are not really independent one cannot count on the polarization effects of P_m^{row}. But, letting $U = P_m^{\text{row}}(Z)$ one can show that most elements of the column of U^j are almost independent of each other, provided m is much larger than the mixing time of the source. (Here we imagine that the entries of Z arrive row-by-row, so that the source outputs within each row are temporally well-separated from most entries of the previous row, when m is large.) Further, this almost independence holds even when conditioning on the columns $U^{<j}$ for most values of j. Thus the operation $P_m^{\text{column}}(\cdot)$ continues to have its polarization effects and this is good enough to get a qualitatively strong polarization theorem (about the operator P_n!).

The above analysis is asymptotic, proving that in the limit of $n \to \infty$, we get optimal compression. However, we do not know how to give an effective finite-length analysis of the polarization process for Markovian process, as the analysis in [5, 6] crucially rely on independence which we lack within a row.

Our Modified Code and Ingredients of Analysis. To enable a finite-length analysis, we make a minor, but quite important, alteration to the polar code: Instead of using $P_n(Z) = P_m^{\text{column}}(P_m^{\text{row}}(Z))^T$ we simply use the transformation $\tilde{P}_n = P_m^{\text{column}}(Z)^T$ (or in other words, we replace the inner function $P_m^{\text{row}}(\cdot)$ in the definition of P_n by the identity function). This implies that we lose whatever polarization effects of P_m^{row} we may have been counting on, but as pointed out above, for Markov sources, we weren't counting on polarization here anyway!

The crucial property we identify and exploit in the analysis is the following: the Markovian nature of the source plus the row-by-row arrival ordering of Z, implies that the distribution of the j'th source column Z^j conditioned on the previous columns $Z^{<j} = z^{<j}$, is a *close to* a product distribution, for all but the last few (say εm) columns. [8]

It turns out that the analysis of the polar transform P_m only needs independent inputs, which however need not be identically distributed. We are then able to apply the recent analysis from [2], essentially as black box, to argue that P_m will compress each of the conditioned sources $Z^j|Z^{<j} = z^{<j}$ to its respective entropy, and also establish fast convergence via quantitatively strong polynomial (in the gap to capacity) upper bounds on the m needed to achieve this. Further, we automatically benefit from the generality of the analysis in [2], which applies not only to the 2×2 transform P_2 at the base case, but in fact any $k \times k$ transform (satisfying some minimal necessary conditions) over an arbitrary prime field $\mathbb{F}_q$. Previous works on polar coding for Markovian sources [4, 9, 12] only applied for Boolean sources.

We remark that the use of the identity transform for the rows in $\tilde{P}_n$ is quite counterintuitive. It implies that the compression matrix is a block diagonal matrix (after some permutation of the rows and columns) – and in turn this seems to suggest that we are compressing different parts of the input sequence "independently". However this is not quite true. The relationship between the blocks ends up influencing the final set S of the bits of $\tilde{P}_n(Z)$ that are output by the compression algorithm. Furthermore the decompression relies on the information obtained from the decompression of the blocks corresponding to $Z^{<j}$ to compute the block Z^j.

Decompression algorithm. Our alteration to apply the identity transform for the rows also helps us with the task of decompression. Toward this, we build on a decompression algorithm for memoryless sources from [2] that is somewhat different looking from the usual ones in the polar coding literature. This algorithm aims to compute $U = P_m^{\mathrm{row}}(Z)$ one *column* at a time, given $P_n(Z)|_S$. Given the first $j-1$ columns $U^{<j} = u^{<j}$, the algorithm first computes the conditional distribution of U^j conditioned on $U^{<j} = u^{<j}$ and then uses a recursive decoding algorithm for P_m to determine U^j. The key to the recursive use is again that the decoding algorithm works as long as the input variables are independent (and in particular, does not need them to be identically distributed).

In our Markovian setting, we now have to compute the conditional distribution of Z^j conditioned on $Z^{<j} = z^{<j}$. But as mentioned above, this conditional distribution is close to a product distribution, say $D_j(z^{<j})$ (except for the last few columns j where decompression is trivial as we output the entire column). Further, the marginals of this product distribution are easily computed using dynamic programming (via what is called the "Forward Algorithm" for hidden Markov models, described for completeness in Appendix A). We can then determine the j'th column Z^j (having already recovered the first $j-1$ columns as $z^{<j}$) by running (in a black box fashion) the polar decompressor from [2] for the memoryless case, feeding this product distribution $D_j(z^{<j})$ as the source distribution.

Computing the output indices. Finally we need one more piece to make the result fully constructive. This is the preprocessing needed to compute the subset S of the coordinates of $\tilde{P}_n(Z)$ that have noticeable conditional entropy. For the memoryless case these computations

[8] We handle the non-independence in the last few columns, by simply outputting those columns $P_m(Z^j)$ in entirety, rather than only a set S_j of entropy-carrying positions. This only adds an ε fraction to the output length, which we can afford.

Algorithm 4.1 POLAR-COMPRESS.

Constants: $M \in \mathbb{F}_q^{k\times k}$, $m = k^t, n = m^2$
Input: $Z = (Z_{11}, Z_{12}, \ldots, Z_{mm}) \in \mathbb{F}_q^n$, and sets $S_j \subseteq [m]$ for $j \in [m]$
Output: $U^j_{S_j} \in \mathbb{F}_q^{s_j}$ for all $j \in [m]$ ▷ $s_j := |S_j|$ for $j \leq (1-\varepsilon)m$, and $s_j := m$ otherwise.

```
procedure POLAR-COMPRESS(Z; {S_j}_{j∈[m]})
    for all j ∈ [m] do
        Compute U^j := P_m(Z^j).
        If j ≤ (1 − ε)m then
            Output U^j_{S_j}
        else
            Output U^j
```

were shown to be polynomial time computable in the works of [8, 5, 11]. We manage to extend the ideas from Guruswami and Xia [5] to the case of Markovian channels as well. It turns out the only ingredients needed to make this computation work are, again, the ability to compute the distributions of Z^j conditioned on $Z^{<j} = z^{<j}$ for typical values of $z^{<j}$. We note that unlike in the setting of memoryless channels (or i.i.d. sources) our preprocessing step is randomized. We believe this is related to the issue that there is no "closed" form solutions to basic questions related to Markovian sources and channels (such as the capacity of the channel in Figure 1) and this forces us to use some random sampling and estimation to compute some of the conditional entropies needed by our algorithms.

Organization of rest of the paper. In the next section (Section 4) we describe our compression and decompression algorithms. In Section 5 we describe a notion of "nice"-ness for the preprocessing stage and show that if the preprocessing algorithm returns a nice output, then the compression and decompression algorithm work correctly with moderately high probability (over the message produced by the source). In Section 6 we describe our preprocessing algorithm that returns a nice set with all but exponentially small failure probability (over its internal coin tosses). Finally in Section 7 we give the formal proofs of Theorems 9 and 10.

4 Construction

4.1 Compression Algorithm

Our compression, decompression and preprocessing algorithms are defined with respect to arbitrary mixing matrices $M \in \mathbb{F}_q^{k\times k}$. (Recall that mixing matrices were defined in Definition 8.) Though a reader seeking simplicity may set $k = 2$ and $M = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. Given integer t, let $m = k^t$ and let $P_m = P_{m,M} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ be the polarization transform given by $P_m = M^{\otimes t}$.

4.2 Fast Decompressor

The decompressor below makes black-box use of the FAST-DECODER from [2, Algorithm 4].

The FAST-DECODER takes as input the description of a *product distribution* $\mathcal{D}_Z$ on inputs in $\mathbb{F}_q^m$, as well as the specified coordinates of the compression U. It is intended to decode from the encoding $U' \in \{\mathbb{F}_q \cup \{\perp\}\}^m$, where $U := M^{\otimes t}Z$, coordinates of Z are independent,

Algorithm 4.2 Polar-Decompress.

Constants: $M \in \mathbb{F}_q^{k\times k}$, $m = k^t, n = m^2$
Input: Markov Source $\mathcal{H}$ and $U^1_{S_1}, U^2_{S_2}, \ldots, U^m_{S_m} \in \mathbb{F}_q^m$
Output: $\hat{Z} \in \mathbb{F}_q^{m\times m}$

1: **procedure** Polar-Decompress($\mathcal{H}; U^1_{S_1}, U^2_{S_2}, \ldots, U^m_{S_m}$)
2: **for all** $j \in [m]$ **do**
3: **If** $j \le (1-\varepsilon)m$ **then**
4: Compute the distribution $\mathcal{D}_{z^j|\hat{z}^{<j}} \equiv \overline{Z}^j | \{\overline{Z}^{<j} = \hat{Z}^{<j}\}$, using the *Forward Algorithm* on Markov Source $\mathcal{H}$.
5: Define $U^j \in \{\mathbb{F}_q \cup \{\perp\}\}^m$ by extending $U^j_{S_j}$ using $\perp$ in the unspecified coordinates.
6: Set $\hat{Z}^j \leftarrow$ Fast-Decoder($\mathcal{D}_{z^j|\hat{z}^{<j}}; U^j$)
7: **else**
8: Set $\hat{Z}^j \leftarrow (M^{-1})^{\otimes t} \hat{U}^j_{S_j}$ ▷ Note here $S_j = [m]$
9: **Return** $\hat{Z}$

and U' is defined by U on the high-entropy coordinates of U (and $\perp$ otherwise). It outputs an estimate $\hat{Z}$ of the input Z.

Note that, for a Markov source $\mathcal{H}$ on ℓ states, Line 4 takes time $\mathcal{O}_q(m^2\ell^2)$ (time $\mathcal{O}_q(m\ell^2)$ per coordinate of $\overline{Z}^j$, using the Forward Algorithm). The Fast-Decoder call in Line 6 takes time $\mathcal{O}_q(m \log m)$. Thus, the total runtime is $\mathcal{O}_q(m^3\ell^2 + m^2 \log m) = \mathcal{O}_q(n^{3/2}\ell^2 + n \log n)$.

5 Analysis

The goal of this section is to prove that the decompressor works correctly, with high probablity, provided the preprocessing stage returns the appropriate sets $\{S_j\}$. Specifically, we prove Theorem 12 as stated below. But first we need a definition of "nice" sets $\{S_j\}$: We will later show that pre-processing produces such sets and compression and decompression work correctly (w.h.p.) given nice sets.

▶ **Definition 11** ((ε,ζ)-niceness)**.** Let $\mathcal{H}$ be a Markov source. For every $m \in \mathbb{N}$ and $n = m^2$, let $\overline{Z} \sim \mathcal{H}_m^{\otimes m}$ be the corresponding "independent" distribution. Let $\overline{U} := P_m^{\text{column}}(\overline{Z})$.

We call sets $S_1, S_2, \ldots S_m \subseteq [m]$ *"(ε,ζ)-nice"* if they satisfy the following:

1. $\sum_j |S_j| \le \overline{H}(Z) + \varepsilon n$
2. $\forall j \in [m], i \notin S_j : \overline{H}(\overline{U}_{(i,j)} | \overline{U}_{\prec(i,j)}) < \zeta$

Now, the rest of this section will show the following.

▶ **Theorem 12.** *There exists a polynomial $p(\cdot)$ such that for every $\varepsilon > 0$, $\tau > 0$, and $n = m^2 > p(\tau/\varepsilon)$ the following holds:*

Let $\mathcal{H}$ be an aperiodic irreducible Markov source with alphabet $\mathbb{F}_q$, mixing time τ and underlying state space $[\ell]$. Define random variables $Z = (Z_{11}, Z_{12} \ldots Z_{mm}) \sim \mathcal{H}_{m^2}$ as generated by $\mathcal{H}$. Then, for all sets $S_1, S_2, \ldots S_m \subseteq [m]$ that are (ε,ζ)-nice as per Definition 11, we have:

$$\Pr_Z[\text{Polar-Decompress}(\text{Polar-Compress}(Z; \{S_j\}_{j\in[m]})) \neq Z] \le n\zeta + m\exp(-\varepsilon m/\tau)$$

5.1 Proof Overview

Throughout this section, let $\mathcal{H}$ be a stationary Markov source with alphabet $\mathbb{F}_q$ and mixing-time τ. The key part of the analysis is showing that compression and decompression succeed when applied to the "independent" distribution $\overline{Z} \sim \mathcal{H}_m^{\otimes m}$. To do this, we first show that the compression transform "polarizes" entropies, which follows directly from the results of [2, 3]. Then we show that, provided "nice" sets can be computed (low-entropy sets, a la Definition 11), the compression and decompression succeed with high probability. This also follows essentially in a black-box fashion from the results of [2]. Finally, we argue that the compression and decompression also work for the actual distribution $Z \sim \mathcal{H}_{m^2}$, simply by observing that the involved variables are close in distribution.

We later describe how such "nice" sets can be computed in polynomial time, given the description of the Markov source $\mathcal{H}$.

5.2 Polarization

In this section, we show that the compression transform P_m^{column} polarizes entropies.

▶ **Lemma 13.** *Let $\mathcal{H}$ be a Markov source, and let $\overline{Z} \sim \mathcal{H}_m^{\otimes m}$. Let $\overline{U} = P_m^{\text{column}}(\overline{Z})$.*

Then, there exists a polynomial $p(\cdot)$ such that for every $\varepsilon > 0$, there exists $\beta > 0$ such that if $m > p(1/\varepsilon)$, the following holds: For all but ε-fraction of indices $i, j \in [m] \times [m]$, the normalized entropy

$$\overline{H}(\overline{U}_{i,j} | \overline{U}_{\prec(i,j)}) \notin (\exp(-m^\beta), 1-\varepsilon)$$

Proof. We will show that for each column $\overline{U}^j$, all but ε-fraction of indices $i \in [m]$ have entropies

$$\overline{H}(\overline{U}_i^j | \overline{U}_{<i}^j, \overline{U}^{<j}) \notin (\exp(-m^\beta), 1-\varepsilon)$$

Indeed, this follows directly from the analysis in [3]. For each j, the set of variables $(\overline{Z}_1^j, \overline{Z}_1^{<j}), (\overline{Z}_2^j, \overline{Z}_2^{<j}), \ldots, (\overline{Z}_m^j, \overline{Z}_m^{<j})$ are independent and identically distributed. Thus, Theorem 14 from [3] (reproduced below) implies that the conditional entropies are polarized. Specifically, let $p(\cdot)$ and β be as guaranteed by Theorem 14, for the distribution $\mathcal{D}) \equiv (\overline{Z}_1^j, \overline{Z}_1^{<j})$. Then, since $P_m = M^{\otimes t}$, we have

$$\begin{aligned}
\overline{H}(\overline{U}_i^j | \overline{U}_{<i}^j, \overline{U}^{<j}) &= \overline{H}(\overline{U}_i^j | \overline{U}_{<i}^j, P_m(\overline{Z}^{<j})) && \text{(by definition)} \\
&= \overline{H}(\overline{U}_i^j | \overline{U}_{<i}^j, \overline{Z}^{<j}) && (P_m \text{ is invertible}) \\
&\notin (\exp(-m^\beta), 1-\varepsilon) && \text{(Theorem 14)}
\end{aligned}$$

◀

The following theorem is direct from the works [3].

▶ **Theorem 14.** *For every $k \in \mathbb{N}$, prime q, mixing-matrix $M \in \mathbb{F}_q^{k \times k}$, discrete set $\mathcal{Y}$, and any distribution $\mathcal{D} \in \Delta(\mathbb{F}_q \times \mathcal{Y})$, the following holds. Define the random vectors $A := (A_1, A_2, \ldots A_n)$ and $B := (B_1, B_2, \ldots B_n)$ where $n = k^t$ and each component (A_i, B_i) is independent and identically distributed $(A_i, B_i) \sim \mathcal{D}$.*

Let $X := M^{\otimes t} A$. Then, the conditional entropies of X are polarized*: There exists a polynomial $p(\cdot)$ and $\beta > 0$ such that for every $\varepsilon > 0$, if $n = k^t > p(1/\varepsilon)$, then all but ε-fraction of indices $i \in [n]$ have normalized entropy*

$$\overline{H}(X_i | X_{<i}, B) \notin (\exp(-n^\beta), 1-\varepsilon) \ .$$

5.3 Independent Analysis

Now we bound the failure probability of the Polar Compressor and Decompresser, when applied to the "independent" input distribution $\overline{Z}$.

▶ **Claim 15.** *Let $\overline{Z} \sim \mathcal{H}_m^{\otimes m}$ and $\overline{U} := P_m^{\text{column}}(\overline{Z})$. Then, for all sets $S_1, S_2, \dots S_m \subseteq [m]$,*

$$\Pr_{\overline{U} \leftarrow P_m^{\text{column}}(\overline{Z})}[\textsc{Polar-Decompress}(U^1_{S_1}, U^2_{S_2}, \dots, U^m_{S_m}) \neq \overline{U}] \leq \sum_{j \in [m], i \notin S_j} \overline{H}(\overline{U}_i^j | \overline{U}_{<i}^j, \overline{U}^{<j})$$

Proof. Appears in the full version of this paper. ◀

5.4 Proof of Main Theorem

At this point, we can show the entire process of compression and decompression succeeds with high probability, proving Theorem 12.

First, we argue $\mathcal{H}_{m^2}$ and $\mathcal{H}_m^{\otimes m}$ are close in the appropriate sense.

▶ **Lemma 16.** *Let $Z \sim \mathcal{H}_{m^2}$ and $\overline{Z} \sim \mathcal{H}_m^{\otimes m}$. Then, for every $\ell \in [m]$, the distribution of $Z^{<m-\ell}$ and $\overline{Z}^{<m-\ell}$ are $m \cdot \exp(-\ell/\tau)$-close in L_1.*

Proof. We proceed by a sequence of m hybrids, changing one row at a time to being independent. Let the i-th hybrid be $H_i := Z_{\le i}^{<m-\ell} \circ \overline{Z}_{>i}^{<m-\ell}$, that is, the first i rows of Z, with the remaining rows replaced by iid copies of Z_1.

Consider moving from H_{i+1} to H_i. Conditioned on the first i rows of $Z^{<m-\ell}$, the distribution of the hidden state of the Markov source, at the beginning of the $(i+1)$th row, is $\exp(-\ell/\tau)$-close to its stationary distribution π (since ℓ steps pass between $Z_{i,m-\ell}$ and $Z_{i+1,1}$). Recall that the distribution of Z_1 is generated by the Markov source starting from π. Thus, the distribution of the $(i+1)$th row of Z, conditioned on the first i rows of $Z^{<m-\ell}$, is $\exp(-\ell/\tau)$-close to the distribution of Z_1. So, $|H_{i+1} - H_i|_1 \le \exp(-\ell/\tau)$. Since we pass through m hybrids, the total L_1 distance is at most $m \cdot \exp(-\ell/\tau)$. ◀

Proof of Theorem 12. First, we show the corresponding claim about the "independent" distribution $\overline{Z} \sim \mathcal{H}_m^{\otimes m}$:

▶ **Claim 17.** *For $\overline{Z} \sim \mathcal{H}_m^{\otimes m}$, we have:*

$$\Pr_{\overline{U} \leftarrow P_m^{\text{column}}(\overline{Z})}[\textsc{Polar-Decompress}(\overline{U}^1_{S_1}, \overline{U}^1_{S_2}, \dots \overline{U}^m_{S_m}) \neq \overline{Z}] \leq n\zeta$$

or equivalently,

$$\Pr_{\overline{Z} \sim \mathcal{H}_m^{\otimes m}}[\textsc{Polar-Decompress}(\textsc{Polar-Compress}(\overline{Z}; \{S_j\}_{j \in [m]})) \neq \overline{Z}] \leq n\zeta$$

Proof. By Claim 15, we have

$$\begin{aligned}\Pr_{\overline{U} \leftarrow P_m^{\text{column}}(\overline{Z})}[\textsc{Polar-Decompress}(\overline{U}^1_{S_1}, \overline{U}^1_{S_2}, \dots \overline{U}^m_{S_m}) \neq \overline{Z}] &\leq \sum_{j \in [m], i \notin S_j} \overline{H}(\overline{U}_i^j | \overline{U}_{<i}^j, \overline{U}^{<j}) \\ &\leq n\zeta \quad \text{(by } (\varepsilon, \zeta)\text{-niceness)}\end{aligned}$$ ◀

Continuing the proof of Theorem 12, notice that the composition of Polar-Compress and Polar-Decompress always operate as the identity transform on the inputs Z^j for $j > (1-\varepsilon)m$. Thus, it suffices to consider the behavior of this composition on inputs $Z^{\le(1-\varepsilon)m}$. In this case, Lemma 16 guarantees that the distributions of $\overline{Z}^{\le(1-\varepsilon)m}$ and $Z^{\le(1-\varepsilon)m}$ are close in L_1, and thus we may conclude by Claim 17:

$$\Pr_{Z\sim\mathcal{H}_{m^2}}[\textsc{Polar-Decompress}(\textsc{Polar-Compress}(Z;\{S_j\}_{j\in[m]}))\neq Z]$$
$$\leq \Pr_{\overline{Z}\sim\mathcal{H}_m^{\otimes m}}[\textsc{Polar-Decompress}(\textsc{Polar-Compress}(\overline{Z};\{S_j\}_{j\in[m]}))\neq \overline{Z}] + m\exp(-\varepsilon m/\tau)$$
(Lemma 16)
$$\leq n\zeta + m\exp(-\varepsilon m/\tau) \quad \text{(Claim 17)}$$ ◀

6 Preprocessing

In this section, we describe a pre-processing algorithm to find the (ε,ζ)-nice sets, as defined in Definition 11, that are required by the compression and decompression algorithms. Recall the notion of a mixing matrix (Definition 8). The following theorem shows that for every prime alphabet $\mathbb{F}_q$ and mixing matrix $M\in\mathbb{F}_q^{k\times k}$, there is an efficient algorithm that can find nice sets in polynomial time. Specifically, we prove the following theorem.

▶ **Theorem 18.** *For every prime q and mixing-matrix $M\in\mathbb{F}_q^{k\times k}$, there exists a polynomial $p(\cdot)$ and a polynomial time preprocessing algorithm* Polar-Preprocess *(Algorithm 6.3), such that for every $\varepsilon>0$ and $m>p(1/\varepsilon)$, the following holds:*

Let $\mathcal{H}$ be a Markov source with mixing-time τ, alphabet $\mathbb{F}_q$, and underlying state space $[\ell]$. Let $\overline{Z}\sim\mathcal{H}_m^{\otimes m}$ for $m=k^t$, and $\overline{U}:=P_m^{\text{column}}(\overline{Z})$.

Let

$$S_1,S_2,\dots S_m \leftarrow \textsc{Polar-Preprocess}(q,k,t,M,\mathcal{H})$$

Then, except with probability $\exp(-\Omega(m))$ over the randomness of the algorithm, the output sets $S_1,S_2,\dots S_m\subseteq[m]$ are $(\varepsilon,\zeta=\mathcal{O}(\frac{1}{n^3}))$-nice for $\mathcal{H}$. Further, the algorithm runs in time $\text{poly}_q(m,\ell,1/\varepsilon)$.

Our main goal will be to estimate the conditional entropies

$$\overline{H}(\overline{U}_{(i,j)}|\overline{U}_{\prec(i,j)}) = \overline{H}(\overline{U}_{i,j}|\overline{U}_{<i,j},\overline{U}^{<j}) = \overline{H}(\overline{U}_{i,j}|\overline{U}_{<i,j},\overline{Z}^{<j})$$

for $\overline{Z}\sim\mathcal{H}_m^{\otimes m}$ and $\overline{U}:=P_m^{\text{column}}(\overline{Z})$. Then, we will construct the "nice" sets by defining, for each j, S_j as the set of indices with high entropy: $S_j:=\{i\in[m]:\overline{H}(\overline{U}_{i,j}|\overline{U}_{<i,j},\overline{Z}^{<j})>\frac{1}{n^3}\}$. By Polarization (Lemma 13), these sets have size at most $\sum_j|S_j|\leq\overline{H}(Z)+\varepsilon n$, since they must have conditional entropies close to 1 (except possibly for some ε fraction of indices $(i,j)\in[m]\times[m]$).

We will estimate conditional entropies $\overline{H}(\overline{U}_{i,j}|\overline{U}_{<i,j},\overline{Z}^{<j})$ by approximately tracking the distribution of variables as we apply successive tensor-powers of M. Since we are only interested in conditional entropies, it is sufficient to "quantize" the true distribution of, for example $U_i|U_{<i}$, into an approximation $U_i|A$, such that $H(U_i|U_{<i})\approx H(U_i|A)$. This algorithm follows the same high-level strategy of [5], of approximating the conditional distributions via quantized bins. It turns out that this strategy can be implemented for Markov sources, using the fact that Markov sources are constructive. We define our notions of approximation, and formalize this strategy below.

6.1 Notation and Preliminaries

▶ **Definition 19** (Associated Conditional Distribution)**.** Let X be a random variable taking values in universe U, and let W be an arbitrary random variable. Let $\mathcal{D}_{X|w}\in\Delta(U)$ denote the conditional distribution of $X|\{W=w\}$. Let $\mathbb{D}_{X|W}\in\Delta(\Delta(U))$ be the distribution over $\mathcal{D}_{X|w}$ defined by sampling $w\sim W$. We call $\mathbb{D}_{X|W}$ the *associated conditional distribution to* $X|W$.

As above, we use boldface $\mathbb{D}$ to denote objects of type $\Delta(\Delta(U))$. Note that we can operate on conditional distributions as we would on their underlying random variables. For example, for random variables (A_1, W) and (A_2, Y) such that $A_1, A_2 \in \mathbb{F}_q$ and (A_1, W) is independent from (A_2, Y), the associated conditional distribution of $A_1 + A_2|Y, W$ can be computed from the associated conditional distributions of $A_1|Y$ and $A_2|W$. To more easily describe such operations on conditional distributions (which may not always arise from underlying random variables), we define the *implicit random variables* associated to a conditional distribution:

▶ **Definition 20** (Implicit Random Variables Associated to Conditional Distribution)**.** For every $\mathbb{D}_{X|W} \in \Delta(\Delta(U))$, define *implicit random variables* X, W *associated to* $\mathbb{D}_{X|W}$ as random variables (X, W) such that the associated conditional distribution to $X|W$ is exactly $\mathbb{D}_{X|W}$. Note that there is not a unique choice of such random variables.

Using this, we can naturally define (for example) $\mathbb{D}_{A_1+A_2|W,Y}$ and $\mathbb{D}_{A_2|W,Y,A_1+A_2}$ from any $\mathbb{D}_{A_1,W}, \mathbb{D}_{A_2,Y} \in \Delta(\Delta(\mathbb{F}_q))$. Note that we will always be performing such operations assuming independence of the involved implicit random variables, ie (A_1, W) and (A_2, Y).

▶ **Definition 21** (Conditional Distance)**.** Let (X, W) and (Y, Z) be two joint distributions, such that X and Y take values in the same universe U. Let $\mathbb{D}_{X|W}$ and $\mathbb{D}_{Y|Z}$ be the associated distributions in $\Delta(\Delta(U))$. Then, define the *conditional distance*

$$d_C(\mathbb{D}_{X|W}, \mathbb{D}_{Y|Z}) := \min_{\substack{(A,B):\text{a distribution in } \Delta(\Delta(U)\times\Delta(U)) \\ \text{s.t. marginals of } A \text{ match } \mathbb{D}_{X|W}, \text{ and} \\ \text{marginals of } B \text{ match } \mathbb{D}_{Y|Z}}} \mathop{\mathbb{E}}_{(D_A,D_B)\sim(A,B)} [||D_A - D_B||_1]$$

Note that d_C can be equivalently defined as an optimal transportation cost between two distributions in $\Delta(\Delta(U))$, where the cost of moving a unit of mass between points $D_i, D_j \in \Delta(U)$ is $||D_i - D_j||_1$.

This metric behaves naturally under post-processing:

▶ **Claim 22.** *For all* $\mathbb{D}_{X|W}, \mathbb{D}_{X'|W'} \in \Delta(\Delta(U))$*, and any* $f : U \to V$*,*

$$d_C(\mathbb{D}_{f(X)|W}, \mathbb{D}_{f(X')|W'}) \le d_C(\mathbb{D}_{X|W}, \mathbb{D}_{X'|W'})$$

For computational purposes, we represent the space of distributions using ε-nets:

▶ **Definition 23** (ε-nets)**.** For every set U and any $\varepsilon > 0$, let $T_\varepsilon(U) \subseteq \Delta(U)$ be an ε-net of $\Delta(U)$ with respect to L_1. That is, for every $\mathcal{D} \in \Delta(U)$, there exists $\hat{\mathcal{D}} \in T_\varepsilon(U)$ such that $||\mathcal{D} - \hat{\mathcal{D}}||_1 \le \varepsilon$.

Note that for $|U| = |\mathbb{F}_q| = q$, $T_\varepsilon(U)$ can be chosen such that $|T_\varepsilon(U)| \le \binom{\frac{q}{\varepsilon}+q}{q} \le (\frac{2q}{\varepsilon})^q = poly_q(1/\varepsilon)$.

Moreover, $\Delta(T_\varepsilon(U))$ is an ε-net of $\Delta(\Delta(U))$ under the d_C-metric.

6.2 Conditional Distribution Approximation

The below procedure takes as input a conditional distribution $\mathbb{D}_{Z|W} \in \Delta(\Delta(\mathbb{F}_q))$, and computes an approximation to the conditional distribution of $U_I|(U_{\prec I}, W_1, \dots W_{k^t})$, for an index $I \in [k]^t$, where $U := M^{\otimes t}Z$ and $\{(Z_i, W_i)\}_{i\in[k^t]}$ are independently defined by $\mathbb{D}_{Z|W}$.

Note that if the input $\mathbb{D}_{Z|W}$ is specified in an ε-net $\Delta(T_\varepsilon(\mathbb{F}_q))$, then the above procedure runs in time $poly_q(m, 1/\varepsilon)$ for $m = k^t$.

Algorithm 6.1 Conditional Distribution Approximation.

Input: Conditional distribution on inputs $\mathbb{D}_{Z|W} \in \Delta(\Delta(\mathbb{F}_q))$, $\varepsilon > 0$, $t \in \mathbb{N}$, index $I \in [k]^t$, and $M \in \mathbb{F}_q^{k \times k}$

Output: Conditional distribution $\tilde{\mathbb{D}}_{U|W} \in \Delta(\Delta(\mathbb{F}_q))$, an approximation to $U_I | (U_{\prec I}, W_1, \dots W_{k^t})$ for $U := M^{\otimes t} Z$ and $\{(Z_i, W_i)\}_{i \in [k^t]}$ independently defined by $\mathbb{D}_{Z|W}$.

1: **procedure** ApproxDist($\mathbb{D}_{Z|W}, \varepsilon, t, I = (I_1, \dots, I_t), M$)
2: **If** $t = 0$ **then**
3: **Return** $\mathbb{D}_{Z|W}$
4: **else**
5: $\hat{\mathbb{D}}_{Z|Y} \leftarrow$ ApproxDist($\mathbb{D}_{Z|W}, \varepsilon/(2k), t-1, I_{<t} = (I_1, \dots, I_{t-1}), M$)
6: $j \leftarrow I_t$.
7: Explicitly compute the following conditional distribution $\hat{\mathbb{D}}_{U_j | U_{<j}, Y_1, \dots Y_k} \in \Delta(\Delta(\mathbb{F}_q))$:
8: Let (Z, Y) be the implicit random variables associated to $\hat{\mathbb{D}}_{Z|Y}$.
9: Let $\{(Z_i, Y_i)\}_{i \in [k]}$ be independent random variables distributed identically to (Z, Y).
10: Define random vector $U := M \cdot Z'$, Where $Z' = (Z_1, \dots Z_k)$.
11: Let $\hat{\mathbb{D}}_{U_j | U_{<j}, Y_1, \dots Y_k}$ be the associated conditional distribution to $U_j | U_{<j}, Y_1, \dots Y_k$.
12: Round $\hat{\mathbb{D}}_{U_j | U_{<j}, Y_1, \dots Y_k}$ to $\tilde{\mathbb{D}}_{U|Y} \in \Delta(T_{\varepsilon/2}(\mathbb{F}_q))$, a point in the $\varepsilon/2$-net of $\Delta(\Delta(\mathbb{F}_q))$ under d_C.
13: **Return** $\tilde{\mathbb{D}}_{U|Y}$.

▶ **Lemma 24.** *For all $\mathbb{D}_{Z|W} \in \Delta(\Delta(U)), \varepsilon > 0, t \in \mathbb{N}, M \in \mathbb{F}_q^{k \times k}$, and $I \in [k]^t$, we have*

$$d_C(\text{ApproxDist}(\mathbb{D}_{Z|W}, \varepsilon, t, I, M) \quad , \quad \mathbb{D}_{U_I | U_{\prec I}, W_1, \dots W_{k^t}}) \leq \varepsilon$$

where $\mathbb{D}_{U_I | U_{\prec I}, W_1, \dots W_{k^t}}$ is the associated conditional distribution to the random variables defined as follows. Let (Z, W) be the implicit random variables associated to $\mathbb{D}_{Z|W}$. Let $\{(Z_i, W_i)\}_{i \in [k^t]}$ be independent random variables distributed identically to (Z, W). Finally, define random vector $U := M^{\otimes t} \cdot Z'$, where $Z' = (Z_1, \dots Z_{k^t})$.

Proof. Appears in the full version of this paper. ◀

6.3 Approximating Conditional Entropies

Here we use Algorithm 6.1 directly to approximate conditional entropies:

▶ **Theorem 25.** *For every field $\mathbb{F}_q$, conditional distribution $\mathbb{D}_{Z|W} \in \Delta(\Delta(\mathbb{F}_q))$, matrix $M \in \mathbb{F}^{k \times k}$, $t \in \mathbb{N}, m = k^t$, and $\gamma > 0$, consider the random variable $U := M^{\otimes t} Z$ where each $\{(Z_i, W_i)\}_{i \in [m]}$ is sampled independently from $\mathcal{D}_{Z,W}$.*

Then, Algorithm 6.2 outputs $\hat{h}_1, \dots \hat{h}_m \leftarrow$ ApproxEntropy($\mathbb{D}_{Z|W}, \gamma, t, M$) such that

$$\forall i \in [m] : \hat{h}_i = \overline{H}(U_i | U_{<i}, W_1, \dots, W_m) \pm \gamma$$

Further, if the input $\mathbb{D}_{Z|W}$ is specified in an ε-net $\Delta(T_\varepsilon(\mathbb{F}_q))$, then the above procedure runs in time $\text{poly}_q(m, 1/\varepsilon, 1/\gamma)$.

Algorithm 6.2 Entropy Approximation.

Input: $\gamma > 0$, $t \in \mathbb{N}$, Conditional distribution $\mathbb{D}_{Z|W} \in \Delta(\Delta(\mathbb{F}_q))$, and $M \in \mathbb{F}_q^{k \times k}$
Output: $\{\hat{h}_i \in \mathbb{R}\}_{i \in [k^t]}$

procedure ApproxEntropy($\mathbb{D}_{Z|W}, \gamma, t, M$)
  $m \leftarrow k^t$
  $\varepsilon \leftarrow \frac{\gamma^2}{16 \log(q)}$
  for all $I \in [k]^t$ **do**
    $\mathbb{D}_{U|Y} \leftarrow$ ApproxDist($\mathbb{D}_{Z|W}, \varepsilon, t, I, M$)
    $\hat{h}_I \leftarrow \overline{H}(U|Y)$, the conditional entropy of the implicit random variables (U, Y) associated to $\mathbb{D}_{U|Y}$.
  Return $\{\hat{h}_i\}_{i \in [k^t]}$ ▷ Abusing notation by identifying $[k]^t$ with $[k^t]$.

Algorithm 6.3 Polar-Preprocess.

Input: $q, k, t \in \mathbb{N}$ with q prime, $M \in \mathbb{F}_q^{k \times k}$, and Markov source $\mathcal{H}$ ▷ $m = k^t, n = m^2$
Output: Sets $S_1, S_2, \ldots S_m \subseteq [m]$

procedure Polar-Preprocess($q, k, t, M, \mathcal{H}$)
  $m \leftarrow k^t$; $\gamma \leftarrow \frac{1}{n^{10}}$; $N \leftarrow |T_\gamma(\mathbb{F}_q)|$; $R \leftarrow n(N/\gamma)^2$ ▷ $N \leq poly_q(1/\gamma)$
  for all $j \in [m]$ **do**
    for all $i = 1, 2, \ldots, R$ **do**
      Sample a sequence $w_i := (y_1, y_2, \ldots y_{j-1})$ from $\mathcal{H}$.
      Compute $\mathcal{D}_{w_i} \in \Delta(\mathbb{F}_q)$, the distribution of $Y_j | Y_{<j} = w_i$, using the Forward Algorithm A.1 for $\mathcal{H}$.
    Let $\widetilde{\mathbb{D}}_{Y|W} \in \Delta(\Delta(F_q))$ be the empirical distribution of $\mathcal{D}_w$, from the samples $\mathcal{D}_{w_i}$ above.
    $\{\hat{h}_1, \ldots \hat{h}_m\} \leftarrow$ ApproxEntropy($\widetilde{\mathbb{D}}_{Y|W}, \gamma = \frac{1}{n^4}, t, M$)
    $S_j \leftarrow \{i \in [m] : \hat{h}_i > \frac{1}{n^3}\}$
  Return $S_1, S_2, \ldots S_j$.

Proof of Theorem 25. Correctness of Algorithm 6.2 follows from the fact that $O(\frac{\gamma^2}{\log(q)})$-closeness in the d_C-metric implies γ-closeness of conditional entropies. Thus, using Algorithm 6.1 to approximate the conditional distributions within $O(\frac{\gamma^2}{\log(q)})$ is sufficient. See the full version of this paper for details. ◀

6.4 Nice Subset Selection

Now we can describe how to find "nice" sets. We first approximate the conditional distribution $\mathbb{D}_{Z_t|Z_{<t}} \in \Delta(\Delta(\mathbb{F}_q))$ for $Z_1, \ldots Z_t \sim \mathcal{H}_t$, by sampling. This crucially relies on the fact that $\mathcal{H}$ is a constructive source (ie, using the Forward Algorithm). Then we use Algorithm 6.2 to estimate conditional entropies, and select high-entropy indices.

The correctness of this algorithm (proof of Theorem 18) appears in the full version of this paper.

7 Proofs of Theorems 9 and 10

Combining Theorem 18 (to compute nice sets) with Theorem 12 (compressing and decompressing assuming nice sets), Theorem 9 follows immediately.

Proof of Theorem 9. The algorithms claimed are Algorithm 6.3 for preprocessing, Algorithm 4.1 for compressing and Algorithm 4.2 for decompression. Theorem 18 asserts that Algorithm 6.3 returns a nice sequence of sets $S_1, \ldots, S_m$ with all but exponentially small probability in n. And Theorem 12 asserts that if $S_1, \ldots, S_m$ are nice then Algorithm 4.1 and 4.2 compress and decompress correctly with high probability over the output of the Markovian source. This yields the theorem. ◀

Finally we show how Theorem 10 follows from Theorem 9.

Proof of Thereom 10. Let $H \in \mathbb{F}_q^{s \times n}$ be the matrix specifying the (linear) compression scheme given by the Preprocessing Algorithm in Theorem 9, when applied to Markov source $\mathcal{H}$. The code C for the additive Markov Channel $\mathcal{C}_{\mathcal{H}}$ is simply specified by the nullspace of H, ie encoding is given by $C(x) := Nx$ where $N \in \mathbb{F}_q^{n \times n-s}$ spans $Null(A)$.

Note that due to the structure of H, a nullspace matrix N can be applied in $\mathcal{O}_q(n \log n)$ time. In particular, H is a subset of rows of the block-diagonal matrix $P \in \mathbb{F}_q^{n \times n}$, where each $\sqrt{n} \times \sqrt{n}$ block is the tensor-power $M^{\otimes t}$. Thus, P^{-1} is also block-diagonal with blocks $(M^{-1})^{\otimes t}$, and so can be applied in time $\mathcal{O}_q(n \log n)$. The matrix N can be chosen as just a subset of columns of P^{-1}, and hence can also be applied in time $\mathcal{O}_q(n \log n)$.

Let $y_1, y_2, \ldots y_n \in \mathbb{F}_q$ be distributed according to $\mathcal{H}$, and $y := (y_1, \ldots y_n) \in \mathbb{F}_q^n$. To decode from $z = Nx + y$, the decoder first applies H (by running the compression algorithm of Theorem 9), to compute $Hz = HNx + Hy = Hy$. Then, the decoder runs the decompression algorithm of Theorem 9 on Hy to determine y. Finally, the decoder can compute $y - z$ to find the codeword sent (Nx), and thus determine x. (Again using the structure of P, as above, to determine x from Nx in $\mathcal{O}_q(n \log n)$ time). ◀

References

1. Erdal Arıkan. Channel Polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, pages 3051–3073, July 2009.
2. Jarosław Błasiok, Venkatesan Guruswami, Preetum Nakkiran, Atri Rudra, and Madhu Sudan. General strong polarization. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 485–492. ACM, 2018. arXiv:1802.02718.
3. Jaroslaw Blasiok, Venkatesan Guruswami, and Madhu Sudan. Polar Codes with Exponentially Small Error at Finite Block Length. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 116. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
4. Eren Şaşoğlu. *Polar coding theorems for discrete systems*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2011.
5. Venkatesan Guruswami and Patrick Xia. Polar Codes: Speed of Polarization and Polynomial Gap to Capacity. *IEEE Trans. Information Theory*, 61(1):3–16, 2015. Preliminary version in Proc. of FOCS 2013.
6. Seyed Hamed Hassani, Kasra Alishahi, and Rüdiger L. Urbanke. Finite-Length Scaling for Polar Codes. *IEEE Trans. Information Theory*, 60(10):5875–5898, 2014. doi:10.1109/TIT.2014.2341919.

7 Satish Babu Korada, Eren Sasoglu, and Rüdiger L. Urbanke. Polar Codes: Characterization of Exponent, Bounds, and Constructions. *IEEE Transactions on Information Theory*, 56(12):6253–6264, 2010. doi:10.1109/TIT.2010.2080990.

8 Ramtin Pedarsani, Seyed Hamed Hassani, Ido Tal, and Emre Telatar. On the construction of polar codes. In *Proceedings of 2011 IEEE International Symposium on Information Theory*, pages 11–15, 2011. doi:10.1109/ISIT.2011.6033724.

9 Eren Sasoglu and Ido Tal. Polar coding for processes with memory. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 225–229, 2016.

10 Boaz Shuval and Ido Tal. Fast Polarization for Processes with Memory. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 851–855, 2018.

11 Ido Tal and Alexander Vardy. How to Construct Polar Codes. *IEEE Transactions on Information Theory*, 59(10):6562–6582, October 2013.

12 Runxin Wang, Junya Honda, Hirosuke Yamamoto, Rongke Liu, and Yi Hou. Construction of polar codes for channels with memory. In *Proceedings of the 2015 IEEE Information Theory Workshop - Fall (ITW)*, pages 187–191, 2015.

13 Runxin Wang, Rongke Liu, and Yi Hou. Joint Successive Cancellation Decoding of Polar Codes over Intersymbol Interference Channels. *CoRR*, 2014. arXiv:1404.3001.

A Forward Algorithm

Algorithm A.1 Forward Algorithm.

Input: $n \in \mathbb{N}$. Markov source $\mathcal{H}$ with state-space $[\ell]$, alphabet Σ, stationary distribution $\pi \in \Delta([\ell])$, transition matrix $\Pi \in \mathbb{R}^{\ell \times \ell}$, and output distributions $\{\mathcal{S}_i \in \Delta(\Sigma)\}_{i \in [\ell]}$. And $y = (y_1, y_2, \ldots y_{n-1})$ for $y_i \in \Sigma$.

Output: Distribution $Y_n \in \Delta(\Sigma)$

1: **procedure** ForwardInfer($\mathcal{H} = (\ell, \Sigma, \pi, \Pi, \{\mathcal{S}_i\}), n, y$)

2: $s_0 \leftarrow \pi$.

3: **for all** $t = 1, 2, \ldots n-1$ **do**

4: Define $s_t \in \Delta([\ell])$ by $s_t(i) \leftarrow \frac{(\Pi s_{t-1})_i \cdot \mathcal{S}_i(y_t)}{\sum_{j \in [\ell]} (\Pi s_{t-1})_j \cdot \mathcal{S}_j(y_t)}$ ▷ Treating s_{t-1} as a vector in the probability simplex embedded in $\mathbb{R}^\ell$

5: $s_n \leftarrow \Pi s_{n-1}$.

6: **Return** The distribution $Y_n := \mathbb{E}_{i \sim s_n}[\mathcal{S}_i]$.

▶ **Claim 26.** *For every Markov source $\mathcal{H} = (\ell, \Sigma, \pi, \Pi, \{\mathcal{S}_i\})$, let random variables $Y_1, \ldots Y_n \sim \mathcal{H}_n$. For every setting $y = (y_1, y_2, \ldots y_{n-1})$ for $y_i \in \Sigma$, let $\mathcal{D}_{Y_n | Y_{<n} = y}$ denote the distribution of Y_n conditioned on $Y_{<n} = y$. Then,*

$$\text{ForwardInfer}(\mathcal{H}, n, y) \equiv \mathcal{D}_{Y_n | Y_{<n} = y}$$

This follows inductively, from the fact that s_t as maintained by the algorithm is exactly the distribution of $S_t | \{Y_{\leq t} = y_{\leq t}\}$, where S_t is the hidden state of $\mathcal{H}$ after t steps.

B Connection to Learning Parity with Noise

The problem of learning parity with noise (LPN) is the following. Fix an (unknown) string $a \in \mathbb{F}_2^\ell$ and $\eta > 0$ and let $D_{a,\eta}$ be the distribution on $\mathbb{F}_2^{\ell+1}$ whose samples (x, y) are generated as follows: Draw $x \in \mathbb{F}_2^\ell$ uniformly and let $z \in Bern(\eta)$ be drawn independent of x and let $y = \langle a, x \rangle + z$ where $\langle a, x \rangle = \sum_{i=1}^{\ell} a_i x_i$. Given samples $(x_1, y_1), \ldots, (x_m, y_m)$ drawn i.i.d. from such a distribution, the LPN problem is the task of "learning" a.

It is well known that a is uniquely determined by $O(\ell)$ samples (i.e., $m = O(\ell)$) where the constant in the $O(\cdot)$ depends on $\eta < 1/2$. However no polynomial time algorithms are known that work with $m = \text{poly}(\ell)$ and determine a for any $\eta > 0$ and indeed this is believed to be a hard task in learning. We refer to this hardness assumption as the LPN hypothesis.

The connection to learning Markovian sources comes from the fact that samples from the distribution $D_{a,\eta}$ can be generated by an $O(\ell)$-state Markov chain. (Briefly the states are indexed (i, b, c) indicating $\sum_{j=1}^{i-1} a_j x_j = b$ and $x_i = c$. For $i < \ell$ the state (i, b, c) outputs c and transtions to $(i + 1, b + c, 0)$ w.p. $1/2$ and to $(i + 1, b + c, 1)$ w.p. $1/2$. When $i = \ell$, the state (i, b, c) outputs $(c, b + c)$ w.p. $1 - \eta$ and $(c, b + c + 1)$ w.p. η and transitions to $(1, 0, 0)$ w.p. $1/2$ and to $(1, 0, 1)$ w.p. $1/2$.) The entropy of this source is $(\ell + H(\eta))/(\ell + 1)$. A compression with $\varepsilon = (1 - H(\eta))/(2(\ell + 1))$ with $\text{poly}(\ell/\varepsilon)$ samples from the source would distinguish this source from purely random strings which in turn enables recovery of a, contradicting the LPN hypothesis.

We thus conclude that compressing an *unknown* Markov source with number of samples that is a polynomial in the mixing time and the inverse of the gap to capacity contradicts the LPN hypothesis.