The Minimum Distance of Turbo-Like Codes

Louay Bazzi, Mohammad Mahdian, Daniel A. Spielman

***Abstract*—Worst-case upper bounds are derived on the minimum distance of parallel concatenated Turbo codes, serially concatenated convolutional codes, repeat-accumulate codes, repeat-convolute codes, and generalizations of these codes obtained by allowing non-linear and large-memory constituent codes. It is shown that parallel-concatenated Turbo codes and repeat-convolute codes with sub-linear memory are asymptotically bad. It is also shown that depth-two serially concatenated codes with constant-memory outer codes and sub-linear-memory inner codes are asymptotically bad. Most of these upper bounds hold even when the convolutional encoders are replaced by general finite-state automata encoders. In contrast, it is proven that depth-three serially concatenated codes obtained by concatenating a repetition code with two accumulator codes through random permutations can be asymptotically good.**

***Index Terms*—Asymptotic growth, concatenated codes, minimum distance, Turbo codes, RAA codes.**

I. Introduction

The low-complexity and near-capacity performance of Turbo codes [3], [9] has led to a revolution in coding theory. The most famous casualty of the revolution has been the idea that good codes should have high minimum distance: the most useful Turbo codes have been observed to have low minimum distance

In this work, we provide general conditions under which many constructions of turbo-like codes, including families of serially-concatenated convolutional codes [2] and Repeat-Accumulate (RA) codes [5], [6], [7], must be asymptotically bad[1]. We also present a simple family of depth-3 serially concatenated convolutional codes that are asymptotically good.

Our work is motivated by the analyses of randomly constructed parallel and serially concatenated convolutional codes by Kahale and Urbanke [7] and of parallel concatenated Turbo codes with two branches by Breiling [4].

Kahale and Urbanke [7] provided probabilistic estimates on the minimum distance of randomly generated parallel concatenated Turbo codes with a constant number of branches. They also provided similar estimates for the minimum distance of the random concatenation of two convolutional codes with bounded memory. In particular, Kahale and Urbanke proved that if one builds a parallel concatenated code with k branches from random permutations and convolutional encoders of memory at most M, then the resulting code has minimum distance at most $2^{O(M)}n^{1-2/k}\log^{O(1)} n$ and at least $\Omega(n^{1-2/k})$ with high probability, where n is the number of message bits. For rate $1/4$ serially concatenated convolutional codes with random interleavers, they proved that the resulting code has minimum distance at most $2^{O(M_i)}n^{1-2/d_o}\log^{O(1)} n$ and at least $\Omega(n^{1-2/d_o})$ with high probability, where d_o is the free distance of the outer code and M_i is the inner code memory.

Breiling [4] proved that the parallel concatenation of two convolutional codes with bounded memory has at most logarithmic minimum distance, regardless of the choice of interleaver. In particular, for parallel concatenated Turbo codes with two branches, Breiling proved that no construction could be much better than a random construction: if the constituent codes have memory M, then the minimum distance of the resulting code is $O(2^{O(M)}\log n)$.

These bounds naturally lead to the following five questions:

(better than random?) Do there exist asymptotically good parallel concatenated Turbo codes with more than two branches or do there exist asymptotically good repeat-convolute or repeat-accumulate codes?
Note that the result of Breiling only applies to Turbo codes with two branches and the results of Kahale and Urbanke do not preclude the existence of codes that are better than the randomly generated codes.
(larger memory?) What happens if we allow the memories of the constituent convolutional codes to grow with the block length?
All the previous bounds become vacuous if the memory even grows logarithmically with the block length.
(non-linearity?) Can the minimum distance of Turbo-like codes be improved by the use of non-linear constituent encoders, such as automata encoders?
(concatenation depth?) Can one obtain asymptotically good codes by serially concatenating a repetition code with two levels of convolutional codes?

We will give essentially negative answers to the first three questions and a positive answer to the last one. For parallel concatenations and depth-2 serial concatenations of convolutional codes and non-linear automata codes, we prove upper bounds on the minimum distance of the resulting codes in terms of the memories of the constituent codes. In Section II-A, we show that parallel concatenated codes and repeat-convolute codes are asymptotically bad if their constituent codes have sub-linear memory. These bounds hold even when the codes are generalized by replacing the constituent convolutional codes by automata codes. In Section II-B, we restrict our attention to concatenations of ordinary convolutional codes, and obtain absolute upper bounds that almost match the high-probability upper bounds for random permutations obtained by Kahale and Urbanke.

L. Bazzi is with the Department of Electrical and Computer Engineering, AUB, Beirut, Lebanon. Work on this paper supported by ARO grant DAA L03-92-G-0115, and MURI grant DAAD19-00-1-0466.

M. Mahdian is with Yahoo! Research. Work on this paper supported by NSF CCR: 9701304.

D. A. Spielman is with the Department of Computer Science and Program in Applied Mathematics, Yale University. Work on this paper supported by NSF CCR: 9701304.

[1] A sequence of codes of increasing block length is called *an asymptotically good code* if the message length and the minimum distance of the codes grows linearly with the block length. Codes for which either the message length or minimum distance grow sub-linearly with the block length are called *asymptotically bad*.

In Section III-A, we show that depth-two serially concatenated codes are asymptotically bad if their inner code has sub-linear memory and their outer code has constant memory. This bound also applies to the generalized case of constituent automata codes.

In contrast, we show in Section III-B that depth-three concatenations of constant-memory codes can be asymptotically good. In particular, we prove this for the random concatenation of a repetition code with two accumulator codes.

A. Turbo-like codes

The fundamental components of the codes we consider in this paper are convolutional codes and their non-linear generalizations, which we call automata codes. The fundamental parameter of a convolutional code that we will measure is its *memory*—the number of registers in its encoder. The amount of memory can also be defined to be the binary logarithm of the number of states in the encoder's state diagram. A general automata encoder is obtained by considering an encoder with any deterministic state diagram. We will consider automata encoders that read one bit at each time step, and output a constant number of bits at each time step. These can also be described as deterministic automata or transducers with one input bit and a constant number of output bits on each transition. We will again define the memory of an automata encoder to be the binary logarithm of its number of states.

Given a convolutional encoder Q and k permutations $\pi_1, \ldots, \pi_k$, each of length n, we can define the *parallel concatenated Turbo code with k branches* [3], [9] to be the code whose encoder maps an input $x \in \{0,1\}^n$ to $(x, Q(\pi_1(x)), \ldots, Q(\pi_k(x)))$, where $\pi_i(x)$ denotes the permutation of the bits in x according to π_i and $Q(y)$ denotes the output of the convolutional code Q on input y.

Given an integer k, we define the repeat-k-times encoder, r_k, to be the encoder that just repeats each of its input bits k times. Given a convolutional encoder Q, a message length n, and a permutation π of length kn, we define the *repeat-convolute code* [5] to be the code whose encoder maps an input $x \in \{0,1\}^n$ to $(x, Q(\pi(r_k(x))))$. That is, each bit of the input is repeated k times, the resulting kn bits are permuted, and then fed through the convolutional encoder. We also assume that the input x is output as well. While some implementations do not include x in the output, its exclusion cannot improve the minimum distance so we assume it appears. The number k is called the *repetition factor* of the code. When the convolutional encoder Q is the accumulator (*i.e.,* the map $Q(x)_j = \sum_{i=1}^{j} x_i$), this code is called a *repeat-accumulate (RA) code* [5].

Given two convolutional encoders Q_o and Q_i that output h_o and h_i bits per time step respectively, an integer n, and a permutation π of length $h_o n$, we define the depth-two *serially concatenated convolutional code* [2], [9] to be the rate $1/h_o h_i$ code whose encoder maps an input $x \in \{0,1\}^n$ to the codeword $Q_i(\pi(Q_o(x)))$. The codes Q_o and Q_i are called *outer* and *inner* codes, respectively. A classical example of serially concatenated convolutional codes, and that considered in [7], is a rate $1/4$ code given by the map $(\pi(x, L_o(x)), L_i(\pi((x, L_o(x)))))$, where L_o and L_i are rate-1 convolutional codes. This fits into our framework with $Q_o(x) = (x, L_o(x))$ and $Q_i(x) = (x, L_i(x))$.

One can allow greater depth in serial concatenation. The only codes of greater depth that we consider will be repeat-accumulate-accumulate codes (RAA). These are specified by a repetition factor k, an integer n, and two permutations π_1 and π_2 of length kn. Setting Q_1 and Q_2 to be accumulators, the resulting code maps an input x to $Q_2(\pi_2(Q_1(\pi_1(r_k(x)))))$.

We can generalize each of these constructions by allowing the component codes to be automata codes. In this case, we will refer to the resulting codes as *generalized parallel concatenated Turbo codes*, *generalized repeat convolute codes*, and *generalized serially concatenated codes*.

In practice, some extra bits are often appended to the input x so as to guarantee that some of the encoders return to the zero state. As this addition does not substantially increase the minimum distance of the resulting code, we will not consider this technicality in this paper.

B. Previous results

Kahale and Urbanke [7] proved that if one builds a parallel concatenated Turbo code from a random interleaver and convolutional encoders of memory at most M, then the resulting code has minimum distance at most $n^{1-2/k} \log^{O(1)} n$ and at least $\Omega(n^{1-2/k})$ with high probability. For rate $1/4$ serially concatenated convolutional codes of the form mentioned in the previous section with a random interleaver, they proved that the resulting code has minimum distance at most $2^{O(M_i)} n^{1-2/d_o} \log^{O(1)} n$ and at least $\Omega(n^{1-2/d_o})$ with high probability, where d_o is the free distance of the outer code and M_i is the inner code memory.

For parallel concatenated Turbo codes with two branches, Breiling [4] proved that no construction could be much better than a random code: if the constituent codes have memory M, then the minimum distance of the resulting code is $O(2^{O(M)} \log n)$.

Serially concatenated codes of depth greater than 2 were studied by Pfister and Siegel [8], who performed experimental analyses of the serial concatenation of repetition codes with l levels of accumulators connected by random interleavers, and theoretical analyses of concatenations of a repetition code with certain rate-1 codes for large l. Their experimental results indicate that the average minimum distance of the ensemble starts becoming good for $l \geq 2$, which is consistent with our theorem. For certain rate-1 codes and l going to infinity, they proved their codes could become asymptotically good.

C. Our results

In Section II-A, we upper bound the minimum distance of generalized repeat-convolute codes and generalized parallel concatenated Turbo codes. We prove that generalized repeat-convolute codes of message length n, memory M, and repetition factor k have minimum distance at most $O(n^{1-1/k} M^{1/k})$. The same bound holds for generalized parallel concatenated Turbo codes with k branches, memory M, and message length n. Therefore such codes are asymptotically bad when k is constant and M is sub-linear in n. Note that M sub-linear in n corresponds to the case when the size of the corresponding trellis is sub-exponential, and so it includes the cases in which the codes have natural sub-exponential time iterative decoding algorithms. This proof uses techniques introduced by Ajtai [1] for obtaining time-space trade-offs for branching programs. Comparing our upper bound with the $2^{O(M)}n^{1-2/k}\log^{O(1)} n$ high-probability upper bound of Kahale and Urbanke for parallel concatenated codes, we see that our bound has a much better dependence on M and a slightly worse dependence on k. A similar relation holds between our bound and the $O(2^{O(M)}\log n)$ upper bound of Breiling [4] for parallel concatenated codes with 2 branches.

In Section II-B, we restrict our attention to linear repeat-convolute codes, and prove that every repeat-convolute code with repetition factor k in which the convolutional encoder has memory M has minimum distance at most $2^{O(M)}n^{1-1/\lceil k/2\rceil}\log n$. For even k, this bound is very close to the high-probability bound of Kahale and Urbanke.

In Section III-A, we study serially concatenated codes with two levels, and prove that if the outer code has memory M_o and the inner code has memory M_i, then the resulting code has minimum distance at most $O(n^{1-1/h_o(M_o+2)}M_i^{1/h_o(M_o+2)})$. Accordingly, we see that such codes are asymptotically bad when M_o, h_o and h_i are constants and M_i is sub-linear in n. The proof uses similar techniques to those used in Section II-A. When specialized to the classical rate $1/4$ construction of serially concatenated convolutional codes considered by Kahale and Urbanke [7], our bound on the minimum distance becomes $O(n^{1-1/(2M_o+4)}M_i^{1/(2M_o+4)})$. Comparing this with the high-probability $O(n^{1-2/d_o}2^{O(M_i)}\log^{O(1)} n)$ upper bound of Kahale and Urbanke, we see that our bound is better in terms of M_i, comparable in terms of d_o, and close to their existential bound of $\Omega(n^{1-2/d_o})$.

Finally, in Section III-B, we show that serially concatenated codes of depth greater than two can be asymptotically good, even if the constituent codes are repetition codes and accumulators. In particular, we prove that randomly constructed RAA codes are asymptotically good with constant probability.

Throughout this paper, our goal is to obtain asymptotic bounds. We make no claim about the suitability of our bounds for any particular finite n.

II. Repeat-convolute-like and parallel Turbo-like codes

In this section we consider codes that are obtained by serially concatenating a repeat-k-times code r_k with any code Q that can be encoded by an automata (transducer) with at most 2^M states and one output bit per transition. More precisely, if Q is such an encoder, π is a permutation of length kn, and r_k is the repeat-k-times map, we define the *generalized repeat-convolute code* to be the code whose encoder $C_{k,\pi,Q}$ maps a string $x \in \{0,1\}^n$ to $C_{k,\pi,Q}(x) := (x, Q(\pi(r_k(x))))$.

We consider also the parallel concatenated variations of these codes. Given an automata Q with at most 2^M states and one output bit per transition and k permutations $\pi_1,\ldots,\pi_k$ each of length n, we define the *generalized parallel concatenated Turbo code* [3], [9] to be the code whose encoder $P_{\pi_1,\ldots,\pi_k,Q}$ maps an input $x \in \{0,1\}^n$ to $P_{\pi_1,\ldots,\pi_k,Q}(x) := (x, Q(\pi_1(x)),\ldots, Q(\pi_k(x)))$.

A. An upper bound on the minimum distance

Theorem 1: Let $k \geq 2$ be a constant integer, Q an automata encoder with at most 2^M states, and n an integer.

Let π be a permutation of length kn. If $n \geq 2^k kM$, then the minimum distance of the generalized repeat-convolute code encoded by $C_{k,\pi,Q}$ is at most

$$3k^2n^{1-1/k}M^{1/k} + 2^k kM + k + 1.$$

The same bound holds for the parallel concatenated Turbo-like code encoded by $P_{\pi_1,\ldots,\pi_k,Q}$, where $\pi_1,\ldots,\pi_k$ are permutations each of length n.

Proof: We first consider the case of repeat-convolute-like codes. We explain at the end of the proof how to modify the proof to handle parallel concatenated Turbo-like codes.

To prove this theorem, we make use of a technique introduced by Ajtai [1] for proving time-space trade-offs for branching programs. In particular, for an input x of length n, the encoding action of Q is naturally divided into kn time steps in which the automata reads a bit of $\pi(x)$, outputs a bit, and changes state. For convenience, we will let $I = \{1,\ldots,kn\}$ denote the set of time steps, and we will let $s_i(x)$ denote the state of Q on input $\pi(r_k(x))$ at the end of the i'th time step.

Let C denote the encoder $C_{k,\pi,Q}$. Following Ajtai [1], we will prove prove the existence of two input strings, x and y, a set $U \subset \{1,\ldots,n\}$ of size at most $3k^2n^{1-1/k}M^{1/k}+1$, and $J \subset I$ of size at most $2^k kM + k$ such that x and y may only differ on bits with indices in U and $s_i(x)$ and $s_i(y)$ may only differ on time steps with indices in J. The claimed bound on the minimum distance of code encoded by C will follow from the existance of these two strings.

To construct the set J, we first divide the set of time steps I into b consecutive intervals, where b is a parameter we will specify later. We choose these intervals so that each has size $\lfloor kn/b \rfloor$ or $\lceil kn/b \rceil$. For example, if $k=2$, $n=4$, and $b=3$ we can divide $I=\{1,\ldots,8\}$ into the intervals $[1,3]$, $[4,6]$, and $[7,8]$.

For each index of an input bit $i \in \{1,\ldots,n\}$, we let S_i denote the multiset of time intervals in which Q reads input bit i (this is a multiset as a bit can appear multiple times in the same interval). As each bit appears k times, the multisets S_i each have size k. As there are b intervals, there are at most b^k possible k-multisets of intervals. So, there exists a set $U \subset \{1,\ldots,n\}$ of size at least n/b^k and a multiset of intervals, S, such that for all $i \in U$, $S_i = S$. Let U be such a set with $|U| = \lceil n/b^k \rceil$ and let T be the corresponding set of intervals. Let $l = |T|$. The set J will be the union of the intervals in T.

Let $t_1,\ldots,t_l$ be the last times in the time intervals in T (e.g., in the above example the last time of the interval $[4,6]$ is 6). For each $x \in \{0,1\}^n$, that is zero outside U, we consider the vector of states of Q at times $t_1,\ldots,t_l$ on input $\pi(r_k(x))$: $\{s_{t_i}(x)\}_{i=1}^l$. As the number of such possible sequences is at most 2^{Ml} and the number of x that are zero outside U is $2^{|U|}$, if

$$2^{|U|} > 2^{Ml}, \tag{1}$$

then there should exist two different strings x and y that are both zero outside of U and such that $s_{t_i}(x) = s_{t_i}(y)$ for $i=1,\ldots,l$. To make sure that (1) is satisfied, we set

$$b = \left\lceil \left(\frac{n}{kM}\right)^{1/k} \right\rceil - 1.$$

Our assumption that $n \geq 2^k kM$ ensures that $b \geq 1$. Now, since

1) x and y agree outside U,
2) the bits in U only appear in time intervals in T, and
3) Q traverses the same states at the ends of time intervals in T on inputs $\pi(r_k(x))$ and $\pi(r_k(y))$,

Q must traverse the same states at all times in intervals outside T on inputs $\pi(r_k(x))$ and $\pi(r_k(y))$. Thus, the bits output by Q in time steps outside intervals in T must be the same on inputs $\pi(r_k(x))$ and $\pi(r_k(y))$. So $Q(\pi(r_k(x)))$ and $Q(\pi(r_k(y)))$ can only disagree on bits output during times in the intervals in T, and hence on at most $l \lceil kn/b \rceil$ bits. This means that the distance between $C(x)$ and $C(y)$ is at most

$$\begin{aligned}
& |U| + l\lceil kn/b \rceil \\
\leq\ & \left\lceil \frac{n}{b^k} \right\rceil + k\lceil kn/b \rceil, \text{ as } |U| = \lceil n/b^k \rceil \text{ and } l \leq k, \\
\leq\ & \frac{n}{b^k} + 1 + \frac{k^2 n}{b} + k \\
\leq\ & \frac{n}{\left\lceil \left(\frac{n}{kM}\right)^{1/k} - 1 \right\rceil^k} + \frac{k^2 n}{\left(\frac{n}{kM}\right)^{1/k} - 1} + k + 1 \\
\leq\ & \frac{n}{\left(\left(\frac{n}{kM}\right)^{1/k} - 1 \right)^k} + \frac{k^2 n}{\left(\frac{n}{kM}\right)^{1/k}} \frac{\left(\frac{n}{kM}\right)^{1/k}}{\left(\frac{n}{kM}\right)^{1/k} - 1} + k + 1 \\
=\ & \frac{n}{\left(\frac{n}{kM}\right)} \left(\frac{\left(\frac{n}{kM}\right)^{1/k}}{\left(\frac{n}{kM}\right)^{1/k} - 1} \right)^k + \frac{k^2 n}{\left(\frac{n}{kM}\right)^{1/k}} \frac{\left(\frac{n}{kM}\right)^{1/k}}{\left(\frac{n}{kM}\right)^{1/k} - 1} + k + 1 \\
\leq\ & \frac{n}{\left(\frac{n}{kM}\right)} 2^k + 2\frac{k^2 n}{\left(\frac{n}{kM}\right)^{1/k}} + k + 1, \text{ as } n \geq 2^k kM \\
=\ & 2^k kM + 2k^2 n^{1-1/k} M^{1/k} k^{1/k} + k + 1, \\
\leq\ & 3k^2 n^{1-1/k} M^{1/k} + 2^k kM + k + 1,
\end{aligned}$$

as $k^{1/k} \leq 3/2$.

Now, we explain how to apply the proof to the generalized parallel concatenated Turbo codes. Let $\pi_1, \ldots, \pi_k$ be permutations each of length n, Q be an automata encoder with at most 2^M states and consider the parallel concatenated Turbo-like code encoded by $P_{\pi_1,\ldots,\pi_k,Q}$. Let π be the length-kn permutation constructed from $\pi_1, \ldots, \pi_k$ and the repetition map r_k in such a way that $\pi(r_k(x)) = (\pi_1(x), \ldots, \pi_k(x))$ for all $x \in \{0,1\}^n$. Let Q' be the time-varying automata that works exactly like Q except that it is goes back to the start state at the time steps $n+1, 2n+1, \ldots, (k-1)n+1$. Thus $P_{\pi_1,\ldots,\pi_k,Q}(x) = (x, Q'(\pi(r_k(x))))$ for all $x \in \{0,1\}^n$. In other words, we can realize $P_{\pi_1,\ldots,\pi_k,Q}$ as a time-varying repeat-convolute-like code whose encoder $C_{k,\pi,Q'}$ maps a string $x \in \{0,1\}^n$ to $C_{k,\pi,Q'}(x) := (x, Q'(\pi(r_k(x))))$. To extend the minimum distance bound to generalized parallel concatenated Turbo codes, it is sufficient to note that the proof of Theorem 1 works without any changes for time-varying generalized repeat-convolute codes, which the are natural generalizations of repeat-convolute-like codes obtained by allowing the automata to be time-varying [2].

■

Corollary 1: Let k be a constant. Then, every generalized repeat-convolute code with input length n and memory M and repetition factor k and every generalized parallel concatenated Turbo code with input length n, convolutional encoder memory M and k branches has minimum distance $O(n^{1-1/k}M^{1/k})$. Thus, such codes cannot be asymptotically good for M sub-linear in n.

This means that if we allow M to grow like $\log n$, or even like $n^{1-\epsilon}$ for some $\epsilon > 0$, the minimum relative distance of the code will still go to zero. Moreover, M sub-linear in n corresponds to the case in which the size of the corresponding trellis is sub-exponential, and therefore it includes all the cases in which such codes have natural sub-exponential-time iterative decoding algorithms.

It is interesting to compare our bound with that obtained by Kahale and Urbanke [7], who proved that a randomly chosen parallel concatenated code with k branches has minimum distance $2^{O(M)} n^{1-2/k} \log^{O(1)} n$ with high probability. Theorem 1 has a much better dependence on M and a slightly worse dependence on n. A similar comparison can be made with the bound of Breiling [4], who proved that every parallel concatenated code with $k = 2$ branches has minimum distance at most $2^{O(M)} \log n$. In the next section, we prove an upper bound whose dependence on M and n is asymptotically similar to that obtained by these authors.

B. Improving the bound in the linear, low-memory case

We now prove that every repeat-convolute code with repetition factor k, memory M, and input length n, and every parallel concatenated Turbo code with k branches, memory M, and input length n has minimum distance at most $O(2^{O(M)} n^{1-1/\lceil k/2 \rceil} \log n)$.

Theorem 2: Let $k \geq 2$ and n be integers and let Q be a convolutional encoder with memory M.

Let π be a permutation of length kn. Assuming $M < (\log_2 n - 3)/k$, the minimum distance of the repeat-convolute code encoded by $C_{k,\pi,Q}$ is at most

$$16k^2 n^{1-1/\lceil k/2 \rceil} 2^{2M} \log_2 n + 6 \log_2 n.$$

Thus, when k is constant, the minimum distance of the repeat-convolute code encoded by $C_{k,\pi,Q}$ is

$$2^{O(M)} n^{1-1/\lceil k/2 \rceil} \log n.$$

If $M < (\log_2 n - 3)/k - \log_2 k$, the same bounds hold for the parallel concatenated Turbo code encoded by $P_{\pi_1,\ldots,\pi_k,Q}$, where $\pi_1, \ldots, \pi_k$ are permutations each of length n.

If we ignore constant factor, we see that our bound asymptotically matches the bound of Breiling for parallel concatenated Turbo codes with two branches [4] (*i.e.* when $k = 2$). The constant factor in our bound is however larger. We have not attempted to optimize the

[2] A time-varying automata is specified by a state transition map $\delta : \mathcal{S} \times \{0,1\} \times \mathcal{T} \to S$ and an output map $\gamma : \mathcal{S} \times \{0,1\} \times \mathcal{T} \to \{0,1\}$, where $\mathcal{S}$ is the set of states and $\mathcal{T} = \{1, 2, 3, \ldots\}$ is the set of time steps

constants in our proof. Our main objective is to establish, when $k > 2$, an asymptotic bound in terms of the growth of n and M.

When k is even, our bound asymptotically matches also the bound for randomly constructed parallel-concatenated Turbo codes proved by Kahale and Urbanke [7]. As Kahale and Urbanke proved similar lower bounds for $k \geq 3$, we learn that the minimum distances of randomly constructed Turbo codes is not too different from that of optimally constructed Turbo codes.

Proof of Theorem 2:

First we note that if the convolutional code is non-recursive, it is trivial to show that on input 10^{n-1} (i.e., a 1 followed by $n-1$ zeros) the output codeword will have weight at most $k2^M$. Thus, without loss of generality, we assume that the convolutional code is recursive.

Our proof of Theorem 2 will make use of the following fact about linear convolutional codes mentioned in Kahale-Urbanke [7]:

Lemma 1: [7] For any recursive convolutional encoder Q of memory M, there is a number $\delta \leq 2^M$ such that, after processing any input of the form $0^*10^{j\delta-1}1$ for any positive integer j, Q comes back to the zero state after processing the second 1. In particular, the weight of the output of Q after processing any such input is at most $j\delta$.

We consider first the case of repeat-convolute codes. We explain at the end of the proof how to customize the proof to the setting parallel concatenated Turbo codes.

Let δ be the number shown to exist in Lemma 1 for convolutional code Q. As in [7] and [4], we will construct a low-weight input x on which $C_{k,\pi,Q}(x)$ also has low-weight by taking the exclusive-or of a small number of weight 2 inputs each of whose two 1s are separated by a low multiple of δ 0s. As the code encoded by $C_{k,\pi,Q}$ is a linear code, its minimum distance equals the minimum weight of its codewords.

To construct this low-weight input, we first note that every bit of the input x appears exactly k times in the string $\pi(r_k(x))$. For every $i \in \{1, \ldots, n\}$ and every $1 \leq j \leq k$, let $\sigma_j(i)$ denote the position of the j'th appearance of the bit i in $\pi(r_k(x))$. For each bit, i, consider the sequence $(\sigma_1(i) \bmod \delta, \sigma_2(i) \bmod \delta, \ldots, \sigma_k(i) \bmod \delta)$. Since there are at most δ^k such possible sequences and n input bits, there exists a set $U \subset \{1, \ldots, n\}$ of size at least $\lceil n/\delta^k \rceil$ such that all of its elements induce the same sequence. That is, for all i and j in U, $\sigma_l(i) - \sigma_l(j)$ is divisible by δ for all $1 \leq l \leq k$. From now on, we will focus on the input bits with indices in U, and construct a low-weight codeword by setting some of these bits to 1.

As in the proof of Theorem 1, we now partition the set of time steps $\{1, \ldots, kn\}$ into b consecutive intervals, $I_1, I_2, \ldots, I_b$, each of length $\lceil kn/b \rceil$ or $\lfloor kn/b \rfloor$, where b is a parameter we will specify later. For every index $i \in U$, we let the *signature* of i be the k-tuple whose j'th component is the index of the interval to which $\sigma_j(i)$ belongs.

Now, we construct a hypergraph $\mathcal{H}$ as follows: $\mathcal{H}$ has k parts, each part consisting of b vertices which are identified with the intervals $I_1, \ldots, I_b$. There are $|U|$ hyperedges in $\mathcal{H}$, one corresponding to each input bit with index in U. The vertices contained in the hyperedge are determined by the signature of the corresponding bit: if input bit i has signature $(i_1, i_2, \ldots, i_k)$, then the i'th hyperedge contains the i_j'th vertex of the j'th part, for every $j = 1, \ldots, k$. Thus, $\mathcal{H}$ is a k-partite k-uniform hypergraph (i.e., each hyperedge contains exactly k vertices, each from a different part) with b vertices in each part and $|U| \geq n/\delta^k$ edges.

We now define a family of subgraphs such that if $\mathcal{H}$ contains one of these subgraphs, then the code encoded by $C_{k,\pi,Q}$ must have a low-weight codeword. We define an ℓ-forbidden subgraph S of $\mathcal{H}$ to be a set of at least one and at most 2ℓ hyperedges in $\mathcal{H}$ such that each vertex in $\mathcal{H}$ is contained in an even number of the hyperedges of S. (One can think of an ℓ-forbidden subgraph as a generalization of a cycle of length ℓ to hypergraphs). In Lemma 2, we prove that if $\mathcal{H}$ contains an ℓ-forbidden subgraph then the code encoded by $C_{k,\pi,Q}$ has a codeword of weight at most $2\ell + \ell \lceil kn/b \rceil$. In Lemma 3, we prove that if $\mathcal{H}$ contains at least $4b^{\lceil k/2 \rceil}$ edges, then it contains a $k \log_2 b$ forbidden subgraph. As $\mathcal{H}$ has at least n/δ^k edges, if we set

$$b = \left\lfloor \left(\frac{n}{4\delta^k} \right)^{1/\lceil k/2 \rceil} \right\rfloor,$$

then $n/\delta^k \geq 4b^{\lceil k/2 \rceil}$; so, Lemma 3 will imply that $\mathcal{H}$ has a $k \log_2 b$-forbidden subgraph and Lemma 2 will imply that the code encoded by $C_{k,\pi,Q}$ has a codeword of weight at most $2k \log_2 b + k \log_2 b \lceil kn/b \rceil$. Plugging in the value we have chosen for b, we find that the minimum distance of the code encoded by $C_{k,\pi,Q}$ is at most

$$\begin{aligned} & 2k \log_2 b + k \lceil kn/b \rceil \log_2 b \\ \leq\ & 4 \log_2 n + 2 \lceil kn/b \rceil \log_2 n, \ \text{ as } k \log_2 b \leq 2 \log_2 n, \\ \leq\ & 2\frac{kn}{b} \log_2 n + 6 \log_2 n \\ \leq\ & 2\frac{kn}{\left(\frac{n}{4\delta^k}\right)^{1/\lceil k/2 \rceil} - 1} \log_2 n + 6 \log_2 n \\ =\ & 2kn\frac{\delta^{k/\lceil k/2 \rceil}}{n^{1/\lceil k/2 \rceil}} \frac{1}{(1/4)^{1/\lceil k/2 \rceil} - (\delta^k/n)^{1/\lceil k/2 \rceil}} \log_2 n + 6 \log_2 n \\ \leq\ & 2kn\frac{\delta^{k/\lceil k/2 \rceil}}{n^{1/\lceil k/2 \rceil}} 8 \lceil k/2 \rceil \log_2 n + 6 \log_2 n, \\ \leq\ & 16k^2 n^{1-1/\lceil k/2 \rceil} 2^{2M} \log_2 n + 6 \log_2 n, \end{aligned}$$

where the last inequality follows from $\delta \leq 2^M$, and the second-to-last inequality follows from combining this inequality with the assumption in the theorem that $n \geq 8 \cdot 2^{kM}$ to show $n \geq 8\delta^k$, and applying the bound $(4^{-x} - 8^{-x})/x \geq 1/8$ for all $0 \leq x \leq 1$.

Now, we explain how to modify the proof to handle parallel concatenated Turbo codes. Let $\pi_1, \ldots, \pi_k$ be permutations each of length n, Q be a recursive convolutional encoder with memory M, and consider the parallel concatenated Turbo code encoded by $P = P_{\pi_1,\ldots,\pi_k,Q}$. We can associate with P the repeat-convolute encoder $C = C_{k,\pi,Q}$, where π is the length-kn permutation constructed from $\pi_1, \ldots, \pi_k$ and the repetition map r_k in such a way that $\pi(r_k(x)) = (\pi_1(x), \ldots, \pi_k(x))$ for all $x \in \{0,1\}^n$. To extend the bound to parallel-concatenated Turbo codes, we will force P and C have the same input-output behavior on the special low-weight inputs considered in this proof. To do this, we set

$$b = k \left\lfloor \left(\frac{n}{4\delta^k} \right)^{1/\lceil k/2 \rceil} / k \right\rfloor,$$

and require that $n+1$, $2n+1$, …, $(k-1)n+1$ be the first times in the intervals in which they appear. We then guarantee that, on the special low-weight inputs considered in the proof, the convolutional encoder will be in the zero state at steps $n+1$, $2n+1$, ..., $(k-1)n+1$, and so it will have the same output as P. The rest of the analysis is similar, except that we use the slightly stronger assumption $M < (\log_2 n - 3)/k - \log_2 k$.

■

Lemma 2: If $\mathcal{H}$ contains an ℓ-forbidden sub-hypergraph, then there is an input sequence of weight at most 2ℓ whose corresponding codeword in $C_{k,\pi,Q}$ has weight at most $2\ell + \ell\lceil kn/b\rceil$.

Proof: Let S denote the set of hyperedges of the ℓ-forbidden sub-hypergraph in $\mathcal{H}$, and consider the set B of bits of the input that correspond to the hyperedges of S. By definition, $B \subseteq U$ and $|B| \leq 2\ell$. We construct an input x of weight at most 2ℓ by setting the bits in B to 1 and other bits to 0, and consider the codeword corresponding to x: $(x, Q(\pi(r_k(x))))$. As each vertex of $\mathcal{H}$ is contained in an even number of the hyperedges in S, each interval in $\mathcal{I}$ contains an even number of bits that are 1 in $\pi(r_k(x))$. Thus, by the definition of U and Lemma 1, $Q(\pi(r_k(x)))$ is zero everywhere except inside those intervals of $\mathcal{I}$ that contain a bit that is 1 in $\pi(r_k(x))$. Since there are at most ℓ such intervals, the weight of $Q(\pi(r_k(x)))$ is at most $\ell\lceil kn/b\rceil$. Therefore, the weight of the codeword corresponding to x is at most $2\ell + \ell\lceil kn/b\rceil$. ■

Lemma 3: Every k-partite k-uniform hypergraph $\mathcal{H}$ with b vertices in each part and at least $4b^{\lceil k/2\rceil}$ hyper-edges contains a $k\log_2 b$-forbidden sub-hypergraph.

Proof: We construct a bipartite graph G from H as follows: For every $\lceil k/2\rceil$-tuple $(i_1, i_2, ..., i_{\lceil k/2\rceil})$ where i_j is a vertex in the j'th part of $\mathcal{H}$, we put a vertex in the first part of G, and for every $\lfloor k/2\rfloor$-tuple $(i_{\lceil k/2\rceil+1}, \ldots, i_k)$ where i_j is a vertex in the j'th part of $\mathcal{H}$, we put a vertex in the second part of G. If there is a hyperedge $\{i_1, i_2, \ldots, i_k\}$ in $\mathcal{H}$, where i_j is a vertex of the j'th part, we connect the vertices $(i_1, i_2, ..., i_{\lceil k/2\rceil})$ and $(i_{\lceil k/2\rceil+1}, \ldots, i_k)$ in G.

By the above construction, each edge in G corresponds to a hyperedge in $\mathcal{H}$. There are at least $4b^{\lceil k/2\rceil}$ edges and at most $2b^{\lceil k/2\rceil}$ vertices in G. Thus, by Lemma 4 below, G has a cycle of length at most $2\log_2(2b^{\lceil k/2\rceil}) < 2k\log_2 b$. It is easy to see that the hyperedges corresponding to the edges of this cycle constitute a $k\log_2 b$-forbidden sub-hypergraph in $\mathcal{H}$. ■

Lemma 4: Let G be a graph on n vertices with at least $2n$ edges. Then, G has a cycle of length at most $2\log_2 n$.

Proof: We first prove the theorem in the case that every vertex of G has degree at least 3. In this case, if the shortest cycle in the graph had length $2d+1$, then a breadth-first search tree of depth d from any vertex of the graph would contain at least $1 + 3\sum_{i=0}^{d-1} 2^i = 3\cdot 2^d - 2$ distinct vertices. As $3 \cdot 2^{\log_2 n} - 2 > n$, this would be a contradiction for $d \geq \log_2 n$. So, the graph must contain a cycle of length at most $2\log_2 n$.

We may prove the lemma in general by induction on n. Assume the lemma has been proved for all graphs with fewer than n vertices, and let G be a graph on n vertices with at least $2n$ edges. If the degree of every node in G is at least 3, then G has a cycle of length at most $2\log_2 n$ by the preceding argument. On the other hand, if G has a vertex of degree 2, we consider the graph G' obtained by deleting this vertex and its two adjacent edges. The graph G' has $n-1$ vertices and at least $2(n-1)$ edges, and so by induction has a cycle of length at most $2\log_2(n-1)$. As G' is a subgraph of G, G also has a cycle of length at most $2\log_2(n-1) \leq 2\log_2 n$, which proves the lemma. ■

III. Serially concatenated codes

In this section, we consider codes that are obtained by serially concatenating convolutional codes and, more generally, automata codes. In Section III-A, we prove an upper bound on the minimum distance of the concatenation of a low-memory outer automata encoder with an arbitrary inner automata encoder. In particular, we prove that if the memory of the outer code is constant and the memory of the inner code is sub-linear, then the code is asymptotically bad. In contrast, in Section III-B, we prove that if the input is first passed through a repetition code and a random permutation, then the code is asymptotically good with constant probability, even if both convolutional encoders are accumulators.

A. Upper bound on the minimum distance when the outer code is weak

In this section, we consider the serial concatenation of automata codes. We assume that each automata outputs a constant number of bits per transition. This class of codes includes the serially concatenated convolutional codes introduced by Benedetto, Divsalar, Montorsi and Pollara [2] and studied by Kahale and Urbanke [7]. If the outer code has constant memory and the inner code has sub-linear memory, then our bound implies that the code cannot be asymptotically good.

Formally, we assume that Q_o (Q_i, respectively) is an automata encoder with at most 2^{M_o} (2^{M_i}, respectively) states and h_o (h_i, respectively) output bits per transition. For an integer n and a permutation π of length $h_o n$, we define $C_{Q_o,Q_i,\pi}$ to be the encoder that maps an input $x \in \{0,1\}^n$ to the codeword $C_{Q_o,Q_i,\pi}(x) := Q_i(\pi(Q_o(x))) \in \{0,1\}^{h_o h_i n}$. We will assume without loss of generality that Q_o, Q_i, and π are such that this mapping is an injective mapping. The encoders Q_o and Q_i are called the outer and inner encoders, respectively.

Theorem 3: Let Q_o be an automata encoder with at most 2^{M_o} states that outputs h_o bits at each time step, and let Q_i be an automata encoder with at most 2^{M_i} states that outputs h_i bits at each time step. For any positive integer n and any permutation π of length nh_o, the minimum distance of the serially-concatenated code encoded by $C_{Q_o,Q_i,\pi}$ is at most

$$3h_o^2 h_i (M_o+2) n^{1-\frac{1}{h_o(M_o+2)}} M_i^{\frac{1}{h_o(M_o+2)}}.$$

In particular, if M_o is constant (and h_i and h_0 are constants), the minimum distance of the serially-concatenated code encoded by $C_{Q_o,Q_i,\pi}$ is

$$O(n^{1-\frac{1}{h_o(M_o+2)}} M_i^{\frac{1}{h_o(M_o+2)}}),$$

and consequently any such family of codes is asymptotically bad as long as M_i is sub-linear in n.

Proof: The proof follows the same outline as the proof of Theorem 1. We begin by setting $I_o = \{1, \ldots, n\}$ to be the set of times steps in the computation of Q_o on input $x \in \{0,1\}^n$, and setting $I_i = \{1, \ldots, h_o n\}$ to be the set of times steps in the computation of Q_i on input $\pi\left(Q_o(x)\right) \in \{0,1\}^{h_o n}$. We similarly, let $\left\{s_o^{(t)}(x)\right\}_{t \in I_o}$ denote the sequence of states traversed by Q_o on input x and $\left\{s_i^{(t)}(x)\right\}_{t \in I_i}$ denote the sequence of states traversed by Q_i on input $\pi\left(Q_o(x)\right)$.

To prove the claimed bound on the minimum distance of the code encoded by $C_{Q_o,Q_i,\pi}$, we will prove the existence of two distinct input strings x and y, a set $V \subset \{1, \ldots, n\}$, a set $J_o \subset I_o$, and a set $J_i \subset I_i$ such that x and y are both 0 on bits not in V, $s_o^{(t)}(x)$ and $s_o^{(t)}(y)$ only differ for $t \in J_o$, and $s_i^{(t)}(x)$ and $s_i^{(t)}(y)$ only differ for $t \in J_i$. The minimum distance bound will then follow from an upper bound on the size of J_i.

To construct these sets, we make use of parameters m_o and m_i to be determined later. We first partition the set I_o into $b_o \overset{\text{def}}{=} \lfloor n/m_o \rfloor$ intervals each of size m_o or m_o+1, and we partition the set I_i into $b_i \overset{\text{def}}{=} \lfloor n h_o/m_i \rfloor$ intervals each of size m_i or m_i+1.

As Q_o outputs at most $(m_o+1)h_o$ bits during the time steps in an interval in I_o, the bits output by Q_o during an interval in I_o are read by Q_i during at most $(m_o+1)h_o$ intervals in I_i. As there are fewer than $(b_i)^{(m_o+1)h_o}$ sets of at most $(m_o+1)h_o$ intervals in I_i, there exists a set of at least $b_o/(b_i)^{(m_o+1)h_o}$ intervals in I_o such that all the bits output by Q_o during these intervals are read by Q_i during a single set of at most $(m_o+1)h_o$ intervals in I_i. Let U denote the set of at least $b_o/(b_i)^{(m_o+1)h_o}$ intervals in I_o and let T denote the corresponding set of at most $(m_o+1)h_o$ intervals in I_i. We then let V denote the set of input bits read by Q_o during the intervals in U. As all the intervals in I_o have size at least m_o, we have $|V| \geq m_o\,|U|$. The set J_o will be the union of the intervals in U and J_i will be the union of the intervals in T.

Let $\{u_j\}_{j=1}^{|U|}$ and $\{t_j\}_{j=1}^{|T|}$ denote the last time steps in the intervals in U and T respectively. For each $x \in \{0,1\}^n$ that is zero outside V, we consider $\left(s_o^{(u_j)}(x)\right)_{j=1}^{|U|}$, the sequence of states traversed by Q_o on x at times $u_1, \ldots, u_{|U|}$, and, $\left(s_i^{(t_j)}(x)\right)_{j=1}^{|T|}$, the sequence of states traversed by Q_i on input $\pi(Q_o(x))$ at times $t_1, \ldots, t_{|T|}$. There are at most $2^{M_o|U|}2^{M_i|T|}$ such pairs of sequences. So, if

$$2^{M_o|U|}2^{M_i|T|} < 2^{|V|}, \tag{2}$$

then there are two distinct x and y in $\{0,1\}^n$ that are both 0 outside V and a pair of sequences $\left(s_i^{(t_j)}\right)_{j=1}^{|T|}$ and $\left(s_o^{(u_j)}\right)_{j=1}^{|U|}$ such that $s_i^{(t_j)}(x) = s_i^{(t_j)}(y) = s_i^{(t_j)}$ for all $1 \leq j \leq |T|$ and $s_o^{(u_j)}(x) = s_o^{(u_j)}(y) = s_o^{(u_j)}$ for all $1 \leq j \leq |U|$. This means that the bits output and states traversed by Q_o on inputs x and y are the same at time steps outside the time intervals in U, and therefore the bits output and states traversed by Q_i on inputs $\pi(Q_o(x))$ and $\pi(Q_o(y))$ are the same outside time steps in intervals in T. Thus

$$0 < d(C_{Q_i,Q_o,\pi}(x), C_{Q_i,Q_o,\pi}(y)) \leq m_i h_i\,|T| \leq (m_o+1)m_i h_o h_i. \tag{3}$$

As this bound assumes (2), we will now show that for

$$m_o = M_o + 1, \text{ and}$$
$$m_i = 3h_o n^{1-\frac{1}{(M_o+2)h_o}} (M_i)^{\frac{1}{(M_o+2)h_o}},$$

this assumption is true.

Our setting of m_o reduces (2) to

$$|U| \geq |T|\,M_i,$$

which would be implied by

$$\frac{b_o}{b_i^{(m_o+1)h_o}} > (m_o+1)h_o M_i. \tag{4}$$

To derive this inequality, we first note that since $x^{2/x} < 3$ for $x \geq 1$,

$$m_i > h_o n^{1-\frac{1}{(M_o+2)h_o}} (((m_o+1)h_o)^2 M_i)^{\frac{1}{(M_o+2)h_o}}.$$

Rearranging terms, we find this implies

$$\left(\frac{n}{(m_o+1)^2 h_o^2 M_i}\right)^{\frac{1}{(m_o+1)h_o}} > \frac{n h_o}{m_i} \geq b_i.$$

Again rearranging terms, we obtain

$$n > b_i^{(m_o+1)h_o}(m_o+1)^2 h_o^2 M_i \geq b_i^{(m_o+1)h_o}(m_o+1)m_o h_o M_i + m_o,$$

which implies

$$\left\lfloor \frac{n}{m_o} \right\rfloor > b_i^{(m_o+1)h_o}(m_o+1)h_o M_i.$$

By now dividing both sides by $b_i^{(m_o+1)h_o}$ and recalling $b_o = \left\lfloor \frac{n}{m_o} \right\rfloor$, we derive (4).

Finally, the bound on the minimum distance of the code now follows by substituting the chosen values for m_o and m_i into (3).

■

We now compare this with the high-probability upper bound of $O(n^{1-2/d_o}2^{M_i}\log^{O(1)} n)$ on the minimum distance of rate $1/4$ random serially concatenated convolutional codes obtained by Kahale and Urbanke [7]. In their case, we have $h_o = h_i = 2$, and our upper bound becomes $O(n^{1-1/(2M_o+4)}\ M_i^{1/(2M_o+4)})$. We note that the dependence of our bound on d_o is comparable, and the dependence of our bound on M_i is much better.

B. A strong outer code: when serially concatenated codes become asymptotically good

The proof technique used in Theorem 3 fails if the outer code is not a convolutional code or encodable by a small finite automata. This suggests that by strengthening the outer code one might be able to construct asymptotically good codes. In fact, we will prove that the serial concatenation of an outer repeat-accumulate code with an inner accumulator yields an asymptotically good code with some positive probability.

Let $k \geq 2$ be an integer, r_k be the repeat-k-times map, Q_1 and Q_2 be accumulators[3], n be an integer, and π_1 and π_2 be

[3] While Q_1 and Q_2 are identical as codes, we give them different names to indicate their different roles in the construction.

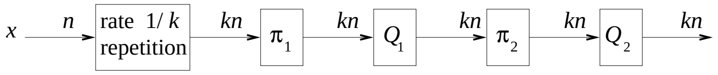

Fig. 1. an RAA code

permutations of length kn. We define C_{k,π_1,π_2} to be the encoder that maps input strings $x \in \{0,1\}^n$ to the codeword $C_{k,\pi_1,\pi_2}(x) := Q_2(\pi_2(Q_1(\pi_1(r_k(x)))))$. We call the code encoded by C_{k,π_1,π_2} an RAA (Repeat, Accumulate, and Accumulate) code (See Figure 1). We note that this code has rate $1/k$.

In contrast with the codes analyzed in Theorem 3, these RAA codes have a repeat-accumulate encoder, $C_{k,\pi_1}(y) = Q_1(\pi_1(r_k(x))$ where those analyzed in Theorem 3 merely have an automata encoder.

Theorem 4: Let $k \geq 2$ and n be integers, and let π_1 and π_2 be permutations of length kn chosen uniformly at random. Then for each constant $\delta > 0$, there exists a constant $\epsilon > 0$ and an integer n_0, such that the RAA code encoded by C_{k,π_1,π_2} has minimum distance at least ϵn with probability at least $1 - \delta$ for all $n \geq n_0$.

So specifically, there exists an infinite family of asymptotically good RAA codes.

Proof: Conditions bounding the size of ϵ will be appear throughout the proof.

Let $E_{\epsilon n}$ denote the expected number of non-zero codewords in the code encoded by C_{k,π_1,π_2} of weight less than or equal to ϵn. Taking a union bound over inputs and applying linearity of expectation, we see that the probability the minimum distance of the code encoded by C_{k,π_1,π_2} is less than ϵn is at most $E_{\epsilon n}$. Thus, we will bound this probability by bounding $E_{\epsilon n}$.

To bound $E_{\epsilon n}$, we use techniques introduced by Divsalar, Jin and McEliece [5] for computing the expected input-output weight enumerator of random Turbo-like codes. For an accumulator code of message length N, let $A^{(N)}_{w,h}$ denote the number of inputs of weight w on which the output of the accumulator has weight h. Divsalar, Jin and McEliece [5] prove that

$$A^{(N)}_{w,h} = \binom{N-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}, \tag{5}$$

where $\binom{a}{b}$ is defined to be zero if $a < b$. Therefore, if the input to Q is a random string of length N and weight w, the probability that the output has weight h is

$$\frac{A^{(N)}_{w,h}}{\binom{N}{w}} = \frac{\binom{N-h}{\lfloor w/2 \rfloor}\binom{h-1}{\lceil w/2 \rceil - 1}}{\binom{N}{w}}. \tag{6}$$

Now consider a fixed input $x \in \{0,1\}^n$. If x has weight w and π_1 is a random permutation of length kn, then $\pi_1(r_k(x))$ is a random string of length kn and weight kw. This random string is the input to the accumulator Q_1. Therefore, by (6), for any h_1 the probability that the output of Q_1 has weight h_1 is $A^{(kn)}_{kw,h_1}/\binom{kn}{kw}$. If this happens, the input to Q_2 will be a random string of weight h_1, and therefore, again by (6), the probability that the output of Q_2 has weight h will be equal to $A^{(kn)}_{h_1,h}/\binom{kn}{h_1}$. Thus, for any fixed input string x of weight w, and any fixed h_1 and h, the probability over the choice of π_1 and π_2 that the output of Q_1 has weight h_1 and the output of Q_2 (which is also the output of C_{k,π_1,π_2}) has weight h is equal to

$$\frac{A^{(kn)}_{kw,h_1} A^{(kn)}_{h_1,h}}{\binom{kn}{kw}\binom{kn}{h_1}}.$$

Thus, by the linearity of expectation, the expected number of non-zero codewords of the code encoded by C_{k,π_1,π_2} of weight at most ϵn equals

$$\begin{aligned} E_{\epsilon n} &= \sum_{w=1}^{n}\sum_{h_1=0}^{kn}\sum_{h=1}^{\epsilon n} \frac{\binom{n}{w} A^{(kn)}_{kw,h_1} A^{(kn)}_{h_1,h}}{\binom{kn}{kw}\binom{kn}{h_1}} \\ &= \sum_{h_1=1}^{2\epsilon n}\sum_{w=1}^{2h_1/k}\sum_{h=1}^{\epsilon n} \frac{\binom{n}{w} A^{(kn)}_{kw,h_1} A^{(kn)}_{h_1,h}}{\binom{kn}{kw}\binom{kn}{h_1}}, \end{aligned}$$

as the terms with $\lceil h_1/2 \rceil > h$ or $\lceil kw/2 \rceil > h_1$ are zero. Using the inequalities $\binom{x}{y} \leq (ex/y)^y$, $\binom{x}{\lfloor y/2 \rfloor} \leq (4ex/y)^y$ and $\binom{x}{\lceil y/2 \rceil - 1} \leq (4ex/y)^y$, for positive integers x and y, we bound this sum by

$$\begin{aligned} &E_{\epsilon n} \\ &= \sum_{h_1=1}^{2\epsilon n}\sum_{w=1}^{2h_1/k}\sum_{h=1}^{\epsilon n} \frac{\binom{n}{w}\binom{kn-h_1}{\lfloor kw/2 \rfloor}\binom{h_1-1}{\lceil kw/2 \rceil - 1}\binom{kn-h}{\lfloor h_1/2 \rfloor}\binom{h-1}{\lceil h_1/2 \rceil - 1}}{\binom{kn}{kw}\binom{kn}{h_1}} \\ &\leq \sum_{h_1=1}^{2\epsilon n}\sum_{w=1}^{2h_1/k}\sum_{h=1}^{\epsilon n} \\ &\quad \frac{\binom{n}{w}\left(\frac{4ekn}{kw}\right)^{kw/2}\left(\frac{4eh_1}{kw}\right)^{kw/2}\left(\frac{4ekn}{h_1}\right)^{\lfloor h_1/2 \rfloor}\left(\frac{4eh}{h_1}\right)^{\lceil h_1/2 \rceil - 1}}{\left(\frac{n}{w}\right)^{kw}\left(\frac{kn}{h_1}\right)^{h_1}} \\ &= \sum_{h_1=1}^{2\epsilon n}\sum_{w=1}^{2h_1/k}\sum_{h=1}^{\epsilon n} \binom{n}{w}\left(\frac{4e\sqrt{h_1}}{\sqrt{kn}}\right)^{kw}\left(\frac{h}{kn}\right)^{\lceil h_1/2 \rceil}\frac{(4e)^{h_1-1}h_1}{h} \end{aligned}$$

The summand in the above expression is at maximum when $h = \epsilon n$. Therefore,

$$\begin{aligned} &E_{\epsilon n} \\ &\leq \epsilon n \sum_{h_1=1}^{2\epsilon n}\left(\frac{\epsilon n}{kn}\right)^{\lceil h_1/2 \rceil}\frac{h_1(4e)^{h_1-1}}{\epsilon n}\sum_{w=1}^{2h_1/k}\binom{n}{w}\left(\frac{4e\sqrt{h_1}}{k\sqrt{n}}\right)^{kw} \\ &\leq \sum_{h_1=1}^{2\epsilon n} h_1\left(4e\sqrt{\epsilon/k}\right)^{h_1}\sum_{w=1}^{2h_1/k}\binom{n}{w}\left(\frac{4e\sqrt{h_1}}{k\sqrt{n}}\right)^{kw} \\ &\leq \sum_{h_1=1}^{2\epsilon n} h_1\left(4e\sqrt{\epsilon/k}\right)^{h_1}\sum_{w=1}^{2h_1/k}\left(\frac{ne}{w}\right)^{w}\left(\frac{4e\sqrt{h_1}}{k\sqrt{n}}\right)^{kw} \\ &= \sum_{h_1=1}^{2\epsilon n} h_1\left(4e\sqrt{\epsilon/k}\right)^{h_1}\sum_{w=1}^{2h_1/k}\left(\frac{e\left(\frac{4e}{k}\right)^{k} n^{1-k/2}h_1^{k/2}}{w}\right)^{w} \\ &\leq \sum_{h_1=1}^{2\epsilon n} h_1\left(4e\sqrt{\epsilon/k}\right)^{h_1}\frac{2h_1}{k}e^{\left(\frac{4e}{k}\right)^{k} n^{1-k/2}h_1^{k/2}}, \text{ as } \left(\frac{y}{x}\right)^{x} \leq e^{y/e} \\ &\leq \frac{2}{k}\sum_{h_1=1}^{2\epsilon n}\left(4e^2\sqrt{\epsilon/k}\right)^{h_1} e^{\left(\left(\frac{4e^2}{k}\right)^{k} n^{1-k/2}\right)h_1^{k/2}}, \end{aligned} \tag{7}$$

since $h_1^2 \le e^{h_1}$ for all $h_1 \ge 1$. To bound (7), note that the sum has the form

$$S = \sum_{x=1}^{m} \alpha^x e^{\beta x^l},$$

where $\alpha = 4e^2\sqrt{\epsilon/k}$, $\beta = \left(\frac{4e^2}{k}\right)^k n^{1-k/2}$, $l = \frac{k}{2}$, and $m = 2\epsilon n$. If we can guarantee that

$$\alpha^{x+1} e^{\beta(x+1)^l} \le \frac{1}{2}\alpha^x e^{\beta x^l}, \tag{8}$$

for all $x = 1, \ldots, m-1$, we can use the bound

$$S \le 2\alpha e^{\beta}. \tag{9}$$

We can express (8) as $\beta((x+1)^l - x^l) \le \ln\frac{1}{2\alpha}$. Thus (8) holds for all the desired values of x if $\beta((m+1)^l - m^l) \le \ln\frac{1}{2\alpha}$, or equivalently

$$\beta m^l \left(\left(1+\frac{1}{m}\right)^l - 1\right) \le \ln\frac{1}{2\alpha},$$

which can be guaranteed when

$$2l\beta m^{l-1} \le \ln\frac{1}{2\alpha} \quad \text{and} \quad l \le m, \tag{10}$$

via the bounds

$$\left(1+\frac{1}{m}\right)^l \le e^{l/m} \le 1 + (e-1)\frac{l}{m} \le 1 + 2\frac{l}{m},$$

where we need $l \le m$ in the second inequality. Going back to (7), we get via (9) and (10) that

$$E_{\epsilon n} \le \frac{2}{k} 2(4e^2\sqrt{\epsilon/k}) e^{\left(\frac{4e^2}{k}\right)^k n^{1-k/2}} = \frac{16e^2\sqrt{\epsilon}}{k\sqrt{k}} e^{\left(\frac{4e^2}{k}\right)^k n^{1-k/2}}, \tag{11}$$

when

$$2\frac{k}{2}\left(\frac{4e^2}{k}\right)^k n^{1-k/2}(2\epsilon n)^{k/2-1} \le \ln\left(\frac{1}{8e^2}\sqrt{\frac{k}{\epsilon}}\right) \quad \text{and} \quad \frac{k}{2} \le 2\epsilon n,$$

or, equivalently, when

$$\ln\frac{1}{\epsilon} \ge \left(2k\left(\frac{4e}{k}\right)^k 2^{k/2-1}\right)\epsilon^{k/2-1} - 2\ln\frac{\sqrt{k}}{8e^2} \quad \text{and} \quad \frac{k}{2} \le 2\epsilon n. \tag{12}$$

It follows from (11) and (12), that for each $k \ge 2$, and for each constant $\delta > 0$, there is constant $\epsilon > 0$ such $E_{\epsilon n} < \delta$ when n is sufficiently large.

$\blacksquare$

While the constants we obtain are not particularly sharp, they are sufficient to prove the existence of asymptotically good families of depth-three serially-concatenated codes based on accumulators.

This result should be compared with the work of Pfister and Siegel [8], who performed experimental analyses of the serial concatenation of repetition codes with l levels of accumulators connected by random interleavers, and theoretical analyses of concatenations of a repetition code with certain rate-1 codes for large l. Their experimental results indicate that the average minimum distance of the ensemble starts becoming good for $l \ge 2$, which is consistent with our theorem. For certain rate-1 codes and l going to infinity, they proved their codes could become asymptotically good. In contrast, we prove this for $l = 2$ and accumulator codes.

IV. Conclusion and Open questions

We derived in Section II-A a worst-case upper bound on the minimum distance of parallel concatenated convolutional codes, repeat convolute codes, and generalizations of these codes obtained by allowing non-linear and large-memory automata-based constituent codes. The bound implies that such codes are asymptotically bad when the underlying automata codes have sub-linear memory. In the setting of convolutional constituent codes, a sub-linear memory corresponds to the case when the size of the corresponding trellis is sub-exponential, and so it includes the cases in which the codes have natural sub-exponential time iterative decoding algorithm. In Section II-B, we improved the bound in the setting of low-memory convolutional constituent codes. We leave the problem of interpolating between the two bounds open:

- Is it possible to interpolate between the bounds of Theorems 1 and 2?

Then, we derived in Section III-A a worst-case upper bound on the minimum distance of depth-2 serially concatenated automata-based codes. Our bound implies that such codes are asymptotically bad when the outer code has a constant memory and the inner code has a sub-linear memory. This suggests the following question:

- If one allows the memory of the outer code in a depth-2 serially concatenated code to grow logarithmically with the block length, can one obtain an asymptotically good code?

In contrast, we proved in Section III-B that RAA codes, which are depth-3 serially concatenated codes obtained by concatenating a repetition code with two accumulator codes through random permutations, can be asymptotically good. This result naturally leads to the following open questions:

- Can one obtain depth-3 serially concatenated codes with better minimum distance by replacing the accumulators in the RAA codes with convolutional codes of larger memory? Also, can one improve the minimum distance bounds on the RAA codes?
- Can the RAA codes be efficiently decoded by iterative decoding, or any other algorithm?

V. Acknowledgment

We thank Sanjoy Mitter for many stimulating conversations and the anonymous reviewers for many useful comments on the paper.

References

[1] Miklós Ajtai. Determinism verus nondeterminism for linear time RAMs with memory restrictions. *J. Comput. Syst. Sci.*, 65(1):2–37, 2002.

[2] Sergio Benedetto, Dariush Divsalar, Guido Montorsi, and Fabrizio Pollara. Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding. *IEEE Transactions on Information Theory*, 44(3):909–926, 1998.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes. In *IEEE Int. Conf. on Communications (ICC-1993)*, pages 1064–1070, 1993.

[4] Marco Breiling. A logarithmic upper bound on the minimum distance of turbo codes. *IEEE Transactions on Information Theory*, 50(8):1692–1710, 2004.

[5] Dariush Divsalar, Hui Jin, and Robert J. McEliece. Coding theorems for "turbo–like" codes. In *Proceedings 36th Annual Allerton Conference on Communication, Control, and Computing*, pages 201–210, Monticello, IL, USA, September 1998.

[6] Hui Jin and Robert J. McEliece. RA codes achieve AWGN channel capacity. In Marc P. C. Fossorier, Hideki Imai, Shu Lin, and Alain Poli, editors, *AAECC*, volume 1719 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1999.

[7] Nabil Kahale and Rudiger Urbanke. On the minimum distance of parallel and serially concatenated codes. In *Proceedings of IEEE International Symposium on Information Theory*, page 31, Cambridge, MA, USA, aug 1998. IEEE.

[8] Henry D. Pfister and Paul H. Siegel. The serial concatenation of rate-1 codes through uniform random interleavers. *IEEE Transactions on Information Theory*, 49(6):1425–1438, 2003.

[9] Branka Vucetic and J.S. Yuan. *Turbo Codes: Principles and Applications*. Kluwer Academic Publishers, 2000.

Louay Bazzi received his Ph.D. degree form the department of Electrical Engineering and Computer Science at MIT in 2003. He is currently an assistant Professor in the Electrical and Computer Engineering department at the American University of Beirut. His research interests include coding theory, psuedorandomness, complexity theory, and algorithms.

Mohammad Mahdian received a B.S. degree in computer engineering from Sharif University of Technology in 1997, an M.S. degree in computer science from University of Toronto in 2000, and a Ph.D. degree in mathematics from Massachusetts Institute of Technology in 2004. He has worked as an intern and a postdoctoral researcher at IBM Research Labs and Microsoft Research, and is currently a Research Scientist at Yahoo! Research Lab in Santa Clara, CA. His current research interests include algorithm design, algorithmic game theory, and applications in online advertising and social networks.

Daniel A. Spielman was born in Philadelphia, Pennsylvania, in 1970. He received the B.A. degree in mathematics and computer science from Yale University in 1992, and the Ph.D. degree in Applied Mathematics from M.I.T. in 1995.

From 1996 to 2005, he was an Assistant and then an Associate Professor of Applied Mathematics at M.I.T. Since 2005, he has been a Professor of Applied Mathematics and Computer Science at Yale University.

Dr. Spielman is the recipient of the 1995 Association for Computing Machinery Doctoral Dissertation Award, the 2002 IEEE Information Theory Society Paper Prize, and the 2008 Gödel Prize, awarded by the European Association for Theoretical Computer Science and the Association for Computing Machinery.