Explicit and Efficient WOM Codes of Finite Length

Yeow Meng Chee, *Senior Member, IEEE*, Han Mao Kiah, Alexander Vardy, *Fellow, IEEE*, and Eitan Yaakobi, *Senior Member, IEEE*

***Abstract*— Write-once memory (WOM) is a storage device consisting of binary cells that can only increase their levels. A *t-write WOM code* is a coding scheme that makes it possible to write t times to a WOM without decreasing the levels of any of the cells. The *sum-rate* of a WOM code is the ratio between the total number of bits written to the memory during the t writes and the number of cells. It is known that the maximum possible sum-rate of a t-write WOM code is $\log(t+1)$. This is also an achievable upper bound, both by information-theoretic arguments and through explicit constructions. While existing constructions of WOM codes are targeted at the sum-rate, we consider here two more figures of merit. The first one is the *complexity* of the encoding and decoding maps. The second figure of merit is the *convergence rate*, defined as the minimum code length $n(\delta)$ required to reach a point that is δ-close to the capacity region. One of our main results in this paper is a capacity-achieving construction of two-write WOM codes which has polynomial encoding/decoding complexity while the block length $n(\delta)$ required to be δ-close to capacity is significantly smaller than existing constructions. Using these two-write WOM codes, we then obtain three-write WOM codes that approach a sum-rate of 1.809 at relatively short block lengths. We also provide several explicit constructions of finite length three-write WOM codes; in particular, we achieve a sum-rate of 1.716 by using only 93 cells. Finally, we modify our two-write WOM codes to construct ϵ-error WOM codes of high rates and small probability of failure.**

***Index Terms*— Binary write-once memory (WOM) codes, spreads, cooling codes, flash memories, ϵ-error WOM codes.**

I. Introduction

WRITE-ONCE memory (*WOM*) is a storage medium consisting of cells that can only increase their level. WOM codes were first introduced by Rivest and Shamir [28] in 1982, and were motivated by storage media such as punch cards and optical disks. These media comprise storage elements, called *cells*, which have an asymmetric programming attribute. In the binary version, it is only allowed to irreversibly program each cell from level zero to level one. If a cell can accommodate more than two levels then, on each programming operation, it is only possible to increase the cell's level. A WOM code is a coding scheme that allows to store multiple messages in a WOM, while making sure that all cells will only increase their level on each write. One famous example of a WOM code was presented by Rivest and Shamir in [28]. This WOM code stores two bits twice using only three cells; other constructions of WOM codes were also presented in [28]. Several more families of WOM codes were studied later in the 1980s and 1990s — see for example [8], [16], [24].

In the past decade, WOM codes have attracted tremendous interest due to their applicability in flash memories [2], [12], [20], [30], [31], [35]–[38]. Flash memory may be regarded as an example of a WOM; its cells are charged with electrons and usually have multiple levels [3]. Increasing a cell level is fast and easy. However, in order to decrease its level, the entire containing block of cells has to be erased first. This not only affects the writing speed of the flash memory but also significantly reduces its lifetime [3]. Thus reducing the number of block erasures in flash memories is crucial in order to improve their lifetime. Implementation of WOM codes in flash memories was recently demonstrated in several works [22], [26], [40]. Another recent paper discusses the benefits of WOM coding for improving the memory lifetime [39].

Assume that a WOM, consisting of n binary cells, is required to accommodate t message writes. For $i = 1, 2, \ldots, t$, let M_i denote the number of possible messages during the i-th write. The *rate* of the i-th write is defined as $\mathcal{R}_i = \log M_i / n$, and the *sum-rate* of the WOM code is $\mathcal{R} = \sum_{i=1}^{t} \mathcal{R}_i$. The capacity region of the WOM is the set of all achievable rate tuples $(\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_t)$. For the binary case, the capacity region was determined in [15], [18], and [28]. It was also proved in these references that the maximum achievable sum-rate for a binary WOM code with t writes is $\log(t+1)$. These results were generalized in [15] for non-binary WOMs: the maximum sum-rate for a WOM with q-ary cells is known to be $\log \binom{t+q-1}{t}$.

The main goal in designing a WOM code is to achieve, through explicit code constructions, all the rate-tuples in the capacity region and, in particular, a high sum-rate.

Manuscript received October 21, 2018; revised August 4, 2019; accepted September 30, 2019. Date of publication October 9, 2019; date of current version April 21, 2020. Y. M. Chee and H. M. Kiah were supported by the Singapore Ministry of Education Research under Grant MOE2015-T2-2-086. A. Vardy was supported by the National Science Foundation under Grant CCF-1405119 and Grant CCF-1719139. E. Yaakobi was supported by the Israel Science Foundation (ISF) under Grant 1624/14. This article was presented in part at the 2017 IEEE International Symposium on Information Theory.

Y. M. Chee was with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371. He is now with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576 (e-mail: pvocym@nus.edu.sg).

H. M. Kiah is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (e-mail: hmkiah@ntu.edu.sg).

A. Vardy was with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371. He is now with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA (e-mail: avardy@ucsd.edu).

E. Yaakobi is with the Department of Computer Science, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: yaakobi@cs.technion.ac.il).

Communicated by A. Jiang, Associate Editor for Coding Theory.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2019.2946483

For the two-write case, such constructions were studied in [31] and [37]. For multiple writes, capacity-achieving constructions were introduced in [30]. There are two more figures of merit that are important when evaluating WOM code constructions, which we investigate in this work. The first one is the *complexity* of the encoding and decoding maps of the WOM code as a function of the code length n. The second one, called the *convergence rate*, is the minimum code length $n(\delta)$ required in order to achieve a rate-tuple that is δ-close to a point in the capacity region. We also consider capacity-achieving constructions for the ϵ-error case. For ϵ-error WOM codes, successive writes are not guaranteed in the worst-case, but only with high probability [2], [12], [20], [35]. For such codes, in addition to the first two measures, we study another figure of merit: the *failure decay rate*, which is the rate at which the probability of a write failure tends to zero.

Prior to this work, known capacity-achieving constructions for two-write zero-error WOM codes featured either fast convergence rate (that is, approaching the capacity region at relatively short code lengths) or low complexity, *but not both*. Specifically, the construction from [37] has fast convergence rate, but suffers from high complexity. On the other hand, the construction of [31] has the opposite attributes: it requires exponential block length in order to achieve a point that is δ-close to capacity. One of our main results in this paper is an explicit construction of two-write WOM codes with fast convergence rate *as well as* low complexity of encoding and decoding. Moreover, our construction makes it possible to design WOM codes of extremely short block length; such codes are attractive due to their very efficient encoding and decoding procedures [4]. In this regard, the best previously known construction of two-write WOM codes [37] had a sum-rate of 1.4928, using 33 binary cells. Herein, we improve this to a sum-rate of 1.509, using 40 cells.

For three-write WOM codes, the upper bound on the sum-rate is $\log(3+1) = 2$. The best known sum-rate of a fully explicit construction [37] is 1.61. The work of [37] also presents a somewhat less explicit construction, based upon the existence of certain ternary matrices, whose sum-rate is 1.66. In [31], the latter result was improved to a sum-rate of 1.809. Yet another improvement was presented in [38], achieving a sum-rate of 1.885. As mentioned earlier, the construction of [30] is capacity-achieving for any number of writes, thus in particular for three writes. However, the block length required in order to get close capacity is extremely large — significantly larger than the block length resulting from the constructions of [31], [37], [38]. Using our two-write WOM codes, we modify the construction of [31] to obtain three-write WOM codes that approach the same sum-rate of 1.809, but much faster than in [31]. We also propose a number of constructions of three-write WOM codes of finite length. Using one of these constructions, we achieve a sum-rate of 1.716 with only 93 cells. For the ϵ-error case, we present three-write WOM codes that approach a sum-rate of 1.936 and have a faster failure decay rate than any previous construction.

The rest of this paper is organized as follows. In Section II, we review the basic theory of WOM codes and define the figures of merit (complexity and convergence rate) that we use in evaluating WOM code constructions. Section III is devoted to two-write WOM codes. We first review the current state-of-the-art, then present our construction of two-write WOM codes with polynomial complexity and fast convergence rate. In Sections IV and V, we present constructions of explicit three-write WOM codes, for the zero-error case and the ϵ-error case, respectively. Lastly, Section VI concludes the paper and lists several open problems.

II. Definitions and Basic Properties

In this work, we assume that the memory consists of n binary cells, where initially all of them are in the zero state. On each write it is only possible to increase the level of each cell to level one. A vector $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ will be called a *cell-state vector*. For two cell-state vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we say that $\boldsymbol{x} \leqslant \boldsymbol{y}$ if $x_i \leqslant y_i$ for all $1 \leqslant i \leqslant n$. For a positive integer n, we use the notation $[n]$ to define the set of integers $\{1, \ldots, n\}$. If x represents a bit value then its complement is $\overline{x} = 1 - x$, and for a binary vector $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\overline{\boldsymbol{x}} = (\overline{x}_1, \ldots, \overline{x}_n)$. For any map $f : A \to B$, $Im(f)$ is the image of the map f. The binary entropy function is defined for every probability $0 < p < 1$ as $h(p) = -p\log(p) - (1-p)\log(1-p)$.

We follow the formal definition of WOM codes from [38].

Definition 1. An $[n, t; M_1, \ldots, M_t]$ *t-write WOM code* is a coding scheme comprising of n binary cells and is defined by t pairs of encoding and decoding maps $(\mathcal{E}_i, \mathcal{D}_i)$, for $1 \leqslant i \leqslant t$. The encoding map $\mathcal{E}_i$ is defined by

$$\mathcal{E}_i : [M_i] \times Im(\mathcal{E}_{i-1}) \to \{0,1\}^n,$$

where, by definition, $Im(\mathcal{E}_0) = \{(0, \ldots, 0)\}$, such that $\mathcal{E}_i(m, \boldsymbol{c}) \geqslant \boldsymbol{c}$ for all $(m, \boldsymbol{c}) \in [M_i] \times Im(\mathcal{E}_{i-1})$. Similarly, the decoding map $\mathcal{D}_i$ is defined by

$$\mathcal{D}_i : Im(\mathcal{E}_i) \to [M_i],$$

such that for all $(m, \boldsymbol{c}) \in [M_i] \times Im(\mathcal{E}_{i-1})$, $\mathcal{D}_i(\mathcal{E}_i(m, \boldsymbol{c})) = m$. The rate on the i-th write is defined by $\mathcal{R}_i = \frac{\log M_i}{n}$, and the sum-rate is $\mathcal{R}_{\text{sum}} = \sum_{i=1}^{t} \mathcal{R}_i$.

In [15] and [18], the capacity region of a binary t-write WOM was found to be

$$C_t = \Big\{(\mathcal{R}_1, \ldots, \mathcal{R}_t) | \mathcal{R}_1 \leqslant h(p_1), \mathcal{R}_2 \leqslant (1-p_1)h(p_2), \ldots, \mathcal{R}_{t-1} \leqslant \Big(\prod_{i=1}^{t-2}(1-p_i)\Big)h(p_{t-1}), \mathcal{R}_t \leqslant \prod_{i=1}^{t-1}(1-p_i), \text{ where } 0 \leqslant p_1, \ldots, p_{t-1} \leqslant 1/2\Big\}, \quad (1)$$

and $\log(t+1)$ was proved to be the maximum sum-rate. Even though it is known that all rate tuples in the capacity region are achievable, the problem of finding efficient code construction remains a challenge. Following [37], we assume that the write number on each write is known since this side information does not affect the achievability of rate tuples in the capacity region.

We evaluate the efficiency of a construction according to the following two figures of merit.

(I) *Encoding / decoding complexity*: the complexity is defined to be the complexity of the encoding and decoding maps as a function of code length n.

(II) *Convergence rate*: the minimum code length $n(\delta)$ in order to be δ-close to a rate tuple $(\mathcal{R}_1, \ldots, \mathcal{R}_t)$ in the capacity region.

More rigorously, the second figure of merit states that the convergence rate of a family of WOM codes to a rate tuple $(\mathcal{R}_1, \ldots, \mathcal{R}_t) \in C_t$ is $n(\delta)$ if the construction yields a WOM code of length $n(\delta)$ with rate tuple at least $(\mathcal{R}_1 - \delta, \ldots, \mathcal{R}_t - \delta)$. Since we are mostly interested in whether $n(\delta)$ is polynomial or exponential in $1/\delta$, we say that a construction approaches a rate tuple $(\mathcal{R}_1, \ldots, \mathcal{R}_t) \in C_t$ with polynomial or exponential rate if $n(\delta)$ is polynomial or exponential in $1/\delta$, respectively. Similarly we say that a construction approaches a sum-rate $\mathcal{R}_{\text{sum}}$ with polynomial or exponential rate. In addition, if $n(\delta) = O((1/\delta)\log(1/\delta))$, we say that the rate tuples converge at an *almost-linear* rate.

We also consider the ϵ-error case of WOM codes where successive writes are not guaranteed in the worst case.

Definition 2. An $[n, t; M_1, \ldots, M_t; \epsilon_1, \epsilon_2, \ldots, \epsilon_t]$ *t-write ϵ-error WOM code* is a coding scheme comprising of n binary cells and is defined by t pairs of encoding and decoding maps $(\mathcal{E}_i, \mathcal{D}_i)$, for $1 \leqslant i \leqslant t$. The encoding map $\mathcal{E}_i$ is defined by

$$\mathcal{E}_i : [M_i] \times Im(\mathcal{E}_{i-1}) \to \{0,1\}^n \cup \{\mathsf{fail}\},$$

where, by definition, $Im(\mathcal{E}_0) = \{(0, \ldots, 0)\}$, such that for all $(m, \boldsymbol{c}) \in [M_i] \times Im(\mathcal{E}_{i-1})$ either $\mathcal{E}_i(m, \boldsymbol{c}) \geqslant \boldsymbol{c}$ or $\mathcal{E}_i(m, \boldsymbol{c}) = \mathsf{fail}$. Similarly, the decoding map $\mathcal{D}_i$ is defined by

$$\mathcal{D}_i : Im(\mathcal{E}_i) \cup \{\mathsf{fail}\} \to [M_i],$$

such that for all $(m, \boldsymbol{c}) \in [M_i] \times Im(\mathcal{E}_{i-1})$, $\mathcal{D}_i(\mathcal{E}_i(m, \boldsymbol{c})) = m$ whenever $\mathcal{E}_i(m, \boldsymbol{c}) \neq \mathsf{fail}$. Furthermore, we require that

$$|\{(m, \boldsymbol{c}) \in [M_i] \times Im(\mathcal{E}_{i-1}) : \mathcal{E}_i(m, \boldsymbol{c}) = \mathsf{fail}\}| \leqslant \epsilon_i M_i |Im(\mathcal{E}_{i-1})| \text{ for } i \in [t].$$

When $\epsilon_i = 0$ for all $i \in [t]$, Definition 2 reduces to Definition 1. Given a construction that yields a family of t-write ϵ-error WOM codes $\{\mathbb{C}_n\}$, where $\mathbb{C}_n$ is an $[n, t;\ M_1(n), \ldots,\ M_t(n);\ \epsilon_1(n),\ \epsilon_2(n),\ \ldots,\ \epsilon_t(n)]$ t-write ϵ-error WOM code, we then require that $\epsilon_i(n)$ converges to zero for all $i \in [t]$. As before, we set $\mathcal{R}_i^{\epsilon} = \lim_{n\to\infty}(\log M_i(n))/n$ for $i \in [t]$ and $\mathcal{R}_{\text{sum}}^{\epsilon} = \sum_{i=1}^{t} \mathcal{R}_i^{\epsilon}$.

Surprisingly, Wolf *et al.* [34] proved that the capacity region of a binary t-write ϵ-error WOM code coincides with the capacity region of a binary t-write WOM code. In other words, $(\mathcal{R}_1^{\epsilon}, \mathcal{R}_2^{\epsilon}, \ldots, \mathcal{R}_t^{\epsilon}) \in C_t$ where C_t is defined in (1).

Therefore, we evaluate our constructions of ϵ-error WOM codes using the previous two figures of merit: (I) encoding/decoding complexity and (II) convergence rate. Additionally, we also evaluate how fast the failure probability converges to zero. Specifically, we consider the following figure of merit.

(III) *Failure decay rate*: set $\epsilon(n) = \max_{i\in[t]} \epsilon_i(n)$ and we are interested in the rate that $\epsilon(n)$ converges to zero.

Finally, motivated by a reviewer, we study the possibility of increasing sum-rates when we allow the failure probability to converge to a small positive constant ϵ'. Unfortunately, the answer is negative. Specifically, we prove the following theorem.

Theorem 1. *Let $\epsilon' < 1/(t-1)$ for $t \geqslant 2$. Suppose that $\{\mathbb{C}_n : \mathbb{C}_n$ is an $[n, t; M_1(n), \ldots, M_t(n); \epsilon_1(n), \epsilon_2(n), \ldots, \epsilon_t(n)]$ t-write ϵ-error WOM code$\}$ is a family of WOM codes where $\limsup_{n\to\infty} \max_{i\in[t]} \epsilon_i(n) \leqslant \epsilon'$. Then $\mathcal{R}_{sum}^{\epsilon} \leqslant \log_2(t+1)$.*

In other words, even if we allow the failure probability to decay to positive $\epsilon' < 1/(t-1)$, the maximum sum-rate remains bounded above by $\log_2(t+1)$ (the sum-rate corresponding to the zero-error case). We defer the proof of Theorem 1 to Section V-C.

III. Capacity-Achieving Two-Write WOM Codes

We present an explicit construction of two-write WOM codes. As opposed to existing constructions, we prove that this construction has both polynomial encoding/decoding complexity and almost-linear convergence rate.

A. Background and State of the Art Results

The capacity region of a two-write WOM is given by

$$C_2 = \{(\mathcal{R}_1, \mathcal{R}_2) | \mathcal{R}_1 \leqslant h(p), \mathcal{R}_2 \leqslant (1-p), 0 \leqslant p \leqslant 1/2\},$$

and the maximum sum-rate is $\log 3 \approx 1.58$. On the other hand, the best sum-rate reported in the literature for a finite-length code (in fact, only 33 cells) is 1.4928 [37]. There are three explicit constructions which achieve the capacity C_2 of two-write WOM. The first one to accomplish this task was presented in [37]. Shortly after, Shpilka [31] presented another capacity-achieving construction for two writes and a general construction for t writes in [30]. There are several other works which achieve the ϵ-error capacity of two-write WOM. Here, ϵ-error implies that the second write does not succeed in the worst case but only with high probability. These constructions are based on polar codes [2], LDPC codes [12], [20], or random matrices [35].

All the aforementioned constructions use a very similar principle. For a given probability $0 < p < 1/2$, if the WOM has n cells, then on the first write at most pn cells are programmed (but not necessarily all such patterns). Thus, it is possible to store roughly $nh(p)$ bits and the rate approaches $h(p)$. The challenge on the second write is to store roughly $(1-p)n$ more bits on the remaining $(1-p)n$ cells.

The construction in [37] accomplishes this task by restricting the cell-state vectors that can be programmed on the first write, such that any pattern of $n(1-p)$ bits can be stored on the second write. This approach significantly simplifies the encoding and decoding on the second write however it incurs an extremely high complexity on the first write since not all vectors of weight at most pn can be programmed. While the convergence rate of this construction is polynomial its complexity is, in general, exponential since it may require a lookup table for the first write encoding and decoding.

On the other hand, in [31] (almost) any pattern of at most pn cells can be programmed and the second write uses a set of "average" MDS codes which are derived from a *Wozencraft ensemble* [19], [23]. This collection of codes guarantees the success of encoding roughly $(1-p)n$ bits on the second write. The encoding and decoding complexities of this construction are polynomial with the code length n. However in order to achieve high sum-rate, the block length has to be extremely large, and in particular the convergence rate of this construction is exponential.

The constructions from [37] and [31] introduce a tradeoff between the complexity and convergence rate. While the first one suffers from high complexity but achieves polynomial convergence rate, the second one has opposite attributes, i.e., low complexity but exponential convergence rate. However, we show that there is no such a tradeoff by presenting a construction which achieves these two goals simultaneously.

B. The Construction

We are now ready to present our construction of two-write WOM codes. In fact, the encoding map for our two-write codes has a certain specific property (that was described in connection with [31] in the preceding subsection) that will be useful for the construction of three-write WOM codes, to be presented in Sections IV and V. We now formally state this property.

Definition 3. An $[n, 2; M_1, M_2]$ two-write WOM code is said to be of *type A* if $M_1 = \sum_{i=0}^{\tau} \binom{n}{i}$ and the encoding map $\mathcal{E}_1$ on the first write is a bijection from $[M_1]$ to $J^+(n,\tau)$. Here, $J^+(n,\tau)$ denotes the set $\{\boldsymbol{x} \in \{0,1\}^n : \mathrm{wt}(\boldsymbol{x}) \leqslant \tau\}$. In other words, after the first write, any pattern of at most τ cells can be programmed, and we denote such a WOM code as an $[n, 2; \tau, M_1, M_2]_A$ WOM code.

Remark 1. The encoding map $\mathcal{E}_1 : [M_1] \to J^+(n,w)$ and its corresponding decoding map can be implemented efficiently, for example using the enumerative coding scheme of Cover [9]. Hence, in the analysis of our two-write WOMs, it remains to determine the complexities of the encoding and decoding maps on the second write.

In [5], the authors introduced in high performance interconnects, and provided constructions of cooling codes that are optimal in terms of redundancy. We summarize these results here, but first we demonstrate the connection between cooling codes and two-write WOM codes of type A.

Definition 4. An (n,τ) *cooling code* $\mathbb{C}$ of size M is defined as a collection of *codesets* $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$, where $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ are disjoint subsets of $\{0,1\}^n$ satisfying the following property: for any set $S \subseteq [n]$ of size at most τ and for all $i \in [M]$, there exists a codeword $\boldsymbol{x} \in \mathcal{C}_i$ with $\mathrm{supp}(\boldsymbol{x}) \cap S = \varnothing$.

The following proposition shows that two-write WOM codes of type A and cooling codes are equivalent.

Proposition 1. *An $[n, 2; \tau, M_1, M_2]_A$ WOM code of type A exists if and only if an (n,τ)-cooling code of size M_2 exists.*

Proof: Suppose an (n,τ)-cooling code of size M_2 exists. To construct an $[n, 2; \tau, M_1, M_2]_A$ two-write WOM code of type A, we describe the encoding and decoding maps for the second write.

Let $m \in [M_2]$ be the message to be written on the second write, and we consider the codeset $\mathcal{C}_m$ in the cooling code $\mathbb{C}$. Let S be the set of indices of the programmed cells on the first write. Since $|S| \leqslant \tau$, there exists a word $\boldsymbol{x} \in \mathcal{C}_i$ with $\mathrm{supp}(\boldsymbol{x}) \cap \mathcal{S} = \varnothing$ by the definition of cooling codes. The programmed cell-state vector is then $\boldsymbol{c} = \overline{\boldsymbol{x}}$. Note that since for all $i \in S$, $x_i = 0$, we get that $c_i = 1$, as required.

Given a codeword $\boldsymbol{c}$, we compute $\boldsymbol{x} = \overline{\boldsymbol{c}}$. Since the codesets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{M_2}$ are disjoint, we identify the unique codeset that the nonzero vector $\boldsymbol{x}$ belongs to and hence decode the value of m. This completes the decoding.

Conversely, suppose an $[n, 2; \tau, M_1, M_2]_A$ WOM code exists. We construct an (n,τ) cooling code of size M_2 by defining the codesets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{M_2}$. Let $m \in [M_2]$ be the message for the second write. We set $\mathcal{C}_m \triangleq \{\overline{\boldsymbol{x}} : \boldsymbol{x} = \mathcal{E}_2(m,\boldsymbol{c}),\ \text{for some } \boldsymbol{c} \in Im(\mathcal{E}_1)\}$.

We demonstrate that $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{M_2}$ form an (n,τ) cooling code. Since $\mathcal{D}_2(\overline{\boldsymbol{x}}) = m$ for all $m \in [M_2]$ and $\boldsymbol{x} \in \mathcal{C}_m$, we have that $\mathcal{C}_m \cap \mathcal{C}_{m'} = \varnothing$ for $m \neq m'$. Next, consider $S \subset [n]$ with $|S| \leqslant \tau$ and $m \in [M_2]$. We set the state-vector $\boldsymbol{c}$ to be the vector whose support is given by S, and so, $\boldsymbol{c} \in Im(\mathcal{E}_1)$. For a type A code, let $\boldsymbol{x} = \mathcal{E}_2(m,\boldsymbol{c})$ and we have $\overline{\boldsymbol{x}} \in \mathcal{C}_m$. Since $\boldsymbol{x} \geqslant \boldsymbol{c}$, we have $\mathrm{supp}(\boldsymbol{x}) \supseteq \mathrm{supp}(\boldsymbol{c}) = S$. In other words, $\mathrm{supp}(\overline{\boldsymbol{x}}) \cap S = \varnothing$, as required. □

The authors of [5] provided a number of constructions of cooling codes that in turn yield two-write WOM codes of type A via Proposition 1. A crucial construction is based on the notion of *spreads* in projective geometry and a specific example for this construction was first developed by Dumer for codes with stuck-at cells [11]. This structure is defined formally as follows.

Definition 5. A collection $V_1, V_2, \ldots, V_M$ of τ-dimensional subspaces of $\mathbb{F}_2^n$ is said to be a *τ-partial spread of $\mathbb{F}_2^n$* of size M if $V_i \cap V_j = \{\boldsymbol{0}\}$ for all $i \neq j$. In addition, if $\bigcup_{i=1}^{M} V_i = \mathbb{F}_2^n$, then the collection is said to be a *τ-spread of $\mathbb{F}_2^n$*.

Spreads were widely studied in the literature and it is well known that τ-spreads of $\mathbb{F}_2^n$ exist if and only if τ divides n (see [1], [13], [29], [32]). In this case, it holds that $M = (2^n-1)/(2^\tau-1) > 2^{n-\tau}$. The case where n is not a multiple of τ was studied in [14] and it was shown that a τ-partial spread of $\mathbb{F}_2^n$ of size $2^{n-\tau}$ exists, where $\tau \leqslant n/2$. Furthermore, it was demonstrated in [5] that a $(\tau+1)$-partial spread of $\mathbb{F}_2^n$ of size M implies the existence of an (n,τ) cooling code of size $M+1$. In particular, when $\tau+1|n$, the authors modified Dumer's encoding/ decoding method [11] to encode / decode a cooling code of size $2^{n-(\tau+1)}$. We extend the method to the case where $\tau+1$ does not necessarily divide n and modify the method to the case where $n/2 - 1 \leqslant \tau \leqslant 0.687n$. We defer the proof of Proposition 2 to Appendix A.

Proposition 2.

(i) Let $\tau + 1 \leqslant n/2$. There exists an (n, τ) cooling code of size $2^{n-(\tau+1)}$ with encoding / decoding complexity $O(n^3)$.

(ii) Let $\tau \leqslant 0.687n$. Set $r = \lceil \log(2n+1) \rceil$ and $\alpha = \tau/n$. There exists n_α (dependent of α only) such that for all $n \geqslant n_\alpha$, an (n, τ) cooling code of size $2^{n-(\tau+r)}$ exists with encoding / decoding complexity $O(n^3)$.

(iii) Let $\tau = \lfloor 0.59n \rfloor$. Set $r = \lceil \log(2n+1) \rceil$. For all $n \geqslant 7430$, an (n, τ) cooling code $\mathbb{C}_n$ of size $2^{n-(\tau+r)}$ exists with encoding / decoding complexity $O(n^3)$. Furthermore, $\mathbb{C}_n$ can be constructed in time polynomial in n.

Following Propositions 1 and 2, we obtain a family of capacity-achieving two-write WOM codes that achieves our two figures of merit: low complexity and almost-linear convergence rate.

Theorem 2 (Construction 1)**.** *Let $0 \leqslant p \leqslant 1/2$. For all n, there exists a two-write WOM code $\mathbb{C}$ of length n such that the following hold.*

(i) $\mathbb{C}$ has encoding / decoding complexity $O(n^3)$.

(ii) Furthermore, if $n \geqslant \lfloor (6/\delta)\log(3/\delta) \rfloor - 1$, its rate-tuple is δ-close to $(h(p), 1-p)$.

Proof: Choose $\tau = \lceil np \rceil$. Apply Propositions 1 and 2 to obtain a two-write WOM code $\mathbb{C}$ with rates

$$\mathcal{R}_1(n) = \frac{\log\left(\sum_{i=0}^{\tau}\binom{n}{i}\right)}{n}, \text{ and } \mathcal{R}_2(n) = \frac{n-(\tau+1)}{n}.$$

According to [27, Lemma 4.8], we use the inequality

$$\sum_{i=0}^{\tau}\binom{n}{i} \geqslant \frac{1}{n+1}2^{nh\left(\frac{\tau}{n}\right)},$$

and conclude that

$$\begin{aligned}\mathcal{R}_1(n) &\geqslant \frac{\log\left(\frac{1}{n+1}2^{nh(\tau/n)}\right)}{n} \\ &= h\left(\frac{\tau}{n}\right) - \frac{\log(n+1)}{n} \geqslant h(p) - \frac{\log(n+1)}{n}. \quad (2)\end{aligned}$$

Set $n_0 \triangleq \lfloor (6/\delta)\log(3/\delta) \rfloor - 1$. Since $\log(n_0+1) \leqslant \log\left((6/\delta)\log(3/\delta)\right) = \log(3/\delta) + 1 + \log\log(3/\delta)$, and $n_0 \geqslant (6/\delta)\log(3/\delta) - 2 \geqslant (3/\delta)\log(3/\delta)$, we have that

$$\frac{\log(n_0+1)}{n_0} \leqslant \frac{\log(3/\delta)+1+\log\log(3/\delta)}{(3/\delta)\log(3/\delta)} \leqslant \frac{3}{3/\delta} = \delta.$$

In other words, we have that $\log(n_0+1) \leqslant \delta n_0$. Since the function $(1/x)\log(x+1)$ is monotone decreasing on $x > 0$, we have that $\log(n+1) \leqslant \delta n$ for $n \geqslant n_0$. Therefore, $\mathcal{R}_1(n) \geqslant h(p) - \delta$ follows from (2).

On the other hand, since $n \geqslant \lfloor (6/\delta)\log(3/\delta) \rfloor - 1 \geqslant 2/\delta$, we have that

$$\mathcal{R}_2 = \frac{n-(\tau+1)}{n} \geqslant 1 - \frac{np+2}{n} = 1 - p - \frac{2}{n} \geqslant 1 - p - \delta.$$

Therefore, the rate tuples converge to $(h(p), 1-p)$ in rate almost linear in $1/\delta$. □

The codes we derived from Proposition 1 are not only capacity-achieving. The construction also provides two-write WOM codes with explicit sum-rates for finite values of n.

Let $S(n, \tau)$ denote the maximum size of an τ-partial spread of $\mathbb{F}_2^n$. The following theorem summarizes the results on two-write WOM codes that can be obtained by applying Proposition 1 to the cooling codes constructed in [5].

Theorem 3. *Let $\tau \leqslant n$. There exists an $[n, 2; \tau, M_1, M_2]_A$ WOM code of type A under the following conditions.*

(i) If $\tau \in \{1, n-1, n\}$, then $M_2 = 2^{n-\tau}$.

(ii) For all $\tau \leqslant n$, we have that $M_2 = n - \tau + 1$.

(iii) If $\tau + 1 \leqslant n/2$, then $M_2 = S(n, \tau+1) + 1$.

(iv) Suppose r, s, and d are integers satisfying: $\tau + r \leqslant (n+s)/2$; a binary linear $[n, s, d]$ code exists; and an binary linear $[n-t, r, d]$ code does not exist. Then $M_2 = S(n-s, \tau+r-s) + 1$.

Hence, to obtain lower bounds for M_2, we require certain knowledge of $S(n, \tau)$. Recently, the values of $S(n, \tau)$ has been almost completely determined. We summarize the current knowledge of the best known estimates for $S(n, \tau)$.

Proposition 3 ([21], [25])**.** *Let $\tau \leqslant n/2$. Let r be the integer such that $r \equiv n \bmod \tau$ and $0 \leqslant r < \tau$. Then*

$$S(n, \tau) = \begin{cases} \dfrac{2^n - 2^{\tau+r}}{2^\tau - 1} + 1, & \text{if } \tau > 2^r - 1, \\ \dfrac{2^n - 32}{7} + 2, & \text{if } \tau = 3 \text{ and } r = 2. \end{cases}$$

For all values n and τ, we have $S(n, \tau) \geqslant (2^\tau - 2^{\tau+r})/(2^\tau - 1) + 1$.

Using Proposition 3, we compute the best possible sum-rates and fixed rates[1] that arise from Theorem 3 for lengths up to 40. We benchmark our results with an online table created by Dobbelaere [10]. The comparisons are given in Table I and we observe that Theorem 3 yields the best known rates in some cases.

We also remark that the two-write WOM codes in [10] are obtained from a computer search aided by a scoring heuristic. It is unclear whether the search extends to larger lengths. In contrast, our construction applies *for all lengths* and converges to capacity. For lengths up to 100, we plot the sum-rates in Figure 1 and observe that we are 0.04-close to optimal when $n = 100$.

Observe that when $\tau \leqslant n/2$, Proposition 3 states that $S(n, \tau) \geqslant 2^{n-\tau}$. Applying Theorem 3(iii), we obtain the following corollary that provides two-write WOM codes whose rates are explicitly given for all values of n.

Corollary 1. *Let $\tau + 1 \leqslant n/2$. There exists an $[n, 2; M_1, M_2]$ two-write WOM, where*

$$M_1 = \sum_{i=0}^{\tau}\binom{n}{i}, \quad M_2 = 2^{n-\tau-1}.$$

[1] An $[n, t; M_1, M_2, \ldots, M_t]$ t-write WOM code has *fixed rates* if $M_1 = M_2 = \cdots = M_t$.

TABLE I
TWO-WRITE WOM: COMPARISON WITH PREVIOUS KNOWN CONSTRUCTIONS

Length	Maximum Fixed Rate from [10]	Maximum Fixed Rate using Theorem 3	Maximum Sum-Rate from [10]	Maximum Sum-Rate using Theorem 3
4	**1.16096**	**1.16096**	**1.33048**	**1.33048**
5	**1.2**	1.03399	**1.36147**	1.31699
6	**1.33333**	1.10731	**1.37465**	1.32955
7	**1.37353**	1.19141	**1.4047**	1.3317
8	**1.30236**	1.28232	**1.40599**	1.33864
9	**1.33333**	1.22746	**1.41572**	1.35221
10	**1.4**	1.20888	**1.42479**	1.36799
11	**1.38315**	1.27679	**1.43104**	1.37989
12	**1.37067**	1.34967	**1.44006**	1.38963
13	**1.38462**	1.31727	**1.44905**	1.40301
14	**1.42857**	1.28652	**1.45419**	1.41237
15	**1.39087**	1.33962	**1.45647**	1.41944
16	**1.38326**	1.38072	**1.4608**	1.42925
17	**1.41176**	1.37061	**1.46362**	1.43608
18	**1.44444**	1.3359	**1.46853**	1.44144
19	**1.40858**	1.37081	**1.4725**	1.44866
20	1.4	**1.40225**	**1.47357**	1.45374
21	**1.42857**	1.40645	**1.47835**	1.45797
22	**1.45455**	1.37393	**1.48053**	1.46345
23	**1.42848**	1.39228	**1.48221**	1.46734
24	1.41667	**1.4176**	**1.4854**	1.47078
25	**1.44**	1.43243	**1.48678**	1.47507
26	**1.46154**	1.40462	**1.48929**	1.47816
27	**1.4441**	1.40782	**1.49169**	1.48101
28	1.42857	**1.42897**	**1.49303**	1.48447
29	1.44828	**1.44866**	**1.49498**	1.48697
30	**1.46667**	1.42794	**1.49584**	1.48939
31	**1.45514**	1.41954	1.48522	**1.49224**
32	1.4375	**1.43768**	1.48243	**1.49431**
33	1.45455	**1.45472**	**1.50049**	1.49639
34	**1.47059**	1.44633	**1.50239**	1.49878
35	**1.46563**	1.42865	1.48803	**1.50053**
36	**1.44547**	1.44452	**1.50344**	1.50233
37	1.45946	**1.45954**	1.48983	**1.50437**
38	**1.47368**	1.46125	1.48429	**1.50587**
39	**1.47421**	1.44231	1.49036	**1.50746**
40	**1.45574**	1.45004	1.50736	**1.50921**

Best rates are highlighted in **boldface**.

IV. THREE-WRITE WOM CODES

In this section, we present a few constructions of three-write WOM codes and focus on obtaining codes with high sum-rates. Suppose a family of three-write WOM codes approaches sum-rate $\mathcal{R}_{\text{sum}}$. As before, we consider the minimum code length $n(\delta)$ in order to be δ-close to $\mathcal{R}_{\text{sum}}$ and are interested in families of three-write WOM codes with $n(\delta)$ which is polynomial in $1/\delta$. In addition, we also examine constructions of three-write WOM codes with high sum-rates for finite blocklengths.

Definition 6. An $[n, 2; M_1, M_2]$ two-write WOM code is said to be of *type B* if the encoding map $\mathcal{E}_2$ on the second write is an injection from $[M_2]$ to $J^+(n, \omega)$. In other words, after the second write, at most ω cells are programmed, and we denote such a WOM code as an $[n, 2; \omega, M_1, M_2]_B$ WOM code.

Suppose that there exists an $[n, 2; \omega, M_1, M_2]_B$ WOM code. Then after the second write, at most ω cells are programmed. As observed in previous works [31], [38], if we have an $[n, 2; \omega, M_1', M_2']_A$ WOM code of type A, we may encode another M_2' messages on the third write. Formally, we have the following construction of three-write WOM codes.

Proposition 4. *Let $\omega \leqslant n$. If there exists an $[n, 2; \omega, M_1, M_2]_B$ WOM code and an $[n, 2; \omega, M_1', M_2']_A$ WOM code, then there exists an $[n, 3; M_1, M_2, M_2']$ three-write WOM code.*

Shpilka [31] modified the Rivest-Shamir two-write WOM code [28] to construct an $[n, 2, \omega; M_1, M_2]_B$ WOM code.

Theorem 4. *[31] Let m and ℓ be integers such that $\ell \leqslant 4m$. Set $n = 12m + 5$ and $\omega = \lceil 8m - 5\ell/4 + 5 \rceil$. Then there exists an $[n, 2; \omega, M_1, M_2]_B$ WOM code with*

$$M_1 = \sum_{i=\ell}^{4m} \binom{4m}{i} 3^{4m-i}, \text{ and } M_2 = 4^{4m}.$$

Setting $\ell = \lfloor 1.768m \rfloor$, Shpilka applied Proposition 4 to construct a family of three-write WOM codes whose sum-rates approach 1.809. However, on the third write, Shpilka applies a two-write WOM code derived from a Wozencraft ensemble. Hence, the sum-rate of the three-write WOM codes approaches 1.809 with exponential rate. A simple modification to this scheme is to apply the two-write WOM code from Theorem 2. Since at most ω cells are programmed after the second write and if we set $\omega + 1 \leqslant n/2$, or $\lceil 8m - 5\ell/4 \rceil \leqslant 6m + 1$, we are able to apply Theorem 2

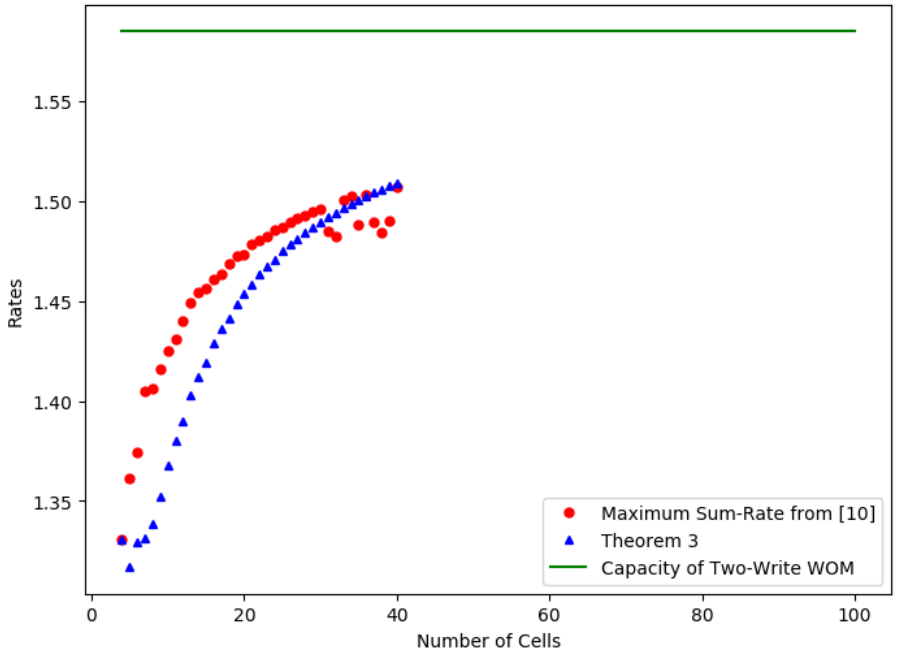

Fig. 1. Rates of known constructions for lengths at most 100.

to encode $2^{n-(\omega+1)}$ messages. Since the two-write WOM code has almost-linear convergence rate, the sum-rate of the modified three-write WOM code approaches 1.809 with almost-linear rate. We summarize this discussion in the following theorem.

Theorem 5 (Construction 2)**.** *Let m and ℓ be integers such that $\ell \leqslant 4m$ and $\lceil 8m - 5\ell/4 \rceil \leqslant 6m+1$. Set $n = 12m+5$ and $\omega = \lceil 8m - 5\ell/4 + 5 \rceil$. There exists an $[n, 3; M_1, M_2, M_3]$ three-write WOM, where*

$$M_1 = \sum_{i=\ell}^{4m} \binom{4m}{i} 3^{4m-i}, \quad M_2 = 4^{4m}, \quad M_3 = 2^{12m-(\omega+1)}.$$

Moreover, the encoding / decoding complexity is polynomial $O(n^3)$ and if we set $\ell = \lfloor 1.768m \rfloor$, these codes approach sum-rate 1.809 *with almost-linear rate.*

While Construction 1 yields a family of WOM codes whose sum-rates approach 1.809 with polynomial rate, at lengths less than 100, the sum-rates are relatively low. For example, for $n = 89$ cells, Construction 1 yields a three-write WOM code with sum-rate 1.673.

The next two constructions yield three-write WOM codes of short lengths with higher sum-rates. Our second construction modifies known two-write WOM codes by expurgating certain messages on the second write so that the resulting two-write WOM codes are of type B.

Specifically, given an $[n, 2; M_1, M_2]$ two-write WOM code $\mathbb{C}$, we consider the set of messages $[M_2]$ on the second write. For an integer $\omega < n$, we say that a message $m \in [M_2]$ is *ω-bad* if there exists a cell-state vector $\boldsymbol{c}$ in $Im(\mathcal{E}_1)$ such that the codeword $\mathcal{E}(m, \boldsymbol{c})$ has weight strictly larger than ω. Define $B(\mathbb{C}, \omega)$ to be the set of all ω-bad words, in other words, $B(\mathbb{C}, \omega) = \{m \in [M_2] : m \text{ is } \omega\text{-bad}\}$.

Now, if we only write the message in $[M_2] \setminus B(\mathbb{C}, \omega)$ on the second write, the number of programmed cells is at most ω. Therefore, we obtain an $[n, 2; \omega, M_1, M_2 - |B(\mathbb{C}, \omega)|]_B$ WOM code. We summarise the discussion in the following theorem.

Theorem 6 (Construction 3)**.** *If an $[n, 2; M_1, M_2]$ two-write WOM code $\mathbb{C}$ exists, then an $[n, 2; \omega, M_1, M_2 - |B(\mathbb{C}, \omega)|]_B$ WOM code exists. Suppose further that there exists an $[n, 2; \omega, M_1', M_3]_A$ WOM code, where $M_1' = M_2 - |B(\mathbb{C}, \omega)|$. Then there exists an $[n, 3; M_1, M_2, M_3]$ three-write WOM code.*

Unfortunately, determining the size of the set $B(\mathbb{C}, \omega)$ is computationally difficult. Nevertheless, since the number of ω-bad messages is at most the number of binary vectors with weight at least $\omega + 1$, we have that

$$B(\mathbb{C}, \omega) \leqslant \sum_{i=\omega+1}^{n} \binom{n}{i}.$$

Corollary 2 (Construction 3a)**.** *If an $[n, 2; M_1, M_2]$ two-write WOM code exists, then an $[n, 2; \omega, M_1, M_2 - \sum_{i=\omega+1}^{n} \binom{n}{i}]_B$ WOM code exists. Suppose further that there exists an $[n, 2; \omega, M_1', M_3]_A$ WOM code. Then there exists an $[n, 3; M_1, M_2, M_3]$ three-write WOM code.*

The next corollary is immediate from Theorem 3.

Corollary 3 (Construction 3b)**.** *Let τ and ω be integers such that $\tau + 1 \leqslant n/2$, and $\sum_{i=\omega+1}^{n} \binom{n}{i} < S(n, \tau+1) + 1$. Then there exists an $[n, 2, \omega; M_1, M_2]_B$ two-write WOM of type B, where*

$$M_1 = \sum_{i=0}^{\tau} \binom{n}{i}, \text{ and } M_2 = 1 + S(n, \tau+1) - \sum_{i=\omega+1}^{n} \binom{n}{i}.$$

Suppose further that an $[n, 2; \omega, M_1', M_3]_A$ *WOM code exists. Then there exists an* $[n, 3; M_1, M_2, M_3]$ *three-write WOM code.*

Our third construction extends a construction of two-write WOM codes which was introduced in [37]. Let $\mathcal{C}$ be a linear $[n, k]$-code with parity check matrix $\boldsymbol{H}$. For any binary word $\boldsymbol{c}$, the $(n-k) \times n$ matrix $\boldsymbol{H}(\boldsymbol{c})$ is defined as follows. The ith column of $\boldsymbol{H}(\boldsymbol{c})$ is set to be the ith column of $\boldsymbol{H}$ if $\boldsymbol{c}_i = 1$ and otherwise it is set to be the zeroes column.

Let t and w be integers. We define $W(t, w; \boldsymbol{H})$ to be a set of binary words so that the following hold for each $\boldsymbol{x} \in W(t, w; \boldsymbol{H})$.

(H1) The weight of $\boldsymbol{x}$ is at most t.
(H2) There exists a word $\boldsymbol{c}$ with weight at most w such that

$$\text{the columns of } \boldsymbol{H}(\boldsymbol{c}) \text{ sum to zero, and} \tag{3}$$

$$\operatorname{rank} \boldsymbol{H}(\boldsymbol{c}) = n - k, \tag{4}$$

$$\operatorname{supp}(\boldsymbol{c}) \cap \operatorname{supp}(\boldsymbol{x}) = \varnothing. \tag{5}$$

Theorem 7 (Construction 4)**.** *Let* $\mathcal{C}$ *be a linear* $[n, k]$*-code with parity check matrix* $\boldsymbol{H}$*. Let* t *and* w *be integers such that the set* $W(t, w; \boldsymbol{H})$ *in nonempty and set* $\omega = t + \lfloor w/2 \rfloor$*. Then there exists an* $[n, 2; \omega, M_1, M_2]_B$ *WOM code of type B, where*

$$M_1 = |W(t, w; \boldsymbol{H})| \text{ and } M_2 = 2^{n-k}.$$

Suppose further that an $[n, 2; \omega, M_1', M_3]_A$ *WOM code exists. Then there exists an* $[n, 3; M_1, M_2, M_3]$ *three-write WOM code.*

Proof: We outline the encoding and decoding maps of the two writes.

First write. We encode one of the M_1 messages by programming a vector $\boldsymbol{x} \in W(t, w; \boldsymbol{H})$.

Second write. We pick a binary vector $\boldsymbol{m}$ of length $n-k$. Making use of the check matrix $\boldsymbol{H}$, we encode $\boldsymbol{m}$ using a word of weight of at most ω. We provide a formal description of this encoding map below.

Suppose that $\boldsymbol{x}$ is written on the first write and $\boldsymbol{m}$ is the message to be written on the second write. Set $\boldsymbol{m}' \triangleq \boldsymbol{H}\boldsymbol{x}^T + \boldsymbol{m}$. By the definition of $W(t, w; \boldsymbol{H})$, there exists a word $\boldsymbol{c}$ that satisfies (3), (4) and (5). Since $\operatorname{rank} \boldsymbol{H}(\boldsymbol{c}) = n-k$, there exists $\boldsymbol{v}_1$ such that $\boldsymbol{H}(\boldsymbol{c})\boldsymbol{v}_1^T = \boldsymbol{m}'$. Set $\boldsymbol{v}_2 \triangleq \boldsymbol{c} + \boldsymbol{v}_1$ and since the columns of $\boldsymbol{H}(\boldsymbol{c})$ sum to zero, we also have that $\boldsymbol{H}(\boldsymbol{c})\boldsymbol{v}_2^T = \boldsymbol{m}'$. Since $\boldsymbol{v}_1 + \boldsymbol{v}_2 = \boldsymbol{c}$ and the weight of $\boldsymbol{c}$ is at most w, one of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ has weight at most $\lfloor w/2 \rfloor$. Let $\boldsymbol{v}$ be the vector and on the second write, we program $\boldsymbol{v} + \boldsymbol{x}$. Since $\operatorname{supp}(\boldsymbol{v}) \subseteq \operatorname{supp}(\boldsymbol{c})$, we have that $\boldsymbol{v} + \boldsymbol{x} \geqslant \boldsymbol{x}$. Also, since $\boldsymbol{x}$ has weight at most t, we check that $\boldsymbol{v} + \boldsymbol{x}$ has weight at most $\omega = \lfloor w/2 \rfloor + t$ as desired.

We present the decoding map of the second write. Given a codeword $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{x}$, we recover the message vector $\boldsymbol{m}$ by multiplying the check matrix $\boldsymbol{H}$. In other words, $\boldsymbol{m} = \boldsymbol{H}\boldsymbol{u}^T$. □

Example 1.

(i) Let $n = 31$ and $k = 17$. We consider the best known $[31, 17]$-linear code [17] with parity check matrix

$$\boldsymbol{H} = \begin{pmatrix}
1\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\\
0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,1\,1\,1\,0\,1\,0\\
0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,0\,1\,1\,1\\
0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,1\,0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,0\,1\,0\,0\,1\,0\\
0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,1\,1\,1\,1\,0\\
0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,1\,1\,0\\
0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,1\,1\,1\,0\,0\,1\,1\,0\,0\,0\,0\\
0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,1\,1\,0\,1\,0\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\,0\,1\\
0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,1\,0\,0\,1\,1\,0\,1\,0\,0\,1\,0\,1\,1\,1\\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,0\,0\,1\,0\\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0\,1\\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,1\,0\,1\,1\\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,1\,1\,0\,0\,1\,1\,1\,0
\end{pmatrix}.$$

We set $t = 10$ and $w = 15$, and so, $\omega = 10 + \lfloor 15/2 \rfloor = 17$. Then a computer search yields the size of $W(t, w; \boldsymbol{H})$ to be 61644301. Hence, we have an $[31, 2; 17, M_1, M_2]_B$ WOM code

$$M_1 \approx 2^{25.877} \text{ and } M_2 = 2^{14}.$$

Setting $n = 31$, $\tau = 17$ with $r = 1$, $s = 5$, $d = 16$, Theorem 3(iv) yields an $[31, 2; 17, \sum_{i=0}^{17} \binom{31}{i}, 2^{13}]_A$ WOM code (see also [5, Table I]). Hence, we obtain an $[31, 3; 2^{25.877}, 2^{14}, 2^{13}]$ three-write WOM code whose sum-rate is 1.706.

(ii) Set $n = 93$. By concatenating three copies of the $[31, 2; 17, 2^{25.877}, 2^{14}]_B$ WOM code constructed in (i), we obtain an $[93, 2; 51, 2^{77.632}, 2^{42}]_B$ WOM code. Similar to (i), we set $n = 93$, $\tau = 51$ with $r = 2$, $s = 13$, $d = 38$ in Theorem 3(iv) and obtain an $[93, 2; 51, \sum_{i=0}^{51} \binom{93}{i}, 2^{40}]_A$ WOM code. Hence, we obtain an $[93, 3; 2^{77.632}, 2^{42}, 2^{40}]$ three-write WOM code whose sum-rate is 1.716. This yields the best known sum-rate for three-write WOM codes whose blocklength is at most 100.

To conclude this section, we determine the maximum sum-rates resulting from Constructions 1, 2a, 2b and 3 for blocklengths up to 50. For Construction 3, we consider the best known linear codes of lengths up to 34 [17]. For each parity check matrix $\boldsymbol{H}$, we set $w = \min\{\operatorname{wt}(\boldsymbol{c}) : \boldsymbol{c} \text{ satisfying (3) and (4)}\}$ and then compute the size of $W(t, w; \boldsymbol{H})$ for values of t. The best sum-rate for a fixed value of n is then reported. As before, we benchmark our results with the online table created by Dobbelaere [10] and the comparisons are given in Table II.

V. ϵ-Error Three-Write WOM Codes

In this section, we present constructions of three-write ϵ-error WOM codes. In addition to polynomial encoding / decoding complexity and almost-linear convergence rate, we also demonstrate that the failure decay rate is faster as compared to certain existing constructions. At the end of this section, we demonstrate Theorem 1.

A. Background and State of the Art Results

The first class of ϵ-error WOM codes was constructed in [15] and [18] to demonstrate the achievability of the

TABLE II
THREE-WRITE WOM: COMPARISON WITH PREVIOUS KNOWN CONSTRUCTIONS

Length	Max Sum-Rate from [10]	Max Sum-Rate Construction 2	Max Sum-Rate Construction 3a	Max Sum-Rate Construction 3b	Max Sum-Rate Construction 4
4	–	–	**1.53232**	1.33048	1.29248
5	1.49837	–	**1.52294**	1.35098	1.31699
6	**1.56539**	–	1.5258	1.3832	1.39872
7	**1.54483**	–	1.53641	1.42078	1.49185
8	1.51696	–	**1.54456**	1.43686	1.29024
9	1.52832	–	**1.55375**	1.46581	1.46826
10	1.55098	–	**1.55608**	1.48258	1.5268
11	**1.65032**	–	1.56222	1.49766	1.54771
12	**1.62432**	–	1.56054	1.50736	1.55836
13	–	–	**1.56545**	1.51228	1.5606
14	–	–	**1.57314**	1.51941	1.55942
15	–	–	**1.57771**	1.52599	1.57231
16	**1.6875**	–	1.5754	1.53408	1.61481
17	–	1.26445	1.57463	1.53993	**1.63447**
18	–	–	1.57619	1.54463	**1.61079**
19	–	–	1.58809	1.5465	**1.61729**
20	–	–	1.59585	1.5539	**1.62959**
21	–	–	1.60638	1.56088	**1.64857**
22	–	–	1.61916	1.57343	**1.66411**
23	1.43478	–	1.63332	1.58215	**1.67166**
24	–	–	**1.64127**	1.58862	1.62631
25	–	–	1.61315	1.589	**1.66923**
26	–	–	1.61808	1.59166	**1.67325**
27	–	–	1.62574	1.59399	**1.67875**
28	–	–	1.62549	1.60173	**1.66284**
29	–	1.4781	1.63274	1.60847	**1.69975**
30	–	–	1.63403	1.61597	**1.7047**
31	–	–	1.63866	1.61852	**1.70574**
32	–	–	1.64105	1.62463	**1.70259**
33	–	–	1.6457	1.62651	**1.69016**
34	–	–	1.6456	1.63148	**1.67511**
35	–	–	**1.64262**	1.63508	–
36	–	–	**1.6503**	1.63959	–
37	–	–	**1.6447**	1.64047	–
38	–	–	**1.6557**	1.64419	–
39	–	–	**1.65696**	1.64503	–
40	–	–	**1.66273**	1.64683	–
41	–	1.57294	–	**1.65228**	–
42	–	–	–	**1.65448**	–
43	–	–	–	**1.65964**	–
44	–	–	–	**1.66124**	–
45	–	–	–	**1.66425**	–
46	–	–	–	**1.66617**	–
47	–	–	–	**1.66845**	–
48	–	–	–	**1.66959**	–
49	–	–	–	**1.67311**	–
50	–	–	–	**1.67409**	–

Best rates are highlighted in **boldface**.

capacity region. However, the encoding is random and therefore, the coding scheme has no efficient encoding / decoding methods.

Recently, a number of capacity-achieving ϵ-error WOM codes with polynomial encoding / decoding methods were proposed. Burshtein and Strugatski [2] used polar codes to construct efficient capacity-achieving ϵ-error WOM codes. They showed that these multi-write WOM codes have polynomial convergence rate and that the probability of failure for each write is at most $2^{-\sqrt{n}}$. Later, Gad *et al.* [12] and Kumar *et al.* [20] used LDPC techniques to construct efficient capacity-achieving ϵ-error two-write WOM codes. However, no estimates on the failure decay and the convergence rates were given. Finally, as mentioned earlier, the zero-error WOM codes in [30] achieve capacity with low complexity encoding / decoding algorithms. However, the sum-rates approach capacity in rate exponential in $1/\delta$.

In the next subsection, we construct efficient ϵ-error three-write WOM codes whose sum-rates converge with an almost-linear rate and whose failure probability is at most $2^{-\lambda n+o(n)}$ for some positive constant λ. In other words, we achieve all three figures of merit. However, in contrast with Burshtein and Strugatsky's construction, even though the failure decay rate is fast, we are unable to achieve full capacity or the maximum sum-rate 2. Instead, we obtain WOM codes with sum-rate approaching 1.936.

B. Construction

We borrow ideas from the zero-error case for three-write WOM codes to construct our three-write ϵ-error WOM codes.

In particular, we use the WOM codes of type A given in Definition 3.

Proposition 5. *Suppose that there exists an* $[n, 2; \tau, M_1, M_2]_A$ *WOM code and* $[n, 2; \tau', M_1', M_2']_A$ *WOM code. Set* $f = \sum_{i=\tau'+1}^{n-\tau} \binom{n-\tau}{i}$. *If* $f < M_2$, *then there exists an* $[n, 3; M_1, M_2, M_2'; 0, f/M_2, 0]$ *three-write* ϵ*-error WOM code.*

Proof: Let $\mathbb{C}$ be an $[n, 2; \tau, M_1, M_2]_A$ WOM code with encoding maps $\mathcal{E}_1, \mathcal{E}_2$ and corresponding decoding maps $\mathcal{D}_1, \mathcal{D}_2$. Similarly, let $\mathbb{C}'$ be an $[n, 2; \tau', M_1', M_2']_A$ WOM code with encoding / decoding maps $\mathcal{E}_1', \mathcal{E}_2', \mathcal{D}_1', \mathcal{D}_2'$. Using these maps, we provide the encoding / decoding maps $\mathcal{E}_i^\epsilon, \mathcal{D}_i^\epsilon, i \in$ [3] for our three-write ϵ-error WOM code.

First write. Set $\mathcal{E}_1^\epsilon = \mathcal{E}_1$ and $\mathcal{D}_1^\epsilon = \mathcal{D}_1$. Hence, after the first write, at most τ cells are programmed and the probability of failure is zero.

Second write. Let $\boldsymbol{c} \in Im(\mathcal{E}_1^\epsilon)$ and so, $\mathrm{wt}(\boldsymbol{c}) \leqslant \tau$. Set $\boldsymbol{c}'$ such that $\mathrm{wt}(\boldsymbol{c}') = \tau$ and $\mathrm{supp}(\boldsymbol{c}') \supseteq \mathrm{supp}(\boldsymbol{c})$. For $m \in [M_2]$, set

$$\mathcal{E}_2^\epsilon(m, \boldsymbol{c}) = \begin{cases} \mathcal{E}_2(m, \boldsymbol{c}'), & \text{if } \mathrm{wt}(\mathcal{E}_2(m, \boldsymbol{c}')) \leqslant \tau', \\ \mathsf{fail}, & \text{otherwise.} \end{cases}, \quad \mathcal{D}_2^\epsilon = \mathcal{D}_2.$$

In other words, we only program the cells on the second write if the codeword written has weight at most τ'. Otherwise, we declare a failure. Therefore, on the second write, at most τ' cells are programmed.

Next, we estimate the failure probability. Suppose that the cell-state vector is $\boldsymbol{c} \in Im(\mathcal{E}_1^\epsilon)$ after the first write and let $\boldsymbol{c}'$ be the word of weight τ with $\mathrm{supp}(\boldsymbol{c}') \supseteq \mathrm{supp}(\boldsymbol{c})$. Consider the set $F_{\boldsymbol{c}} = \{m \in [M_2] : \mathcal{E}_2^\epsilon(m, \boldsymbol{c}') = \mathsf{fail}\}$. Since $\boldsymbol{m} \in F_{\boldsymbol{c}}$ implies that $\mathcal{E}_2(\boldsymbol{m}, \boldsymbol{c}')$ has weight at least $\tau' + 1$ and $\mathcal{E}_2(\boldsymbol{m}, \boldsymbol{c}') \neq \mathcal{E}_2(\boldsymbol{m}', \boldsymbol{c}')$ for $\boldsymbol{m} \neq \boldsymbol{m}'$, we have that the size of $F_{\boldsymbol{c}}$ is upper bounded by the set of words with weight at least $\tau' + 1$ whose support contain $\mathrm{supp}(\boldsymbol{c}')$. In other words,

$$|F_{\boldsymbol{c}}| \leqslant \sum_{i=\tau'+1}^{n-\tau} \binom{n-\tau}{i} = f.$$

Therefore,

$$\begin{aligned} &|\{(m, \boldsymbol{c}) \in [M_2] \times Im(\mathcal{E}_1^\epsilon) : \mathcal{E}_2^\epsilon(m, \boldsymbol{c}) = \mathsf{fail}\}| \\ &= \sum_{\boldsymbol{c} \in Im(\mathcal{E}_1^\epsilon)} |F_{\boldsymbol{c}}| \quad \leqslant f|Im(\mathcal{E}_1^\epsilon)| = (f/M_2)M_2|Im(\mathcal{E}_1^\epsilon)|, \end{aligned}$$

as required.

Third write. Set $\mathcal{E}_3^\epsilon = \mathcal{E}_2'$ and $\mathcal{D}_3^\epsilon = \mathcal{D}_2'$. Since after the second write, at most τ' cells are programmed, the encoding map $\mathcal{E}_3^\epsilon = \mathcal{E}_2'$ always succeeds and so, the probability of failure is zero. □

As before, we use the efficient cooling codes given in Proposition 2 to seed Proposition 5.

Theorem 8. *Choose* $p, \kappa > 0$ *such that* $p < 1/2$ *and* $(1 + p)/2 + \kappa \leqslant 0.687$. *Set* $\tau_1 = \lceil pn \rceil$ *and* $\tau_2 = \lceil ((1 + p)/2 + \kappa)\, n \rceil$. *Then an* $[n, 3; M_1, M_2, M_3; \epsilon_1, \epsilon_2, \epsilon_3]$ *three-write* ϵ*-error WOM code* $\mathbb{C}_n$ *exists with*

$$M_1 = \sum_{i=0}^{\tau_1} \binom{n}{i}, \quad M_2 = 2^{n-(\tau_1+1)}, \quad M_3 = 2^{n-(\tau_2+1)},$$

and $\epsilon_1 = \epsilon_3 = 0$, $\epsilon_2 = 2^{-\lambda n + o(n)}$ *for some* λ *(dependent on* p *and* κ *only). Furthermore,* $\mathbb{C}_n$ *has encoding / decoding complexity* $O(n^3)$ *and the rate tuple approaches* $\big(h(p), (1 - p), (1 - p)/2 - \kappa\big)$ *at almost-linear rate.*

Proof: Proposition 1 with Propositions 2(i) and (ii) provide an $[n, 2; \tau_1, \sum_{i=0}^{\tau_1} \binom{n}{i}, 2^{n-(\tau_1+1)}]_A$ WOM code and an $[n, 2; \tau_2, \sum_{i=0}^{\tau_2} \binom{n}{i}, 2^{n-(\tau_2+1)}]_A$ WOM code. Applying Proposition 5 yields an $[n, 3; M_1, M_2, M_3; 0, \epsilon_2, 0]$ three-write ϵ-error WOM code $\mathbb{C}_n$.

Observe that the probability of failure on the second write is at most

$$\begin{aligned} \frac{\sum_{i=\tau_2+1}^{n-\tau_1} \binom{n-\tau_1}{i}}{M_2} &\leqslant \frac{2^{(n-\tau_1)h(\tau_2/(n-\tau_1))}}{2^{n-(\tau_1+1)}} \\ &\leqslant 2^{(n-\tau_1)(h(\tau_2/(n-\tau_1))-1)} \\ &\leqslant 2^{-((1-p)n-1)(1-h(\tau_2/(n-\tau_1)))}. \end{aligned}$$

On the other hand,

$$\begin{aligned} 1 - \frac{\tau_2}{n - \tau_1} = \frac{n - \tau_1 - \tau_2}{n - \tau_1} &\leqslant \frac{n - pn - ((1 + p)/2 + \kappa)n}{n - pn + 1} \\ &= \left(\frac{1}{2} - \frac{\kappa}{1 - p}\right)(1 - o(1)). \end{aligned}$$

Therefore, the probability of failure is at most $2^{-\lambda n + o(n)}$ when we choose

$$\lambda = (1 - p)\left(1 - h\left(\frac{1}{2} - \frac{\kappa}{1 - p}\right)\right).$$

Next, $\mathbb{C}_n$ has encoding / decoding complexity $O(n^3)$ as the type-A WOM codes have encoding / decoding complexities $O(n^3)$. Finally, we have that the rates

$$\begin{aligned} \frac{\log \sum_{i=0}^{\tau_1} \binom{n}{i}}{n} &\to h(p), \\ \frac{\log 2^{n-(\tau_1+1)}}{n} &\to (1 - p), \\ \frac{\log 2^{n-(\tau_2+1)}}{n} &\to \frac{1 - p}{2} - \kappa, \end{aligned}$$

as $n \to \infty$. Similar to the proof of Theorem 2, we can demonstrate that the rate tuple converges at almost-linear rate. □

Theorem 8 provides a family of ϵ-error WOM codes whose sum-rates approach $h(p) + 3(1 - p)/2 - \kappa$. This sum-rate achieves a maximum of $1.936 - \kappa$ when p is chosen to be $(2\sqrt{2} - 1)/7 \approx 0.261$. However, the encoding / decoding maps for the third write may not be constructible in polynomial time (see Appendix A-C for a discussion).

Nevertheless, we can modify the method in Proposition 2(ii) to construct cooling codes, or equivalently, type-A WOM codes, in polynomial time. However, this alters the range of values for p that is applicable in Theorem 8. In particular, to maximize the sum-rate for this new range, we can only set $p = 0.18$ and the corresponding sum-rate is 1.910. We summarize the discussion in the following corollary.

Corollary 4. *Fix* $\kappa > 0$.

(i) There exists a family of three-write ϵ*-error codes with encoding / decoding complexity* $O(n^3)$ *whose sum-rates approach* $1.936 - \kappa$ *at almost-linear rate.*

(ii) There exists a family of three-write ϵ-error codes $\mathbb{C}_n$ with encoding / decoding complexity $O(n^3)$ whose sum-rates approach $1.910 - \kappa$ at almost-linear rate. Furthermore, $\mathbb{C}_n$ can be constructed in polynomial time.

C. Proof of Theorem 1

We end this section by providing a complete proof of Theorem 1. To this end, we demonstrate the following lemma.

Lemma 1. *Let $\{\mathbb{C}_n\}$ be a family of WOM codes such that each $\mathbb{C}_n$ is an $[n, t; M_1(n), \ldots, M_t(n); \epsilon_1(n), \epsilon_2(n), \ldots, \epsilon_t(n)]$ t-write ϵ-error WOM code and $\mathcal{R}^{\epsilon}_{sum} > \log_2(t+1)$. Then $\lim_{n\to\infty} \sum_{i\in[t]} \epsilon_i(n) \geqslant 1$.*

Proof: Let $\log_2(t+1) < R' < \mathcal{R}^{\epsilon}_{\text{sum}}$. Then for sufficiently large n, we have that $\prod_{i\in[t]} M_i(n) \geqslant 2^{nR'}$.

We mimic the proof of Wolf *et al.* [34]. For each t-tuple of messages $\boldsymbol{m} = (m)_{i=1}^{t} \in [M_1(n)] \times [M_2(n)] \times \cdots \times [M_t(n)]$, we recursively define $\mathbf{C}(\boldsymbol{m}) = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_t)$, where

$$\boldsymbol{c}_i = \begin{cases} \mathcal{E}_i(m_i, \boldsymbol{c}_{i-1}), & \text{if } \boldsymbol{c}_{i-1} \neq \mathsf{fail}, \\ \mathsf{fail}, & \text{if } \boldsymbol{c}_{i-1} = \mathsf{fail}. \end{cases}$$

We say that $\mathbf{C}(\boldsymbol{m})$ is *good* if $\boldsymbol{c}_i \neq \mathsf{fail}$ for all $i \in [t]$. Otherwise, we say that $\mathbf{C}(\boldsymbol{m})$ is *bad*. We claim that the fraction of bad $\mathbf{C}(\boldsymbol{m})$'s approaches one as n grows. Equivalently, we show that the fraction of good $\mathbf{C}(\boldsymbol{m})$'s approaches zero.

When $\mathbf{C}(\boldsymbol{m})$ is good, we can arrange the t binary vectors of length n in a $t \times n$-matrix. Furthermore, since all t writes are successful, all n columns belong to the following set of cardinality $t+1$:

$$\left\{ \begin{bmatrix}0\\0\\\vdots\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix}, \begin{bmatrix}0\\0\\\vdots\\1\\1\end{bmatrix}, \cdots, \begin{bmatrix}0\\1\\\vdots\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\\vdots\\1\\1\end{bmatrix} \right\}.$$

In other words, the number of good $\mathbf{C}(\boldsymbol{m})$'s is at most $(t+1)^n$. Hence, the fraction of good $\mathbf{C}(\boldsymbol{m})$'s is at most

$$\frac{(t+1)^n}{\prod_{i\in[t]} M_i(n)} \leqslant \left(\frac{t+1}{2^{R'}}\right)^n.$$

Since $\log_2(t+1) < R'$, this fraction of good $\mathbf{C}(\boldsymbol{m})$'s tends to zero as n grows.

Finally, using the union bound, we have that $\sum_{i\in[t]} \epsilon_i(n)$ is at least the fraction of bad $\mathbf{C}(\boldsymbol{m})$'s and hence, the theorem follows. □

We complete the proof of Theorem 1.

Proof: [Proof of Theorem 1] First, observe that $\epsilon_1(n) = 0$ is zero for all n. Hence, we have that $\lim_{n\to\infty} \sum_{i\in[t]} \epsilon_i(n) \leqslant \lim_{n\to\infty} \sum_{i=2}^{t} \epsilon_i(n) \leqslant (t-1)\epsilon' < 1$. Lemma 1 then implies that $\mathcal{R}^{\epsilon}_{\text{sum}} \leqslant \log_2(t+1)$. □

VI. Conclusions and Open Problems

We studied explicit WOM codes with low encoding / decoding complexity and polynomial convergence rate. In the two-write case, we provided a capacity-achieving construction which has polynomial complexity and almost-linear convergence rate. Unfortunately, our method was unable to provide capacity-achieving codes for the case of more than two writes. Nevertheless, for three writes, we constructed zero-error WOM codes that improved the best known sum-rates of explicit WOM codes and ϵ-error WOM codes with high sum-rates and faster failure decay rates.

While the results in the paper provide a significant contribution in the area of WOM codes, there are still several interesting problems which are left open.

- Finding better explicit constructions for the three-write case and capacity-achieving construction with polynomial complexity and polynomial convergence rate. This is challenging and stimulating for the case of more than three writes, where less is known.
- Recently, in order to combat the limited-endurance and overshooting problems in multilevel cell (MLC) flash memories, a new subclass of *non-binary* WOM codes, called *write ℓ-step-up memories (WℓM) codes*, was proposed and studied. In this subclass, each cell has $q \geqslant 2$ levels and each write can only increase a cell level by at most $\ell < q$. Techniques that rely on cooling codes (see Proposition 2) were used to construct explicit high-rate WℓM codes. Specifically, in [7], the authors provided an explicit construction of three-write ternary WℓM codes with $\ell = 1$ and sum-rates approaching 2.77. Interested readers are refered to [7] for the detailed constructions.

Appendix A
Efficient Encoding and Decoding for Cooling Codes

We provide a detailed proof for Proposition 2.

A. When $\tau + 1 \leqslant n/2$

Suppose that $\tau + 1 \leqslant n/2$. The case for $\tau + 1 | n$ is detailed in [5] and [11]. Here, we generalize the previous encoding and decoding methods to all pairs of $\tau + 1$ and n satisfying $\tau + 1 \leqslant n/2$.

Given $\tau + 1 \leqslant n/2$, set $r \equiv n \pmod{\tau+1}$ with $0 \leqslant r < \tau + 1$ and $s = \lfloor n/(\tau+1) \rfloor$. Hence, $n = s(\tau+1) + r$ and $s \geqslant 2$.

Here, we identify our messages in $[2^{n-(\tau+1)}]$ with binary strings of length $n - (\tau+1) = (s-1)(\tau+1) + r$. We also identify binary strings of length $\tau+1$ and $\tau+1+r$ with field elements in $\mathbb{F}_{2^{\tau+1}}$ and $\mathbb{F}_{2^{\tau+1+r}}$, respectively.

So, for a message $\boldsymbol{m}$ of length $(s-1)(\tau+1)+r$, we partition it into $(s-2)$ blocks of length $\tau+1$ and one block of length $\tau+1+r$. We identify $\boldsymbol{m}$ with the $(s-1)$-tuple $(\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{s-1}) \in \mathbb{F}_{2^{\tau+1}}^{s-2} \times \mathbb{F}_{2^{\tau+1+r}}$. In other words, $\boldsymbol{m}_i \in \mathbb{F}_{2^{\tau+1}}$ for $i \in [s-2]$ and $\boldsymbol{m}_{s-1} \in \mathbb{F}_{2^{\tau+1+r}}$. Finally, we define the codeset $\mathcal{C}_{\boldsymbol{m}}$ as follow:

$$\mathcal{C}_{\boldsymbol{m}} = \Big\{ \boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m}) = (\boldsymbol{\beta}\boldsymbol{m}_1, \boldsymbol{\beta}\boldsymbol{m}_2, \ldots, \boldsymbol{\beta}\boldsymbol{m}_{s-2}, \boldsymbol{\beta}\boldsymbol{m}_{s-1}, \boldsymbol{\beta}) : \boldsymbol{\beta} \in \mathbb{F}_2^{\tau+1} \setminus \{\mathbf{0}\} \Big\},$$

where $\boldsymbol{\beta}\boldsymbol{m}_{s-1}$ is obtained by padding $\boldsymbol{\beta}$ with r zeroes and then regarding $\boldsymbol{\beta}$ as a field element in $\mathbb{F}_2^{\tau+1+r}$.

Set $\mathbb{C} = \left\{\mathcal{C}_{\boldsymbol{m}} : \boldsymbol{m} \in \mathbb{F}_2^{n-(\tau+1)}\right\}$ and we demonstrate that $\mathbb{C}$ is an (n, τ) cooling code by providing the encoding / decoding algorithms.

Encoding. Given $S \subseteq [n]$ with $|S| \leqslant \tau$, our encoding task is to compute $\boldsymbol{\beta} \in \mathbb{F}_{2^{\tau+1}} \setminus \{\mathbf{0}\}$ such that $\mathrm{supp}(\boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m})) \cap S = \varnothing$.

To find $\boldsymbol{\beta}$, we consider the map $\psi : \mathbb{F}_{2^{\tau+1}} \to \mathbb{F}_2^{|S|}$ such that $\psi(\boldsymbol{\beta}) = \boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m})|_S$, where $\boldsymbol{w}|_S$ is the projection of $\boldsymbol{w}$ to the coordinate set S. Since $\psi(\boldsymbol{\beta} + \boldsymbol{\beta}') = \psi(\boldsymbol{\beta}) + \psi(\boldsymbol{\beta}')$, ψ is an $\mathbb{F}_2$-linear map. Furthermore, since $|S| < \tau + 1$, the kernel of ψ is nontrivial and our encoding task is reduced to finding a nonzero element $\boldsymbol{\beta}$ in the kernel. As $\tau + 1 = O(n)$ and $|S| = O(n)$, an element $\boldsymbol{\beta}$ can be computed in $O(n^3)$ time.

Decoding. Given a codeword $\boldsymbol{c} \in \mathbb{F}_2^n$, we partition $\boldsymbol{c}$ into $(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{s-1}, \boldsymbol{c}_s)$ such that the length of $\boldsymbol{c}_i$ is $\tau + 1$ for $i \in [s] \setminus \{s-1\}$ and the length of $\boldsymbol{c}_{s-1}$ is $\tau + 1 + r$. To recover the message $\boldsymbol{m} \in \mathbb{F}_2^{n-(\tau+1)}$, we set $\boldsymbol{\beta} = \boldsymbol{c}_s$ and compute $\boldsymbol{m}_i = \boldsymbol{\beta}^{-1}\boldsymbol{c}_i$ for $i = [s-1]$. Note that for $i = s-1$, we regard as $\boldsymbol{\beta}$ as an element in $\mathbb{F}_{2^{\tau+1+r}}$.

Therefore, $\mathbb{C}$ is an (n, τ) cooling code of size $2^{n-(\tau+1)}$. Encoding and decoding for the code $\mathbb{C}$ has complexity $O(n^3)$.

B. When $\tau \leqslant 0.687n$

Let $\tau \leqslant 0.687n$ and set $\alpha = \tau/n$ and $r = \lceil \log(2n+1) \rceil$. In [5], the authors provided a construction of an (n, τ) cooling code of size $2^{n-(\tau+r)}$ and hinted the existence of efficient encoding and decoding methods. Here, we modify the methods in the previous subsection and provide a pair of explicit encoding and decoding algorithms.

We set $d = \lceil (n-\tau)/2 \rceil$ and choose $s = \lceil (1 - h(d/n))n \rceil = \lceil (1 - h((1-\alpha)/2))n \rceil$. By the Gilbert-Varshamov bound (see [27]), there exists an $[n, s, d]$-linear code $\mathcal{K}$. Rewrite $\mathbb{F}_2^n$ as a direct sum of $\mathcal{K}' \oplus \mathcal{K}$ and let $\Phi : \mathbb{F}_2^{n-s} \to \mathcal{K}'$ be an $\mathbb{F}_2$-linear bijection from $\mathbb{F}_2^{n-s}$ to $\mathcal{K}'$.

Furthermore, given α, r and s, we can choose n to be sufficiently large such that

$$(\tau + r) - s \leqslant n - (\tau + r). \tag{6}$$

We state the following lemma and defer its technical proof to the end of this appendix.

Lemma 2. *Let $\alpha = \tau/n \leqslant 0.65$, $r = \lceil \log(2n+1) \rceil$ and $s = \lceil (1 - h((1-\alpha)/2))n \rceil$. Then there exists n_α such that* (6) *holds for all $n \geqslant n_\alpha$.*

As we before, we identify messages in $[2^{n-(\tau+r)}]$ with field elements in $\mathbb{F}_{2^{n-(\tau+r)}}$. Let $\boldsymbol{m} \in \mathbb{F}_{2^{n-(\tau+r)}}$ and $\boldsymbol{\beta} \in \mathbb{F}_{2^{(\tau+r)-s}}$. Lemma 2 then implies that $(\tau + r) - s \leqslant n - (\tau + r)$. Hence, we are able to pad $\boldsymbol{\beta}$ with zeroes and regard $\boldsymbol{\beta}$ as an element in $\mathbb{F}_{2^{n-(\tau+r)}}$. Set $\boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m}) = (\boldsymbol{\beta}\boldsymbol{m}, \boldsymbol{\beta}) \in \mathbb{F}_2^{n-s}$ and we define the codeset

$$\mathcal{C}_{\boldsymbol{m}} = \left\{\Phi(\boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m})) + \boldsymbol{k} : \boldsymbol{\beta} \in \mathbb{F}_{2^{(\tau+r)-s}} \setminus \{\mathbf{0}\}, \boldsymbol{k} \in \mathcal{K}\right\}.$$

We further set $\mathbb{C} = \left\{\mathcal{C}_{\boldsymbol{m}} : \boldsymbol{m} \in \mathbb{F}_{2^{n-(\tau+r)}}\right\}$. As before, we demonstrate that $\mathbb{C}$ is an (n, τ) cooling code by providing the encoding / decoding algorithms.

Encoding. Given $S \subseteq [n]$ with $|S| = \tau$, our encoding task is to find a nonzero $\boldsymbol{\beta} \in \mathbb{F}_{2^{(\tau+r)-s}}$ and $\boldsymbol{k} \in \mathcal{K}$ such that $\mathrm{supp}(\Phi(\boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m})) + \boldsymbol{k}) \cap S = \varnothing$. As in the previous subsection, we consider an $\mathbb{F}_2$-linear map $\phi : \mathbb{F}_{2^{(\tau+r)-s}} \times \mathcal{K} \to \mathbb{F}_2^{|S|}$. where $\phi(\boldsymbol{\beta}, \boldsymbol{k})$ is the projection of the binary word $\Phi(\boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m})) + \boldsymbol{k}$ onto the coordinate set S. Notice that ϕ is a linear map from a vector space of dimension $(\tau + r - s) + s = \tau + r$ to a subspace of dimension τ. Hence, the kernel of ϕ has dimension at least r.

We find a subspace $\mathcal{N}$ of this kernel with dimension r and consider the space $\mathrm{Im}(\mathcal{N}) = \{\Phi(\boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m})) + \boldsymbol{k} \in \mathbb{F}_2^n : (\boldsymbol{\beta}, \boldsymbol{k}) \in \mathcal{N}\}$. Since $r = \lceil \log(2n+1) \rceil$, the size of $\mathrm{Im}(\mathcal{N})$ is at least $2n+1$ and all words in $\mathrm{Im}(\mathcal{N})$ have zeroes at S. Shortening the words at these τ coordinates, we obtain a linear code of length $n - \tau$ and size at least $2n+1$. By Plotkin bound [27], we have that there exists a word $\boldsymbol{x} \in \mathrm{Im}(\mathcal{N})$ with weight at most $d-1$. In other words, $\boldsymbol{x} \notin \mathcal{K}$ and hence, $\boldsymbol{x} = \Phi(\boldsymbol{c}(\boldsymbol{\beta}, \boldsymbol{m})) + \boldsymbol{k}$ for some nonzero element $\boldsymbol{\beta}$.

The linear space $\mathcal{N}$ can be computed in $O(n^3)$ time. Since $|\mathcal{N}| = 2^r \leqslant 4n + 2$, finding a word $\boldsymbol{x}$ of weight at most $d-1$ can be done in $O(n)$ time. Hence, the encoding complexity is $O(n^3)$.

Decoding. Given some codeword $\boldsymbol{x} \in \mathbb{F}_2^n$, we write $\boldsymbol{x} = \boldsymbol{k}' + \boldsymbol{k}$ with $\boldsymbol{k}' \in \mathcal{K}'$ and $\boldsymbol{k} \in \mathcal{K}$. Applying Φ^{-1}, we obtain $\boldsymbol{c} \in \mathbb{F}_2^{n-s}$ and write $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$. To recover the message $\boldsymbol{m} \in \mathbb{F}_2^{n-(\tau+r)}$, we set $\boldsymbol{\beta} = \boldsymbol{c}_2$ and compute $\boldsymbol{m} = \boldsymbol{\beta}^{-1}\boldsymbol{c}_1$ in $O(n^3)$ time.

Finally, we provide the technical proof for Lemma 2.

Proof: [Proof of Lemma 2] First, observe that $(\tau + r) - s \leqslant n - (\tau + r)$ is equivalent to

$$1 + \frac{s}{n} - \frac{2\tau}{n} \geqslant \frac{2r}{n}. \tag{7}$$

Since $s = \lceil (1 - h((1-\alpha)/2))n \rceil$, we have $s/n \geqslant 1 - h((1-\alpha)/2))$. In other words,

$$1 + \frac{s}{n} - \frac{2\tau}{n} \geqslant 2 - h\left(\frac{1 - \tau/n}{2}\right) - 2(\tau/n) \triangleq h_\alpha. \tag{8}$$

Since $\alpha \leqslant 0.687$, we have that the value h_α is strictly positive.

On the other hand, $r = \lceil \log(2n+1) \rceil \leqslant 1 + \log(2n+1)$. Since $(2 + 2\log(2n+1))/n$ converges to zero, we can find n_α such that $(2 + 2\log(2n+1))/n \leqslant h_\alpha$ for $n \geqslant n_\alpha$. Hence,

$$\frac{2r}{n} \leqslant \frac{2 + 2\log(2n+1)}{n} \leqslant h_\alpha \text{ for } n \geqslant n_\alpha. \tag{9}$$

Therefore, (8) and (9) imply (7). □

C. When $\tau = \lfloor 0.59n \rfloor$ and Polynomial Time Construction

In the previous subsection, to construct an (n, τ) cooling code for $\tau \geqslant n/2$, we require a linear code $\mathcal{K}$ whose parameters satisfy the GV bound. Unfortunately, no efficient method to construct $\mathcal{K}$ is known. Hence, we borrow a construction by Vlăduţ *et al.* [33] that constructs high-rate linear codes in polynomial time.

Lemma 3 (Vlăduţ *et al.* [33, Corollary 2])**.** *Set $0 \leqslant \delta \leqslant 1/2$. Suppose that there exists an $[n_0, 2m, d]$ linear code with $d/n \geqslant \delta$. Then there is an infinite family of $[n, \lceil \alpha n \rceil, \lceil \delta n \rceil]$ linear codes with*

$$\alpha \geqslant \frac{2m(2^m - 2)}{n_0(2^m - 1)} - \frac{2m\delta}{d}.$$

For simplicity, we focus on the case where $\tau = \lfloor 0.59n \rfloor$ and our method can be generalized for $\tau \leqslant 0.59n$. When $\tau = \lfloor 0.59n \rfloor$, we apply Lemma 3 with an $[24, 12, 8]$ binary Golay code to construct the following family of linear codes.

Corollary 5. *Let $s = \lceil 0.184n \rceil$ and $d = \lceil 0.205n \rceil$. For infinite values of n, an $[n, s, d]$ linear code can be constructed in time polynomial in n.*

For the encoding / decoding method in the previous subsection to work, we require (6) or equivalently, (7) to hold. From our choice of τ and s, we have that

$$1 + \frac{s}{n} - \frac{2\tau}{n} \geqslant 0.004. \tag{10}$$

On the other hand,

$$\frac{2r}{n} \leqslant \frac{2 + 2\log(2n+1)}{n} \leqslant 0.004 \text{ for } n \geqslant 7430. \tag{11}$$

Therefore, (10) and (11) imply (7) and we apply the encoding / decoding algorithms in the previous subsection for our cooling codes.

REFERENCES

[1] T. Bu, "Partitions of a vector space," *Discrete Math.*, vol. 31, no. 1, pp. 79–83, 1980.

[2] D. Burshtein and A. Strugatski, "Polar write once memory codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 5088–5101, Aug. 2013.

[3] P. Cappelletti, C. Golla, P. Olivo, and E. Zanoni, *Flash Memories*. Boston, MA, USA: Kluwer, 1999.

[4] Y. Cassuto and E. Yaakobi, "Short q-ary fixed-rate WOM codes for guaranteed rewrites and with hot/cold write differentiation," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3942–3958, Jul. 2014.

[5] Y. M. Chee, T. Etzion, H. M. Kiah, and A. Vardy, "Cooling codes: Thermal-management coding for high-performance interconnects," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3062–3085, Apr. 2018.

[6] Y. M. Chee, H. M. Kiah, A. Vardy, and E. Yaakobi, "Explicit constructions of finite-length WOM codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Aachen, Germany, Jun. 2017, pp. 2870–2874.

[7] Y. M. Chee, H. M. Kiah, A. J. H. Vinck, V. V. Khu, and E. Yaakobi, "Coding for write ℓ-step-up memories," in *Proc. IEEE Int. Symp. Inf. Theory*, Apr. 2019, pp. 1597–1601.

[8] G. Cohen, P. Godlewski, and F. Merkx, "Linear binary code for write-once memories," *IEEE Trans. Inf. Theory*, vol. IT-32, no. 5, pp. 697–700, Sep. 1986.

[9] T. M. Cover, "Enumerative source encoding," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 1, pp. 73–77, Jan. 1973.

[10] B. Dobbelaere, *A Heuristic Approach to Find Short Efficient WOM Codes*, document Draft Rev. 1.1, 2017. [Online]. Available: http://users.telenet.be/bertdobbelaere/WOM/

[11] I. I. Dumer, "Asymptotically optimal codes correcting memory defects of fixed multiplicity," *Problems Peredachi Inform.*, vol. 25, no. 4, pp. 3–10, Oct. 1989.

[12] E. En Gad, W. Huang, Y. Li, and J. Bruck, "Rewriting flash memories by message passing," in *Proc. IEEE Int. Symp. Inf. Theory*, Hong Kong, Jun. 2015, pp. 646–650.

[13] T. Etzion, "Perfect byte-correcting codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 3140–3146, Nov. 1998.

[14] T. Etzion and A. Vardy, "Error-correcting codes in projective space," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1165–1173, Feb. 2011.

[15] F.-W. Fu and A. J. H. Vinck, "On the capacity of generalized write-once memory with state transitions described by an arbitrary directed acyclic graph," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 308–313, Jan. 1999.

[16] P. Godlewski, "WOM-codes construits à partir des codes de Hamming," *Discrete Math.*, vol. 65, no. 3, pp. 237–243, Jul. 1987.

[17] M. Grassl. *Bounds on the Minimum Distance of Linear Codes and Quantum Codes*. Accessed: Dec. 27, 2016. [Online]. Available: http://www.codetables.de

[18] C. Heegard, "On the capacity of permanent memory," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 1, pp. 34–42, Jan. 1985.

[19] J. Justesen, "Class of constructive asymptotically good algebraic codes," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 5, pp. 652–656, Sep. 1972.

[20] K. Santhosh, V. Avinash, N. Krishna, and P. Henry, "Spatially-coupled codes for write-once memories," in *Proc. 53rd Annu. Allerton Conf. Commun., Control Comput. (Allerton)*, Oct. 2015, pp. 125–131.

[21] S. Kurz, "Improved upper bounds for partial spreads," *Des., Codes Cryptogr.*, vol. 85, no. 1, pp. 97–106, Oct. 2017.

[22] F. Margaglia, G. Yadgar, E. Yaakobi, Y. Li, A. Schuster, and A. Brinkmann, "The devil is in the details: Implementing flash page reuse with WOM codes," in *Proc. USENIX FAST*, Santa Clara, CA, USA, Feb. 2016, pp. 1–16.

[23] J. L. Massey, "Threshold decoding," Res. Lab. Electron., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 410, 1963.

[24] F. Merkx, "Womcodes constructed with projective geometries," *Traitement Signal*, vol. 1, no. 2, pp. 227–231, 1984.

[25] E. L. Năstase and P. A. Sissokho, "The maximum size of a partial spread in a finite projective space," *J. Combinatorial Theory, A*, vol. 152, no. 1, pp. 353–362, Jul. 2017.

[26] S. Odeh and Y. Cassuto, "NAND flash architectures reducing write amplification through multi-write codes," in *Proc. 30th Symp. Mass Storage Syst. Technol. (MSST)*, May 2014, pp. 1–10.

[27] R. M. Roth, *Introduction to Coding Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[28] R. L. Rivest and A. Shamir, "How to reuse a 'write-once' memory," *Inf. Control*, vol. 55, nos. 1–3, pp. 1–19, Dec. 1982.

[29] M. Schwartz and T. Etzion, "Codes and anticodes in the Grassman graph," *J. Combinatorial Theory, A*, vol. 97, no. 1, pp. 27–42, 2002.

[30] A. Shpilka, "Capacity-achieving multiwrite WOM codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1481–1487, Mar. 2014.

[31] A. Shpilka, "New constructions of WOM codes using the Wozencraft ensemble," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4520–4529, Jul. 2013.

[32] S. Thomas, "Designs over finite fields," *Geometriae Dedicata*, vol. 21, no. 2, pp. 237–242, 1987.

[33] S. G. Vlăduţ, G. L. Katsman, and M. A. Tsfasman, "Modular curves and codes with polynomial construction complexity," *Problems Peredachi Inform.*, vol. 20, no. 1, pp. 47–55, 1984.

[34] J. K. Wolf, A. D. Wyner, J. Ziv, and J. Korner, "Coding for a write-once memory," *AT T Bell Labs. Tech. J.*, vol. 63, no. 6, pp. 1089–1112, Aug. 1984.

[35] Y. Wu, "Low complexity codes for writing a write-once memory twice," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, Jun. 2010, pp. 1928–1932.

[36] Y. Wu and A. Jiang, "Position modulation code for rewriting write-once memories," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3692–3697, Jun. 2011.

[37] E. Yaakobi, S. Kayser, P. H. Siegel, A. Vardy, and J. K. Wolf, "Codes for write-once memories," *IEEE Trans. Inf. Theory*, vol. 58, no. 9, pp. 5985–5999, Sep. 2012.

[38] E. Yaakobi and A. Shpilka, "High sum-rate three-write and nonbinary WOM codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 7006–7015, Nov. 2014.

[39] E. Yaakobi, A. Yucovich, G. Maor, and G. Yadgar, "When do WOM codes improve the erasure factor in flash memories?" in *Proc. IEEE Int. Symp. Inf. Theory*, Hong Kong, Jun. 2015, pp. 2091–2095.

[40] G. Yadgar, E. Yaakobi, and A. Schuster, "Write once, get 50% free: Saving SSD erase costs using WOM codes," in *Proc. USENIX FAST*, Santa Clara, CA, USA, Feb. 2015, pp. 257–271.

Yeow Meng Chee (SM'08) received the B.Math. degree in computer science and combinatorics and optimization and the M.Math. and Ph.D. degrees in computer science from the University of Waterloo, Waterloo, ON, Canada, in 1988, 1989, and 1996, respectively.

Currently, he is Associate Vice President (Innovation and Enterprise), and Professor in the Department of Industrial Systems Engineering and Management, at the National University of Singapore, Singapore. Prior to this, he was Program Director of Interactive Digital Media R&D in the Media Development Authority of Singapore, Postdoctoral Fellow at the University of Waterloo and IBMs Zürich Research Laboratory, General Manager of the Singapore Computer Emergency Response Team, Deputy Director of Strategic Programs at the Infocomm Development Authority, Singapore, and Professor at the Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore.

His research interest lies in the interplay between combinatorics and computer science/engineering, particularly combinatorial design theory, coding theory, extremal set systems, and electronic design automation.

Han Mao Kiah received his Ph.D. degree in mathematics from Nanyang Technological University (NTU), Singapore, in 2014. From 2014 to 2015, he was a Postdoctoral Research Associate at the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign. From 2015 to 2018, he was a Lecturer at the School of Physical and Mathematical Sciences (SPMS), NTU, Singapore. Currently, he is an Assistant Professor at SPMS, NTU, Singapore. His research interests include combinatorial design theory, coding theory, and enumerative combinatorics.

Alexander Vardy (S'88–M'91–SM'94–F'99) was born in Moscow, U.S.S.R., in 1963. He earned his B.Sc. *(summa cum laude)* from the Technion, Israel, in 1985, and Ph.D. from the Tel-Aviv University, Israel, in 1991. During 1985–1990 he was with the Israeli Air Force, where he worked on electronic counter measures systems and algorithms. During the years 1992 and 1993 he was a Visiting Scientist at the IBM Almaden Research Center, in San Jose, CA. From 1993 to 1998, he was with the University of Illinois at Urbana-Champaign, first as an Assistant Professor then as an Associate Professor. Since 1998, he has been with the University of California San Diego (UCSD), where he is the Jack Keil Wolf Endowed Chair Professor in the Department of Electrical and Computer Engineering and the Department of Computer Science. While on sabbatical from UCSD, he has held long-term visiting appointments with CNRS, France, the EPFL, Switzerland, the Technion, Israel, and Nanyang Technological University, Singapore.

His research interests include error-correcting codes, algebraic and iterative decoding algorithms, lattices and sphere packings, coding for storage systems and devices, cryptography and computational complexity theory, as well as fun math problems.

He received an IBM Invention Achievement Award in 1993, and NSF Research Initiation and CAREER awards in 1994 and 1995. In 1996, he was appointed Fellow in the Center for Advanced Study at the University of Illinois, and received the Xerox Award for faculty research. In the same year, he became a Fellow of the David and Lucile Packard Foundation. He received the IEEE Information Theory Society Paper Award (jointly with Ralf Koetter) for the year 2004. In 2005, he received the Fulbright Senior Scholar Fellowship, and the Best Paper Award at the IEEE Symposium on Foundations of Computer Science (FOCS). In 2017, his work on polar codes was recognized by the the IEEE Communications & Information Theory Societies Joint Paper Award. During 1995–1998, he was an Associate Editor for Coding Theory and during 1998–2001, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY. From 2003 to 2009, he was an Editor for the *SIAM Journal on Discrete Mathematics*. He is currently serving on the Executive Editorial Board for the IEEE TRANSACTIONS ON INFORMATION THEORY. He has been a member of the Board of Governors of the IEEE Information Theory Society during 1998–2006, and again during 2011–2017.

Eitan Yaakobi (S'07–M'12–SM'17) is an Associate Professor at the Computer Science Department at the Technion Israel Institute of Technology. He received the B.A. degrees in computer science and mathematics, and the M.Sc. degree in computer science from the Technion — Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. Between 2011-2013, he was a postdoctoral researcher in the department of Electrical Engineering at the California Institute of Technology. His research interests include information and coding theory with applications to non-volatile memories, associative memories, data storage and retrieval, and voting theory. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010-2011.