Sublinear-Time Non-Adaptive Group Testing With $O(k \log n)$ Tests via Bit-Mixing Coding

Steffen Bondorf, *Member, IEEE*, Binbin Chen, *Member, IEEE*, Jonathan Scarlett, *Member, IEEE*, Haifeng Yu, and Yuda Zhao

***Abstract*— The group testing problem consists of determining a small set of defective items from a larger set of items based on tests on groups of items, and is relevant in applications such as medical testing, communication protocols, pattern matching, and many more. While rigorous group testing algorithms have long been known with runtime at least linear in the number of items, a recent line of works has sought to reduce the runtime to $\mathrm{poly}(k \log n)$, where n is the number of items and k is the number of defectives. In this paper, we present such an algorithm for non-adaptive group testing termed *bit mixing coding* (BMC), which builds on techniques that encode item indices in the test matrix, while incorporating novel ideas based on erasure-correction coding. We show that BMC achieves asymptotically vanishing error probability with $O(k \log n)$ tests and $O(k^2 \cdot \log k \cdot \log n)$ runtime, in the limit as $n \to \infty$ (with k having an arbitrary dependence on n). This closes an open problem of simultaneously achieving $\mathrm{poly}(k \log n)$ decoding time using $O(k \log n)$ tests without any assumptions on k. In addition, we show that the same scaling laws can be attained in a commonly-considered noisy setting, in which each test outcome is flipped with constant probability.**

Manuscript received January 22, 2020; revised August 11, 2020; accepted December 5, 2020. Date of publication December 21, 2020; date of current version February 17, 2021. This work was supported in part by the Singapore Ministry of Education Academic Research Fund Tier-2 under Research Grant MOE2017-T2-2-031. The work of Binbin Chen was supported by the National Research Foundation, Prime Minister's Office, Singapore, through the Energy Programme administrated by the Energy Market Authority under EP Award NRF2017EWT-EP003-047 and through the Campus for Research Excellence and Technological Enterprise (CREATE) Programme. The work of Jonathan Scarlett was supported by the National University of Singapore (NUS) Early Career Research Award. A preliminary conference version of part of this work was presented and published in the International Conference on Information Processing in Sensor Networks (IPSN), Montreal, 2019 [1]. *(Corresponding author: Jonathan Scarlett.)*

Steffen Bondorf was with the Department of Computer Science, National University of Singapore, Singapore 117417. He is now with the Faculty of Mathematics, Ruhr University Bochum, 44801 Bochum, Germany (e-mail: steffen.bondorf@rub.de).

Binbin Chen is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore 487372 (e-mail: binbin_chen@sutd.edu.sg).

Jonathan Scarlett is with the Department of Computer Science, National University of Singapore, Singapore 117417, also with the Department of Mathematics, National University of Singapore, Singapore 119076, and also with the Institute of Data Science, National University of Singapore, Singapore 117602 (e-mail: scarlett@comp.nus.edu.sg).

Haifeng Yu is with the Department of Computer Science, National University of Singapore, Singapore 117417 (e-mail: haifeng@comp.nus.edu.sg).

Yuda Zhao was with the Department of Computer Science, National University of Singapore, Singapore 117417. He is now with Advance.AI, Singapore 068898 (e-mail: yudazhao@gmail.com).

Communicated by V. Sidorenko, Associate Editor for Coding Theory.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2020.3046113.

Digital Object Identifier 10.1109/TIT.2020.3046113

***Index Terms*— Group testing, sublinear-time decoding, sparsity.**

I. Introduction

THE group testing problem consists of determining a small subset of defective items within a larger set of items, based on tests performed on groups of items, and corresponding outcomes that indicate whether the group contains at least one defective item. This problem has a history in medical testing [2], and has regained significant attention following new applications in areas such as communication protocols [3], pattern matching [4], and database systems [5], and connections with compressive sensing [6], [7].

The design and analysis of group testing algorithms remains an active ongoing area of research; see [8], [9] for comprehensive surveys. In this paper, building on a recent line of works, we consider group testing algorithms with *low decoding time*, alongside the usual requirement of a low number of tests. Before outlining the related work and specific contributions in detail, we formally introduce the problem setup.

A. Problem Setup

The group testing problem consists of n items labeled $\{1, \dots, n\}$, a subset $\mathcal{K}$ of which are *defective*. We seek to identify $\mathcal{K}$ via a series of suitably-chosen tests. We initially focus on the noiseless setting (see Section V for the noisy setting), in which each test takes the form

$$Y = \bigvee_{j \in \mathcal{K}} X_j, \tag{1}$$

where $X = (X_1, \dots, X_n) \in \{0,1\}^n$ indicates which items are included in the test, and $Y \in \{0,1\}$ is the outcome.

We focus on *non-adaptive* test designs, in which all tests must be chosen prior to observing any outcomes. Accordingly, the tests $X^{(1)}, \dots, X^{(t)}$ are represented by a *test matrix* $\mathbf{X} \in \{0,1\}^{t \times n}$ whose i-th column is $X^{(i)} \in \{0,1\}^t$. The corresponding test outcomes are denoted by $\mathbf{Y} = (Y^{(1)}, \dots, Y^{(t)})$, with $Y^{(i)} \in \{0,1\}$ generated from $X^{(i)}$ according to the model (1). Given the tests and their outcomes, a *decoder* forms an estimate $\widehat{\mathcal{K}}$ of $\mathcal{K}$. We consider the exact recovery criterion, in which the error probability is given by

$$P_{\mathrm{e}} := \mathbb{P}[\widehat{\mathcal{K}} \neq \mathcal{K}]. \tag{2}$$

We assume that $|\mathcal{K}| \leq k$ for some k that is known to the group testing algorithm. That is, the algorithm knows an upper bound

TABLE I

OVERVIEW OF THE MOST RELEVANT EXISTING NON-ADAPTIVE GROUP TESTING RESULTS HAVING $\mathrm{poly}(k \log n)$ DECODING TIME, WITH n ITEMS, k DEFECTIVES, AND t TESTS, UNDER THE HIGH-PROBABILITY EXACT RECOVERY CRITERION

References	Number of tests t	Runtime	Construction
Lower Bound [18]	$\Omega\big(k \log \frac{n}{k}\big)$	-	-
GROTESQUE [28]	$O(k \cdot \log k \cdot \log n)$	$O(k \cdot \log k \cdot \log n)$	Randomized
SAFFRON [29]	$O(k \cdot \log k \cdot \log n)$	$O(k \cdot \log k \cdot \log n)$	Randomized
Inan *et al.* [11]	$O\big(k \cdot \log n \cdot \log \frac{\log n}{\log k}\big)$	$O\big(k^3 \cdot \log n \cdot \log \frac{\log n}{\log k}\big)$	Explicit
This paper	$O(k \log n)$	$O(k^2 \cdot \log k \cdot \log n)$	Randomized[1]

on $|\mathcal{K}|$ is known but not necessarily the exact value. This is a standard assumption in the literature, and is necessary to avoid scenarios such as $|\mathcal{K}| \geq \frac{n}{2}$ that require n tests [10]. Our analysis will hold for an arbitrary *fixed* defective set $\mathcal{K}$ with cardinality at most k, meaning that the probability in (2) is only with respect to our randomized test design $\mathbf{X}$.

Throughout the paper, we use the standard asymptotic notation $O(\cdot)$, $o(\cdot)$, $\Theta(\cdot)$, $\Omega(\cdot)$ and $\omega(\cdot)$.

B. Summary of Results

In this paper, we introduce a non-adaptive group testing procedure termed *bit mixing coding* (BMC) that attains asymptotically vanishing error probability as $n \to \infty$ with $O(k \log n)$ tests and $O(k^2 \cdot \log k \cdot \log n)$ decoding time. The $O(k \log n)$ number of tests is known to be order-optimal whenever $k \leq O(n^{1-\epsilon})$ for some $\epsilon > 0$. BMC is the first algorithm to achieve such optimal number of tests together with $\mathrm{poly}(k \log n)$ decoding time, resolving an open problem recently posed in [11]; more detailed comparisons are given below. We additionally show that BMC permits conditions for success that hold with high probability and can be verified in time $\mathrm{poly}(k \log n)$, and that BMC can easily be adapted to combat random noise in the test outcomes.

II. RELATED WORK

A. Existing Group Testing Results

Here we provide an overview of some existing works. We focus on those that are most related, in the sense of attaining high-probability recovery with non-adaptive testing and $\mathrm{poly}(k \log n)$ decoding time, while only providing a brief outline of other less closely-related settings.

A complementary goal to attaining small error probability (the *for-each guarantee*, or *probabilistic group testing*) is the *for all guarantee*, or *combinatorial group testing*: Find a test matrix that deterministically permits the recovery of *any* defective set of size k. While this is a significantly stronger guarantee, it comes at the expense of requiring $t = \Omega\big(\min\big\{k^2 \frac{\log n}{\log k}, n\big\}\big)$ tests, as opposed to $t = O(k \log n)$. Upper bounds of $O(k^2 \log n)$ for this setting were originally attained with $\Omega(n)$ decoding time [12], [13], and more recently with $\mathrm{poly}(k \log n)$ time [14]–[17].

In the probabilistic setting, the $\Omega\big(k \log \frac{n}{k}\big)$ lower bound [18] on the number t of tests indicates that the scaling $t = O(k \log n)$ is optimal whenever $k \leq O(n^{1-\epsilon})$ for some constant $\epsilon > 0$. Under a randomized test design and with $\Omega(n)$ decoding time, numerous results attaining asymptotically vanishing error probability with $t = O(k \log n)$ have been obtained, with various scaling laws and specific sparse regimes considered in early works [18], [19], and more precise characterizations of the regime $k \leq O(n^{1-\epsilon})$ given in a recent line of works [20]–[27].

The most relevant existing works to this paper are those attaining $\mathrm{poly}(k \log n)$ decoding time in the probabilistic setting (see Table I for a summary), particularly GROTESQUE [28], SAFFRON [30], and the study of the Kautz-Singleton construction (originally proposed in [31]) by Inan *et al.* [11]. GROTESQUE and SAFFRON attain $O(k \cdot \log k \cdot \log n)$ for both the number of tests and runtime using randomized designs, whereas [11] attains $t = O\big(k \cdot \log n \cdot \log \frac{\log n}{\log k}\big)$ with $O\big(k^3 \cdot \log n \cdot \log \frac{\log n}{\log k}\big)$ decoding time using an explicit design (i.e., $\mathbf{X} \in \{0,1\}^{t \times n}$ can be constructed deterministically in time polynomial in t and n). Variations of GROTESQUE and SAFFRON additionally permit $t = O(k \log n)$ under adaptive testing [28] and/or approximate recovery [29], but not in the non-adaptive setting with exact recovery, which is the main focus of existing works and the present paper.

Comparison with our results. As outlined above, our main contribution is to bring the number of tests down to the optimal $t = O(k \log n)$ scaling while maintaining efficient decoding time (namely, $O(k^2 \log k \cdot \log n)$).

Follow-up works. After the initial posting of this work, a distinct algorithmic approach was proposed in the follow-up works of [32] and [33] that attains $O(k \log n)$ scaling in both the number of tests and decoding time. This approach can be viewed as a randomized non-adaptive variant of binary splitting, which is a popular adaptive testing technique [34]. In addition, the $O(k \log n)$ scaling laws in [32], [33] apply even when the error probability scales as $O(n^{-c})$ for any fixed $c > 0$, whereas BMC's decoding time is inversely proportional to the error probability (see Section III-G).

Nevertheless, BMC retains some advantages over the algorithm proposed in [32], [33]. In Section V, we demonstrate that BMC is resilient to noise in the form of randomly flipped tests, whereas [32] only considers the noiseless setting, and [33] allows for false positive or false negative tests individually, but not both simultaneously. Another advantage is the availability of sufficient conditions for the success of BMC that can be verified in $\mathrm{poly}(k \log n)$ time. Finally,

[1]Despite being randomized, we also provide sufficient conditions that hold with high probability, that ensure success, and that can be verified in time $\mathrm{poly}(k \log n)$; see the discussion following Lemma 1.

in another follow-up work, it has been shown that BMC can be adapted to obtain similar guarantees for an *explicit* construction, with $O(k \log n)$ tests and $O(k^3 \log k + k \log n)$ decoding time [35].

B. Existing Group Testing Techniques

The above-outlined group testing algorithms with low decoding time are predominantly either based *code concatenation* [11], [14]–[16], or related techniques that *directly encode item indices* into the test matrix [17], [28], [29]. Our approach builds on the latter of these, but has some important differences, outlined below.

We briefly mention that such techniques appeared prior to these group testing works in the context of *compressive sensing* (CS) [7], [17], [36]–[39]. However, the two problems bear fundamental differences; for instance, unlike group testing, the CS problem permits uniform (for-all) recovery with $O(k \log n)$ tests. In addition, the linear nature of the CS model permits "subtracting off" previously-found values in the sparse vector, which is not possible in group testing. In light of these differences, transferring CS algorithms directly to group testing is typically not possible.

To facilitate a comparison with our own techniques, we describe the "singleton-only" version of the SAFFRON algorithm [29], but its limitations preventing $t = O(k \log n)$ scaling equally apply to the more sophisticated version of SAFFRON, as well as the closely-related GROTESQUE algorithm [28]. Briefly, singleton-only SAFFRON works as follows: One forms $O(k \log k)$ "bundles" of tests of size $2 \log_2 \ n$ each, and assigns each item to any given bundle with probability $O\big(\frac{1}{k}\big)$. If a given item is assigned to a given bundle, then its $(\log_2 \ n)$-bit description is encoded into the first $\log_2 \ n$ tests (i.e., the item is included in the test if and only if its bit description contains a 1 at the corresponding location), and its bit-wise complement is encoded into the last $\log_2 \ n$ tests. The following decoding procedure ensures the identification of any defective item for which there exists a bundle in which it is included without any other defective items:

- For each bundle, check whether the first $\log_2 \ n$ test outcomes equal the bit-wise negation of the next $\log_2 \ n$.
- If so, add the item with bit representation given by the first $\log_2 \ n$ outcomes into the defective set estimate.

Due to the first step, the second step will never erroneously be performed on a bundle without defective items, nor on a bundle with multiple defective items.

To frame SAFFRON using similar terminology to our approach, we envision the assignments of n items to $B = O(k \log k)$ bundles as forming a $n \times B$ matrix with rows $\mathbf{c}_1, \ldots, \mathbf{c}_n$. The final (transpose of the) test matrix is obtained by expanding each entry into a length-$(2 \log_2 \ n)$ vector as described above.

Our strategy, BMC, is outlined in Section III-A, but we can already highlight some of the key differences:

- While the randomly-generated strings $\mathbf{c}_1, \ldots, \mathbf{c}_n$ are mutually independent for SAFFRON (and GROTESQUE) described above, this is far from being true in BMC. Instead, we independently generate a *much smaller* number of strings, and then let each $\mathbf{c}_i$ equal one of these strings selected uniformly at random. Hence, there are a large number of *repeated strings*; we only seek to ensure that there are no repetitions *among the defectives*.
- The first step of our decoding algorithm is to *identify the strings associated with defectives*, whereas in SAFFRON the goal is to *identify the bundles corresponding to isolated defectives* (i.e., being alone in the bundle). These are distinct goals, and are solved using different techniques: In contrast with the above-outlined bit-negation approach (or an analogous *location test* in GROTESQUE), we achieve our goal by performing a simple one-by-one check on the small set of strings mentioned in the previous dot point.
- We not only seek for each defective item $i \in \mathcal{K}$ to have a single isolated index in $\mathbf{c}_i$, but rather, $O(\log n)$ of them. This may sound like a more restrictive condition that potentially *increases* the number of tests, but it is made up for by the following crucial observation: We do *not* blow up the number of tests by a factor of $O(\log n)$ in order to ensure $O(\log n)$ "collision-free" tests for each defective item. Instead, we treat any collisions as erasures, and control for them using erasure-correcting coding.[2] Hence, instead of seeking $O(\log n)$ specific collision-free tests, we allow the defectives to *share the damage of collisions* in a controlled manner.

To our knowledge, these unique aspects of BMC have not appeared in any existing works, including those on CS. BMC is also significantly different from the list-decoding approach [16], [40], [41], which first identifies a superset of $\mathcal{K}$ of size $O(k)$ (or similar) and then resolves the false positives. In fact, our analysis will reveal that after the first step of BMC, there are still $O\big(\frac{n}{k \log k}\big)$ items that could potentially be defective, which typically far exceeds $O(k)$.

III. Bit Mixing Coding: Test Design and Decoding

In this section, we provide the details of BMC, as well as formally stating its theoretical guarantees.

A. Overview of Bit Mixing Coding

Given integers t_1, t_2, and w, the testing is done in two (non-adaptive) batches, described below A rough illustration of these batches is shown in Figure 1. Subsequently, the function $\log(\cdot)$ has base e.

In the first batch, each item is assigned a binary string of length t_1 and weight w, chosen uniformly at random with replacement from a carefully designed set $\mathcal{S} \subseteq \{0,1\}^{t_1}$.[3] We refer to these strings as *masking strings* (see Section III-B). The number of strings in $\mathcal{S}$ is typically much smaller than

[2] GROTESQUE [28] employs expander codes to combat *random noise*, but this is a distinct notion to our idea of using erasure-correction to combat collisions, and the former technique does not transfer readily to the latter.

[3] Since we seek the high-probability recovery of any fixed defective set $\mathcal{K}$, and $\mathbf{X}$ is deterministic given the masking string allocations, randomized allocation is required to avoid falling under the uniform (for-all) recovery guarantee requiring $t = \Omega\big(\min\big\{k^2 \frac{\log n}{\log k}, n\big\}\big)$ tests [12], in contrast with our goal of $t = O(k \log n)$.

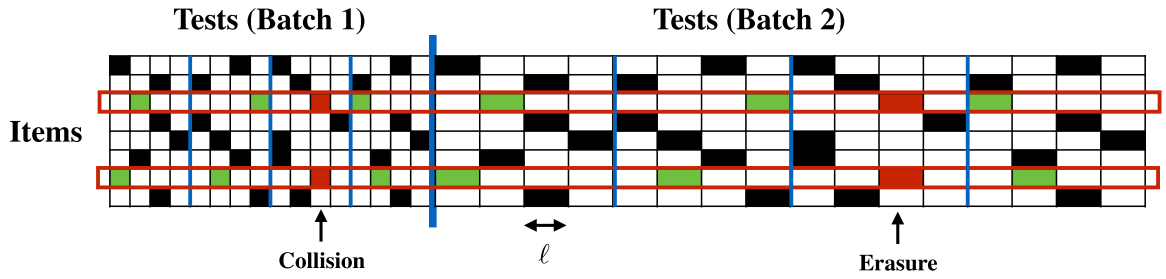

Fig. 1. Illustration of the BMC-based (transpose of the) group testing matrix, with $n = 8$ items, $k = 2$ defectives (the two highlighted rows), and weight $w = 4$. For compactness, each masking string is shown with length $t_1 = 16$ even though the choice in our analysis would correspond to $t_1 = 4kw = 32$.

the number of items, implying that a given item's string is unlikely to be unique. However, we do seek uniqueness *among the defective items*. The testing sub-matrix $\mathbf{X}_1 \in \{0,1\}^{t_1 \times n}$ simply arranges the items' strings in columns (or rows in Figure 1, which shows $\mathbf{X}^T$). Given the resulting t_1 test outcomes, the decoder searches through the strings in $\mathcal{S}$ and seeks to determine which ones were assigned to *some* defective item, without attempting to identify the index of that item.

In the second batch, the testing sub-matrix $\mathbf{X}_2 \in \{0,1\}^{t_2 \times n}$ has a similar structure to $\mathbf{X}_1$, but with each bit replaced by a constant number ℓ of bits; hence, $t_2 = \ell t_1$. Any entry that was zero in $\mathbf{X}_1$ is simply replaced by a string of ℓ zeros. On the other hand, for any given column, each of the w entries equal to one is replaced by the binary description of a symbol from a codeword. Specifically, each item has a *unique* codeword of length w on an alphabet $\mathcal{A}$ of size 2^ℓ, and that codeword is an erasure-coded representation of the item's index.

The idea of the decoding procedure is to identify which k (or fewer) masking strings were assigned to defective items in the first batch, and to learn the defective item indices from the corresponding codewords in the second batch. While some parts of the codeword may be lost due to collisions, the erasure coding controls for this.

In the following, we focus on the case that $k \to \infty$ as $n \to \infty$. The case $k = O(1)$ is in fact much simpler, but also more convenient to handle separately, so it is deferred to Appendix B.

B. Masking Strings and Low Collision Sets

A key technical challenge in our analysis is proving the existence of the set $\mathcal{S} \subseteq \{0,1\}^{t_1}$ satisfying the properties overviewed in Section III-A. We proceed by presenting the relevant definitions and results for achieving this goal.

We begin with the formal definition of a masking string. This definition depends on the maximum number of defectives k and a length parameter w, and leads to a number of tests in the first batch given by $t_1 = 4kw$.

Definition 1: *We say that* $\mathbf{s} \in \{0,1\}^{t_1}$ *is a* (k, w) masking string *if it is the concatenation of* w *(typically different) binary substrings of length* $4k$*, with each substring having a Hamming weight of* 1*, for a total weight of* w*.*

Our group testing design will rely crucially on a subset $\mathcal{S}$ of masking strings that are sufficiently "well-separated on average", as formalized in the following definition.

Definition 2: *A set* $\mathcal{S} \subseteq \{0,1\}^{t_1}$ *of* (k, w) *masking strings is a* (k, w, δ) low collision set *(LCS) if it satisfies the following for any given integer* $k' \le k$ *and any given index* $i \in \{1, \dots, k'\}$*: If we choose* k' *strings* $\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}$ *from* $\mathcal{S}$ *uniformly at random with replacement, then the following conditions hold with probability at least* $1 - \delta$*:*

1. *The multi-set* $\widetilde{\mathcal{S}} = \{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$ *is such that all* $\tilde{\mathbf{s}} \in \mathcal{S} \setminus \widetilde{\mathcal{S}}$ *satisfy* $\sum_{j=1}^{k'} \tilde{\mathbf{s}}^T \tilde{\mathbf{s}}_j \le \frac{w}{2}$*;*
2. *The multi-set* $\widetilde{\mathcal{S}}^{(-i)} = \{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{i-1}, \tilde{\mathbf{s}}_{i+1}, \dots, \tilde{\mathbf{s}}_{k'}\}$ *is such that* $\sum_{1 \le j \le k' \,:\, j \ne i} \tilde{\mathbf{s}}_i^T \tilde{\mathbf{s}}_j \le \frac{w}{2}$*.*

The bulk of our technical analysis is devoted to proving the following lemma, establishing the existence of an LCS with certain requirements on the size $|\mathcal{S}|$ and parameters (k, w, δ). To simplify the analysis, we state the result in an asymptotic form, but non-asymptotic variants can be deduced from the proof. In addition, we make no effort to optimize the constant factors, which could be improved by refining our analysis.

Lemma 1: *Consider any sequence of* (k, δ) *pairs such that* $k \to \infty$*,* $\delta \to 0$*, and* $\delta \ge \frac{1}{k^2}$*. If* w *satisfies*

$$w \ge 70 \log \frac{k}{\delta}, \tag{3}$$

then for sufficiently large k *there exists a* (k, w, δ) *low collision set (LCS)* $\mathcal{S}$ *with cardinality* $|\mathcal{S}| = \frac{2k}{\delta}$*.*

Proof: See Section IV. □

While the construction used to prove Lemma 1 is randomized, the proof provides sufficient conditions for being an LCS that hold with high probability, and that can be verified in time $O\big(|\mathcal{S}|^2 \cdot w\big)$. We will later set $w = O(\log n)$ and $\delta = \frac{1}{k \log k}$, in which case substituting $|\mathcal{S}| = \frac{2k}{\delta}$ gives verification time $O(k^4 (\log k)^2 \log n) = \mathrm{poly}(k \log n)$. In contrast, given a set $\mathcal{S}$ of masking strings, it appears difficult to efficiently verify directly whether $\mathcal{S}$ is an LCS.

C. Encoding and Decoding: First Batch of Tests

The test design and decoding procedure associated with the first batch of tests are depicted in Algorithm 1. Masking strings are assigned uniformly at random from $\mathcal{S}$ with replacement and arranged to form $\mathbf{X}_1$, and the decoder operates by finding a list of masking strings whose support is a subset of that of the test outcome vector.

The following lemma provides a formal statement of successful masking string identification.

Algorithm 1 Test design (encoding) and masking string identification (decoding) for the first batch of tests.

Global parameters: Number of items n, triplet (k, w, δ), low collision set $\mathcal{S}$ of size $\frac{2k}{\delta}$

Test design

foreach $i = 1, \dots, n$ **do**
  Let the i-th column of $\mathbf{X}_1$ be a uniformly random element from the set $\mathcal{S}$
endfor

Masking string identification (**input:** A received binary string $\mathbf{y}_1$ of $4kw$ bits; **output:** A list $\mathcal{L}$ of masking strings)

foreach $\mathbf{s} \in \mathcal{S}$ **do**
  if $\mathbf{s}^T \mathbf{y}_1 = w$ **then** include $\mathbf{s}$ in the output list $\mathcal{L}$
endfor

Lemma 2: *Suppose that there are $k' \leq k$ defective items, and their associated masking strings $\{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$ satisfy the first condition of Definition 2. Then, the test design and masking string identification procedure in Algorithm 1 lead to an estimate $\mathcal{L}$ containing $\{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$ and no other elements of $\mathcal{S}$.*

Proof: It is trivial that any masking string $\tilde{\mathbf{s}}_i$ assigned to a defective item will be included in $\mathcal{L}$: Any index where its masking string is 1 will lead to a positive test, yielding $\tilde{\mathbf{s}}_i^T \mathbf{y}_1 = w$ since the weight of each masking string is w.

On the other hand, if $\tilde{\mathbf{s}} \in \mathcal{S}$ is not assigned to any defective item, then the first property of Definition 2 ensures that the sum of overlaps between $\tilde{\mathbf{s}}$ and the elements of $\{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$ is at most $\frac{w}{2}$. Since $\mathbf{y}_1$ is the bit-wise "OR" of $\{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$, this implies that $\tilde{\mathbf{s}}^T \mathbf{y}_1 \leq \frac{w}{2}$. □

D. Encoding and Decoding: Second Batch of Tests

The test design and decoding procedure associated with the second batch of tests are depicted in Algorithm 2. As discussed in Section III-A, the idea is to copy the structure of $\mathbf{X}_1$, but replace each bit by a sequence of ℓ bits. Any "0" bit is trivially mapped to a string of ℓ zeros, whereas any "1" bit is replaced by the binary representation of a codeword symbol. The codeword has length w and alphabet $\mathcal{A}$, whose size is $|\mathcal{A}| = 2^\ell$, and the corresponding codebook $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_n\}$ is chosen to have good worst-case erasure correction guarantees (see Section III-E).

For item identification, any collisions between masking strings in $\mathcal{L}$ are treated as erasures, whereas in the absence of a collision, the corresponding length-ℓ binary string from the test outcome vector $\mathbf{y}_2$ is mapped to a symbol from $\mathcal{A}$. For each $\mathbf{s} \in \mathcal{L}$, if there are sufficiently few erasures, then we can recover the corresponding codeword $\mathbf{c}_i$ via erasure-correcting decoding, and hence identify the defective item index $i \in \{1, \dots, n\}$.

The following lemma states the requirements on $\mathcal{C}$, along with sufficient conditions for successful decoding.

Lemma 3: *Suppose that there are $k' \leq k$ defective items, and their associated masking strings $\{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$ satisfy the second condition of Definition 2. If the first batch successfully produces $\mathcal{L} = \{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$, and the decoder of $\mathcal{C}$ is able to correct an arbitrary pattern of $\frac{w}{2}$ erasures, then the test design and item identification procedure in Algorithm 2 lead to successful recovery, i.e., $\widehat{\mathcal{K}} = \mathcal{K}$.*

Algorithm 2 Test design (encoding) and item identification (decoding) for the second batch of tests.

Global parameters: Number of items n, triplet (k, w, δ), multi-set $\{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ of masking strings assigned to items in first batch, parameter ℓ and alphabet $\mathcal{A}$ of size 2^ℓ, codebook $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_n\}$ with n codewords in $\mathcal{A}^w$.

Test design

foreach $i = 1, \dots, n$ **do**
  Initialize $\mathbf{x}$ to be the empty string
  Let $\mathbf{s} = (s_1, \dots, s_{4kw})$ be the masking string assigned to item i in the first batch
  foreach $j = 1, \dots, 4kw$ **do**
    if $s_j = 0$ **then** append ℓ zeros to $\mathbf{x}$
    else Append length-ℓ binary representation of the next symbol of $\mathbf{c}_i$ to $\mathbf{x}$
  endfor
  Fill in the i-th column of $\mathbf{X}_2$ with the entries of $\mathbf{x}$
endfor

Item identification (**input:** Received string $\mathbf{y}_2$ of length $4kw\ell$, list $\mathcal{L}$ of decoded masking strings returned by Algorithm 1; **output:** Estimate $\widehat{\mathcal{K}}$ of the defective set)

Construct $\tilde{\mathbf{y}} \in \mathcal{A}^{4kw}$ by converting $\mathbf{y}_2 \in \{0,1\}^{4kw\ell}$ from binary to the alphabet $\mathcal{A}$
foreach $\mathbf{s} \in \mathcal{L}$ **do**
  Initialize $\mathbf{u}$ to be the empty string
  foreach $i = 1, \dots, 4kw$ **do**
    if $(s_i = 1)$ and (there exists no $\tilde{\mathbf{s}} \in \mathcal{L}$ such that $\tilde{\mathbf{s}} \neq \mathbf{s}$ and $\tilde{s}_i = 1$) **then**
      Append the i-th symbol of $\tilde{\mathbf{y}}$ to $\mathbf{u}$;
    else if $(s_i = 1)$ **then**
      Append the erasure symbol to $\mathbf{u}$;
  endfor
  $\mathcal{I} \leftarrow$ decoder for $\mathcal{C}$ applied to $\mathbf{u}$ to return an index in $\{1, \dots, n\}$
  Include $\mathcal{I}$ in the output set $\widehat{\mathcal{K}}$
endfor

Proof: The second condition of Definition 2 implies that $\mathcal{L} = \{\tilde{\mathbf{s}}_1, \dots, \tilde{\mathbf{s}}_{k'}\}$ contains no duplicates, and also that for any such $\tilde{\mathbf{s}}_i \in \mathcal{L}$, at most $\frac{w}{2}$ of the indices of 1's collide with those of *any* of the other strings in $\mathcal{L}$. Hence, when $\tilde{\mathbf{s}}_i$ is processed in the outer loop of item identification in Algorithm 2, we have the following:

- Whenever there is a collision, an erasure symbol is added to $\mathbf{u}$, and this occurs at most $\frac{w}{2}$ times;
- Whenever there is no collision, the correct codeword symbol from $\mathbf{c}_i$ is added to $\mathbf{u}$.

Hence, $\mathbf{u}$ equals the desired length-w codeword $\mathbf{c}_i$ with at most $\frac{w}{2}$ entries replaced by the erasure symbol, and by our assumption on the decoder of $\mathcal{C}$, the correct codeword $\mathbf{c}_i$ (or equivalently, the correct index i) is identified. □

E. Choice of Erasure-Correcting Code

In principle, any erasure-correcting code could be utilized in Algorithm 2. For the purpose of proving our main theoretical result, we find the following code construction from [42] to

be convenient, providing near-MDS erasure correction with a bounded alphabet size.[4]

Lemma 4: (Corollary of [42, Thm. 1]) *For any $r \in (0,1)$ and arbitrarily small $\epsilon > 0$, there exists an alphabet $\mathcal{A}$ whose size is a constant depending only on ϵ, and a codebook $\mathcal{C}$ (with codeword symbols on $\mathcal{A}$) and associated encoder/decoder pair, such that the following properties hold*:

- *$\mathcal{C}$ has rate r, i.e., the number of codewords is $|\mathcal{A}|^{wr}$, where w is the block length;*
- *The decoder corrects any (worst-case) fraction $1-r-\epsilon$ of erasures;*
- *The encoding and decoding time are linear in the block length.*

In our analysis, we will take $r = \frac{1}{3}$ and $\epsilon = \frac{1}{6}$, so that a fraction $\frac{1}{2}$ of erasures is tolerated.

F. Statement of Main Result

We are now ready to state our main theorem. For simplicity, we set the relevant parameters to ensure $P_{\mathrm{e}} \leq \frac{1}{\log k}$, but with simple modifications to the constant factors (here and in the auxiliary results), we can improve this to $P_{\mathrm{e}} \leq \frac{1}{k^c}$ for any fixed $c > 0$, albeit with a $\frac{1}{P_{\mathrm{e}}}$ dependence of the runtime on P_{e}. We also re-iterate that we have made no effort to optimize constant factors, and we recall that the case $k = O(1)$ is handled in Appendix B.

Theorem 1: *Under the choices $w = \max\left\{\frac{3}{\ell}\log_2 n, 70\log\frac{k}{\delta}\right\}$ and $\delta = \frac{1}{k\log k}$, and a code $\mathcal{C}$ chosen suitably according to Lemma 4, the BMC group testing procedure described in Algorithms 1 and 2 with an LCS constructed according to Lemma 1 yields $P_{\mathrm{e}} \leq \frac{1}{\log k}$ for any $k \to \infty$, and the resulting number of tests used satisfies*

$$t \leq 12k \max\left\{\frac{\ell+1}{\ell}\log_2 n,\ 50\,(\ell+1)\log k\right\}(1+o(1)), \quad (4)$$

where $\ell \geq 2$ is a constant (not depending on n or k) corresponding to $\log_2|\mathcal{A}|$ in Lemma 4. In addition, with probability at least $1-\frac{1}{\log k}$, the decoding time is $O(k^2 \cdot \log k \cdot \log n)$.

Proof: By Lemma 1, there exists an LCS with $\delta = \frac{1}{k\log k}$ as long as $w \geq 70\log\frac{k}{\delta}$. The choice of w in the theorem statement ensures that this condition is true. Then, since the high-probability event in Definition 2 holds with probability at least $1-\delta$ for each defective item i, it holds simultaneously for all defective items with probability at least $1-k\delta = 1-\frac{1}{\log k}$. Under this high-probability event, assuming the codebook $\mathcal{C}$ in Algorithm 2 corrects $\frac{w}{2}$ worst-case erasures, we deduce from Lemmas 2 and 3 that the final estimate of the defective set is indeed correct.

It remains to choose the parameters to ensure that $w \geq 70\log\frac{k}{\delta}$, and to characterize the total number of tests and runtime. Suppose that, as stated following Lemma 4, we use a code of rate $\frac{1}{3}$. Since identifying an item requires $\frac{1}{\ell}\log_2 n$ symbols from $\mathcal{A}$ (with alphabet size 2^ℓ), a rate-$\frac{1}{3}$ code yields $w = \frac{3}{\ell}\log_2 n$.[5] This is consistent with the condition $w \geq 70\log\frac{k}{\delta}$ whenever $k \leq \delta n^{\frac{3}{70\ell\cdot\log 2}}$. Alternatively, if this condition on k fails to hold, we can simply replace the rate-$\frac{1}{3}$ code by a (potentially much) lower rate code such that $w = 70\log\frac{k}{\delta}$; by Lemma 4, such a code still exists with the required erasure-correcting properties. Combining these two cases, we obtain

$$w = \max\left\{\frac{3}{\ell}\log_2 n, 70\log\frac{k}{\delta}\right\}. \quad (5)$$

The number of tests is equal to $t_1 = 4kw$ in the first batch, and $t_2 = 4kw\ell$ in the second batch, for a total of

$$\begin{aligned} t &= 4kw(\ell+1) \\ &= 4k\max\left\{\frac{3(\ell+1)}{\ell}\log_2 n, 70(\ell+1)\log\frac{k}{\delta}\right\}. \end{aligned} \quad (6)$$

Substituting $\delta = \frac{1}{k\log k}$, taking a factor of 3 out the front, and writing $\frac{2\times 70}{3} \leq 50$, we obtain (4).

Decoding time. For decoding in Algorithm 1, we need to compute an inner product between $\mathbf{y}_1$ and every $\mathbf{s} \in \mathcal{S}$. To do so, we use the w positions of the "1" bits in $\mathbf{s}$ to index the required entries of $\mathbf{y}_1$. This leads to $O(w) = O(\log n)$ complexity for each $\mathbf{s}$, or $O(|\mathcal{S}|\cdot\log n) = O(k^2\cdot\log k\cdot\log n)$ for all $\mathbf{s} \in \mathcal{S}$ (since $|\mathcal{S}| = \frac{2k}{\delta}$). The decoding in Algorithm 2 has a total of $|\mathcal{L}|$ iterations. In each iteration, it constructs a sequence $\mathbf{u}$ while incurring $O(w|\mathcal{L}|) = O(|\mathcal{L}|\log n)$ complexity,[6] and then invokes decoding on $\mathbf{u}$, whose time is linear in the length $w = O(\log n)$ (see Lemma 4). Hence, the total decoding time for Algorithm 2 is $O(|\mathcal{L}|^2\log n)$. By Lemma 2 and our choice of δ, we know that with probability at least $1-\frac{1}{\log k}$, we have $|\mathcal{L}| \leq k$. It is then easy to see that the decoding time is dominated by Algorithm 1, and the overall complexity is $O(k^2\cdot\log k\cdot\log n)$. □

G. Limitations of BMC

The most immediate limitation of BMC is that it has a higher decoding time than the fastest existing algorithms (GROTEQUE [28] and SAFFRON [29]) by a factor of k. Hence, it remains an open problem as to whether one can further reduce the decoding time while still maintaining $t = O(k\log n)$. Another important limitation is the dependence on the error probability. We focused on the goal of attaining asymptotically vanishing error probability, and accordingly only targeted $P_{\mathrm{e}} \leq \frac{1}{\log k}$ in our main result with $k \to \infty$. However, in finite-size systems, the speed of convergence to zero can be important, and faster convergence such as $P_{\mathrm{e}} \leq k^{-\tau}$ is generally preferable. While our algorithm and analysis can be adapted to achieve this, the decoding time increases to $k^{2+\tau}\log n$. Finally, we re-iterate that the constants factors in Theorem 1 are fairly high, since our focus is on the scaling laws.

[4] As pointed out by a reviewer, the alphabet size has a $\frac{1}{\epsilon^4}$ dependence on the parameter ϵ appearing in Lemma 4, leading to large implied constants. However, we note that this code is only being used for its convenience in an analysis that does not optimize constants; in practice, one could instead utilize a more practical erasure-correcting code.

[5] Here and subsequently, we ignore rounding issues, as these do not impact the final result.

[6] The loop from $i = 1, \ldots, 4kw$ need not be done explicitly; instead, this can be thought of as a loop over w locations of 1's.

IV. Proof of Lemma 1 (Finding a Low Collision Set)

Algorithm 1 takes as input an LCS, whose properties play a crucial role in proving our main result, Theorem 1. In this section, we prove the existence of an LCS under suitable parameters, as stated in Lemma 1. Specifically, we show that if we construct a multi-set in a certain randomized way, then with probability close to 1, this multi-set will satisfy some *sufficient conditions* for being an LCS. In addition, these sufficient conditions will be verifiable in $\mathrm{poly}(k \log n)$ time, which is beneficial from a practical point of view. We emphasize that the LCS is constructed "offline" prior to forming the test matrix, and needs to be done only once.

A. A Random Construction

We will analyze a randomized construction of masking strings (see Definition 1). To construct a *single* masking string of length $t_1 = 4kw$, for each $4k$-bit segment of the string, we set a uniformly random bit in the segment to be "1" and all remaining bits to be "0". To construct a multi-set $\mathcal{S}$ containing $|\mathcal{S}| = \frac{2k}{\delta}$ random masking strings, we simply repeat this procedure independently $\frac{2k}{\delta}$ times. This means that $\mathcal{S}$ may contain duplicates; however, we will later prove that with high probability, there are no duplicates, so that $\mathcal{S}$ is a set.

B. Overview of the Proof

To decouple the randomness in the masking string construction from the randomness in Definition 2 itself, we introduce the concept of a *promising set* (see Section IV-C). In contrast with LCS, the definition of a promising set does not contain any probability terms. We will prove the following: (i) With probability approaching one, the multi-set returned by the random construction in Section IV-A is a promising set (see Lemma 6 below); (ii) A promising set must be an LCS (see Lemma 7 below). We will prove these claims for $w \geq 70 \log \frac{k}{\delta}$, $k \to \infty$, $\delta \to 0$, and $\delta \geq \frac{1}{k^2}$, as stated in Lemma 1. In addition to the assumption $w \geq 70 \log \frac{k}{\delta}$, we can further restrict our attention to $w = C \log \frac{k}{\delta}$ for some constant $C = \Theta(1)$ with $C \geq 70$. Once this is established, we can easily get an LCS for larger w values (e.g., $C \to \infty$ as $k \to \infty$) by repeating each masking string; this is formally stated as follows.

Lemma 5: *Given any (k, w, δ) low collision set $\mathcal{S}$ and any positive integer c, we can construct a (k, cw, δ) low collision set $\mathcal{S}^c$.*

Proof: For compactness, throughout this proof we use the terminology that a (k, w) masking string $\tilde{\mathbf{s}}$ is *w-compatible* with a multi-set $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ if $\sum_{i=1}^{m} \tilde{\mathbf{s}}^T \mathbf{s}_i \leq \frac{w}{2}$. Let $\mathcal{S}^c = \{\mathbf{s}^c \mid \mathbf{s} \in \mathcal{S}\}$, where $\mathbf{s}^c$ denotes the concatenation of c copies of $\mathbf{s}$. For all $\mathbf{s}$ and $\tilde{\mathbf{s}}$, we trivially have $(\mathbf{s}^c)^T \tilde{\mathbf{s}}^c = c \times (\mathbf{s}^T \tilde{\mathbf{s}})$. Fix an integer m and a multi-set $\{\tilde{\mathbf{s}}_1^c, \ldots, \tilde{\mathbf{s}}_m^c\} \subseteq \mathcal{S}^c$. It is easy to verify that: (i) $\widetilde{\mathcal{S}}^c = \{\tilde{\mathbf{s}}_1^c, \ldots, \tilde{\mathbf{s}}_m^c\}$ is (cw)-compatible with all $\mathbf{s}^c \in \mathcal{S}^c \setminus \widetilde{\mathcal{S}}^c$ if and only if $\widetilde{\mathcal{S}} = \{\tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_m\}$ is w-compatible with all $\mathbf{s} \in \mathcal{S} \setminus \widetilde{\mathcal{S}}$, and (ii) for all $i = 1, \ldots, m$, $(\widetilde{\mathcal{S}}^{(-i)})^c = \{\tilde{\mathbf{s}}_1^c, \ldots, \tilde{\mathbf{s}}_{i-1}^c, \tilde{\mathbf{s}}_{i+1}^c, \ldots, \tilde{\mathbf{s}}_m^c\}$ is (cw)-compatible with $\tilde{\mathbf{s}}_i^c$ if and only if $\widetilde{\mathcal{S}}^{(-i)} = \{\tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_{i-1}, \tilde{\mathbf{s}}_{i+1}, \ldots, \tilde{\mathbf{s}}_m\}$ is w-compatible with $\tilde{\mathbf{s}}_i$. From Definition 2, we deduce that since $\mathcal{S}$ is a (k, w, δ) LCS, $\mathcal{S}^c$ is a (k, cw, δ) LCS. □

Hence, we proceed by assuming that $w = C \log \frac{k}{\delta}$ with $C = \Theta(1)$ and $C \geq 70$. In particular, we will use the fact that $\frac{w}{k} \to 0$, obtained by combining this assumption with $k \to \infty$ and $\delta \geq \frac{1}{k^2}$.

C. The Concept of a Promising Set

Given a set $\mathcal{S}$ of masking strings and any $\tilde{\mathbf{s}} \in \mathcal{S}$, we define

$$\mu(\tilde{\mathbf{s}}, \mathcal{S}) = \frac{1}{|\mathcal{S}| - 1} \sum_{\mathbf{s} \in \mathcal{S} \setminus \{\tilde{\mathbf{s}}\}} \tilde{\mathbf{s}}^T \mathbf{s}. \tag{7}$$

In the following, we define the concept of a *promising set*, as a stepping stone to proving the existence of an LCS.

Definition 3: *A set $\mathcal{S}$ of (k, w) masking strings is a (k, w, δ)* promising set *if the following holds for all $\tilde{\mathbf{s}} \in \mathcal{S}$:*

$$\left| \mu(\tilde{\mathbf{s}}, \mathcal{S}) - \frac{w}{4k} \right| \leq \frac{0.04w}{4k} \tag{8}$$

$$\max_{\mathbf{s} \in \mathcal{S} \setminus \{\tilde{\mathbf{s}}\}} |\tilde{\mathbf{s}}^T \mathbf{s} - \mu(\tilde{\mathbf{s}}, \mathcal{S})| \leq 6.1 \tag{9}$$

$$\sum_{\mathbf{s} \in \mathcal{S} \setminus \{\tilde{\mathbf{s}}\}} (\tilde{\mathbf{s}}^T \mathbf{s} - \mu(\tilde{\mathbf{s}}, \mathcal{S}))^2 \leq (|\mathcal{S}| - 1) \frac{w}{2k}. \tag{10}$$

To gain some intuition behind this definition, note that $\mu(\tilde{\mathbf{s}}, \mathcal{S})$ is the average number of collisions between $\tilde{\mathbf{s}}$ and other masking strings in $\mathcal{S}$. Hence, (8) requires the average to be close to $\frac{w}{4k}$. Similarly, (9) requires the maximum number of collisions to be close to this average, and (10) bounds the "variance" of the number of collisions.

As we stated previously, the conditions in Definition 3 can be verified in a computationally efficient manner. Computing the inner product between two masking strings can be done in $O(w)$ time, since each has only w non-zero entries. Then, each mean value $\mu(\tilde{\mathbf{s}}, \mathcal{S})$ (for $\mathbf{s} \in \mathcal{S}$) can be computed in time $O(|\mathcal{S}| \cdot w)$, for a total of $O(|\mathcal{S}|^2 \cdot w)$. Finally, conditions (8)–(10) can similarly be directly checked for all $\tilde{\mathbf{s}} \in \mathcal{S}$ in time $O(|\mathcal{S}|^2 \cdot w)$.

D. Probability of Being a Promising Set

The following lemma states that the construction in Section IV-A yields a promising set with high probability.

Lemma 6: *Consider any sequence of triplets (k, w, δ) such that $k \to \infty$, $\delta \to 0$, $\delta \geq \frac{1}{k^2}$, and $w = C \log \frac{k}{\delta}$ with $C = \Theta(1)$ and $C \geq 70$. For sufficiently large k, with probability*[7] *approaching one as $k \to \infty$ the multi-set $\mathcal{S}$ is a (k, w, δ) promising set of size $\frac{2k}{\delta}$.*

Proof: Let $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{\frac{2k}{\delta}}\}$ be the multi-set constructed in Section IV-A. With a slight abuse of notation, for any i, we define $\mu(\mathbf{s}_i, \mathcal{S}) = \frac{1}{|\mathcal{S}|-1} \sum_{j\,:\,j \neq i} \mathbf{s}_i^T \mathbf{s}_j$. We will prove that, with probability approaching one, the following conditions hold simultaneously for all i:

$$\left| \mu(\mathbf{s}_i, \mathcal{S}) - \frac{w}{4k} \right| \leq \frac{0.04w}{4k} \tag{11}$$

$$\max_{j\,:\,j \neq i} \left| \mathbf{s}_i^T \mathbf{s}_j - \frac{w}{4k} \right| \leq 6.05 \tag{12}$$

[7]The probability is with respect to the randomness in the construction in Section IV-A.

$$\sum_{j\,:\,j\neq i}\Big(\mathbf{s}_i^T\mathbf{s}_j-\frac{w}{4k}\Big)^2\le(|\mathcal{S}|-1)\frac{w}{2k}. \tag{13}$$

Note that (12) implies that $\mathcal{S}$ is a set (i.e., there are no duplicates): If there existed i and j such that $i\neq j$ and $\mathbf{s}_i=\mathbf{s}_j$, then we would have $\max_{j\,:\,j\neq i}|\mathbf{s}_i^T\mathbf{s}_j-\frac{w}{4k}|=w-\frac{w}{4k}=w(1+o(1))$, violating (12). Given that $\mathcal{S}$ is a set, (11) becomes equivalent to (8). Then, combining (11) and (12) leads to (9), since

$$\begin{aligned}&\max_{\mathbf{s}\in\mathcal{S}\setminus\{\tilde{\mathbf{s}}\}}|\tilde{\mathbf{s}}^T\mathbf{s}-\mu(\tilde{\mathbf{s}},\mathcal{S})|\\&=\max_{j\,:\,j\neq i}|\mathbf{s}_i^T\mathbf{s}_j-\mu(\mathbf{s}_i,\mathcal{S})| &(14)\\&\le\Big|\mu(\mathbf{s}_i,\mathcal{S})-\frac{w}{4k}\Big|+\max_{j\,:\,j\neq i}\Big|\mathbf{s}_i^T\mathbf{s}_j-\frac{w}{4k}\Big| &(15)\\&\le\frac{0.04w}{4k}+6.05\le 6.1, &(16)\end{aligned}$$

where the final step holds for sufficiently large k since $\frac{w}{k}\to 0$. Finally, using the standard fact that an expectation $\mathbb{E}[(Z-a)^2]$ is always smallest when $a=\mathbb{E}[Z]$, and noting that $\mu(\mathbf{s}_i,\mathcal{S})$ is the average of the $\mathbf{s}_i^T\mathbf{s}_j$ values for $j\neq i$, we deduce that $\sum_{j\,:\,j\neq i}(\mathbf{s}_i^T\mathbf{s}_j-\mu(\mathbf{s}_i,\mathcal{S}))^2\le\sum_{j\,:\,j\neq i}(\mathbf{s}_i^T\mathbf{s}_j-a)^2$ for any $a\in\mathbb{R}$. For $a=\frac{w}{4k}$, (13) implies (10).

To complete the proof, we show that (11), (12), and (13) each hold (simultaneously for all i) with probability approaching one as $k\to\infty$. By the union bound, the three hold simultaneously with probability approaching one.

For (11), consider any fixed i and fixed $\mathbf{s}_i$, and view the remaining masking strings in $\mathcal{S}$ as random variables (according to the randomness in the construction). The quantity $\sum_{j\,:\,j\neq i}\mathbf{s}_i^T\mathbf{s}_j$ follows a binomial distribution with parameters $(|\mathcal{S}|-1)w$ and $\frac{1}{4k}$. By the Chernoff bound (see Appendix A), we have

$$\begin{aligned}&\mathbb{P}\bigg[\Big|\mu(\mathbf{s}_i,\mathcal{S})-\frac{w}{4k}\Big|\ge\frac{0.04w}{4k}\bigg]\\&=\mathbb{P}\bigg[\Big|\sum_{j\,:\,j\neq i}\mathbf{s}_i^T\mathbf{s}_j-(|\mathcal{S}|-1)\frac{w}{4k}\Big|\ge(|\mathcal{S}|-1)\frac{0.04w}{4k}\bigg] &(17)\\&\le 2\exp\bigg(-\frac{1}{3}\cdot(0.04^2)\cdot(|\mathcal{S}|-1)\frac{w}{4k}\bigg) &(18)\\&\le 2\exp\bigg(-\frac{70\cdot(0.04^2)}{12\delta}\log\frac{k}{\delta}\bigg) &(19)\\&=\Big(\frac{\delta}{k}\Big)^{\omega(1)}, &(20)\end{aligned}$$

where (19) uses $|\mathcal{S}|-1=\frac{2k}{\delta}-1\ge\frac{k}{\delta}$ and $w\ge 70\log\frac{k}{\delta}$, and (20) uses $\delta\to 0$. By a union bound across all $\frac{2k}{\delta}$ values of i, we deduce that (11) holds with probability approaching one.

For (12), first observe that $\mathbf{s}_i^T\mathbf{s}_j\ge\frac{w}{4k}-6.05$ holds trivially for sufficiently large k, since $\frac{w}{k}\to 0$ and $\mathbf{s}_i^T\mathbf{s}_j\ge 0$. To establish the other direction $\mathbf{s}_i^T\mathbf{s}_j\le\frac{w}{4k}+6.05$, consider any fixed i and fixed $\mathbf{s}_i$, and view $\mathbf{s}_j$ as a random variable. The quantity $\mathbf{s}_i^T\mathbf{s}_j$ follows a binomial distribution with parameters w and $\frac{1}{4k}$, so its mean is $\frac{w}{4k}$. By the Chernoff bound (see Appendix A), we have for any $\eta>0$ that

$$\begin{aligned}&\mathbb{P}\Big[\mathbf{s}_i^T\mathbf{s}_j\ge\frac{w}{4k}(1+\eta)\Big]\\&\qquad\le\exp\bigg(-\frac{w}{4k}\big((1+\eta)\log(1+\eta)-\eta\big)\bigg). &(21)\end{aligned}$$

Recalling that $w=C\log\frac{k}{\delta}$ with $C=\Theta(1)$, we set $\eta=\frac{24.2}{C}\cdot\frac{k}{\log\frac{k}{\delta}}$, so that the event in the probability (21) is indeed the complement of the event $\mathbf{s}_i^T\mathbf{s}_j<\frac{w}{4k}+6.05$. This choice satisfies $\eta\to\infty$, and hence $(1+\eta)\log(1+\eta)-\eta=(\eta\log\eta)(1+o(1))$. Also noting that $\log\eta=(\log k)(1+o(1))$, we find that (21) simplifies to

$$\begin{aligned}&\mathbb{P}\Big[\mathbf{s}_i^T\mathbf{s}_j\ge\frac{w}{4k}(1+\eta)\Big]\\&\le\exp\bigg(-\frac{C\log\frac{k}{\delta}}{4k}\cdot\frac{24.2}{C}\cdot\frac{k}{\log\frac{k}{\delta}}\cdot(\log k)(1+o(1))\bigg) &(22)\\&=\exp\Big(-\big(6.05\log k\big)(1+o(1))\Big) &(23)\\&=k^{-6.05(1+o(1))}. &(24)\end{aligned}$$

There are $\binom{|\mathcal{S}|}{2}\le(\frac{2k}{\delta})^2$ possible combinations of i and j, which we can further upper bound by $4k^6$ since $\delta\ge\frac{1}{k^2}$. Taking a union bound over all combinations, we deduce that (12) holds for all i with probability approaching one.

Finally, for (13), consider any fixed i and $\mathbf{s}_i$, and view the remaining masking strings in $\mathcal{S}$ as random variables. Under the given i and $\mathbf{s}_i$, define the random variable $Z_j=\frac{(\mathbf{s}_i^T\mathbf{s}_j-\frac{w}{4k})^2}{(0.05^2)}$ for $j\neq i$. The quantity $\mathbf{s}_i^T\mathbf{s}_j$ is a binomial random variable with parameters w and $\frac{1}{4k}$, and hence

$$\mathbb{E}[Z_j]=\frac{\mathbb{E}[(\mathbf{s}_i^T\mathbf{s}_j-\frac{w}{4k})^2]}{(6.05^2)}=\frac{\mathrm{Var}[\mathbf{s}_i^T\mathbf{s}_j]}{(6.05^2)}=\frac{w\cdot\frac{1}{4k}\cdot(1-\frac{1}{4k})}{(6.05^2)}, \tag{25}$$

which implies

$$\mathbb{E}\Big[\sum_{j\,:\,j\neq i}Z_j\Big]\le\frac{w(1-\frac{1}{4k})}{4k}\cdot\frac{|\mathcal{S}|-1}{(6.05^2)}. \tag{26}$$

By (24) and the union bound, we know that with probability at least $1-k^{-3.05(1+o(1))}$, it holds that $|\mathbf{s}_i^T\mathbf{s}_j-\frac{w}{4k}|<6.05$ for all $j\neq i$, and hence $Z_j\le 1$. It will be useful to condition on the corresponding event $\mathcal{B}=\bigcap_{j\,:\,j\neq i}\{Z_j\le 1\}$. Since $\{Z_j\}$ are independent, they remain independent after this conditioning. In addition, (26) implies that

$$\mathbb{E}\Big[\sum_{j\,:\,j\neq i}Z_j\,\Big|\,\mathcal{B}\Big]\le\frac{w(1-\frac{1}{4k})}{4k}\cdot\frac{|\mathcal{S}|-1}{(6.05^2)}, \tag{27}$$

since $\mathcal{B}$ conditions on being smaller than a given threshold, and thus cannot increase the average.

Conditioned on $\mathcal{B}$, we invoke the Chernoff bound (see Appendix A) and get

$$\begin{aligned}&\mathbb{P}\Big[\sum_{j\,:\,j\neq i}\Big(\mathbf{s}_i^T\mathbf{s}_j-\frac{w}{4k}\Big)^2\ge(|\mathcal{S}|-1)\frac{w}{2k}\,\Big|\,\mathcal{B}\Big] &(28)\\&\qquad=\mathbb{P}\Big[\sum_{j\,:\,j\neq i}Z_j\ge\frac{|\mathcal{S}|-1}{(6.05^2)}\frac{w}{2k}\,\Big|\,\mathcal{B}\Big] &(29)\end{aligned}$$

$$\leq \exp\Big(-\frac{1}{3}\frac{|\mathcal{S}|-1}{(6.05^2)}\frac{w(1-\frac{1}{4k})}{4k}\Big) \tag{30}$$

$$\leq \exp\Big(-\frac{70}{3(6.05^2)}\cdot\frac{k(1-\frac{1}{4k})\log\frac{k}{\delta}}{4k\delta}\Big) \tag{31}$$

$$= \Big(\frac{\delta}{k}\Big)^{\omega(1)}, \tag{32}$$

where the application of the Chernoff bound in (30) also uses (27), (31) uses $|\mathcal{S}|-1=\frac{2k}{\delta}-1\geq\frac{k}{\delta}$ and $w\geq 70\log\frac{k}{\delta}$, and (32) uses $\delta=o(1)$. Using $\mathbb{P}[\mathcal{B}]\geq 1-k^{-3.05(1+o(1))}$ and taking a union bound across all values of i, we deduce that (13) holds with probability approaching one. □

E. A Promising Set Must Be an LCS

The following lemma establishes that any promising set is an LCS.

Lemma 7: *Consider any sequence of triplets (k,w,δ) such that $k\to\infty$, $\delta\to 0$, $\delta\geq\frac{1}{k^2}$, and $w\geq 70\log\frac{k}{\delta}$. For sufficiently large k, a (k,w,δ) promising set $\mathcal{S}$ of size $\frac{2k}{\delta}$ must be a (k,w,δ) LCS.*

Proof: In accordance with Definition 2, fix $k'\leq k$, and select $\tilde{\mathbf{s}}_1,\dots,\tilde{\mathbf{s}}_{k'}$ from $\mathcal{S}$ uniformly at random with replacement to form the multi-set $\widetilde{\mathcal{S}}=\{\tilde{\mathbf{s}}_1,\dots,\tilde{\mathbf{s}}_{k'}\}$. Note that here $\mathcal{S}$ is already fixed; only $\tilde{\mathbf{s}}_1,\dots,\tilde{\mathbf{s}}_{k'}$ are random.

We first prove that $\mathcal{S}$ satisfies the first requirement of LCS. Specifically, we will show that with probability at least $1-\frac{\delta}{4}$, we have $\sum_{i=1}^{k'}\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}_i\leq\frac{w}{2}$ for all $\tilde{\mathbf{s}}\in\mathcal{S}\setminus\widetilde{\mathcal{S}}$. We consider a binary matrix whose $|\mathcal{S}|^{k'}$ columns correspond to all the possible $\widetilde{\mathcal{S}}$, and whose $|\mathcal{S}|$ rows correspond to all the possible $\tilde{\mathbf{s}}\in\mathcal{S}$. We say that a matrix entry corresponding to a given $\widetilde{\mathcal{S}}$ and $\tilde{\mathbf{s}}$ is *bad* if $\sum_{i=1}^{k'}\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}_i>\frac{w}{2}$ and $\tilde{\mathbf{s}}\in\mathcal{S}\setminus\widetilde{\mathcal{S}}$. To prove the desired claim, it suffices to show that at least $|\mathcal{S}|^{k'}\times(1-\frac{\delta}{4})$ columns contain no bad entries. Directly proving this appears to be challenging, so we instead prove that for each row, at most a $\frac{\delta}{4|\mathcal{S}|}$ fraction of the entries are bad. This will then imply that the total number of bad entries in the matrix is at most $|\mathcal{S}|^{k'}\times|\mathcal{S}|\times\frac{\delta}{4|\mathcal{S}|}=|\mathcal{S}|^{k'}\times\frac{\delta}{4}$, and hence there can be at most $|\mathcal{S}|^{k'}\times\frac{\delta}{4}$ columns containing bad entries.

To prove that each row has at most $\frac{\delta}{4|\mathcal{S}|}$ fraction of its entries being bad, it suffices to prove that for any given $\tilde{\mathbf{s}}$, when we choose $\tilde{\mathbf{s}}_1$ through $\tilde{\mathbf{s}}_{k'}$ from $\mathcal{S}\setminus\{\tilde{\mathbf{s}}\}$ uniformly at random with replacement, we have

$$\mathbb{P}\Big[\sum_{i=1}^{k'}\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}_i\geq\frac{w}{2}\Big]\leq\frac{\delta}{4|\mathcal{S}|}. \tag{33}$$

To prove (33), define $Z_i=\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}_i-\mu(\tilde{\mathbf{s}},\mathcal{S})$ for $i=1,\dots,k'$, where $\mu(\tilde{\mathbf{s}},\mathcal{S})$ is defined in (7). Hence, we have $\mathbb{E}[Z_i]=0$. (Note, however, that $\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}_i$ does *not* follow a binomial distribution.) Since $\mathcal{S}$ is a promising set, (8) yields

$$\sum_{i=1}^{k'}\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}_i=\sum_{i=1}^{k'}(Z_i+\mu(\tilde{\mathbf{s}},\mathcal{S}))=k'\cdot\mu(\tilde{\mathbf{s}},\mathcal{S})+\sum_{i=1}^{k'}Z_i$$

$$\leq\frac{1.04w}{4}+\sum_{i=1}^{k'}Z_i. \tag{34}$$

In addition, for all $i=1,\dots,k'$, (9) and (10) tell us that $|Z_i|\leq 6.1$ and $\mathbb{E}[Z_i^2]\leq\frac{w}{2k}$. Hence, by Bernstein's inequality (see Appendix A), we have

$$\mathbb{P}\Big[\sum_{i=1}^{k'}Z_i>\frac{0.96w}{4}\Big]\leq\exp\Big(-\frac{\frac{(0.96w)^2}{16}}{2k'\cdot\frac{w}{2k}+\frac{2}{3}\cdot 6.1\cdot\frac{0.96w}{4}}\Big) \tag{35}$$

$$\leq\exp\Big(-\frac{\frac{0.96^2}{16}w}{1+\frac{2}{3}\cdot 6.1\cdot\frac{0.96}{4}}\Big) \tag{36}$$

$$\leq\exp\Big(-\frac{w}{35}\Big) \tag{37}$$

$$\leq\Big(\frac{\delta}{k}\Big)^2 \tag{38}$$

$$\leq\frac{\delta}{4|\mathcal{S}|}, \tag{39}$$

where (36) uses $k'\leq k$, (37) uses a numerical calculation, (38) uses $w\geq 70\log\frac{k}{\delta}$, and (39) holds since $|\mathcal{S}|=\frac{2k}{\delta}$ and $k\to\infty$. In turn, for any given $\tilde{\mathbf{s}}\in\mathcal{S}$, (33) follows since

$$\mathbb{P}\Big[\sum_{i=1}^{k'}\tilde{\mathbf{s}}^T\tilde{\mathbf{s}}_i\geq\frac{w}{2}\Big]\leq\mathbb{P}\Big[\sum_{i=1}^{k'}Z_i>\frac{0.96w}{4}\Big]\leq\frac{\delta}{4|\mathcal{S}|}, \tag{40}$$

where the first inequality uses (34).

Next, we prove that $\mathcal{S}$ satisfies the second requirement for an LCS. Specifically, we show that for any given $i\in\{1,\dots,k'\}$, with probability at least $1-0.6\,\delta$, the multi-set $\{\tilde{\mathbf{s}}_1,\dots,\tilde{\mathbf{s}}_{i-1},\tilde{\mathbf{s}}_{i+1},\dots,\tilde{\mathbf{s}}_{k'}\}$ and $\tilde{\mathbf{s}}_i$ satisfy $\sum_{j\,:\,j\neq i}\tilde{\mathbf{s}}_i^T\tilde{\mathbf{s}}_j\leq\frac{w}{2}$. We clearly only need to prove this for $k'\geq 2$. In addition, since all the $\tilde{\mathbf{s}}_i$'s are generated in a symmetric manner, we can assume without loss of generality that $i=k'$.

Define $\widetilde{\mathcal{S}}^{(-k')}=\{\tilde{\mathbf{s}}_1,\tilde{\mathbf{s}}_2,\dots,\tilde{\mathbf{s}}_{k'-1}\}$. We claim that with probability at least $1-0.5\delta$, $\tilde{\mathbf{s}}_{k'}\notin\widetilde{\mathcal{S}}^{(-k')}$. To see this, note that $\tilde{\mathbf{s}}_1,\dots,\tilde{\mathbf{s}}_{k'-1}$ correspond to at most $k'-1$ distinct elements form $\mathcal{S}$, and hence $\mathbb{P}[\tilde{\mathbf{s}}_{k'}\in\widetilde{\mathcal{S}}^{(-k')}]\leq\frac{k-1}{|\mathcal{S}|}\leq 0.5\delta$. Conditioned on $\tilde{\mathbf{s}}_{k'}\notin\widetilde{\mathcal{S}}^{(-k')}$, each $\tilde{\mathbf{s}}_j$ for $1\leq j\leq k'-1$ is a uniformly random string in $\mathcal{S}\setminus\{\tilde{\mathbf{s}}_{k'}\}$. As a result, one can apply the same analysis as that for (33) (after replacing k' by $k'-1$), and deduce that

$$\mathbb{P}\Big[\sum_{j=1}^{k'-1}(\tilde{\mathbf{s}}_{k'}^T\tilde{\mathbf{s}}_j)\geq\frac{w}{2}\Big]\leq\frac{\delta}{4|\mathcal{S}|}=o(\delta), \tag{41}$$

where we used the fact that $|\mathcal{S}|=\frac{2k}{\delta}\to\infty$. Hence, we know that with probability at least $(1-0.5\delta)\cdot(1-o(\delta))\geq 1-0.6\delta$ (for sufficiently large k), the multi-set $\widetilde{\mathcal{S}}^{(-k')}$ and $\tilde{\mathbf{s}}_{k'}$ satisfy $\sum_{1\leq j\leq k'-1}\tilde{\mathbf{s}}_{k'}^T\tilde{\mathbf{s}}_j\leq\frac{w}{2}$. Finally, a union bound over the two requirements shows that they hold simultaneously with probability at least $1-\delta$, so $\mathcal{S}$ is an LCS. □

V. Extension to the Noisy Setting

In this section, we outline a simple extension of BMC (i.e., Algorithms 1 and 2) and Theorem 1 to the noisy setting. Generalizing (1), we consider the following widely-adopted symmetric noise model:

$$Y=\Big(\bigvee_{j\in\mathcal{K}}X_j\Big)\oplus Z, \tag{42}$$

where $Z \sim \text{Bernoulli}(\xi)$ for some constant $\xi \in \big[0, \frac{1}{2}\big)$, and $\oplus$ denotes modulo-2 addition. We assume that the noise is independent between tests, i.e., we have i.i.d. bit flips.

In Sections III and IV, we used masking strings with length $t_1 = 4kw$, and showed that this leads to at most $\frac{w}{2}$ collisions in each defective item's masking string, with high probability. In the following, we make use of the following more general statement: For masking strings of length $t_1 = c_1 kw$ constructed by concatenating w unit-weight substrings of length $c_1\ k$ for some constant $c_1 \geq 4$, we have

$$t_1 = c_1 kw \implies \text{At most } \frac{2w}{c_1} \text{ collisions} \tag{43}$$

in each defective item's masking string, with high probability. This follows from straightforward modifications of our previous analysis, including its associated constant factors.

For the first batch of tests, we can modify the decision step (Line 2 of the second part of Algorithm 1) to the following for improved robustness:

$$\text{if } \mathbf{s}^T \mathbf{y}_1 \geq \frac{3w}{4} \text{ then include } \mathbf{s} \text{ in the output list } \mathcal{L}. \tag{44}$$

As seen in the proof of Lemma 2, the values of $\mathbf{s}^T\mathbf{y}_1$ that we obtain in the absence of noise are exactly w for masking strings of defective items, and at most $\frac{w}{2}$ for the other masking strings. Hence, as long as fewer than $\frac{w}{4}$ bit flips occur in the entries of $\mathbf{y}_1$ corresponding to ones in $\tilde{\mathbf{s}}$, the correct decision is still made.

Under the above model of i.i.d. bit flips, we can simply use the Chernoff bound for an i.i.d. sum of w random variables (see Appendix A), and deduce that if $\xi < \frac{1}{4}$, then the mis-classification event resulting from Algorithm 1 has probability $O(n^{-c})$, where c can be set to an arbitrary value by choosing the implied constant in $w = \Theta(\log n)$ large enough. Choosing c large enough, the error probability remains small even after a union bound over the $\frac{2k}{\delta}$ masking strings. In the case that $\xi \in \big[\frac{1}{4}, \frac{1}{2}\big)$, we can increase the value of $c_1 \geq 4$ and use (43), so that $\tilde{\mathbf{s}}^T\mathbf{y}_1$ reduces from $\frac{w}{2}$ to $\frac{2w}{c_1}$ (or less) for masking strings not assigned to defective items. Upon changing the threshold from $\frac{3w}{4}$ to $\frac{1}{2}\big(\frac{2w}{c_1} + w\big)$ in (44), the preceding argument generalizes easily to this case, permitting any $\xi \in \big[0, \frac{1}{2}\big)$.

For the second batch of tests, when noise is present, we can no longer assume that the symbols at any non-collided locations are received perfectly. However, since this part is based on erasure-correcting coding, we can easily generalize to *erasure and error correcting coding* to achieve tolerance to noise.

In the presence of noise, the use of non-binary codes with symbols mapped directly to $\ell > 1$ bits (see Algorithm 2) may not be ideal, since even a single flip among these ℓ bits will cause the symbol to be changed. We therefore favor the use of a binary code $\mathcal{C}$ in the noisy setting, along with a suitable modification of the constants. In this case, we again use the more general statement in (43) with $c_1 \geq 4$, ensuring at most $\frac{2w}{c_1}$ erasures with high probability. While a code with minimum distance exceeding $\frac{2w}{c_1}$ would suffice for correcting these erasures alone, here we further increase the target minimum distance beyond $\frac{2w}{c_1}$ in order to account for the bit flips. As a specific example of a suitable binary code, the Blokh-Zyablov bound [43, Fig. 1] is achieved in [44] with linear-time encoding and decoding, and permits a positive rate as long as $\frac{d_{\min}}{w}$ is a constant strictly less than $\frac{1}{2}$.

To simplify the discussion, suppose that we naively replace all erasures by arbitrary bit values (0 or 1), so that we only have bit flips; this allows us to use codes from [44] that permit efficiently decoding any number of worst-case bit flips less than half the minimum distance. Since the bit flips are i.i.d., we can characterize the number of flips using a concentration argument: With a low enough code rate to make the code length long enough (i.e., a large enough implied constant in $w = O(\log n)$), the number of bit flips is at most $(\xi + \eta)w$ with probability $O(n^{-c})$ for any target $c > 0$, where $\eta > 0$ is any (small) constant. With at most $(\xi + \eta)w$ bit flips coming from the noise, and at most $\frac{2w}{c_1}$ bit flips coming from the collisions in the first batch, we find that the errors can be corrected as long as $\big(\xi + \eta + \frac{2}{c_1}\big)w < \frac{d_{\min}}{2}$. Since $d_{\min}$ can be arbitrarily close to $\frac{w}{2}$, this condition can always be satisfied for sufficiently large c_1 and a sufficiently low code rate as long as $\xi < \frac{1}{4}$. One can also handle the case $\xi \in \big[\frac{1}{4}, \frac{1}{2}\big)$ using a code that corrects both erasures and bit flips, instead of naively treating erasures as flips.

In summary, under i.i.d. noise of the form (42), by modifying only the constant factors and the code $\mathcal{C}$ used, we can achieve the same scaling laws as Theorem 1 in terms of both tests and runtime.

VI. Conclusion

We have introduced a novel scheme for sublinear-time non-adaptive group testing, and established that it attains asymptotically vanishing error probability with $t = O(k \log n)$ tests and $O(k^2 \cdot \log k \cdot \log n)$ runtime. Our algorithm and analysis use coding-based subroutines that permit straightforward extensions to the noisy setting. In future work, it may be interesting to provide a refined analysis with more precise constant factors, or to seek variants of BMC with lower scaling in the decoding time.

Appendix A
Concentration Inequalities

We make use of several standard concentration bounds for sums of independent random variables, e.g., see [45, Sec. 4.1] and [46, Ch. 2]. For clarity, in this section we summarize the specific bounds used. Letting $Z_1, \dots, Z_n$ be a sequence of independent and identically distributed random variables, we have the following:

- (Chernoff bound) If $Z_i \in [0, 1]$ almost surely, and $\mathbb{E}[Z_i] = \mu$, then for any $\alpha > 0$, we have

$$\mathbb{P}\bigg[\sum_{i=1}^{n} Z_i \geq (1+\alpha)n\mu\bigg] \leq \exp\Big(-\mu n\big((1+\alpha)\log(1+\alpha) - \alpha\big)\Big), \tag{45}$$

and for any $\alpha \in (0,1]$, we have

$$\mathbb{P}\Big[\sum_{i=1}^{n} Z_i \le (1-\alpha)n\mu\Big] \le \exp\Big(-\mu n\big((1-\alpha)\log(1-\alpha)+\alpha\big)\Big). \quad (46)$$

- (Weakened Chernoff bound) If $Z_i \in [0,1]$ almost surely and $\mathbb{E}[Z_i] = \mu$, then for any $\alpha \in (0,1]$, we have

$$\mathbb{P}\Big[\sum_{i=1}^{n} Z_i \ge (1+\alpha)n\mu\Big] \le \exp\Big(-\frac{1}{3}\alpha^2\mu n\Big), \quad (47)$$

$$\mathbb{P}\Big[\sum_{i=1}^{n} Z_i \le (1-\alpha)n\mu\Big] \le \exp\Big(-\frac{1}{3}\alpha^2\mu n\Big). \quad (48)$$

  In addition, we use the fact that (47) holds even when μ is replaced by any upper bound $\mu_{\rm ub} \ge \mu$ on both sides. While we expect this variant to have appeared previously, we provide a proof below for completeness.
- (Bernstein's inequality) Suppose that $|Z_i| \le M$ almost surely, and that $\mathbb{E}[Z_i] = 0$. For any $\delta > 0$, we have

$$\mathbb{P}\Big[\sum_{i=1}^{n} Z_i \ge t\Big] \le \exp\Bigg(-\frac{t^2}{2\big(\sum_{i=1}^{n}\mathrm{Var}[Z_i] + \frac{1}{3}Mt\big)}\Bigg). \quad (49)$$

Proof of generalized version of (47)**.** We begin by bounding $\Psi(t) := \mathbb{P}\big[\sum_{i=1}^{n} Z_i \ge n\mu + t\big]$ for a generic choice of t. First, by Bernstein's inequality in (49), we have $\Psi(t) \le \exp\big(-\frac{t^2/2}{\sum_{i=1}^{n}\mathbb{E}[Z_i^2]+t/3}\big)$. Since $z^2 \le z$ for $z \in [0,1]$, we have $\sum_{i=1}^{n}\mathrm{Var}[Z_i] \le \sum_{i=1}^{n}\mathbb{E}[Z_i^2] \le \sum_{i=1}^{n}\mathbb{E}[Z_i] = n\mu \le n\mu_{\rm ub}$, yielding $\Psi(t) \le \exp\big(-\frac{t^2/2}{n\mu_{\rm ub}+t/3}\big)$. Setting $t = \alpha n\mu_{\rm ub}$ for some $\alpha \in [0,1]$ and simplifying via $\frac{1/2}{1+\alpha/3} \ge \frac{1}{3}$, we obtain $\Psi(\alpha n\mu_{\rm ub}) \le \exp(-n\alpha^2\mu_{\rm ub}/3)$. Finally, since $\mu_{\rm ub} \ge \mu$, we have $\mathbb{P}\big[\sum_{i=1}^{n} Z_i \ge n\mu_{\rm ub} + \alpha n\mu_{\rm ub}\big] \le \Psi(\alpha n\mu_{\rm ub})$, which gives the desired upper bound.

APPENDIX B
THE VERY SPARSE REGIME $k = O(1)$

In our main result (Theorem 1), we assumed that $k \to \infty$ as $n \to \infty$. Here we describe how to use BMC to attain $P_{\rm e} \to 0$ as $n \to \infty$ in the case that $k = O(1)$, while using $t = O(\log n)$ tests and $O((\log n)^2)$ decoding time.

We again use Definition 1, letting each masking string contain $w = \log n$ segments of length $4k$ and weight one, so that the total length is $t_1 = 4\ k\log n$. We again use the random construction from Section III-B, this time with a multi-set of size $|\mathcal{S}| = \log n$. For two such random masking strings $\mathbf{s}$ and $\mathbf{s}'$, the average number of collisions (i.e., 1's in common) follows a binomial distribution with parameters $\log n$ and $\frac{1}{4k}$, so the mean is $\frac{\log n}{4\ k}$. Hence, by the Chernoff bound (see Appendix A), the probability of the number of collisions exceeding $\frac{\log n}{2k}$ is $O(n^{-c})$ for some $c > 0$ (here c depends on k, but is still $\Omega(1)$ since $k = O(1)$). By a union bound over $O(\log^2\ n)$ pairs, we deduce that the probability of *any* two $\mathbf{s}, \mathbf{s}' \in \mathcal{S}$ having more than $\frac{\log n}{2k}$ collisions tends to zero as $n \to \infty$. We henceforth condition on the (high-probability) complement of this event.

Due to this conditioning, we find that *any* $\mathbf{s} \in \mathcal{S}$ collides with *any* subset $\widetilde{\mathcal{S}} \subseteq \mathcal{S}\backslash\{\mathbf{s}\}$ of cardinality k (or less) in at most $k \times \frac{\log n}{2k} = \frac{1}{2}\log n = \frac{w}{2}$ positions. Hence, the two conditions in Definition 2 hold for *any* $k' \le k$ distinct strings $\tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_{k'}$ from $\mathcal{S}$. As a result, when we assign strings from $\mathcal{S}$ to the n items uniformly at random with replacement, the only case that causes excessive collisions is that in which two defective items are assigned the same masking string. Since $|\mathcal{S}| = \log n$ and $k = O(1)$, this occurs with probability $O\big(\frac{1}{\log n}\big)$.

Given $\mathcal{S}$ satisfying the preceding properties, the proof of Theorem 1 goes through essentially unchanged with $w = O(\log n)$. The number of tests is $O(w) = O(\log n)$, and the decoding time is $O(|\mathcal{S}| \cdot \log n) = O((\log n)^2)$.

ACKNOWLEDGMENT

The authors would like to thank Rui Zhang for helpful discussions, Sidharth Jaggi for helpful comments regarding the sublinear-time group testing literature, and Mahdi Cheragchi for helpful comments regarding efficient erasure-correcting codes. The authors of this paper are alphabetically ordered.

REFERENCES

[1] S. Bondorf, B. Chen, J. Scarlett, H. Yu, and Y. Zhao, "Cross-sender bit-mixing coding," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2019, pp. 205–216.

[2] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Statist.*, vol. 14, no. 4, pp. 436–440, Dec. 1943.

[3] A. F. Anta, M. A. Mosteiro, and J. R. Munoz, "Unbounded contention resolution in multiple-access channels," in *Distributed Computing*, vol. 6950. Berlin, Germany: Springer, 2011, pp. 225–236.

[4] R. Clifford, K. Efremenko, E. Porat, and A. Rothschild, "Pattern matching with don't cares and few errors," *J. Comput. Syst. Sci.*, vol. 76, no. 2, pp. 115–124, 2010.

[5] G. Cormode and S. Muthukrishnan, "What's hot and what's not: Tracking most frequent items dynamically," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 249–278, Mar. 2005.

[6] A. C. Gilbert, M. A. Iwen, and M. J. Strauss, "Group testing and sparse signal recovery," in *Proc. 42nd Asilomar Conf. Signals, Syst. Comput.*, Oct. 2008, pp. 1059–1063.

[7] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "One sketch for all: Fast algorithms for compressed sensing," in *Proc. ACM-SIAM Symp. Discrete Alg. (SODA)*, New York, NY, USA, 2007, pp. 237–246.

[8] D. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*, vol. 12. Singapore: World Scientific, 2000.

[9] M. Aldridge, O. Johnson, and J. Scarlett, "Group testing: An information theory perspective," *Found. Trend. Comms. Inf. Theory*, vol. 15, nos. 3–4, pp. 196–392, 2019.

[10] M. Aldridge, "Individual testing is optimal for nonadaptive group testing in the linear regime," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2058–2061, Apr. 2019.

[11] H. A. Inan, P. Kairouz, M. Wootters, and A. Ozgur, "On the optimality of the Kautz-Singleton construction in probabilistic group testing," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5592–5603, Sep. 2019.

[12] A. G. D'yachkov and V. V. Rykov, "Bounds on the length of disjunctive codes," *Problemy Peredachi Informatsii*, vol. 18, no. 3, pp. 7–13, 1982.

[13] E. Porat and A. Rothschild, "Explicit nonadaptive combinatorial group testing schemes," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 7982–7989, Dec. 2011.

[14] M. Cheraghchi, "Noise-resilient group testing: Limitations and constructions," in *Proc. Int. Symp. Fundamentals Comput. Theory*, 2009, pp. 62–73.

[15] P. Indyk, H. Q. Ngo, and A. Rudra, "Efficiently decodable non-adaptive group testing," in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2010, pp. 1126–1142.

[16] H. Q. Ngo, E. Porat, and A. Rudra, "Efficiently decodable error-correcting list disjunct matrices and applications," in *Proc. Int. Colloq. Automata, Lang., Prog.*, 2011, pp. 557–568.

[17] M. Cheraghchi and J. Ribeiro, "Simple codes and sparse recovery with fast decoding," 2019, *arXiv:1901.02852*. [Online]. Available: http://arxiv.org/abs/1901.02852

[18] M. B. Malyutov, "The separating property of random matrices," *Math. Notes Acad. Sci. USSR*, vol. 23, no. 1, pp. 84–91, Jan. 1978.
[19] V. L. Freidlina, "On a design problem for screening experiments," *Theory Probab. Appl.*, vol. 20, no. 1, pp. 102–115, 1975.
[20] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama, "Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2011, pp. 1832–1839.
[21] M. Cheraghchi, A. Hormati, A. Karbasi, and M. Vetterli, "Group testing with probabilistic tests: Theory, design and application," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 7057–7067, Oct. 2011.
[22] M. B. Malyutov, "Search for sparse active inputs: A review," in *Information Theory, Combinatorics, and Search Theory*. Berlin, Germany: Springer, 2013, pp. 609–647.
[23] M. Aldridge, L. Baldassini, and O. Johnson, "Group testing algorithms: Bounds and simulations," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3671–3687, Jun. 2014.
[24] J. Scarlett and V. Cevher, "Phase transitions in group testing," in *Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2016, pp. 40–53.
[25] J. Scarlett and V. Cevher, "Near-optimal noisy group testing via separate decoding of items," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 4, pp. 625–638, 2018.
[26] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Information-theoretic and algorithmic thresholds for group testing," in *Proc. Int. Colloq. Automata, Lang. Program. (ICALP)*, 2019, pp. 7911–7928.
[27] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Optimal group testing," 2019, *arXiv:1911.02287*. [Online]. Available: http://arxiv.org/abs/1911.02287
[28] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi, "Efficient algorithms for noisy group testing," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2113–2136, Apr. 2017.
[29] K. Lee, K. Chandrasekher, R. Pedarsani, and K. Ramchandran, "SAFFRON: A fast, efficient, and robust framework for group testing based on sparse-graph codes," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4649–4664, Sep. 2019.
[30] J. D. Lee, Y. Sun, and J. E. Taylor, "On model selection consistency of regularized M-estimators," *Electron. J. Statist.*, vol. 9, no. 1, pp. 608–642, 2015.
[31] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 4, pp. 363–377, Oct. 1964.
[32] E. Price and J. Scarlett, "A fast binary splitting approach to non-adaptive group testing," in *Proc. Int. Conf. Randomization Comput. (RANDOM)*, 2020, pp. 13:1–13:20.
[33] M. Cheraghchi and V. Nakos, "Combinatorial group testing and sparse recovery schemes with near-optimal decoding time," 2020, *arXiv:2006.08420*. [Online]. Available: http://arxiv.org/abs/2006.08420
[34] F. K. Hwang, "A method for detecting all defective members in a population by group testing," *J. Amer. Stat. Assoc.*, vol. 67, no. 339, pp. 605–608, 1972.
[35] H. A. Inan and A. Ozgur, "Strongly explicit and efficiently decodable probabilistic group testing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 525–530.
[36] G. Cormode and S. Muthukrishnan, "Combinatorial algorithms for compressed sensing," in *Proc. Int. Colloquium Struct. Inf. Commun. Complex.* Berlin, Germany: Springer, 2006, pp. 280–294.
[37] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, "Algorithmic linear dimension reduction in the l_1 norm for sparse vectors," in *Proc. Allerton Conf. Commun., Control Comput.*, 2006, pp. 1411–1418.
[38] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Proc. Allerton Conf. Commun., Control Comput.*, Sep. 2008, pp. 798–805.
[39] M. Cheraghchi and P. Indyk, "Nearly optimal deterministic algorithm for sparse Walsh-Hadamard transform," *ACM Trans. Algorithms*, vol. 13, no. 3, p. 34, 2017.
[40] A. G. D'yachkov and V. V. Rykov, "A survey of superimposed code theory," *Problems Control Inf. Theory*, vol. 12, no. 4, pp. 1–13, 1983.
[41] A. D. Bonis, A. Gąsieniec, and U. Vaccaro, "Optimal two-stage algorithms for group testing problems," *SIAM J. Comput.*, vol. 34, no. 5, pp. 1253–1270, 2005.
[42] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1732–1736, Nov. 1996.
[43] I. Dumer, "Concatenated codes and their multilevel generalizations," in *Handbook of Coding Theory*, vol. 2. Amsterdam, The Netherlands: Elsevier, 1998, pp. 1911–1988.
[44] V. Guruswami and P. Indyk, "Linear-time encodable/decodable codes with near-optimal rate," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3393–3400, Oct. 2005.
[45] R. Motwani and P. Raghavan, *Randomized Algorithms*. London, U.K.: Chapman & Hall/CRC, 2010.
[46] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford, U.K.: OUP Oxford, 2013.

Steffen Bondorf (Member, IEEE) received the Dr.-Ing. degree in computer science from TU Kaiserslautern, Germany, in 2016. He is currently an Assistant Professor of distributed and networked systems with the Faculty of Mathematics, Ruhr University Bochum, Germany. After graduation, he was a Carl-Zeiss Fellow at TU Kaiserslautern, a Research Fellow with the School of Computing, National University of Singapore, and an ERCIM Fellow with the Department of Information Security and Communication Technology, NTNU Trondheim, Norway. His research interests include performance modeling and analysis of communication networks.

Binbin Chen (Member, IEEE) received the B.Sc. degree in computer science from Peking University and the Ph.D. degree in computer science from the National University of Singapore. Since July 2019, he has been an Associate Professor with the Information Systems Technology and Design (ISTD) Pillar, Singapore University of Technology and Design (SUTD). He currently holds a joint appointment as a Principal Research Scientist at the Advanced Digital Sciences Center, which is a research center of the University of Illinois in Singapore. His current research interests include wireless networks, cyber-physical systems, and cyber security for critical infrastructures.

Jonathan Scarlett (Member, IEEE) received the B.Eng. degree in electrical engineering and the B.Sc. degree in computer science from the University of Melbourne, Australia, and the Ph.D. degree from the Signal Processing and Communications Group, University of Cambridge, U.K., in August 2014. From September 2014 to September 2017, he was Post-Doctoral Researcher with the Laboratory for Information and Inference Systems, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Since January 2018, he has been an Assistant Professor with the Department of Computer Science, the Department of Mathematics, and the Institute of Data Science, National University of Singapore (NUS). His research interests include the areas of information theory, machine learning, signal processing, and high-dimensional statistics. He received the Singapore National Research Foundation (NRF) Fellowship and the NUS Early Career Research Award.

Haifeng Yu is currently an Associate Professor (with tenure) with the Department of Computer Science, National University of Singapore. His current research interests include the general areas of distributed algorithms, distributed systems security, and applied algorithms in networking. He has published in various premier publication venues, such as JACM, PODC, SIGCOMM, and S&P. Some of his papers have won best paper awards, including a Best Paper Award in SIGCOMM.

Yuda Zhao received the B.Sc. degree in computer science from Tsinghua University and the Ph.D. degree in computer science from the National University of Singapore. He is currently working as the Head of Risk in Atome Finance, a fintech company based in Singapore.