# Community-Aware Group Testing

Pavlos Nikolopoulos, *Member, IEEE*, Sundara Rajan Srinivasavaradhan, Tao Guo, *Member, IEEE*, Christina Fragouli, *Fellow, IEEE*, and Suhas N. Diggavi, *Fellow, IEEE*

***Abstract*— Group testing is a technique that can reduce the number of tests needed to identify infected members in a population, by pooling together multiple diagnostic samples. Despite the variety and importance of prior results, traditional work on group testing has typically assumed independent infections. However, contagious diseases among humans, like SARS-CoV-2, have an important characteristic: infections are governed by community spread, and are therefore correlated. In this paper, we explore this observation and we argue that taking into account the community structure when testing can lead to significant savings in terms of the number of tests required to guarantee a given identification accuracy. To show that, we start with a simplistic (yet practical) infection model, where the entire population is organized in (possibly overlapping) communities and the infection probability of an individual depends on the communities (s)he participates in. Given this model, we compute new lower bounds on the number of tests for zero-error identification and design community-aware group testing algorithms that can be optimal under assumptions. Finally, we demonstrate significant benefits over traditional, community-agnostic group testing via simulations using both noiseless and noisy tests.**

***Index Terms*— Coding, group testing.**

## I. Introduction

GROUP testing can identify the infected individuals in a population using much fewer tests than individual testing. The idea is based on *pooled tests*, which are tests applied on *groups* of diagnostic samples from multiple individuals. So, if infections are sparse, then many pooled tests are likely to be negative and large parts of the population can be massively identified as healthy. Interestingly, group testing has become popular in the context of COVID-19 [3], [4], [5], [6], [7], and several countries (including India, Germany, US, and China) have already deployed preliminary group-testing strategies [8], [9]. Also, companies and schools use pooled tests to regularly monitor parts of their population, and then do individual tests once a pooled test comes out positive.

Manuscript received 1 March 2022; revised 6 September 2022; accepted 16 January 2023. Date of publication 1 March 2023; date of current version 16 June 2023. This work was supported in part by NSF under Grant 2146828, Grant 1705077, and Grant 2007714. Earlier versions of this paper were presented in part at the 2021 International Conference on Artificial Intelligence and Statistics and at the 2021 IEEE International Conference on Communications [DOI: 10.1109/ICC42927.2021.9500791]. *(Corresponding author: Pavlos Nikolopoulos.)*

Pavlos Nikolopoulos is with EPFL, School of Computer and Communication Sciences, 1015 Lausanne, Switzerland (e-mail: pavlos.nikolopoulos@epfl.ch).

Sundara Rajan Srinivasavaradhan, Tao Guo, Christina Fragouli, and Suhas N. Diggavi are with the Department of Electrical and Computer Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA.

Communicated by R. Gabrys, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3250119.

Digital Object Identifier 10.1109/TIT.2023.3250119

Group testing has a rich history in academic literature dating back to R. Dorfman in 1943, who first introduced the concept [10]. One can find nice summaries of the various setups examined so far in [11], [12], [13]. Simply stated, the traditional group-testing problem assumes a population of $n$ individuals out of which a few are infected *independently*, and the goal is to design testing strategies to identify the infected individuals from the pooled-test results. In this regard, most works propose a particular test design (e.g. Bernoulli) coupled with a decoding strategy (e.g. Definite Defectives), and guarantees are provided on the number of tests required to achieve zero- or small-error identification. Additionally, order-optimality results have been proved for the asymptotic regime, where the population size tends to infinity (see Section II for more details).

The new observation we make in this paper is that viral diseases like SARS-CoV-2 are governed by community spread, hence are *not* independent. So, we ask: if infections are based on a known community structure, can we leverage that structure to make group testing more efficient, i.e. achieve the same identification accuracy as traditional group testing, but with even fewer tests?

Knowing the overall community structure (at least to some extent) is not unrealistic today [14], [15], [16], but simpler, more easily acquired structures are also important. As a use case, consider an apartment building consisting of $F$ families that have practiced social distancing; clearly, there is a strong correlation on whether members of the same family are infected or not. Assume that the building management would like to test all members to enable access to common facilities. We ask: what is the most test-efficient way to do so? how many tests do we actually need?

We argue that taking into account the community structure may lead to significant savings in terms of the number of tests required to guarantee a given identification accuracy. Using entropy arguments, it is easy to see that accounting for individual correlations can help coming up with a lower bound for the number of tests that can be less than the traditional counting bound: if we represent the state (infected or not) of each individual as a binary variable, the joint entropy of correlated variables can be much smaller than the sum of the individual entropies. This indicates that there may be room for improvement on the algorithmic side as well. As an extreme case, in the above example, assume that in each family, either all or no members are infected; then clearly, it is enough to test a single member from each family.

We also argue that leveraging the community structure can enlarge the regime, where group testing offers significant

benefits over individual testing. Indeed, classical group testing offers much greater benefits in the sparse regime, where $k = \Theta(n^\alpha)$ and $\alpha < 1$. [11], [17], [18], [19], [20]. However, taking into account the community structure allows us to identify and remove from the population large groups of infected members, thus reducing their proportion and converting a possibly linear- to a sparse-regime identification. Essentially, the community structure can guide us on when to use individual and when group testing.

Our results are as follows: Suppose that $n$ population members are organized into $F$ (possibly overlapping) *communities*, out of which $k_f$ have at least one infected member. The latter may hold exactly or on average. First, we derive a lower bound on the number of tests needed to identify all infected members without error; for some communities and infection regimes, the bound can be shown to increase (almost) linearly with $k_f$ (the number of infected communities) as opposed to $k$ (the number of infected members). Second, we propose an adaptive algorithm that achieves the lower bound in specific parameter regimes. Third, we propose a nonadaptive algorithm that accounts for the community structure and reduces the number of tests at the expense of few false positives. Fourth, we propose a new decoder based on loopy belief propagation that is generic enough to accommodate any community structure and can be combined with any test design (encoder) to achieve low error rates. Last, we numerically validate that leveraging the community structure can offer significant benefits either when tests are noiseless or not.

The paper is structured as follows: We start with background and related work (Section II). Next, we present our community infection model (Section III) and compute the corresponding lower bound (Section IV). Then, we describe our community-aware (non)adaptive test designs for the case where communities have no overlap (Section V), which offers useful insights for designing tests in the general (overlapping) case (Section VI). Finally, we present our loopy belief propagation (LBP) decoder (Section VII), and we close with numerical evaluation (Section VIII) and conclusions (Section IX).

## II. Background and Related Work

### A. Traditional Group Testing

In mathematical terms, a pooled test indexed by $\tau$ takes as input samples from a set of individuals $\delta_\tau$ and outputs a binary value: 1 ("positive") if at least one of the samples is infected, and 0 ("negative") if none of them is infected. More precisely, let $U_i = 1$ if individual $i$ is infected, and 0 otherwise. The output of pooled test $\tau$ is calculated as $Y_\tau = \bigvee_{i\in\delta_\tau} U_i$, where $\bigvee$ stands for the OR operator (disjunction).

Group testing typically considers two models for the infections in a population of $n$ members: (i) a *combinatorial priors* model, where a fixed number of infected individuals $k$ is randomly selected among all sets of size $k$; (ii) an *i.i.d probabilistic priors* model, where each individual is i.i.d. infected with probability $p$, hence the expected number of infected members is $\bar{k} = np$.

In each model, of critical interest is the minimum number of group tests $T = T(n)$ needed to identify the infected members without error or with high probability. In the combinatorial model (i), since $T$ tests allow to distinguish among $2^T$ combinations of test outputs, we need $T \geq \log_2 \binom{n}{k}$ to identify $k$ randomly infected individuals out of $n$. This is known as the *counting bound* and implies that in a sparse regime (i.e. $k = \Theta(n^\alpha)$ and $\alpha \in [0, 1)$), no algorithm can use less than $\mathcal{O}(k \log \frac{n}{k})$ tests to achieve (almost) zero-error identification [12], [21]. In the probabilistic model (ii), a similar bound has been derived for the number of tests needed on average: $T \geq nh_2(p)$, where $h_2$ is the binary entropy function [11].

The usual goal in group testing is to design a testing algorithm that is able to identify all infection statuses $\mathbf{U} = (U_1, U_2, \ldots, U_n)$. Testing algorithms can be adaptive or non-adaptive. Adaptive testing uses the outcome of previous tests to decide what tests to perform next. One such example is the *binary splitting algorithm (BSA)*, which implements a form of binary search [22], [23]. Nonadaptive testing constructs, in advance, a test matrix $\mathbf{G} \in \{0, 1\}^{T\times n}$ where each row corresponds to a test $\tau$, each column to a member, and the non-zero elements determine the sets $\delta_\tau$. Although adaptive testing typically needs fewer tests, nonadaptive testing is often more practical as all tests can be executed in parallel.

**Known results (for noiseless group testing):** In the combinatorial model (i), if the number of infected individuals follows a sparse regime (i.e. $k = \Theta(n^\alpha)$ and $\alpha \in [0, 1)$), adaptive group testing, and more specifically Hwang's generalized binary splitting algorithm (HGBSA), is asymptotically optimal w.r.t. the counting bound [11], [23]. Moreover, if $\alpha \in [0, 0.409]$ there exists a nonadaptive, randomized test design, coupled with decoder, which can identify all infected individuals from the test outcomes with high probability, using a number of tests that asymptotically matches the counting bound. However, the latter is not possible for nonadaptive group testing whenever $\alpha > 0.409$; i.e., no test design, randomized or not, with a number of tests matching the counting bound allows to infer the infected members with a non-vanishing probability. Therefore, in this regime at least two stages of testing are necessary [24].

Conversely, classic individual testing has been proved to be order-optimal in the linear regime (i.e. $k = \Theta(n)$). In fact, if the infection rate $k/n$ is more than 0.38, group testing does not use fewer tests than one-to-one (individual) testing unless high identification error rates are acceptable [17], [18], [19], [20]. Moreover, it has been recently shown that individual testing is asymptotically optimal among non-adaptive designs in the mildly sublinear regime (where $k = \omega(\frac{n}{\log n})$) [25].

The above achievability/converse results for the combinatorial priors are directly applicable to the probabilistic model (ii), by considering $p = k/n$. In fact, Theorems 1.7 and 1.8 from [11] imply that any algorithm that attains a vanishing probability of error on the combinatorial priors, also attains a vanishing probability of error on the corresponding i.i.d. probabilistic priors.

Evidently, despite its thorough analysis, prior work has focused on independent infections. This is perhaps because the group-testing problem has been motivated so far by its interesting mathematical aspect. Group testing is a form of

inference in sparsity regimes, such as compressed sensing, but with an interesting difference: all operations are in Boolean (as opposed to real-valued) algebra, which makes the problem significantly harder. However, the practical challenges of the current pandemic (e.g. scale/cost of testing) and the fact that viral diseases are spread according to people's interactions have naturally brought up the need for similar results in the case of correlated/community-based infections.

### B. Related Work

The idea of community-aware group testing was explored in our conference papers [1], [2], and also our earlier preprints [26], [27]. A similar idea of using side-information from contact tracing in decoding was proposed in [28], [29], independently from our work. In our opinion, that work is complementary to ours; we focus more on test designs rather than decoding, for which we use known algorithms such as COMP and LBP. Follow-up works also share similar goals [30], [31], [32], [33], [34], [35].

Our community-based approach can be viewed as an instantiation of a recent trend in the group-testing literature that examines variations motivated by "real-world" scenaria. For example, graph-constrained group testing considers the case where samples cannot be pooled together arbitrarily in tests, but must conform to constraints imposed by a graph [36], [37], [38], [39]. Sparse group testing considers cases where individuals can participate in a limited number of tests, or tests cannot pool more than a limited number of samples; such constraints can significantly affect the scaling laws [40]. However, in our context, individuals can be grouped into tests freely.

Another related line of work is the work on graph-constrained group testing (e.g., see [36], [38], [39]) that solves the problem of how to design group tests when there are constraints on which samples can be pooled together, provided in the form of a graph; in our case, individuals can be pooled together into tests freely.

Further related is the work on independent but not identical priors [41], [42], as well as the work on models for the test outcomes or noisy tests. For example, [43] proposes a test model specifically tailored to COVID-19 testing, where the test outcomes may also provide a rough estimate of the number of infected samples. In addition, following up on a rich literature on noisy group testing (see for example [44], [45], [46]), generalized group testing [47] subsumes as special cases a variety of noisy group-testing models; it assumes the test outcome is positive with some probability $f(x)$, where $x$ is the number of defectives tested in a pool, and $f(\cdot)$ is an arbitrary monotonically increasing (stochastic) test function. In this work, we do not further expand on these complementary and interesting directions.

Finally, belief propagation has been considered in the past in the context of noiseless [48] as well as noisy [49] nonadaptive group testing, but in the traditional setting, where infections are independent. In this work, we modify loopy belief propagation (LBP) to incorporate the structure of the community-based, correlated infections into the structure of the factor graph; in particular, we add more variable nodes representing the infection status of the communities and we compute the variable-to-factor-node and factor-to-variable-node messages accordingly. Our community-aware LBP decoder is generic enough to accommodate any community structure and can be combined with any test design (encoder).

## III. Model and Problem Formulation

### A. Community Model

Our work extends the results of the traditional setting in Section II-A by assuming a possibly-overlapping community structure: members may belong to one or more communities—hence they are infected according to new combinatorial and probabilistic models, depending on how the communities overlap (Section III-B).

More formally, we assume that all members of the entire population $\mathcal{V} = \{1, 2, \cdots, n\}$ are organized in a known community structure, which can be perceived in the form of a hypergraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$: each vertex $v \in \mathcal{V}$ corresponds to an individual, that we simply call a *member*, and each edge $e \in \mathcal{E}$ indicates which members belong to the same *community*. Since $\mathcal{G}$ is a hypergraph, an edge may connect any number of vertices; hence, a member may belong to one or multiple communities. The number of communities that a member belongs to is called the *degree* of the member.

There exist $F$ communities in total, and each community $e$ has $|\mathcal{V}_e|$ members.

The hypergraph $\mathcal{G}$ may be decomposed into connected components, where each component $C(\mathcal{V}_C, \mathcal{E}_C)$ is a sub-hypergraph. For each component $C$, we define a partition $D_C$ of $\mathcal{V}_C$ as the smallest collection of nonempty subsets of members/vertices that participate in *exactly* the same subset of communities/hyperedges. More specifically, consider the following social preorder on $\mathcal{V}_C$: a member $v$ is said to be *not more socialized* than another member $\nu$, if $\nu$ participates in all the communities that also $v$ participates in. If $v$ is not more socialized than $\nu$ and $\nu$ is not more socialized than $v$, we say that $v$ and $\nu$ are *socially equivalent*—they participate in the same set of communities. The latter defines an equivalent relation, and $D_C$ is the collection of the equivalent classes.

For each part $d \in D_C$, $\mathcal{V}_d$ denotes the set of members it contains, and $\mathcal{E}_{\mathcal{V}_d}$ is the (common) set of communities/hyperedges they belong to. Clearly, all members in $\mathcal{V}_d$ have the same degree of at least one; and as described next, these members are infected according to some common infection principle.

We distinguish 2 kinds of sets in $D_C$: (a) the "outer" sets: $D_{C,out} \triangleq \{d \in D_C : \nexists b \in D_C \text{ s.t. } \mathcal{E}_{\mathcal{V}_b} \subset \mathcal{E}_{\mathcal{V}_d}\}$, and (b) the "inner" sets: $D_{C,in} \triangleq D_C \setminus D_{C,out}$. In other words, given the social preorder defined above, when an equivalent class $d$ is such that $d$ itself is the only class that is not more socialized than $d$, then $d$ is called an outer set. Hence, outer sets are the minimal elements in the social preorder; the rest are inner sets.

Figure 1 depicts a simple example: $\mathcal{E}_C$ consists of 3 hyperedges, and $D_C$ contains 7 disjoint sets/ equivalent classes: 3 outer (yellow) and 4 inner (green, blue) sets. Note that the members of inner sets have a higher degree than those of the outer sets, while the degree of an outer set is not necessarily equal to one.

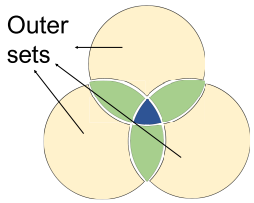



Fig. 1. An example partition with 3 outer (yellow) and 4 inner (green, blue) sets.

### B. Infection Models

The following community-based infection models parallel the classic ones (Section II-A):

• **Combinatorial Model (I).** $k_f$ of the communities have at least one infected member (we will call these *infected communities*). The rest of the communities have no infected members. Any combination of infected communities has the same chance of occurring. In each infected community, there are $k_m^e$ infected members, out of which $k_m^{\mathcal{I}}$ are shared among a subset of infected communities $\mathcal{I} \subseteq \mathcal{E}$, that happen to intersect in the underlying community model. I.e., in each subset of members $\mathcal{V}_{\mathcal{I}} = \cap_{e\in\mathcal{I}}\mathcal{V}_e \neq \emptyset$), there exist $k_m^{\mathcal{I}}$ infected members. Similarly, in each disjoint subset $d \in D_C$ of the partition described above, the number of infected members is denoted by $k_m^d$. The infected communities (resp. infected community members) are randomly chosen out of all communities (resp. members that belong to the same communities), and all combinations are possible.

As also discussed below, to better capture practical infection scenaria, we further assume that $k_m^{\mathcal{I}}/|\mathcal{V}_{\mathcal{I}}|$ is an increasing function of the number of intersecting communities $|\mathcal{I}|$. Similarly, $k_m^d/|\mathcal{V}_d|$ is a non-decreasing function of the degree $|\mathcal{E}_{\mathcal{V}_d}|$.

• **Probabilistic Model (II).** Each community $e$ is infected with probability $q$ i.i.d. If a member $v$ of an infected community $e$ belongs *only* to that community (i.e. has degree 1), then it is infected with probability $p_v = p_e$, independently from the other members/communities. If $v$ belongs to a subset of infected communities $\mathcal{I} \subseteq \mathcal{E}$, it is considered to be infected by either of these communities; so, given their infection probabilities $\{p_e : e \in \mathcal{I}\}$, we say that $v$ becomes infected with probability: $p_v = 1 - \prod_{e\in\mathcal{I}}(1 - p_e)$. If $v$ does not belong to any infected community, then $p_v = 0$. Note that a community $e$ may be labeled "infected" without having any infected members; however, the probability of this is negligible for reasonably high infection probabilities $p_e$ and practical values of $|\mathcal{V}_e|$.

*Some useful remarks:* First, note that although the communities are infected independently, their structure causes a dependent infection model; in fact, the way communities overlap determines the infection probability of their shared members.

Second, our models capture situations where the infection is determined by the participation in a community rather than the status of community members. Albeit simplistic, we believe that this model can be useful in real pandemics. Since the exact community structure of the entire population may be hard to be known in practice, we expect that a graph such as $\mathcal{G}$ can only partially describe the reality; there might be additional members that do not belong to $\mathcal{V}$ but interact with the ones in it in various unknown ways, or there might be additional communities that are simply not captured due to unknown member interactions. Hence, assuming that communities become infected independently seems a simple yet reasonable model to use.

However, once a few communities in $\mathcal{G}$ get infected, we expect that the infection probability of a member will increase with the number of infected communities it belongs to, which is captured by our models in the dependence of $k_m^d/|\mathcal{V}_d|$ on $|\mathcal{E}_{\mathcal{V}_d}|$ and the computation of $p_v$. In Section VI, we leverage this in order to design our adaptive and non-adaptive testing strategies.

Third, both models (I) and (II) allow communities to have quite different infection levels from each other (e.g., different infection probabilities); this is important, as, if we view the setting we examine here as a "static" snapshot of how infections dynamically evolve over time, our models enable to capture many different paths and ways to arrive at the current snapshot state.

A special case of our model is the *non-overlapping* case, where the population is partitioned in $F$ disjoint communities. This case is more amenable to analysis (e.g. if $k_m^e = p_e|\mathcal{V}_e|$, both models I and II behave similarly) and offers useful insights; so, we will examine it separately (Section V).

### C. Problem Formulation

Given the above community infection model, our goal is two-fold: (a) provide new lower bounds on the number of tests $T$ needed for zero-error identification; and (b) design community-aware testing algorithms that are more efficient than traditional group testing, i.e. they can achieve the same identification accuracy using significantly fewer tests.

*Assumptions:* We assume that there is no dilution noise; that is, the performance of a test does not depend on the number of samples pooled together. This is a reasonable assumption with genetic RT-PCR tests, where even small amounts of viral nucleotides can be amplified to be detectable [7], [50]. However, we do consider noisy tests in our numerical evaluation using a Z-channel noise model (Section VIII). We remark that this is simply a model one may use; our algorithms are agnostic to this and can be used with other noise models.

*Terminology:* $\hat{U}_v$ denotes the estimated state of $U_v$ after running our group testing algorithm. *Zero error* means that $\hat{U}_v = U_v$ for all $v \in \mathcal{V}$. *Vanishing error* requires that the overall error probability goes to zero with $n$. We distinguish between *False Negative (FN)* and *False Positive (FP)* errors: FN errors occur when infected members are identified as non-infected (and vice-versa for FP).

## IV. Lower Bound on the Number of Tests

We now compute the minimum number of tests needed to identify all infected members under the zero-error criterion in both models (I) and (II). All proofs are in appendix A.

*Theorem 1 (Combinatorial Community Bound):* Consider combinatorial model (I). Any algorithm that identifies all $k$

infected members without error requires a number of tests $T$ satisfying:

$$T \geq \log_2 \binom{F}{k_f} + \sum_{C\in\mathcal{G}} \sum_{d\in D_C} \log_2 \binom{|\mathcal{V}_d|}{k_m^d}, \tag{1}$$

where $|\mathcal{V}_d|$ (resp. $k_m^d$) is the number of members (resp. infected members) in each disjoint set $d \in D_C$. In the non-overlapping case, the above reduces to:

$$T \geq \log_2 \binom{F}{k_f} + \sum_{e=1}^{k_f} \log_2 \binom{|\mathcal{V}_e|}{k_m^e}. \tag{2}$$

**Observations:** We can make two observations in the case where the number of members in infected communities follows a "strongly" linear regime (i.e. $k_m^d \approx |\mathcal{V}_d|$) and the number of infected communities $k_f$ follows a sparse regime (i.e., $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0,1)$):

(a) The bound increases almost *linearly* with $k_f$ (the number of infected communities), as opposed to $k$ (the overall number of infected members). This is because, if the infection regime about communities is sparse, the following asymptotic equivalence holds: $\log_2 \binom{F}{k_f} \sim k_f \log_2 \frac{F}{k_f} \sim (1-\alpha_f) k_f \log_2 F$.

(b) If additionally to the sparse regime about communities, an overall sparse regime ($k = \Theta(n^\alpha)$ for $\alpha \in [0,1)$) holds, then the community bound may be significantly lower than the (community-agnostic) counting bound. Consider, for example, a symmetric non-overlapping case, where in (2) $|\mathcal{V}_e| = M$ and $k_m^e = k_m$ for all $e \in \mathcal{E}$: The asymptotic behavior of the counting bound in the sparse regime is $\log_2 \binom{n}{k} \sim k \log_2 \frac{n}{k} \sim k_f k_m \log_2 \frac{F}{k_f}$, where the latter is because $k_m \approx M$. So, the ratio of the counting bound to the combinatorial community bound in (2) scales (as $F$ gets large) as:

$$\frac{\log_2 \binom{n}{k}}{\log_2 \binom{F}{k_f} + k_f \log_2 \binom{M}{k_m}} \sim \frac{k_f k_m \log_2 \frac{F}{k_f}}{k_f \log_2 \frac{F}{k_f}} = k_m.$$

This observation is relevant for practical scenarios, as many times, the population is composed of communities with members in close contact (e.g. relatives, work colleagues, etc.)—hence, almost all members of infected communities are expected to be infected (i.e. $k_m \approx M$), even if the overall infection regime may still be sparse.

A similar bound exists for the probabilistic model (II). By rephrasing [41, Theorem 1], any probabilistic group testing algorithm, whose average success probability is at least $\mathbb{P}_{suc}$, requires at least $T \geq \mathbb{P}_{suc} \cdot H(\mathbf{U})$ tests.

Accordingly, we state the following theorem:

*Theorem 2 (Probabilistic Community Bound (II)):* Consider probabilistic model (II). Any algorithm with noiseless measurements, whose average success probability is at least $\mathbb{P}_{suc}$, requires a number of tests:

$$T \geq \mathbb{P}_{suc} \cdot \Bigg[ F h_2(q) + \sum_{v=1}^{n} \sum_{\mathcal{I}\subseteq\mathcal{E}_v} q^{|\mathcal{I}|}(1-q)^{|\mathcal{E}_v|-|\mathcal{I}|} h_2\left(\prod_{e\in\mathcal{I}}(1-p_e)\right) - \sum_{e=1}^{F}(1-q+q(1-p_e)^{|S_e|}) \cdot h_2\left(\frac{1-q}{1-q+q(1-p_e)^{|S_e|}}\right)\Bigg], \tag{3}$$

where $\mathcal{E}_v$ is the set of communities that member $v$ belongs to, $\mathcal{I}$ is the subset of infected communities in $\mathcal{E}_v$, and $S_e$ is the set of members who *only* belong to community $e$.

In the non-overlapping case, the above reduces to:

$$T \geq \mathbb{P}_{suc} \cdot \Bigg[ F h_2(q) + \sum_{e=1}^{F} q|\mathcal{V}_e| h_2(p_e) - w_e h_2\left(\frac{1-q}{w_e}\right)\Bigg], \tag{4}$$

where $w_e = 1-q+q(1-p_e)^{|\mathcal{V}_e|}$.

As said, we consider both zero-error recovery ($\mathbb{P}_{suc} = 1$) and recovery with errors; the former is related to our adaptive test designs, the latter to our nonadaptive ones.

**Algorithm 1** Non-Overlapping Community Testing

$\hat{U}_v$ is the estimated infection status of member $v$.
$\hat{U}_x$ is the estimated infection status of a mixed sample $x$.
$SelectRepresentatives()$ is a function that selects a representative subset from a set of members.
$AdaptiveTest()$ is a classic adaptive algorithm that tests a set of items (mixed samples or members).

**for** $e \in \mathcal{E}$ **do**
  $r_e = SelectRepresentatives\left(\{v : v \in e\}\right)$
**end for**
$\left[\hat{U}_{x(r_1)}, \ldots, \hat{U}_{x(r_F)}\right] = AdaptiveTest\left(x(r_1), \ldots, x(r_F)\right)$
Set $A := \emptyset$
**for** $e = 1, \ldots, F$ **do**
  **if** $\hat{U}_{x(r_e)} =$ "positive" **then**
    Use a noiseless, individual test for each community member: $\hat{U}_v = U_v, \forall v \in e$.
  **else**
    $A := A \cup \{v : v \in e\}$
  **end if**
**end for**
$\left\{\hat{U}_v : v \in A\right\} = AdaptiveTest\left(A\right)$
**return** $\left[\hat{U}_1, \ldots, \hat{U}_n\right]$

## V. Algorithms for Non-Overlapping Communities

In this section, we examine the case of non-overlapping communities, which is not only a realistic scenario (e.g. consider the apartment building from our introduction), but also offers useful insights for the general case described in the next section. All proofs can be found in appendix B.

### A. Adaptive Algorithm

Algorithm 1 is our adaptive algorithm for the non-overlapping case. We next sketch its main points, but the interested reader may also find a detailed rationale about it in Section B-C).

The algorithm consists of two parts; both make use of some traditional adaptive group-testing algorithm, say

$AdaptiveTest()$ (such as BSA). We use $AdaptiveTest()$ as an abstraction for any existing (or future) adaptive group-testing algorithm that assumes independent infections. We distinguish between 2 different kinds of input for $AdaptiveTest()$: (a) a set of selected members, which is the typical input of group-testing algorithms; (b) a set of selected *mixed samples*. A mixed sample is created by pooling together samples from multiple members that usually have some common characteristic. For example, mixed sample $x(r_e)$ denotes an aggregate sample of a set of representative members $r_e$ from community $e$. A mixed sample is "positive," if at least one of the members that compose it is infected, and "negative" otherwise. Because in some cases we care about mixed samples, we can treat them in the same way as individual samples—hence use group testing to identify the infection state of mixed samples as we do for individuals.

**Part 1 (lines 1-4):** The goal of this part is to detect the infection *regime* inside each community $e$, so that the community is tested accordingly in Part 2: i.e., via group testing, if $e$ is "lightly" infected, and via individual testing, otherwise. Our idea is motivated by the result presented in Section II-A that group testing is beneficial, only if the infection rate is low (i.e. $p_e \leq 0.38$). Thus, the only remaining challenge is to accurately detect the infection regime spending only a limited number of tests. In this paper, we limit our exploration to using only one mixed sample in this regard, but more sophisticated techniques are also possible (see for example appendix B-D).

More specifically, a representative subset $r_e$ of community $e$ is selected using a sampling function $SelectRepresentatives()$ (lines 1-3). Then, a mixed sample $x(r_e)$ is produced, and an $AdaptiveTest()$ is applied to the representative mixed samples (line 4). If $AdaptiveTest()$ achieves exact reconstruction (which is usually the case), then: $\hat{U}_{x(r_e)} = U_{x(r_e)}$.

**Part 2 (lines 5-13):** We treat $\hat{U}_{x(r_e)}$ as an estimate of the infection regime inside community $e$: if $\hat{U}_{x(r_e)}$ is positive, then we consider the community as heavily infected (i.e ${}^{k^e_m}/_{|\mathcal{V}_e|}$ or $p_e \geq 0.38$), otherwise lightly infected (i.e. ${}^{k^e_m}/_{|\mathcal{V}_e|}$ or $p_e < 0.38$). Since group testing performs better than individual testing only in the latter case (Section II-A), we use individual testing for each heavily-infected community (lines 7-8), and group testing for all the lightly-infected ones (line 13).

**Analysis for the number of tests.** We now compute the maximum expected number of tests needed by our algorithm to detect the infection status of all members without error. We present our results using the symmetric (a.k.a. uniform) case, where $|\mathcal{V}_e| = M$, $k^e_m = k_m$ (for the combinatorial model) or $p_e = p$ (for the probabilistic model), and $|r_e| = R$ for all communities: Let $SelectRepresentatives()$ be a simple function that performs uniform (random) sampling without replacement, and consider 2 choices for the $AdaptiveTest()$ algorithm: (i) Hwang's generalized binary splitting algorithm (HGBSA) [23], which is optimal if the number of infected members of the tested group is known in advance; and (ii), traditional binary-splitting algorithm (BSA) [22], which performs well, even if little is known about the number of infected members.

*Lemma 1 (Expected Tests - Symmetric Combinatorial Model):* Consider the choices (i) and (ii) for the $AdaptiveTest()$ defined above. Algorithm 1 succeeds using an expected number of tests:

$$\begin{aligned}\bar{T}_{(i)} &\leq k_f\phi_c\left(\log_2\frac{F}{k_f\phi_c}+1+M\right)\\ &\quad+k\left(1-\phi_c\right)\left(\log_2\frac{n-k_fM\phi_c}{k\left(1-\phi_c\right)}+1\right) \qquad (5)\\ \bar{T}_{(ii)} &\leq k_f\phi_c\left(\log_2 F+1+M\right)\\ &\quad+k\left(1-\phi_c\right)\left(\log_2 n+1\right), \qquad (6)\end{aligned}$$

where $\phi_c$ is the expected fraction of infected communities whose mixed sample is positive, computed by the hypergeometric distribution $Hyper(M, k_m, R)$.

*Lemma 2 (Expected Tests - Symmetric Probabilistic Model):* If Algorithm 1 uses BSA in place of $AdaptiveTest()$, then it succeeds using an expected number of tests:

$$\begin{aligned}\bar{T} &\leq Fq\phi_p\left(\log_2 F+1+M\right)\\ &\quad+nqp\left(1-\phi_p\right)\left(\log_2 n+1\right), \qquad (7)\end{aligned}$$

where $\phi_p = 1-(1-p)^R$ is the expected fraction of infected communities whose mixed sample is positive.

Lemmas 1, 2 are derived in appendix B, as a repeated application of the performance bounds of HGBSA and BSA: if out of $n$ members, $k$ are infected, then HGBSA (resp. BSA) achieves exact identification using at most: $\log_2\binom{n}{k}+k$ (resp. $k\log_2 n+k$) tests [11], [51].

**Observations:**

(a) For certain community structures and given that heavily/lightly infected communities are detected without errors in Part 1, our algorithm can asymptotically achieve (up to a constant) the lower combinatorial bound of Theorem 1. We show this via 2 examples:

First, consider a sparse regime for communities (i.e. $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0,1)$) and a moderately linear regime within each community (i.e. ${}^{k_m}/_{M} \approx 0.5$). Thus: $\log_2\binom{F}{k_f} \sim k_f\log_2({}^{F}/_{k_f})$, $\log_2\binom{M}{k_m} \sim Mh_2({}^{k_m}/_{M}) \sim M$ and the bound in Theorem 1 becomes: $k_f\left(\log_2{}^{F}/_{k_f}+M\right)$. If $R$ is chosen such that all infected communities (which are also heavily infected as ${}^{k_m}/_{M} > 0.38$) are detected without errors (e.g. if $R > M-k_m$), then $\phi_c = 1$; thus, the RHS of (5) becomes almost equal (up to constant $k_f$) to the lower bound in Theorem 1.

Second, consider the opposite example, where the infection regime for communities is very high, while each separate community is lightly infected. In this case, $k = k_fk_m \approx k_f$; therefore, the lower bound becomes: $T \sim k_f\log_2({}^{F}/_{k_f}) + k_fk_m\log_2({}^{M}/_{k_m}) \approx k\log_2({}^{n}/_{k})$. If $R$ is chosen such that none of the (lightly infected) communities is marked as heavily infected in Part 1 (e.g. if $R=0$, which reduces to using traditional community-agnostic group testing), then $\phi_c = 0$, and the RHS of (5) is almost equal (up to $k$) to the bound in (2).

(b) The upper bound in (6) shows that our algorithm achieves significant benefits compared to BSA when the

infected communities are heavily-infected and $R$ is chosen such that $\phi_c = 1$ (e.g. $R > M - k_m$); this is because $\bar{T}_{(ii)} \leq k_f (\log_2 F + 1 + M) \ll k \log_2 n + k$. Also, it achieves the same performance as BSA, when communities are lightly-infected and $R$ is chosen such that $\phi_c = 0$ (e.g. $R = 0$); this is because $\bar{T}_{(ii)} \leq k \log_2 n + k$. Since the former case is more realistic, our algorithm is expected to perform much better than classic BSA in practice.

The examples in observation (a) and the above analysis indicate two things: First, the knowledge of the community structure is more beneficial when communities are heavily infected; in case of very light infections, traditional group testing performs equally well. Our experiments showed that the community structure helps whenever $p > 0.15$, with benefits increasing with $p$. Second, a rough estimate of the communities' infection rate has to be available in order to optimally choose $R$. In appendix B-C, we demonstrate that this is unavoidable in the symmetric scenario we examine and when only one mixed sample per community is used to identify which communities are heavily/lightly infected.

(c) Even in the most favorable regime for our community-aware group testing, where very few communities are infected with almost all their members infected (i.e. $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0, 1)$ and $k_m \approx M$), even if $R$ is chosen optimally such that $\phi_c = 1$, the ratio of the expected number of tests needed by Algorithm 1 (see (5)) and HGBSA cannot be less than $1/\log(n/k)$, which upper bounds the benefits one may get. In appendix B-D, we detail this observation and provide an optimized version of our algorithm that slightly improves upon the gain of $1/\log(n/k)$.

### B. Non-Adaptive Algorithm

For simplicity of notation, consider the symmetric case, where $|\mathcal{V}_e| = M$ for all communities.

**Test Matrix.** We split $\mathbf{G}$ into two sub-matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ of sizes $T_1 \times n$ and $T_2 \times n$.

▷ The sub-matrix $\mathbf{G}_1$ identifies the infected communities using one mixed sample from each community, akin to line 4 of Algorithm 1. We want $\mathbf{G}_1$ to identify all (non-)infected communities with small error probability. If there are many tests available, we set $T_1 = F$ and use one row for each community test. Otherwise, in sparse $k_f$ regimes, we set $T_1$ closer to $\mathcal{O}(k_f \log \frac{F}{k_f})$.

▷ The sub-matrix $\mathbf{G}_2$ has a block matrix structure and contains $F$ identity matrices $I_M$, one for each community. $\mathbf{G}_2$ is designed as follows: (i) each block column contains only one identity matrix $I_M$, i.e., each member is tested only once; (ii) each block row $i$ ($i \in \{1, 2, \cdots, b\}$) contains $c_i$ identity matrices $I_M$, i.e., there are $c_i$ members included in the corresponding tests. As a result: $T_2 = bM$. An example with $F = 6$, $b = 3$, $c_1 = 2$, $c_2 = 1$, $c_3 = 3$ is:

$$\mathbf{G}_2 = \begin{bmatrix} I_M & 0_{M\times M} & 0_{M\times M} & I_M & 0_{M\times M} & 0_{M\times M} \\ 0_{M\times M} & I_M & 0_{M\times M} & 0_{M\times M} & 0_{M\times M} & 0_{M\times M} \\ 0_{M\times M} & 0_{M\times M} & I_M & 0_{M\times M} & I_M & I_M \end{bmatrix}.$$

Our detailed rationale about $\mathbf{G}_2$ can be found in appendix B-E.

**Decoding.** We use the test outcomes of $\mathbf{G}_1$ to identify the non-infected communities and we remove the corresponding columns from $\mathbf{G}_2$. We next use the remaining columns of $\mathbf{G}_2$ and combinatorial orthogonal matching pursuit (COMP) [52], [53] to identify the infected members, namely: $(i)$ A member is identified as non-infected if it is included in at least one negative test in $\mathbf{G}_2$. $(ii)$ All other members, that are only included in positive tests in $\mathbf{G}_2$, are identified as infected.

**Error Probability.** After the removal of the columns, the block structure of $\mathbf{G}_2$ helps us obtain a test matrix that is close to an identity matrix—hence perform "almost" individual testing. Also, note that our decoding strategy for $\mathbf{G}_2$ leads to zero FN errors. But, FP errors may happen if identity matrices $I_M$ corresponding to two or more infected communities appear in the same block row—we call this event "covering". In this case, some non-infected members may be included in the same test with infected members from other communities and falsely identified as infected. Building on these ideas, the following lemmas guide us through a design of $\mathbf{G}_2$ that minimizes the (FP) error probability:

*Lemma 3:* Under models (I) and (II), the probability of a "covering" event, where there is some block row containing two or more infected communities is:

$$\mathbb{P}^{I}_{\text{covering}} = 1 - \frac{\sum_{|\mathcal{B}|=k_f:\, \mathcal{B}\subseteq\{1,2,\cdots,b\}} \prod_{i\in\mathcal{B}} c_i}{\binom{F}{k_f}}, \tag{8}$$

$$\mathbb{P}^{II}_{\text{covering}} = 1 - \prod_{i=1}^{b} \left[(1-q)^{c_i} + c_i q (1-q)^{c_i - 1}\right]. \tag{9}$$

*Lemma 4:* $\mathbb{P}_{covering}$ is minimized for both models (I) and (II), if $c_i = c$, $\forall i \in \{1, \cdots, b\}$.

Given a $\mathbf{G}_2$ sub-matrix as in Lemma 4, we now compute the system FP probability:

$$\mathbb{P}(\text{any-FP}) \triangleq \mathbb{P}(\exists v : \hat{U}_v = 1 \text{ and } U_v = 0). \tag{10}$$

We do so, under the assumption that $F$ is a multiple of $b$ and $c$: i.e., $b = T_2/M$ and $c = FM/T_2$. If $F$ cannot be factorized, we can simply pad our design with $F'$ fictitious communities without infected members, so that $F + F' = bc$.

*Lemma 5:* For $\mathbf{G}_2$ as in Lemma 4 and $F = bc$, the system FP probability for models (I) and (II) equals:

$$\mathbb{P}^{I}(\text{any-FP}) = \left[1 - \frac{1}{\binom{M}{k_m}}\right]\left[1 - \frac{\binom{T_2/M}{k_f}(FM/T_2)^{k_f}}{\binom{F}{k_f}}\right]$$

$$\mathbb{P}^{II}(\text{any-FP}) = \left[1 - \sum_{i=1}^{M} \left[p^i (1-p)^{M-i}\right]^2 \frac{1}{\binom{M}{i}}\right] \cdot \left[1 - \left((1-q)^{\frac{FM}{T_2}-1}\left(1 - q + \frac{FMq}{T_2}\right)\right)^{T_2/M}\right]$$

$\mathbb{P}(\text{any-FP})$ can be pessimistic; a more practical metric is the average fraction of members that are misidentified (error rate): $R(\text{error}) \triangleq 1/n \cdot |\{v : \hat{U}_v \neq U_v\}|$.

*Lemma 6:* For $\mathbf{G}_2$ as in Lemma 4, the error rate is calculated for models (I) and (II) as:

$$R_I(\text{error}) < \frac{k_f(M - k_m)}{FM} \cdot \mathbb{P}^{I}_{\text{joint}}, \tag{11}$$

**Algorithm 2** Adaptive Community Testing

$\hat{U}_v$ is the estimated infection state of member $v$.
$\hat{U}_x$ is the estimated state of mixed sample $x$.

1: **for** $d \in D_{C,out}$, $\forall$ connected component $C$ of $\mathcal{G}$ **do**
2: $\quad r_d \leftarrow SelectRepresentatives\,(\mathcal{V}_d)$
3: **end for**
4: $\left\{\hat{U}_{x(r_d)}\right\} \leftarrow AdaptiveTest\,(\{x(r_d)\})$
5: Set $A := \emptyset$
6: **for** each connected component $C$ of $\mathcal{G}$ **do**
7: $\quad$ **for** $d \in D_{C,out}$ **do**
8: $\quad\quad$ **if** $\hat{U}_{x(r_d)} =$ "positive" **then**
9: $\quad\quad\quad$ Individually test $\mathcal{V}_d$: $\hat{U}_v \leftarrow U_v, \forall v \in \mathcal{V}_d$.
10: $\quad\quad\quad \hat{p}_d \leftarrow {}^1\!/\!_{|\mathcal{V}_d|} \cdot \sum_{v\in\mathcal{V}_d} \mathbf{1}_{\{\hat{U}_v = \text{"positive"}\}}$
11: $\quad\quad$ **else**
12: $\quad\quad\quad A \leftarrow A \cup \{v : v \in \mathcal{V}_d\}$
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad$ **for** $b \in D_{C,in}$ (in increasing order of degree) **do**
16: $\quad\quad$ **if** $\exists d \in D_C$ s.t. $\mathcal{E}_{\mathcal{V}_d} \subset \mathcal{E}_{\mathcal{V}_b}$ & $\hat{p}_d > \theta$ **then**
17: $\quad\quad\quad$ Individually test $\mathcal{V}_b$: $\hat{U}_v \leftarrow U_v, \forall v \in \mathcal{V}_b$.
18: $\quad\quad\quad \hat{p}_b \leftarrow {}^1\!/\!_{|\mathcal{V}_b|} \cdot \sum_{v\in\mathcal{V}_b} \mathbf{1}_{\{\hat{U}_v = \text{"positive"}\}}$
19: $\quad\quad$ **else**
20: $\quad\quad\quad A \leftarrow A \cup \{v : v \in \mathcal{V}_b\}$
21: $\quad\quad$ **end if**
22: $\quad$ **end for**
23: **end for**
24: $\left\{\hat{U}_v : v \in A\right\} = AdaptiveTest\,(A)$
25: **return** $\left[\hat{U}_1, \ldots, \hat{U}_n\right]$

$$R_{II}(\text{error}) < (1-p)q\left[1-(1-q)^{c-1}\right]. \tag{12}$$

## VI. Extension to Overlapping Communities

### A. Adaptive Algorithm

Algorithm 2 describes our generalized adaptive algorithm. It is built on Algorithm 1, with the main difference being that the representatives are selected from the outer sets (Section III-A) of the communities. More specifically, the algorithm is again split into 2 parts:

**Part 1:** For each component of the graph $\mathcal{G}$, we first identify the outer sets $D_{C,out}$. Then, from each outer set $d$, we select a representative subset of members $r_d$, and create the mixed sample $x(r_d)$ (lines 1-2). Finally, all mixed samples are identified (line 4).

**Part 2:** We treat $\hat{U}_{x(r_d)}$ as a rough estimate of the infection regime inside each set $d$: if $\hat{U}_{x(r_d)}$ is positive, we consider $d$ to be heavily infected and we individually test its members (line 9); otherwise, we consider it lightly infected and we include its members in set $A$ (line 12). For our rough estimate of the infection regime to be a good one, we choose the number of representatives based on some prior information about infection rate of each outer set; for example if $p_e < 0.38$ then only one representative is enough, otherwise pooling together the entire set is one's best option. Note that the exact knowledge of $p_e$ and a rough prior may be easily acquired. For example, in realistic scenarios, where the community infection rates are not expected to be low, pooling together the entire outer set is a good heuristic.

Due to individually testing the heavily-infected outer sets, we obtain more accurate estimates of their infection rates, $\hat{p}_d$, by computing the average proportion of infected members (line 10). We use these estimates to decide how to test the inner sets of the component: if an outer set $d$ exists whose members belong to a subset of communities in $\mathcal{E}_{\mathcal{V}_b}$ and its estimated infection rate $\hat{p}_d$ is above a threshold $\theta$, then members of $\mathcal{V}_b$ are tested individually (line 17) and a new estimate $\hat{p}_b$ for the infection rate of that set is computed (line 18). Else, members of $\mathcal{V}_b$ are included in set $A$. Our rationale follows the infection model described in Section III-B, which implies that the infection probability of the members of an inner set $b$ will be at least equal to the infection probability of the members (of an outer set $d$) whose community(ies) are a subset of $\mathcal{E}_{\mathcal{V}_d}$. Hence, if an outer set is heavily infected then a corresponding inner set will be heavily infected, too. In our experiments, we numerically examine the impact of $\theta$.

Finally, we test all members of set $A$ that are not tested individually (because infection probability is presumably low) using traditional group testing (line 23).

### B. Non-Adaptive Algorithms

For simplicity, we describe our non-adaptive algorithm using the symmetric case.

**Test Matrix.** We again split $\mathbf{G}$ into two sub-matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ of sizes $T_1 \times n$ and $T_2 \times n$.

▷ The sub-matrix $\mathbf{G}_1$ identifies the non-infected outer sets using one mixed sample for each outer set. If the number of tests available is large, we set $T_1$ to be the number of outer sets; otherwise, in sparse $k_f$ regimes, $T_1$ can be closer to $\mathcal{O}(k_f \log \frac{F}{k_f})$.

▷ Suppose $T_2 = \frac{n}{c}$, with $c$ being an integer parameter value. The sub-matrix $\mathbf{G}_2$ of size $T_2 \times n$ has one "1" in each column (each of the $n$ member participates in one test) and $c$ "1"s in each row (each test pools together $c$ members). For $c = 1$, this reduces to individual testing.

The design of $\mathbf{G}_2$ amounts to deciding which members are placed in the same test. We propose that no two members from the same outer/inner set are placed in the same test and that (assuming set sizes are equal) we randomly select which of the sets with equal degrees will be tested together, i.e. the corresponding identity matrices will be in the same block row; equivalently, $\mathbf{G}_2$ is a concatenation of $c$ identity matrices $I_{T_2}$, i.e., $\mathbf{G}_2 = [I_{T_2}\ I_{T_2} \cdots I_{T_2}]$. If set sizes are different, we try to put sets of similar size and degree in the same block rows and compensate missing rows with zero-padding.

**Decoding.** We use the same decoder from Section V-B that follows the logic of COMP decoder.

**Intuition.** Suppose that there are only few infected communities, each one having a large percentage (say $> 0.38$) of infected members. The idea is similar to the non-overlapping case: ideally, once the members of the non-infected outer sets are removed because of the decoding phase of $\mathbf{G}_1$, we would like each row to have only a single member (instead of being empty or having more members). The proposed structure attempts to balance exactly this: having a high enough number of members in each row so that, once the

non-infected community members are removed in the first decoding phase, no row remains empty; and having a low enough number of members in each row, so that, once the non-infected community members are removed, each row has a small number of members.

**Example.** We here illustrate for a special case our proposed design for $\mathbf{G}_2$ and the resulting error rate our algorithm achieves. Assume that we have $F$ communities, where $2F_o$ communities pairwise overlap (each community overlaps with exactly one other community) and the remaining $F-2F_o$ communities do not overlap with any other community. Assume each community has $M$ members, and overlapping communities share $M_o$ members. We construct the sub-matrix $\mathbf{G}_2$ of size $T_2 \times n$ as in the following example that uses $F = 6$, $F_o = 2$, $M = 3$, $M_o = 1$:

$$\mathbf{G}_2 = \begin{bmatrix} I_3 & & & I_3 & & \\ & I_2 & & & I_2 & \\ & & I_1 & & & I_1 \\ & & & I_2 & & I_2 \end{bmatrix}.$$

This matrix starts with $b_1 = \frac{F-2F_o}{c}$ block-rows that each contains $c$ identity matrices $I_M$, one corresponding to each non-overlapping community. We then have $b_2 = \frac{F_o}{c}$ block-rows each containing $c$ identity matrices $I_{2M-M_o}$, one for each pair of overlapping communities. Each $I_{2M-M_o}$ matrix contains three matrices $I_{M-M_o}$, $I_{M_o}$, and $I_{M-M_o}$ corresponding to the members that belong only in one of the communities, or in both. Note that $F = (b_1 + 2b_2)c$ and $T_2 = b_1 M + b_2(2M - M_o)$.

**Error Rate.** Note that the (COMP) decoding strategy that we use leads to zero FN errors. The following lemma provides an analysis of the error (FP) rate for the design of $\mathbf{G}_2$ in the example which is defined as: $R(\text{error}) \triangleq {}^1\!/_n \cdot |\{v : \hat{U}_v \neq U_v\}|$. We provide the expected error rate for only the probabilistic model (II) for the purpose of comparison with traditional Bernoulli design in Fig. 2.

*Lemma 7:* Consider the probabilistic model (II). For the community structure and $\mathbf{G}_2$ as described in the above example, the error rate is:

$$R_{II}(\text{error}) = \frac{1}{n}\Big[\left(1-(1-pq)^{c-1}\right)\cdot N_1 + \left(1-(1-pq)^{2(c-1)}\right)\cdot N_2\Big], \quad (13)$$

where $N_1$ and $N_2$ are the expected number of non-overlapped and overlapped members in infected communities that are non-infected, respectively, and can be obtained as

$$N_1 = (F-2F_o)q(1-p)M + 2F_o q(1-p)(M-M_o)$$
$$N_2 = F_o\left(1-(1-q)^2\right)(1-p)M_o.$$

The error rate of traditional group testing using Bernoulli design (with parameter $\frac{1}{k}$) and COMP decoding has an error rate of $R_{\text{tradition}}(\text{error}) = {}^1\!/_n \cdot (n-k)\left(1-1/k(1-1/k)^k\right)^T$. Fig. 2 depicts $R(\text{error})$ for parameters $F = 150$, $F_o = 60$, $M = 10$, $M_o = 2$, $q = 0.2$, and $p = 0.2$.

**Remark:** Our nonadaptive designs for both the non-overlapping as well as the overlapping cases use identity blocks in $\mathbf{G}_2$ (i.e. each individual is tested once).

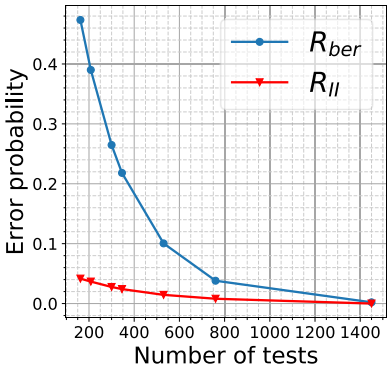



Fig. 2. Error rate for Bernoulli design vs $G_1G_2$ design for the example.

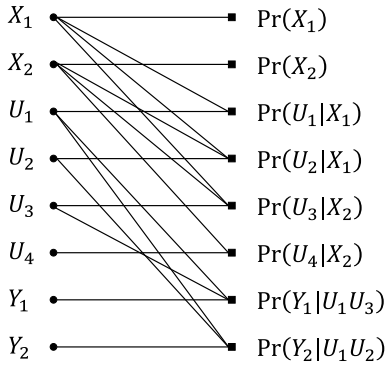



Fig. 3. An example of factor graph.

This is because our designs want to capture well the case, where the fraction of infected members within each infected community or outer/inner disjoint set is large (say $> 0.38$). If we considered sparse infection regimes and very light infection rates, we could perhaps replace each identity block matrix in $\mathbf{G}_2$ with a nonadaptive group testing matrix that is known to be optimal in the particular regime of operation. In significantly sparse regimes, even traditional nonadaptive designs (that are agnostic to the community structure) might perform equally well. However, in this paper and our experiments (Section VIII), we focus more on the former case, because this is the most interesting scenario.

## VII. Loopy Belief Propagation Decoder

We now describe our new algorithm for decoding infection status of the individuals (and communities). This is accomplished by estimating the posterior probability of the corresponding individual (or community) being infected via *loopy belief propagation* (LBP). LBP computes the posterior marginals exactly when the underlying factor graph describing the joint distribution is a tree (which is rarely the case) [54]. Nevertheless, it is a practical algorithm that has achieved success on various applications. Also, LBP offers soft information (posterior distributions), which can be proved more useful than hard decisions in the context of disease-spread management.

We use LBP for our probabilistic model, because it is fast and can be easily configured to take into account the community structure leading to more reliable identification. Many inference algorithms exist that estimate the posterior

marginals, some of which have also been employed for group testing. For example, GAMP [28] and Monte-Carlo sampling [55] may yield more accurate decoders. However, the focus of our work is to examine whether benefits from accounting for the community structure (both at the test design and the decoder) exist; hence we think that considering a simple (possibly sub-optimal) decoder based on LBP is a good first step—we defer more complex designs to future work.

We next describe the factor graph and the belief propagation update rules for our probabilistic model (II). Let the infection status of each community $e$ be $X_e \sim \text{Ber}(q)$. Moreover, let $S_v$ denote the set of communities that $U_v$ belongs to. Then:

$$\mathbb{P}(X_1,\ldots,X_F,U_1,\ldots,U_n,Y_1,\ldots,Y_T) = \prod_{e=1}^{F}\mathbb{P}(X_e)\prod_{v=1}^{n}\mathbb{P}(U_v|X_{S_v})\prod_{\tau=1}^{T}\mathbb{P}(Y_\tau|U_{\delta_\tau}), \quad (14)$$

where $\delta_\tau$ is the group of individuals included in test $\tau$. Equation (14) can be represented by a factor graph, where the variable nodes correspond to the variables $X_e, U_v, Y_\tau$ and the factor nodes correspond to $\mathbb{P}(X_e), \mathbb{P}(U_v|X_{S_v}), \mathbb{P}(Y_\tau|U_{\delta_\tau})$. Figure 3 shows an example of 2 communities, 4 members and 2 tests.

Given the result of each test is $y_\tau$, i.e., $Y_\tau = y_\tau$, LBP estimates the marginals $\mathbb{P}(X_e = v|Y_1 = y_1,\ldots,Y_T = y_T)$ and $\mathbb{P}(U_v = u|Y_1 = y_1,\ldots,Y_T = y_T)$, by iteratively exchanging messages across the variable and factor nodes. The messages are viewed as *beliefs* about that variable or distributions (a local estimate of $\mathbb{P}(\text{variable}|\text{observations})$). Since all random variables are binary, each message is a 2-dimensional vector.

We use the factor graph framework from [54] to compute the messages: Variable nodes $Y_\tau$ continually transmit the message $[0,1]$ if $Y_\tau = 1$ and $[1,0]$ if $Y_\tau = 0$ on its incident edge, at every iteration. Each other variable node ($X_e$ and $U_v$) uses the following rule: for incident each edge $e$, the node computes the elementwise product of the messages from every other incident edge $e'$ and transmits this along $e$. For the factor node messages, we derive closed-form expressions for the sum-product update rules (akin to equation (6) in [54]). The exact messages are described in appendix D.

## VIII. Numerical Evaluation

### A. Non-Overlapping Case

**Experimental setup I: Symmetric.** In our simulations, we consider 2 different use cases about the community structure: (Community 1) a neighborhood with $F = 200$ families of $M = 5$ members each, and (Community 2) a university department with $F = 20$ classes of $M = 50$ students each. In each use case, we also examine 2 different infection regimes: (a) a high-infection regime, where $\bar{k}/n = 0.1$; and (b) a low-infection regime, where $\bar{k} = \sqrt{n} = 32$. Finally, we consider both noiseless tests that have perfect accuracy and noisy tests that follow the Z-channel model from Section III-C. For each scenario, we average over 500 randomly generated community structures, in which the members/students are infected according to the symmetric probabilistic model (II): first a family/class is chosen at random w.p. $q$ to be infected and then each of its members/students gets randomly infected w.p. $p$.

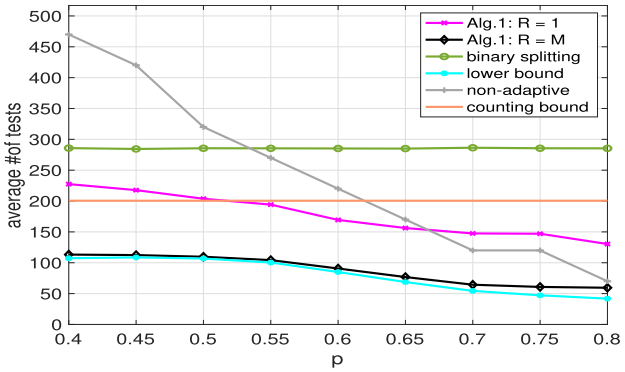



Fig. 4. Noiseless non-overlapping case: Average number of tests.

**Results.** For brevity, we show only the low-infection regime—complete results are in [1, App. E].

(i) *Noiseless testing – Average number of tests:* In this experiment, we measure the average number of tests needed by 3 algorithms that achieve zero-error reconstruction (Algorithm 1 with $R = 1$, $R = M$, and classic BSA), and a nonadaptive algorithm (Section V-B) that uses $T_1 = F$ tests for $\mathbf{G}_1$ and has FP rate around $0.5\%$. Algorithm 1 assumes no prior knowledge of the number of infected families/classes or members/students, hence uses $AdaptiveTest() = \text{BSA}$.

Fig. 4 depicts our results about Community 2 and for $p \in [0.4, 0.8]$. Both versions of Algorithm 1 need significantly fewer tests compared to classic BSA, while staying below the counting bound. This indicates the potential benefits from the community structure, even when the number of infected members is unknown. More interestingly, when $R = M$, Algorithm 1 performs close to the lower bound in most realistic scenarios $p \in [0.5, 0.8]$ (as also explained in Section V-A). The relevant result in the high-infection regime, was slightly worse: 50-70 tests above the lower bound. Last, the grey line shows the tests needed by our nonadaptive algorithm; even that algorithm can perform better than BSA, when $p > 0.55$ and small FP rates are tolerated.

(ii) *Noiseless testing – Average error rate:* We here quantify the additional cost in terms of error rate, when one goes from a two-stage adaptive algorithm that achieves zero-error identification to much faster single-stage nonadaptive algorithms. In each run, we first run a two-stage algorithm (that is similar to Algorithm 1, but uses a classic constant-column-weight, non-adaptive test design at each part, i.e., in the place of $AdaptiveTest()$ at lines 4 and 13) and we measure the number of tests it requires to achieve zero errors. Then, we use the *same* number of tests to infer the members' infection status through 2 nonadaptive algorithms that account for the community structure either at the test matrix (encoding) part or the decoding and a traditional one that does not consider it at all: "COMP with C-encoder" is our nonadaptive algorithm that uses a COMP decoder as described in Section V-B; "C-LBP with NC-encoder" is an algorithm that uses classic constant-column-weight test design combined with our LBP

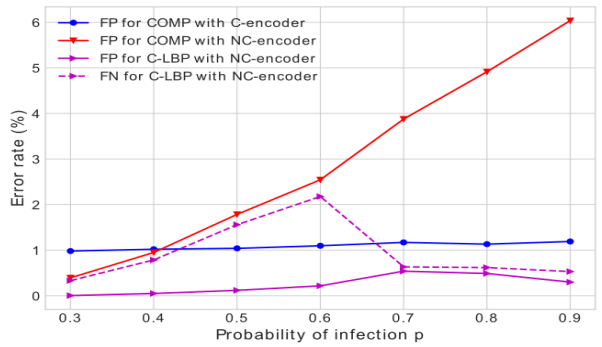



Fig. 5. Noiseless non-overlapping case: Average error rate (fixed $T$).

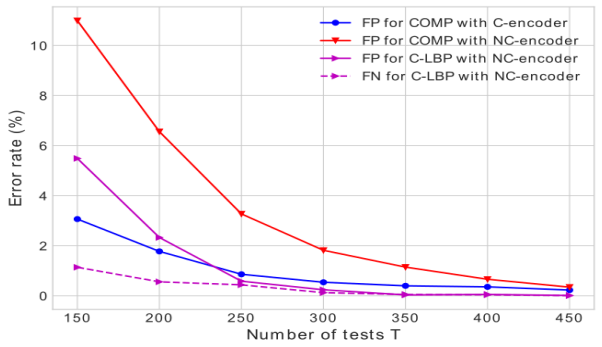



Fig. 6. Noiseless non-overlapping case: Average error rate ($p = 0.6$).

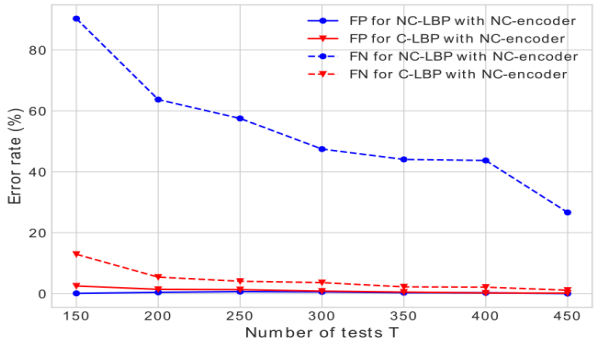



Fig. 7. Noisy non-overlapping case: Average error rate ($p = 0.8$).

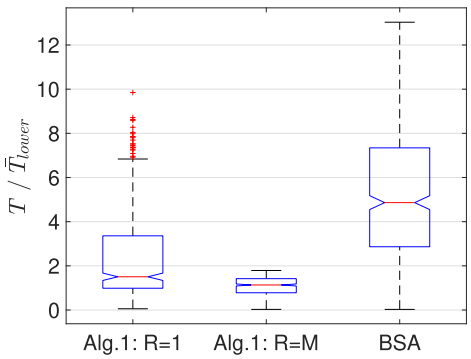



Fig. 8. Non-overlapping asymmetric case: Ratio of the number of tests needed to the lower bound (4).

decoder form Section VII; and "COMP with NC-encoder" is a traditional nonadaptive algorithm, that we use as a benchmark and uses a constant-column-weight test matrix with a COMP decoder. "C" denotes that the community is taken into account, while "NC" denotes that it is ignored. It is important to note that the number of tests needed by the two-stage algorithm (and therefore all other algorithms) gets lower as $p$ gets large, something that affects the results (as discussed further below).

Fig. 5 depicts the FP and FN error rates (averaged over 500 runs) as a function of $p \in [0.3, 0.9]$ for Community 1. FN rate is the percentage of *infected* individuals identified as negative and vice versa for FP. We observe that any community-aware nonadaptive algorithm performs better than traditional nonadaptive group testing (red line) when $p > 0.4$—the absolute performance gap ranges from $0.4\%$ (when $p = 0.3$) to $5.5\%$ (when $p = 0.9$). "COMP with C-encoder" has a stable FP rate across for all $p$ values that was close to $1\%$, and a zero FN rate by construction. Our LBP decoder, may yield both FN and FP errors. Also, being an approximate inference algorithm, it may produce worse results than COMP when $p \in [0.42, 0.67]$, but performs better when the infection rate is higher.

Fig. 6 examines the effect of the number of tests. Starting from the average number of tests used by the two stage algorithm when $p = 0.6$, we compute the FP and FN rates for larger numbers of tests. Our experiment shows a transition around $T = 240$, after which point "C-LBP with NC-encoder" performs better than "COMP with C-encoder". In fact, "COMP with C-encoder" seems to converge to zero FP errors much slower. This result was common for other $p$ values, the transition just occurred at different $T$. We thus conclude that one may use our "COMP with C-encoder" if the available tests are limited or if they just want to use a simple decoder; otherwise if the testing budget is larger, one would prefer "C-LBP with NC-encoder".

(iii) *Noisy testing:* Assuming the Z-channel[1] noise with parameter $z = 0.15$, we evaluate the performance of our community-based LBP decoder of Section VII against a LBP that does not account for community—namely its factor graph has no $X_e$ nodes. Fig. 7 depicts our results for Community 1 and for a selected $p = 0.8$ and a number of tests as given from the two-stage algorithm of the previous experiments. We observe that the knowledge of the community structure (in C-LBP) reduces both FP and FN rates achieved community-unaware NC-LBP. Especially, FN error rates drop significantly (up to $80\%$ when tests are few), which is important in our context since FN errors lead to further infections. Our results were similar for other $p$ values.

**Experimental setup II: Asymmetric.** In our asymmetric setup, infections follow again the probabilistic model (II), but this time for each community $e$, $|\mathcal{V}_e|$ and $p_e$ are selected uniformly at random from the intervals $[5, 50]$ and $[0.4, 0.8]$, respectively. Fig. 8 is a boxplot depicting our results for the low-infection regime ($q = 3\%$) over 500 random instances, generated as described above. The middle line of each box represents the mean, the edges represent the lower and upper

[1]In a Z-channel noise model, a test output that should be positive, flips and appears as negative with probability $z$, while a test output that is negative cannot flip. Thus: $\mathbb{P}(Y_\tau = 1 | U_{\delta_\tau}) = \left(\bigvee_{i \in \delta_\tau} U_v\right)(1 - z)$.

quartile, and the crosses represent outlier points. BSA needs on average $5.23\times$ (that can reach up to $13\times$) more tests compared to the probabilistic bound, while the two versions of Algorithm 1 with $R = 1$ and $R = M$ need only $2.4\times$ and $1.11\times$ (that can reach up to $9.85\times$ and $1.8\times$) more tests, respectively. Also, the smaller range between the 25-th and 75-th percentiles for Algorithm 1 indicates a more predictable performance compared to BSA.

### B. General (Overlapping) Case

**Experimental setup.** We generate 100 random community structures, each having $n = 3000$ members participating in about 200 overlapping communities, by using the following rules: the size of each community is selected uniformly at random from the range $[15, 25]$, and each member is randomly allocated in at most 4 communities with a probability that decreases exponentially with the number of communities (such that eventually, most members belong to a single community and fewer member belong to more communities). Then, the members become infected according to the probabilistic model (II): each community $e$ gets infected w.p. $q = 0.05$; and if infected, then its infection rate $p_e$ is randomly chosen from the interval $[0.3, 0.9]$. We remark that our experimental setup yields a high-infection infection regime; the fraction of infected members is about 5%. We preferred such a setup in order to stress the performance of our algorithms, as group testing generally shows less benefits in such regimes.

For the adaptive algorithms, we compare: the binary splitting algorithm (BSA) [11], which is the best traditional alternative when the number of infections is unknown; Algorithm 1 that assumes non-overlapping communities; and Algorithm 2 (with $AdaptiveTest() = \text{BSA}$).

For the non-adaptive test matrix designs, we compare: $G_1G_2$, our proposed test design in Section VI-B; and *CCW*, constant-column-weight algorithms, where each item is included in a fixed number $w$ of tests selected uniformly at random. $w$ is assumed to be of the form $w = \alpha\frac{T}{k}$, where $k$ is an estimate of the number of defectives in the population. We exhaustively search to find the best value of $\alpha \in [0, 1]$. We also compare LBP and COMP decoding: *C-LBP* is our proposed algorithm in Section VII, that accounts for the community structure. *NC-LBP*, does not take into account the community structure, i.e., assumes that each individual is i.i.d infected with the same probability $p_{iid}$. *COMP*, described in [11], has a zero FN probability by design.

**Results.** $(i)$ *Adaptive test designs.* For each community structure, we measured the number of tests needed by each algorithm to achieve zero-error identification. Since Algorithm 2 depends on $\theta$, the threshold used at line 16, we evaluated its performance for various values of $\theta$. Figure 9 depicts the average performance of our algorithm (for each $\theta$, we average over 100 randomly generated structures). Algorithm 2 was proved resilient to the choice of $\theta$ and needed on average $> 55\%$ fewer tests than the other algorithms. Its performance was also better than the counting bound, which is our best hope with traditional group testing. Our findings were similar for sparser infection regimes (see results in extended version [27]),

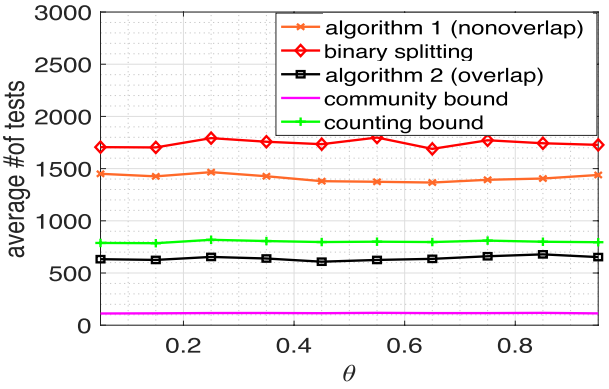



Fig. 9. General case: Average number of tests ($n = 3000$, $F \sim \text{Uniform}[15, 25]$, $q = 0.05$, $p_e \sim \text{Uniform}[0.3, 0.9]$.

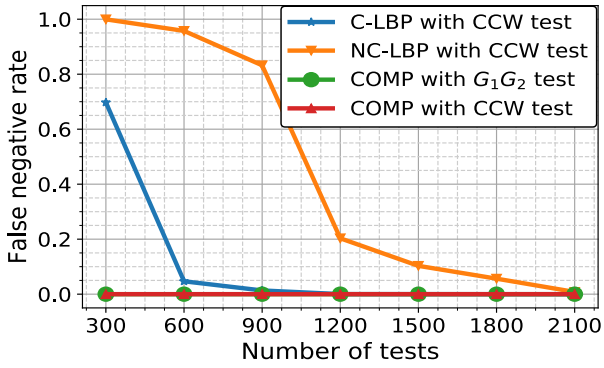



Fig. 10. General case: average FN rate of various non-adaptive test designs.

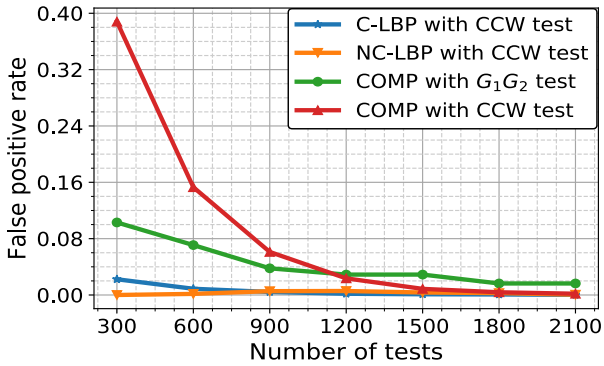



Fig. 11. General case: average FP rate of various non-adaptive test designs.

and there were cases where our algorithm performed closer to the community bound (1).

$(ii)$ *Non adaptive test designs.* In our experiments, we measured the FN/FP rates achieved by the non-adaptive test designs and the corresponding decoders. Fig. 10 and Fig. 11 depict FN and FP rates as a function of $T \in [300, 2100]$, respectively. The key takeaways are:

- C-LBP with CCW attains zero FP and FN at 1200 tests while COMP and NC-LBP with CCW (which are community-agnostic) attain zero FP and FN only at 1800 and 2100 tests respectively. This illustrates potential benefits of making the decoder aware of the community structure.
- If we desire a zero FN rate (or if we would like to use a simple decoder) and we are constrained to use less than 1000 tests, the $G_1G_2$ test design with COMP gives the lowest

FP rates. This illustrates the benefit of designing tests matrices that take into account the community structure.

## IX. Conclusion

The new observation we make in this paper is that taking into account infection correlations, as dictated by a known community structure, enables to reduce the number of group tests required to identify the infected members of a population and can improve the identification accuracy when the number of tests is fixed. We make this point assuming an overlapping community structure (where an individual belongs to one or more communities, and the infection probability depends on the infected communities (s)he participates in), a specific noise model and binary group testing. We considered a combinatorial and probabilistic model, derived lower bounds on the number of tests needed, explored adaptive, two-stage and non-adaptive algorithms for the noiseless case, and we evaluated our algorithms for the noisy case. Our algorithms are not always optimal w.r.t. the lower bounds, but perform significantly better than community-agnostic group testing; per our experiments, they need up to $30-75\%$ fewer tests (on average) to achieve the same identification accuracy. We posit that such benefits are possible in a number of other noise or group test models. Understanding what are benefits in more sophisticated community models remains as an open question.

## Appendix A
## The Lower Bound

### A. Proof of Theorem 1

*Proof:* There exist $\binom{F}{k_f}\cdot\prod_{C\in\mathcal{G}}\prod_{d\in D_C}\binom{|\mathcal{V}_d|}{k_m^d}$ possible combinations of infected members. This is because there are $\binom{F}{k_f}$ combinations of infected communities, each of which has the same chance of occurring, and is associated with a structure of connected components. In each disjoint set $d\in D_C$ of every connected component $C\in\mathcal{G}$, there are $\binom{|\mathcal{V}_d|}{k_m^d}$ possible combinations of infected members, each of which has the same chance of occurring.[2]

To achieve zero-error identification, each combination of infected members must give a different set of test results. Given that there are only $2^T$ possible results, we need: $2^T\geq\binom{F}{k_f}\cdot\prod_{C\in\mathcal{G}}\prod_{d\in D_C}\binom{|\mathcal{V}_d|}{k_m^d}$, which completes the proof.

Similarly, in the non-overlapping case, there are $\binom{F}{k_f}\cdot\prod_{e=1}^{k_f}\binom{|\mathcal{V}_e|}{k_m^e}$ possible sets of infected members that each must give a different set of results. Thus, $2^T\geq\binom{F}{k_f}\cdot\prod_{e=1}^{k_f}\binom{|\mathcal{V}_e|}{k_m^e}$. □

### B. Proof of Theorem 2

*Proof:* Let $\mathbf{X}$ be the indicator random vector for the infection status of all communities. By rephrasing [41, Theorem 1], any probabilistic group testing algorithm using $T$ noiseless tests can achieve reconstruction of $\mathbf{U}$ with success probability at least $\mathbb{P}_{\text{suc}}$, only if $T\geq\mathbb{P}_{\text{suc}}H(\mathbf{U})$. For the entropy term, we have:

$$H(\mathbf{U})=H(\mathbf{X})+H(\mathbf{U}|\mathbf{X})-H(\mathbf{X}|\mathbf{U}). \tag{A.1}$$

The first term is: $H(\mathbf{X})=\sum_{e=1}^{F}H(X_e)=Fh_2(q)$.

The second term is calculated as:

$$\begin{aligned}H(\mathbf{U}|\mathbf{X})&\overset{(a)}{=}\sum_{v=1}^{n}H(U_v|\mathbf{X}_{\mathcal{E}_v})\\&=\sum_{v=1}^{n}\sum_{\mathbf{x}\in\{0,1\}^{|\mathcal{E}_v|}}\mathbb{P}(\mathbf{X}_{\mathcal{E}_v}=\mathbf{x})H(U_v|\mathbf{X}_{\mathcal{E}_v}=\mathbf{x})\\&\overset{(b)}{=}\sum_{v=1}^{n}\sum_{\mathcal{I}\subseteq\mathcal{E}_v}q^{|\mathcal{I}|}(1-q)^{|\mathcal{E}_v|-|\mathcal{I}|}\\&\qquad\cdot H(U_v|\mathbf{X}_{\mathcal{I}}=\mathbf{1},\mathbf{X}_{\mathcal{E}_v\setminus\mathcal{I}}=\mathbf{0})\\&=\sum_{v=1}^{n}\sum_{\mathcal{I}\subseteq\mathcal{E}_v}q^{|\mathcal{I}|}(1-q)^{|\mathcal{E}_v|-|\mathcal{I}|}\\&\qquad\cdot h_2\Big(\prod_{e\in\mathcal{I}}(1-p_e)\Big),\end{aligned}$$

where in $(a)$, $\mathcal{E}_v$ refers to the set of communities member $v$ belongs to, and in $(b)$ the subset $\mathcal{I}$ is the subset of infected communities in $\mathcal{E}_v$.

Finally, we upper bound the third term as:

$$\begin{aligned}H(\mathbf{X}|\mathbf{U})&\leq\sum_{e=1}^{F}H(X_e|\mathbf{U})\leq\sum_{e=1}^{F}H(X_e|\mathbf{U}_{\mathcal{S}_e})\\&=\sum_{e=1}^{F}\mathbb{P}(\mathbf{U}_{S_e}=\mathbf{0})h_2(\mathbb{P}(X_e=0|\mathbf{U}_{S_e}=\mathbf{0}))\\&=\sum_{e=1}^{F}(1-q+q(1-p_e)^{|S_e|})\\&\qquad\cdot h_2\left(\frac{1-q}{1-q+q(1-p_e)^{|S_e|}}\right),\end{aligned}$$

where $S_e$ is the set of members who *only* belong to community $e$. Combining all the 3 terms concludes the proof for the general overlapping case.

In the non-overlapping case, the second term can be expressed more concisely, while the third term can be computed exactly.

The second term is calculated as:

$$\begin{aligned}H(\mathbf{U}|\mathbf{X})&=\sum_{v=1}^{n}H(U_v|X_{\mathcal{E}_v})\\&=\sum_{v=1}^{n}\sum_{x\in\{0,1\}}\mathbb{P}(X_{\mathcal{E}_v}=x)H(U_v|X_{\mathcal{E}_v}=x)\\&=\sum_{v=1}^{n}\left(qH(U_v|X_{\mathcal{E}_v}=1)+(1-q)H(U_v|X_{\mathcal{E}_v}=0)\right)\\&=\sum_{v=1}^{n}qh_2(p_{\mathcal{E}_v})=q\sum_{e=1}^{F}|\mathcal{V}_e|h_2(p_e),\end{aligned}$$

where $\mathcal{E}_v$ is the community containing vertex $v$.

[2]Note that the product is over all disjoint sets instead of only the infected ones, because $\binom{|\mathcal{V}_d|}{k_m^d}=1$, whenever $k_m^d=0$.

The third term is:

$$\begin{aligned} H(\mathbf{X}|\mathbf{U}) &= \sum_{e=1}^{F} H(X_e|\mathbf{U}) = \sum_{e=1}^{F} H(X_e|\mathbf{U}_{\mathcal{V}_e}) \\ &= \sum_{e=1}^{F} \mathbb{P}(\mathbf{U}_{\mathcal{V}_e} = \mathbf{0}) h_2(\mathbb{P}(X_e = 0|\mathbf{U}_{\mathcal{V}_e} = \mathbf{0})) \\ &= \sum_{e=1}^{F} (1 - q + q(1-p_e)^{|\mathcal{V}_e|}) \\ &\quad \cdot h_2\left(\frac{1-q}{1-q+q(1-p_e)^{|\mathcal{V}_e|}}\right) \end{aligned}$$

Combining all the 3 terms concludes the proof. □

## APPENDIX B
## NON-OVERLAPPING CASE

### A. Proof of Lemma 1

*Proof:* We start from the known performance guarantees about HGBSA and BSA: Given a problem with $K$ infected items in a population of size $N$, HGBSA is guaranteed to succeed using $T = K \log_2 \frac{N}{K} + K$ tests, while BSA is guranteed to succeed using $T \leq K \log_2 N + K$ [11], [12], [51].

Algorithm 1 performs testing at lines 4, 8, 13. Let the expected numbers of tests at these lines be $\bar{T}_4$, $\bar{T}_8$ and $\bar{T}_{13}$, respectively.

• At line 4, let $K_4$ be the number of communities whose mixed sample is positive (that is the number of "infected" items in the population of $F$ communities).

If HGBSA is used for $AdaptiveTest()$, we have:

$$\bar{T}_4 = \mathbb{E}\left[K_4 \log_2 \frac{F}{K_4} + K_4\right] \leq \mathbb{E}[K_4]\left(\log_2 \frac{F}{\mathbb{E}[K_4]} + 1\right), \tag{B.1}$$

where the inequality holds because of Jensen's inequality, as $f(x) = x\left(\log_2 \frac{F}{x} + 1\right)$ is a concave function of $x$ (its second derivative is $f''(x) = -\frac{1}{\ln(2)\, x} \leq 0$).

Similarly, if BSA is used for $AdaptiveTest()$, then:

$$\bar{T}_4 \leq \mathbb{E}[K_4 \log_2 F + K_4] = \mathbb{E}[K_4](\log_2 F + 1), \tag{B.2}$$

• At line 8, the expected number of individual tests is:

$$\bar{T}_8 = \mathbb{E}[K_4 \cdot M] = M\mathbb{E}[K_4], \tag{B.3}$$

regardless of whether $AdaptiveTest()$ is HGBSA or BSA.

• At line 13, let the number of infected individuals and the population size be $K_{13}$ and $N_{13}$, respectively.

If HSBSA is used for $AdaptiveTest$:

$$\begin{aligned} \bar{T}_{13} &= \mathbb{E}\left[K_{13} \log_2 \frac{N_{13}}{K_{13}} + K_{13}\right] \\ &\overset{(a)}{=} \mathbb{E}\left[(k - K_4 k_m)\left(\log_2 \frac{n - K_4 M}{k - K_4 k_m} + 1\right)\right] \\ &\overset{(b)}{\leq} (k - \mathbb{E}[K_4] k_m)\left(\log_2 \frac{n - \mathbb{E}[K_4] M}{k - \mathbb{E}[K_4] k_m} + 1\right), \end{aligned} \tag{B.4}$$

where (a) is because $K_{13} = k - K_4 k_m$ and $N_{13} = n - K_4 M$, and (b) holds because of Jensen's inequality, as $f(x) = (k - x k_m)\left(\log_2 \frac{n - xM}{k - x k_m} + 1\right)$ is a concave function of $x$ for $x \geq 0$ and $x \leq k_f \Leftrightarrow k_m x \leq k$ (the second derivative is $f''(x) = -\frac{(k_m n - Mk)^2}{\ln(2)(n - Mx)^2 (k - x k_m)} \leq 0$).

If BSA is used for $AdaptiveTest$:

$$\begin{aligned} \bar{T}_{13} &\leq \mathbb{E}[K_{13} \log_2 N_{13} + K_{13}] \\ &\overset{(a)}{=} \mathbb{E}\left[(k - K_4 k_m)\left(\log_2 (n - K_4 M) + 1\right)\right] \\ &\overset{(b)}{\leq} (k - \mathbb{E}[K_4] k_m)\left(\log_2 n + 1\right) \end{aligned} \tag{B.5}$$

where (a) is because $K_{13} = k - K_4 k_m$ and $N_{13} = n - K_4 M$, and in (b) $(n - K_4 M)$ is upper-bounded by $n$.

We now compute $\mathbb{E}[K_4]$ as follows: Let $\phi_c$ be the expected fraction of infected communities whose mixed sample is positive. Since $SelectRepresentatives()$ is uniform random sampling without replacement, we can compute $\phi_c$ when $1 \leq R \leq M - k_m$ using the hypergeometric distribution $Hyper(M, k_m, R)$, as follows: the probability of a random mixed sample $x(r_e)$ being negative (i.e. all members of $r_e$ are negative) is given by the PMF of $Hyper(M, k_m, R)$ evaluated at 0, and it is therefore equal to $\binom{M-k_m}{R}/\binom{M}{R}$, which yields $\phi_c = 1 - \binom{M-k_m}{R}/\binom{M}{R}$. We also define the following for completeness: $\phi_c = 0$ when $R = 0$ and $\phi_c = 1$ when $M - k_m < R \leq M$. Thus:

$$\mathbb{E}[K_4] = k_f \phi_c. \tag{B.6}$$

To conclude, we add all the above terms ($\bar{T}_4$, $\bar{T}_8$, $\bar{T}_{13}$) that are related to HGBSA or BSA, by also taking into account that $k = k_f k_m$, and the result follows. □

### B. Proof of Lemma 2

*Proof:* Similarly to the proof of Lemma 1, let $\phi_p$ be the expected fraction of infected communities whose mixed sample is positive. Then, because of the probabilistic setting, $\phi_p = 1 - (1-p)^R$.

Algorithm 1 performs testing at lines 4, 8, 13.

• At line 4, let $K_4$ be the number of communities whose mixed sample is positive (that is the number of "infected" items in the population of $F$ communities).

If BSA is used for $AdaptiveTest()$, then:

$$\bar{T}_4 \leq \mathbb{E}[K_4 \log_2 F + K_4] = \mathbb{E}[K_4](\log_2 F + 1), \tag{B.7}$$

• At line 8, the expected number of individual tests is:

$$\bar{T}_8 = \mathbb{E}[K_4 \cdot M] = M\mathbb{E}[K_4]. \tag{B.8}$$

• At line 13, let the number of infected individuals and the population size be $K_{13}$ and $N_{13}$, respectively.

If BSA is used for $AdaptiveTest$:

$$\begin{aligned} \bar{T}_{13} &\leq \mathbb{E}[K_{13} \log_2 N_{13} + K_{13}] \\ &\overset{(a)}{=} \mathbb{E}\left[(k - K_4 k_m)\left(\log_2 (n - K_4 M) + 1\right)\right] \\ &\overset{(b)}{\leq} \mathbb{E}\left[(k_f - K_4) k_m\right]\left(\log_2 n + 1\right) \\ &= \left(\mathbb{E}[k_f] - \mathbb{E}[K_4]\right)\mathbb{E}\left[k_m\right]\left(\log_2 n + 1\right) \end{aligned} \tag{B.9}$$

where (a) is because $K_{13} = k - K_4 k_m$ and $N_{13} = n - K_4 M$, and in (b) $(n - K_4 M)$ is upper-bounded by $n$.

To obtain the result, we add all the above terms ($\bar{T}_4, \bar{T}_8, \bar{T}_{13}$) that are related to HGBSA or BSA, by also taking into account the following: $\mathbb{E}[k_f] = Fq$, $\mathbb{E}[K_4] = Fq\phi_p$, $\mathbb{E}[k_m] = Mp$, and $n = FM$. □

*A remark about upper-bounding* $(n - K_4M)$ *by* $n$ *in (B.5):* In the symmetric case (where $|\mathcal{V}_e| = M$ and $k_m^e = k_m$ for each community $e$), we practically consider two choices for the number of representatives, $R = M$ or $R = 0$ (for details, see observation (b) in Section V-A and the discussion about the selection of representatives in Section B-C). In these two cases, upper bounding $(n - K_4M)$ by $n$ does not lose too much: If $R = 0$, then $K_4 = 0$—hence, Algorithm 1 just reduces to classic BSA and at line 13 the size of the population under testing is indeed $n$. If $R = M$, then $K_4 = k_f$—hence, $\bar{T}_{13}$ becomes 0 regardless of the log term.

### C. Rationale for Algorithm 1

Group testing already has a rich body of literature with near-optimal test designs in the case of independent infections, we do not try to improve upon them. Instead, we adapt these ideas to incorporate the correlations arisen from the community structure. All test designs described in this section are conceptually divided into two parts. This split is guided by the community structure and attempts to identify the different infection regimes inside the community, so that the best testing method (individual or classic group testing) is used. We show that such a two-part design is enough to significantly reduce the cost of group testing and also achieve the lower bound in some cases.

**Two-part design:** Two parts of Algorithm 1 serve complimentary goals:

The goal of Part 1 is to detect the infection *regime* inside each community $e$: i.e., to accurately estimate which of the $F$ communities have a high infection rate ("heavily" infected) and which are have a low or zero infection rate ("lightly" infected). Our interest in detecting the infection regime is motivated by prior work [17], [18], which has shown that group testing offers benefits over individual testing, only if the infection rate is low ($p_e \leq 0.38$). This allows us to define the two regimes as follows: In the combinatorial model I (resp. probabilistic model II), a community is considered heavily infected when $k_m^e/|\mathcal{V}_e| \geq 0.38$ (resp. $p_e \geq 0.38$); conversely, it is considered lightly infected community when $k_m^e/|\mathcal{V}_e| < 0.38$ (resp. $p_e < 0.38$).

For each community $e$, we regard $\hat{U}_{x(r_e)}$ as an estimate of the community's infection regime. If $\hat{U}_{x(r_e)}$ is positive, we consider the community to be highly infected and therefore perform individual testing for all of its members. Otherwise, if $\hat{U}_{x(r_e)}$ is negative, we consider the community to be lightly infected and group test its members with all other lightly infected communities. The challenge is therefore to produce accurate enough regime estimates, such that the overall number of tests that are needed from Algorithm 1 to achieve exact infection-status reconstruction for all members $v = 1, \ldots, n$ is minimal. We discuss this challenge further below.

Given all estimates $\hat{U}_{x(r_e)}$ from Part 1, the goal of the Part 2 is then to identify all infected members, by using the appropriate testing method (group or individual testing) according to the infection regime of each community (light or heavy). In this way, at the end of Part 2, the algorithm returns an estimate $\hat{U}_v$ of the true infection status $U_v$ of each individual member $v$.

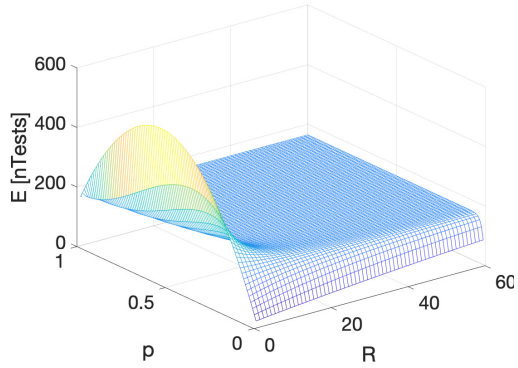



Fig. 12. Expected number of tests from (7) as a function of size of representative set and probability of infection inside a community.

**Selection of community representatives:** Function $SelectRepresentatives()$ at line 2 refers to *any* sampling function on a set of community members, as long as it returns a fixed number of members from community $e$. That is, one may use their own sampling function, as long as the accuracy of Part 1 is well defined. In this paper, we consider only random-sampling functions without replacement (i.e. $|r_e|$ members are randomly chosen from the community members and each subset of that size has the same probability of being selected as the representative subset). But perhaps, more elaborate sampling functions may be considered in other contexts. For example, if the internal structure of community $e$ can be represented through a contact graph, in which only specific community nodes have external contacts with other communities, it may make sense to include (some of) these nodes into the representative group with certainty.

When only one mixed sample per community is used to identify the heavily/lightly infected communities, the cardinality of the representative subset $|r_e|$ is essential, but the optimal choice of it is not trivial. $|r_e|$ affects the accuracy of regime estimate—hence the performance of our algorithm in terms of the expected number of tests that it uses. Unfortunately, choosing the number of representatives optimally is not easy even in the symmetric case that is examined in Section V-A. Ideally, in the symmetric case, we would like to choose $|r_e| = R$ such that the bounds in Lemmas 1 and 2 are minimized. However, this requires solving equations of the form $ye^y = x$, which is generally possible through Lambert functions for $x \geq -\frac{1}{e}$, but the latter does not hold in our case. Fig. 12 demonstrates that there exists no unique $R$ that is optimal for any infection probability $p$ in $(0, 1)$ through an example of $F = 50$ communities with $M = 60$ members each. The figure plots the bound of Lemma 2 as a function of $p$ and $R$. As we can see, there is no single minimizer $R^\star$: if $p < 0.15$, then $R$ must be picked equal to 0 (which yields traditional group testing); otherwise, if $p > 0.15$, then $R$ must be selected equal to $M$.

Therefore, in order to optimally choose $R$, a rough estimate about $p$ has to be known a priori. If the latter is not possible, then one may use a few more tests at the first stage of our algorithm to better detect whether a community is heavily infected. We provide such an optimization in the next section.

**Function** $AdaptiveTest()$**:** In both parts of our algorithm, we make use of a classic adaptive-group-testing algorithm, which we call $AdaptiveTest()$. This may be regarded as an abstraction for any existing (or future) adaptive algorithm in the group-testing literature. In our analysis, however, we mostly focus on the classic binary splitting algorithm because of its good performance in realistic cases, where the numbers of infected communities and/or members ($k_f$, $k_m^e$) are unknown [22].

In this section, we consider only adaptive algorithms that offer noiseless (zero-error) reconstruction. Note, however, the fact that $AdaptiveTest()$ offers exact reconstruction is not enough to guarantee an accurate detection of any community's infection regime in Part 1. For example, consider the following case, where the true infection rate within a community $e$ is not very low (say $p_e = 0.6$), yet none of the community representative in set $r_e$ happened to be infected. Intuitively, the error probability of detection in Part 1 should depend on the number of selected representatives $|r_e|$ from each community $e$ and the infection rate among its members $p_e$. In our analysis, we examine different scenaria w.r.t. these parameters and discuss which parametrization (i.e. value of $|r_e|$) optimizes the expected number of the tests required by our algorithm.

### D. Modified/Optimized Versions of Algorithm 1

• One modification of our algorithm is the following: In Part 1, instead of selecting only one representative group for each community, we select $m_s$ representative subgroups, each of size $s$, and we treat each of these subgroups as a single "(super)-member". That is, we identify whether each subgroup is positive (has at least one positive member) or not, and based on this information, using for example majority vote, we can classify the community as heavily or lightly infected; essentially we can solve an estimation problem as in [11] (see Chapter 5.3), [56], [57]. In this regard, Algorithm 1 is just a special case of this approach, with $m_s = 1$ and $s = |r_e|$.

Intuitively, we expect that such a modification would increase the estimation accuracy of $\hat{p_e}$ and reduce the error of the related hypothesis test, at the cost of few more tests. As a result, it could need fewer tests on expectation than Algorithm 1, hence perform better in some cases. However, the potential improvement would depend on parameters such as the community size - for instance for small size communities it is not expected to be large. To keep things simple, we prefer not to analyze this algorithm in this paper and defer it to future work.

• Another modification could be the following: instead of leveraging the community structure to perform individual tests where needed, we could use it to improve traditional binary splitting algorithm by running it on multiple testing groups that are related to the community structure. For example, consider a symmetric case where: we split all $n = FM$ members into $M$ groups of $F$ individuals (one from each community), then run binary splitting to each of these groups.

This modification is also related to Hwang's Generalized Binary Splitting (HGBSA), but achieves only logarithmic benefits compared to binary splitting, as opposed to our algorithm that may perform much better in real cases (see Section V-A). In fact, the expected number of tests needed by this modified algorithm would be at most $k \log_2 ({}^n/_M) + O(k)$: each group $g$ has $k_g$ infected member and binary splitting needs $k_g \log_2 ({}^n/_M) + O(k_g)$ tests to identify all of them. By adding together the number of tests for each group $g$, we deduce the result.

• A last modification occurred to us after a related comment of one of our reviewers, who we thank. As discussed in Section V-A, even when a sparse regime holds for communities (i.e. $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0, 1)$) and a heavily linear regime holds within each community (i.e. $k_m \approx M$, the benefits of Lemma 1 with regard to HGBSA cannot be more than $1/\log(n/k)$. This is because, in the first term of Eq. (5) we get an additive term $k_f \phi_c M \geq k_f \phi_c k_m = k\phi_c$, and in the second term, we get another additive term that is no less than $k(1-\phi_c)$. Thus, the ratio of the expected number of tests of Algorithm 1 over the expected number of tests of HGBSA is no less than $1/\log(n/k)$.

Nevertheless, in certain cases where $k_m \gg M - k_m$ (i.e., the infection regime inside each infected community[3] is heavy), in the second part of our algorithm it makes more sense to look for non-infected members and stop testing once we find them all. We next show that this results in higher benefits than $1/\log(n/k)$.

Consider a community of $M$ members, with $k_m$ infected and $M - k_m$ non-infected members, and suppose we test members individually until we find a fixed number $r = M - k_m$ of non-infected members. The random variable $K$—i.e., the number of infected items in the tests—follows a negative hypergeometric distribution, and therefore the mean of $K+r$—i.e., the expected number of tests until we find $r = M - k_m$ non-infected members—is equal to $(M - k_m)(\frac{k_m}{M-k_m+1} + 1)$. Depending on the exact value of $k_m$, the latter can be less than $k_m$, as $k_m/M$ goes to 1. As a result, the overall expected number of tests in the second part of our modified adaptive algorithm becomes less than $k_f \phi_c k_m = k_f k_m = k$ (if we further assume that all heavily infected communities are identified correctly in the first part—i.e., we use $R = M$ so that $\phi_c = 1$). Hence, the overall benefits compared to HGBSA can be better than $1/\log(n/k)$, in certain cases.

As an example, let $k_m = M - 1$. The expected number of individual tests needed to find the non-infected member inside each infected community is $(M + 1)/2$. Thus, the expected number of tests in the second part of our modified adaptive algorithm would be: $k_f(M + 1)/2 < k_f(M - 1) = k_f k_m = k$. Hence, in this particular regime, we could achieve higher benefits than $1/\log(n/k)$.

In the more extreme case, where for each infected community $k_m = M$, all we need to do is to identify the

[3]The symmetric case is used here only for illustration purposes; the idea is similar for the asymmetric case.

infected communities. In that case, the benefit would be approximately $k_f/k$.

We remark that for the above technique to work, the knowledge of the number of infected members per community is necessary, but this is also the case for HGBSA. Also, the offered benefits do get diminished as $k_m$ goes away from $M$.

### E. Rationale for the Structure of $\mathbf{G}_2$

Our goal is to design a non-trivial matrix $\mathbf{G}_2$ that can identify almost all the infected members with high probability and a small number of tests. Asking for zero-error requirements is not reasonable in our nonadaptive setting; e.g., in the probabilistic case (model II), every sparsity pattern is possible and thus zero-error recovery can be guaranteed for $\mathbf{G}_2$ only with a number of tests that is at least equal to the number of individuals. We next discuss two intuitive properties we would like our designs to have to minimize the error probability.

*Desirable Property 1:* Use identity matrices as building blocks.

*Intuition:* ideally, after removing the $(F-k_f)M$ columns corresponding to the members in non-infected communities, we would like the remaining columns to form an identity matrix so that we can identify all the infected members correctly. To reduce the number of tests, there should be more than one member included in each test. Thus we use overlapping identity matrices, one corresponding to each community. We assume the index for the $n$ members is community-by-community, i.e., the indices for the members in the same community are consecutive. Then each community corresponds to an identity sub-matrix $I_M$ in $\mathbf{G}_2$. Now the problem becomes how to arrange the identity sub-matrices.

*Desirable Property 2:* The identity matrices corresponding to different communities either appear in the same set of $M$ rows (i.e. block row) in $\mathbf{G}_2$ or they do not appear in any shared rows.

*Intuition:* Otherwise, a community would share tests with more other communities, and therefore, the probability that this community shares tests with infected communities would become larger. This would increase the probability that two infected communities share tests after removing all the non-infected community columns, which in turn would increase the FP probability.

### F. Proof of Lemma 3

*Proof:* The probabilities can be explained as follows:

(i) For $\mathbb{P}^{I}_{\text{covering}}$ in (8), the numerator gives the number of possibilities that each block row contains at most one infected community, which is obtained by randomly choosing $k_f$ block rows (the summation) and then from each chosen block row choosing one community to be infected ($c_i$ possible choices for $i$-th block row). The denominator is the total number of infection possibilities, and then the fraction denotes the probability that each block row contains at most one infected community. Thus, $\mathbb{P}^{I}_{\text{covering}}$ is obtained as the probability that there is some block row that contains two or more infected communities.

(ii) For $\mathbb{P}^{II}_{\text{covering}}$ in (9), $(1-q)^{c_i}$ is the probability that there is no infected community in the $i$-th block row, and $c_i q(1-q)^{c_i-1}$ is the probability that there is only one infected community in the $i$-th block row. The multiplication $\prod$ denotes the probability that any one block row contains at most one infected community. Thus, $\mathbb{P}^{II}_{\text{covering}}$ is obtained as the probability that there is some block row that contains two or more infected communities.

□

### G. Proof of Lemma 4

*Proof:* In this proof, we show that symmetric choice of $c_i$ (i.e. $c_i = c$, $\forall i \in \{1,\cdots,b\}$) is a minimizer of $\mathbb{P}^{I}_{\text{covering}}$ and $\mathbb{P}^{II}_{\text{covering}}$, even though it may not be unique.

Suppose that $c_i$'s are not symmetric, then there exist $i$ and $j$ such that: $c_i \geq c_j + 1$.

Let $c_i' = c_i - 1$ and $c_j' = c_j + 1$.

For the combinatorial model, using (8), $\mathbb{P}^{I}_{\text{covering}}(\{c_i\}) - \mathbb{P}^{I}_{\text{covering}}(\{c_i'\})$ is given by:

$$\sum_{\substack{|\mathcal{B}|=k_f:\\ \mathcal{B}\subseteq\{1,2,\cdots,b\}}} \prod_{\ell\in\mathcal{B}} c_\ell' - \sum_{\substack{|\mathcal{B}|=k_f:\\ \mathcal{B}\subseteq\{1,2,\cdots,b\}}} \prod_{\ell\in\mathcal{B}} c_\ell = \\ = (c_i' c_j' - c_i c_j)\cdot X \\ = (c_i - c_j - 1)\cdot X \geq 0,$$

where $X$ is a positive value independent of $c_i$ and $c_j$. By iterating this multiple times and across other pairs of such indices, this implies that the symmetric case where all $c_i$'s are equal, i.e., $c_i = c$ for all $i \in \{1,2,\cdots,b\}$ is a minimizer (not necessarily a unique one).

Similarly, using (9), the difference $\mathbb{P}^{II}_{\text{covering}}(\{c_i\}) - \mathbb{P}^{II}_{\text{covering}}(\{c_i'\})$ is given by:

$$\prod_{\ell=1}^{b}\left[(1-q)^{c_\ell'} + c_\ell' q(1-q)^{c_\ell'-1}\right] - \\ \prod_{\ell=1}^{b}\left[(1-q)^{c_\ell} + c_\ell q(1-q)^{c_\ell-1}\right] = \\ = [(c_i - c_j) - (1-q)^2] q^2 (1-q)^{c_i+c_j-2}\cdot Y > 0,$$

where $Y = \prod_{\ell\neq i,j}\left[(1-q)^{c_\ell} + c_\ell q(1-q)^{c_\ell-1}\right] \geq 0$ is independent of $c_i$ and $c_j$. By iterating this multiple times and across other pairs of such indices, this implies that the symmetric case where all $c_i$'s are equal, i.e., $c_i = c$ for all $i \in \{1,2,\cdots,b\}$ is a minimizer (not necessarily a unique one). □

### H. Proof of Lemma 5

*Proof:* In the symmetric case, i.e., $c_i = c$ for all $i \in \{1,2,\cdots,b\}$, the probabilities in (8) and (9) become

$$\mathbb{P}^{I}_{\text{covering}} = 1 - \frac{\binom{b}{k_f} c^{k_f}}{\binom{F}{k_f}}, \tag{B.10}$$

$$\mathbb{P}^{II}_{\text{covering}} = 1 - \left((1-q)^{c-1}(1-q+cq)\right)^{b}. \tag{B.11}$$

In the symmetric combinatorial model, all infected communities have the same number of infected members: $k_m^e = k_m,\ \forall e \in \mathcal{E}$. If two communities appear in the same set of $M$ tests, the probability that all infected members of one community share the same $k_m$ tests with the infected members of the other community (i.e., no false positive occurs) is simply:

$$\mathbb{P}(\text{no FP}|\text{covering}) = \frac{1}{\binom{M}{k_m}}. \tag{B.12}$$

Thus the probability that FPs happen is:

$$\begin{aligned}\mathbb{P}_{\text{FP}} &= \mathbb{P}(\text{FP}|\text{covering}) \cdot \mathbb{P}^{I}_{\text{covering}} \\ &= \left[1 - \frac{1}{\binom{M}{k_m}}\right]\left[1 - \frac{\binom{b}{k_f} c^{k_f}}{\binom{F}{k_f}}\right].\end{aligned} \tag{B.13}$$

In the symmetric probabilistic model, all infected communities have the same infection probability: $p_e = p,\ \forall e \in \mathcal{E}$. If two communities appear in the same set of $M$ tests, then there are no false positives, only if the two communities have the same number of infected members and the infected (non-infected) members in one community appear in the same set of tests as infected (non-infected) members of the other community. The probability that two communities both have $i$ infected members is $\left[p^i(1-p)^{M-i}\right]^2$, and the probability that all infected members in one community share tests with only infected members in the other community is simply $\frac{1}{\binom{M}{i}}$. Thus, the probability that there are no false positives is given by:

$$\mathbb{P}(\text{no FP}|\text{covering}) = \sum_{i=1}^{M} \left[p^i(1-p)^{M-i}\right]^2 \frac{1}{\binom{M}{i}}. \tag{B.14}$$

Therefore, the probability that a false positive happens can be obtained as

$$\begin{aligned}\mathbb{P}_{\text{FP}} &= \mathbb{P}(\text{FP}|\text{covering}) \cdot \mathbb{P}^{II}_{\text{covering}} \\ &= \left[\sum_{i=1}^{M} \left[p^i(1-p)^{M-i}\right]^2 \frac{1}{\binom{M}{i}}\right] \\ &\quad \cdot \left[1 - \left((1-q)^{c-1}(1-q+cq)\right)^b\right].\end{aligned} \tag{B.15}$$

Replacing $b$ by $T_2/M$ and $c$ by $FM/T_2$ completes the result. □

### I. Proof of Lemma 6 and Discussions

*Proof:* For the combinatorial model (I), it is hard to explicitly calculate the expected error rate. The upper bound in (11) is obtained by assuming that if there exist errors (FPs), then all non-infected members in infected communities are misidentified as infected in the decoding of $\mathbf{G}_2$. (Note that all non-infected members in non-infected communities are correctly identified by decoding of $\mathbf{G}_1$.)

For the probabilistic model (II), the upper bound for the expected error rate in (13) is obtained by

$$\begin{aligned}R_{II}(\text{error}) = \frac{1}{n} \cdot b \cdot \Bigg[\sum_{j=2}^{c} \binom{c}{j} q^j (1-q)^{c-j} \\ \cdot \left(\sum_{i=1}^{j} \binom{j}{i} p^i (1-p)^{j-i}(j-i)\right) \cdot M\Bigg]\end{aligned} \tag{B.16}$$

$$\begin{aligned}= \frac{bM}{n} \cdot \Bigg[\sum_{j=2}^{c} \binom{c}{j} q^j (1-q)^{c-j} \\ \cdot \big(j(1-p) - j(1-p)^j\big)\Bigg]\end{aligned} \tag{B.17}$$

$$\begin{aligned}&< \frac{(1-p)\,T_2}{n} \cdot \left[\sum_{j=2}^{c} \binom{c}{j} q^j (1-q)^{c-j} \cdot j\right] \\ &= \frac{(1-p)\,T_2}{n} \cdot \left[cq - cq(1-q)^{c-1}\right], \\ &= (1-p)q\left[1 - (1-q)^{c-1}\right],\end{aligned} \tag{B.18}$$

where the expression in the bracket in (B.16) for each $j$ denotes the expected number of FPs in one block row if there are $j$ communities infected in this block row, (B.17) is obtained from the expected value of binomial distribution, and (B.18) follows by substituting $c = \frac{n}{T_2}$. □

We here make the following observation about the system FP probability $\mathbb{P}$(any-FP): As we explore further in Section VIII-A, non-adaptive group testing requires more tests than adaptive. Assume that $k_f = \Theta(F^{\alpha_f})$ for $\alpha_f \in [0, 1)$ and choose $R = M - 1$ in Algorithm 1. Adaptive testing allows to achieve zero error with $k_f \log_2 F + k_f M$ tests; if we use the same (order) number of tests with a non-adaptive strategy, i.e., $T_1 = k_f \log_2 \frac{F}{k_f}$ and $T_2 = k_f (\log_2 k_f + M)$, we get $\mathbb{P}$(any-FP) in Lemma 5 approximately equal to $\left(1 - \frac{1}{M}\right)\left[1 - \frac{\binom{T_2/M}{k_f}}{\left(\frac{T_2/M}{k_f}\right)^{k_f}} \frac{(F/k_f)^{k_f}}{\binom{F}{k_f}}\right]$ which is bounded away from 0. The latter can be seen as follows: i) $T_2/M \approx k_f \ll F$; ii) $\frac{\binom{n}{k}}{\left(\frac{n}{k}\right)^k} / \frac{\binom{n+m}{k}}{\left(\frac{n+m}{k}\right)^k} = \left(\frac{n}{n+m}\right)^k \cdot \prod_{i=1}^{m} \frac{n+i-k}{n+i}$ is decreasing with $m$ and can be very small when $m \gg n$.

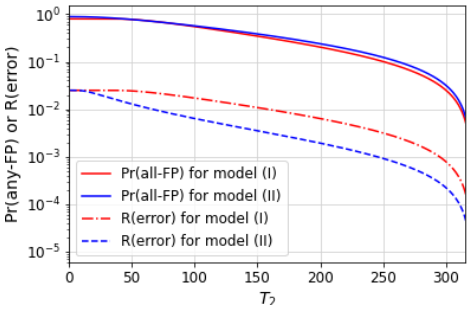



Fig. 13. System FP probability and FP error rate.

Fig. 13 depicts $\mathbb{P}$(any-FP) and $R$(error) for parameters $F = 64$, $k_f = 6$, $k_m = 4$, $M = 5$, $q = 1/8$, and $p = 0.8$.

## APPENDIX C
## GENERAL (OVERLAPPING) CASE

### A. Proof of Lemma 7

*Proof:* For a non-overlapped non-infected member $v$ that belongs to only one community, the probability that $v$ is

misidentified as infected is $1-(1-pq)^{c-1}$. For an overlapped non-infected member $v$ that belongs to more than one community, the probability that $v$ is misidentified as infected is $1-(1-pq)^{2(c-1)}$. Note that we assume the decoding of $\mathbf{G}_1$ has no errors, i.e., it identifies all non-infected outer sets correctly. Then for the pairwise overlap structure in the example, the infection status of all non-overlapped communities and non-overlapped parts are identified correctly. The COMP decoding of $\mathbf{G}_2$ has no FNs. The expected total number of FPs $N_0$ can be obtained as $N_0 \leq \left(1-(1-pq)^{c-1}\right)\cdot N_1 + \left(1-(1-pq)^{2(c-1)}\right)\cdot N_2$, where the inequality is because the RHS have not used the testing resluts of $\mathbf{G}_1$, $N_1$ and $N_2$ are the expected number of non-overlapped and overlapped members in infected communities, respectively. We can calculate $N_1$ as follows,

$$N_1 = (F-2F_o)q(1-p)M + 2F_o q(1-p)(M-M_o), \tag{C.1}$$

where $(F-2F_o)q$ is the expected number of infected non-overlapped communities, $(1-p)M$ is the expected number of non-infected members in each infected non-overlapped community, $2F_o q$ is the expected number of infected overlapped communities, and $(1-p)(M-M_o)$ is the expected number of non-infected members in each infected overlapped community. Similarly, $N_2$ can be calculated as

$$N_2 = F_o\left(1-(1-q)^2\right)(1-p)M_o, \tag{C.2}$$

where $F_o\left(1-(1-q)^2\right)$ is the expected number of overlaps, and $(1-p)M_o$ is the expected number of non-infected members in each overlapped part. □

## APPENDIX D
## LOOPY BELIEF PROPAGATION ALGORITHM

### A. LBP: Message Passing Rules

We here describe our loopy belief propagation algorithm (LBP) and update rules for our probabilistic model (II). We use the factor graph framework of [54] and derive closed-form expressions for the sum-product update rules (see equations (5) and (6) in [54]).

The LBP algorithm on a factor graph iteratively exchanges messages across the variable and factor nodes. The messages to and from a variable node $X_e$ or $U_v$ are *beliefs* about the variable or distributions (a local estimate of $\mathbb{P}(X_e|\text{observations})$ or $\mathbb{P}(U_v|\text{observations})$). Since all the random variables are binary, in our case each message would be a 2-dimensional vector $[a,b]$ where $a,b\geq 0$. Suppose the result of each test is $y_t$, i.e., $Y_t = y_t$ and we wish to compute the marginals $\mathbb{P}(X_e = x|Y_1 = y_1, Y_2 = y_2, \ldots, Y_T = y_T)$ and $\mathbb{P}(U_v = u|Y_1 = y_1, Y_2 = y_2, \ldots, Y_T = y_T)$ for $x,u\in\{0,1\}$. The LBP algorithm proceeds as follows:

1) *Initialization:* The variable nodes $X_e$ and $U_v$ transmit the message $[0.5, 0.5]$ on each of their incident edges. Each variable node $Y_\tau$ transmits the message $[1-y_\tau, y_\tau]$, where $y_\tau$ is the observed test result, on its incident edge.
2) *Factor node messages:* Each factor node receives the messages from the neighboring variable nodes and computes a new set of messages to send on each incident edge. The rules on how to compute these messages are described next.
3) *Iteration and completion.* The algorithm alternates between steps 2 and 3 above a fixed number of times (in practice 10 or 20 times works well) and computes an estimate of the posterior marginals as follows – for each variable node $X_e$ and $U_v$, we take the coordinatewise product of the incoming factor messages and normalize to obtain an estimate of $\mathbb{P}(X_e = x|y_1\ldots y_T)$ and $\mathbb{P}(U_v = u|y_1\ldots y_T)$ for $x,u\in\{0,1\}$.

Next we describe the simplified variable and factor node message update rules. We use equations (5) and (6) of [54] to compute the messages.

*Leaf node messages:* At every iteration, the variable node $Y_\tau$ continually transmits the message $[0,1]$ if $Y_\tau = 1$ and $[1,0]$ if $Y_\tau = 0$ on its incident edge. The factor node $\mathbb{P}(X_e)$ continually transmits $[1-q, q]$ on its incident edge; see Fig. 14 (a) and (b).

*Variable node messages:* The other variable nodes $X_e$ and $U_v$ use the following rule to transmit messages along the incident edges: for incident each edge $e$, a variable node takes the elementwise product of the messages from every other incident edge $e'$ and transmits this along $e$; see Fig. 14 (c).

*Factor node messages:* For the factor node messages, we calculate closed form expressions for the sum-product update rule (equation (6) in [54]). The simplified expressions are summarized in Fig. 14 (d) and (e). Next we briefly describe these calculations.

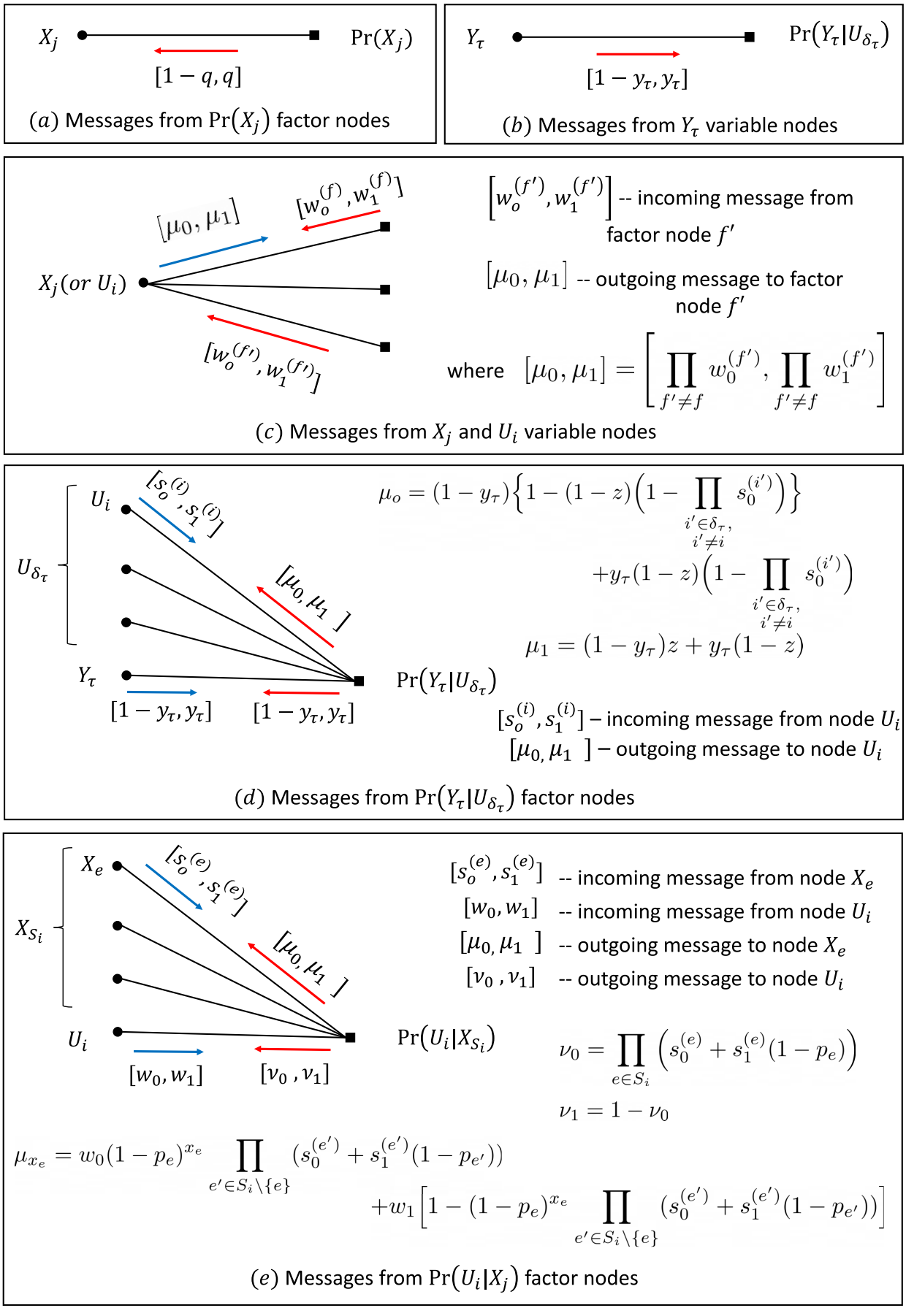
Firstly, we note that each message represents a probability distribution. One could, without loss of generality, normalize each message before transmission. Therefore, we assume that each message $\mu = [a,b]$ is such that $a+b=1$. Now, the the leaf nodes labeled $\mathbb{P}(V_j)$ perennially transmit the prior distribution corresponding to $V_j$.

Next, consider the factor node $\mathbb{P}(U_i|X_{S_i})$ as shown in Fig. 14 (e). The message sent to $U_i$ is calculated as

$$\begin{aligned}\nu_0 &= \sum_{\{x_e\in\{0,1\}:e\in S_i\}} \mathbb{P}(U_i = 0|X_{S_i} = x_{S_i}) \prod_{e\in S_i} s^{(e)}_{x_e}\\ &= \sum_{\{x_e\in\{0,1\}:e\in S_i\}} \prod_{e\in S_i} (s^{(e)}_{x_e}(1-p_e)^{x_e})\\ &= \prod_{e\in S_i}(s^{(e)}_0 + s^{(e)}_1(1-p_e)).\end{aligned}$$

Similarly, $\nu_1$ can be computed to be $\nu_1 = 1-\nu_0$. Now, the message sent to each $X_e$ is

$$\begin{aligned}\mu_{x_e} &= \sum_{\substack{u\in 0,1,\\ \{x_{e'}\in\{0,1\}:e'\in S_i\setminus\{e\}\}}} \mathbb{P}(U_i = u|X_{S_i} = x_{S_i}) w_u \prod_{e'\in S_i\setminus\{e\}} s^{(e')}_{x_{e'}}\\ &= \sum_{\{x_{e'}\in\{0,1\}:e'\in S_i\setminus\{e\}\}} \Big(\prod_{e'\in S_i\setminus\{e\}} s^{(e')}_{x_{e'}}\Big)\\ &\quad\cdot\Big(w_0\prod_{e'\in S_i}(1-p_{e'})^{x_{e'}} + w_1(1-\prod_{e'\in S_i}(1-p_{e'})^{x_{e'}})\Big)\\ &= w_0(1-p_e)^{x_e}\prod_{e'\neq e}(s^{(e')}_0 + s^{(e')}_1(1-p_{e'})) +\\ &\quad w_1\Big[1-(1-p_e)^{x_e}\prod_{e'\neq e}(s^{(e')}_0 + s^{(e')}_1(1-p'_e)\Big].\end{aligned}$$




(a) Messages from $\Pr(X_j)$ factor nodes

(b) Messages from $Y_\tau$ variable nodes

(c) Messages from $X_j$ and $U_i$ variable nodes

(d) Messages from $\Pr(Y_\tau|U_{\delta_\tau})$ factor nodes

(e) Messages from $\Pr(U_i|X_j)$ factor nodes

Fig. 14. The update rules for the factor and variable node messages.

Finally for the factor nodes $\mathbb{P}(Y_\tau|U_{\delta_\tau})$ as shown in Fig. 14 (d), note that the messages to $Y_\tau$ play no role since they are never used to recompute the variable messages. The messages to $U_i$ nodes are expressed as

$$\begin{aligned}\mu_u = &\sum_{\substack{y\in\{0,1\},\\ \{u_{i'}\in\{0,1\}:i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = y|U_{\delta_\tau} = u_{\delta_\tau}) \\ &\qquad \cdot (1-y_\tau)^{1-y} y_\tau^{y} \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big) \\ = &(1-y_\tau) \\ &\cdot \sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big) \\ &+ y_\tau \\ &\cdot \sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 1|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big).\end{aligned}$$

From our Z-channel model, recall that $\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) = 1$ if $u_i = 0\ \forall\ i \in \delta_\tau$ and $\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) = z$ otherwise. Thus we split the summation terms into 2 cases – one where $u_{i'} = 0$ for all $i'$ and the other its complement. Also combining this with the assumption that the messages are normalized, i.e., $s_0^{(i)} + s_1^{(i)} = 1$, we get

$$\begin{aligned}&\sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 0|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big) \\ &\qquad = \mathbb{1}_{u=1} z + \mathbb{1}_{u=0}\Big\{1 - (1-z)\Big(1 - \prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')}\Big)\Big\},\end{aligned}$$

and

$$\begin{aligned}&\sum_{\substack{\{u_{i'}\in\{0,1\}:\\ i'\in\delta_\tau\setminus\{i\}\}}} \Big(\mathbb{P}(Y_\tau = 1|U_{\delta_\tau} = u_{\delta_\tau}) \prod_{i'\in\delta_\tau\setminus\{i\}} s_{u_{i'}}^{(i')}\Big) \\ &\qquad = \mathbb{1}_{u=1}(1-z) + \mathbb{1}_{u=0}\Big((1-z)\Big(1 - \prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')}\Big)\Big).\end{aligned}$$

Substituting $u = 0$, and $u = 1$ we obtain the messages

$$\begin{aligned}\mu_0 &= (1-y_\tau)\Big\{1 - (1-z)\Big(1 - \prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')}\Big)\Big\} \\ &\quad + y_\tau(1-z)\Big(1 - \prod_{\substack{i'\in\delta_\tau\\ i'\neq i}} s_0^{(i')}\Big), \\ \mu_1 &= (1-y_\tau)z + y_\tau(1-z).\end{aligned}$$

For our probabilistic model, the complexity of computing the factor node messages increases only linearly with the factor node degree.

## ACKNOWLEDGMENT

The authors would like to thank Katerina Argyraki for her ongoing support and the interesting discussions about this project. Finally, they would also like to thank their anonymous reviewers for their thoughtful suggestions and valuable feedback.

## REFERENCES

[1] P. Nikolopoulos, S. R. Srinivasavaradhan, T. Guo, C. Fragouli, and S. Diggavi, "Group testing for connected communities," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, vol. 130, 2021.

[2] P. Nikolopoulos, S. R. Srinivasavaradhan, T. Guo, C. Fragouli, and S. Diggavi, "Group testing for overlapping communities," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–7.

[3] M. Broadfoot. (May 2020). *Coronavirus Test Shortages Trigger a New Strategy: Group Screening*. [Online]. Available: https://www.scientificamerican.com/article/coronavirus-test-shortages-trigger-a-new-strategy-group-screening2/

[4] J. Ellenberg, "Five people. One test. This is how you get there," New York Times, May 2020. [Online]. Available: https://www.nytimes.com/2020/05/07/opinion/coronavirus-group-testing.html

[5] C. M. Verdun et al., "Group testing for SARS-CoV-2 allows for up to 10-fold efficiency increase across realistic scenarios and testing strategies," *Frontiers Public Health*, vol. 9, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpubh.2021.583377, doi: 10.3389/fpubh.2021.583377.

[6] S. Ghosh et al., "Tapestry: A single-round smart pooling technique for COVID-19 testing," medRxiv, Cold Spring Harbor Lab. Press, 2020. [Online]. Available: https://www.medrxiv.org/content/early/2020/05/02/2020.04.23.20077727, doi: 10.1101/2020.04.23.20077727.

[7] L. M. Kucirka, S. A. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, "Variation in false-negative rate of reverse transcriptase polymerase chain reaction–based SARS-CoV-2 tests by time since exposure," *Ann. Internal Med.*, vol. 173, no. 4, pp. 262–267, Aug. 2020.

[8] S. Mallapaty. (2020). *The Mathematical Strategy That Could Transform Coronavirus Testing*. [Online]. Available: https://www.nature.com/articles/d41586-020-02053-6

[9] FDA. (2020). *Pooled Sample Testing and Screening Testing for COVID-19*. [Online]. Available: https://www.fda.gov/medical-devices/coronavirus-covid-19-and-medical-devices/pooled-sample-testing-and-screening-testing-covid-19

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

[11] 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, doi: 10.1561/0100000099.

[12] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications* (Applied Mathematics). Singapore: World Scientific, 1993. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/1936, doi: 10.1142/1936.

[13] Y. Malinovsky and P. S. Albert, "Revisiting nested group testing procedures: New results, comparisons, and robustness," 2016, *arXiv:1608.06330*.

[14] C. Troncoso et al., "Decentralized privacy-preserving proximity tracing," 2020, *arXiv:2005.12273*.

[15] S. Azad and S. Devi, "Tracking the spread of COVID-19 in India via social networks in the early phase of the pandemic," *J. Travel Med.*, vol. 27, no. 8, pp. 1–9, Dec. 2020.

[16] A. Aktay et al., "Google COVID-19 community mobility reports: Anonymization process description," 2020, *arXiv:2004.04145*.

[17] L. Riccio and C. J. Colbourn, "Sharper bounds in adaptive group testing," *Taiwanese J. Math.*, vol. 4, no. 4, pp. 669–673, Dec. 2000.

[18] M. C. Hu, F. K. Hwang, and J. K. Wang, "A boundary problem for group testing," *SIAM J. Algebr. Discrete Methods*, vol. 2, no. 2, pp. 81–87, Jun. 1981.

[19] P. Ungar, "The cutoff point for group testing," *Commun. Pure Appl. Math.*, vol. 13, pp. 49–54, Feb. 1960.

[20] 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.

[21] O. Johnson, "Strong converses for group testing from finite blocklength results," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5923–5933, Sep. 2017.

[22] M. Sobel and P. A. Groll, "Group testing to eliminate efficiently all defectives in a binomial sample," *Bell Labs Tech. J.*, vol. 38, no. 5, pp. 1179–1252, Sep. 1959.

[23] 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.

[24] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Optimal group testing," in *Proc. 33rd Conf. Learn. Theory*, vol. 125, J. Abernethy and S. Agarwal, Eds. Jul. 2020, pp. 1374–1388.
[25] W. H. Bay, E. Price, and J. Scarlett, "Optimal non-adaptive probabilistic group testing in general sparsity regimes," 2020, *arXiv:2006.01325*.
[26] P. Nikolopoulos, T. Guo, S. R. Srinivasavaradhan, C. Fragouli, and S. Diggavi, "Community aware group testing," 2020, *arXiv:2007.08111*.
[27] P. Nikolopoulos, S. R. Srinivasavaradhan, T. Guo, C. Fragouli, and S. Diggavi, "Group testing for overlapping communities," 2020, *arXiv:2012.02804*.
[28] J. Zhu, K. Rivera, and D. Baron, "Noisy pooled PCR for virus testing," 2020, *arXiv:2004.02689*.
[29] R. Goenka, S.-J. Cao, C.-W. Wong, A. Rajwade, and D. Baron, "Contact tracing enhances the efficiency of COVID-19 group testing," 2020, *arXiv:2011.14186*.
[30] S. Ahn, W.-N. Chen, and A. Ozgur, "Adaptive group testing on networks with community structure: The stochastic block model," 2021, *arXiv:2101.02405*.
[31] B. Arasli and S. Ulukus, "Group testing with a graph infection spread model," 2021, *arXiv:2101.05792*.
[32] P. Bertolotti and A. Jadbabaie, "Network group testing," 2020, *arXiv:2012.02847*.
[33] S. R. Srinivasavaradhan, P. Nikolopoulos, C. Fragouli, and S. Diggavi, "An entropy reduction approach to continual testing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 611–616.
[34] S. R. Srinivasavaradhan, P. Nikolopoulos, C. Fragouli, and S. Diggavi, "Dynamic group testing to control and monitor disease progression in a population," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2022, pp. 2255–2260.
[35] S. R. Srinivasavaradhan, P. Nikolopoulos, C. Fragouli, and S. Diggavi, "Improving group testing via gradient descent," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2022, pp. 2243–2248.
[36] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama, "Graph-constrained group testing," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 248–262, Jan. 2012.
[37] B. Spang and M. Wootters, "Unconstraining graph-constrained group testing," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)* (Leibniz International Proceedings in Informatics), vol. 145. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 46:1–46:20. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/11261, doi: 10.4230/LIPIcs.APPROX-RANDOM.2019.46.
[38] A. Karbasi and M. Zadimoghaddam, "Sequential group testing with graph constraints," in *Proc. IEEE Inf. Theory Workshop*, Sep. 2012, pp. 292–296.
[39] S. Luo, Y. Matsuura, Y. Miao, and M. Shigeno, "Non-adaptive group testing on graphs with connectivity," *J. Combinat. Optim.*, vol. 38, no. 1, pp. 278–291, Jul. 2019.
[40] V. Gandikota, E. Grigorescu, S. Jaggi, and S. Zhou, "Nearly optimal sparse group testing," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2760–2773, May 2019.
[41] T. Li, C. L. Chan, W. Huang, T. Kaced, and S. Jaggi, "Group testing with prior statistics," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 2346–2350.
[42] T. Kealy, O. Johnson, and R. Piechocki, "The capacity of non-identical adaptive group testing," in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2014, pp. 101–108.
[43] R. Gabrys et al., "AC–DC: Amplification curve diagnostics for COVID-19 group testing," 2021, *arXiv:2011.05223*.
[44] J. Scarlett, "Noisy adaptive group testing: Bounds and algorithms," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3646–3661, Jun. 2019.
[45] J. Scarlett, "An efficient algorithm for capacity-approaching noisy adaptive group testing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 2679–2683.
[46] G. Atia and V. Saligrama, "Noisy group testing: An information theoretic perspective," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2009, pp. 355–362.
[47] X. Cheng, S. Jaggi, and Q. Zhou, "Generalized group testing," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, vol. 151, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds. Mar. 2022, pp. 10777–10835. [Online]. Available: https://proceedings.mlr.press/v151/cheng22a.html
[48] M. Mézard, M. Tarzia, and C. Toninelli, "Group testing with random pools: Phase transitions and optimal strategy," *J. Stat. Phys.*, vol. 131, no. 5, pp. 783–801, 2008, doi: 10.1007/s10955-008-9528-9.
[49] D. Sejdinovic and O. Johnson, "Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction," 2010, *arXiv:1010.2441*.
[50] R. K. Saiki et al., "Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia," *Science*, vol. 230, pp. 1350–1354, Dec. 1985.
[51] L. Baldassini, O. Johnson, and M. Aldridge, "The capacity of adaptive group testing," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 2676–2680.
[52] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 4, pp. 363–377, Oct. 1964.
[53] 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.
[54] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.
[55] M. Cuturi, O. Teboul, Q. Berthet, A. Doucet, and J.-P. Vert, "Noisy adaptive group testing using Bayesian sequential experimental design," 2020, *arXiv:2004.12508*.
[56] S. D. Walter, S. W. Hildreth, and B. J. Beaty, "Estimation of infection rates in populations of organisms using pools of variable size," *Amer. J. Epidemiol.*, vol. 112, no. 1, pp. 124–128, 1980.
[57] M. Sobel and R. M. Elashoff, "Group testing with a new goal, estimation," *Biometrika*, vol. 62, no. 1, pp. 181–193, 1975.

**Pavlos Nikolopoulos** (Member, IEEE) received the M.Sc. degree from the University of Athens, Greece, and the Ph.D. degree from the École Polytechnique Fédérale de Lausanne (EPFL), under the supervision of Prof. Katerina Argyraki. He is currently a Post-Doctoral Researcher with the Network Architecture Laboratory (NAL), EPFL, and the Algorithmic Research in Network Information Flow Laboratory (ARNI), University of California at Los Angeles (UCLA). Before that, he had worked for several years as a Communications Engineer with the Greek Air Force. For the past two years, he has also worked on information theory and group testing. His research interests include intersection of network systems design and statistics, with a focus on network measurements, neutrality, and transparency.

**Sundara Rajan Srinivasavaradhan** received the B.Tech. and M.Tech. degrees in electrical engineering from the Indian Institute of Technology, Madras, India, in 2016, and the Ph.D. degree from the ECE Department, University of California at Los Angeles, Los Angeles, CA, USA. He is currently a Research Scientist with Meta working on recommendation systems. He was previously a Research Intern with Edwards Lifesciences and Microsoft Research. His research interests include statistics, machine learning, and discrete mathematics.

**Tao Guo** (Member, IEEE) received the B.S. degree in telecommunications engineering from Xidian University, Xi'an, China, in 2013, and the Ph.D. degree from the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China, in 2018. He was a Visiting Scholar with the Technical University of Munich from April 2016 to September 2016. From December 2018 to January 2021, he was a Post-Doctoral Researcher with Texas A&M University and the University of California at Los Angeles, successively. Then, he worked with Huawei Theory Lab until August 2022. After that, he joined the School of Cyber Science and Engineering, Southeast University, Nanjing, China, where he is currently an Associate Professor. His research interests include network information theory, information theory security, privacy protection, and semantic communication.

**Christina Fragouli** (Fellow, IEEE) received the B.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles (UCLA). She has worked with the Information Sciences Center, AT&T Labs, Florham Park NJ, USA, and the National University of Athens. She also visited Bell Laboratories, Murray Hill, NJ, USA, and DIMACS, Rutgers University. From 2006 to 2015, she was an Assistant Professor and an Associate Professor with the School of Computer and Communication Sciences, EPFL, Switzerland. She is currently a Professor with the Electrical and Computer Engineering Department, UCLA. Her research interests include coding theory, algorithms for networking, and network security. She has served as the President for the IEEE Information Theory Society in 2022, as an Information Theory Society Distinguished Lecturer, and as an Associate Editor for IEEE COMMUNICATIONS LETTERS, *Computer Communications* (Elsevier), IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION THEORY, and IEEE TRANSACTIONS ON MOBILE COMMUNICATIONS. She has also served in several IEEE committees, and received awards for her work.

**Suhas N. Diggavi** (Fellow, IEEE) received the bachelor's degree from IIT, Delhi, and the Ph.D. degree from Stanford University.

He has worked as a Principal Member Research Staff with AT&T Shannon Laboratories and directed the Laboratory for Information and Communication Systems (LICOS), EPFL. At UCLA, he directs the Information Theory and Systems Laboratory. He is currently a Professor of electrical and computer engineering with UCLA. He has eight issued patents. His research interests include information theory and its applications to several areas, including machine learning, security and privacy, wireless networks, data compression, cyber-physical systems, bio-informatics, and neuroscience.

Dr. Diggavi was selected as a Guggenheim Fellow in 2021. He has received several recognitions for his research from IEEE and ACM, including the 2013 IEEE Information Theory Society & Communications Society Joint Paper Award, the 2021 ACM Conference on Computer and Communications Security (CCS) Best Paper Award, the 2013 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) Best Paper Award, and the 2006 IEEE Donald Fink Prize Paper Award, among others. He also received the 2019 Google Faculty Research Award, the 2020 Amazon Faculty Research Award, and the 2021 Facebook/Meta Faculty Research Award. He served on the Board of Governors for the IEEE Information Theory Society (2016–2021). He has also helped organize IEEE and ACM conferences, including serving as the Technical Program Co-Chair for the 2012 IEEE Information Theory Workshop (ITW), the Technical Program Co-Chair for the 2015 IEEE International Symposium on Information Theory (ISIT), and the General Co-Chair for ACM Mobihoc 2018. He has been an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY, ACM/IEEE TRANSACTIONS ON NETWORKING, and other journals and special issues, as well as in the program committees of several IEEE conferences. He served as an IEEE Distinguished Lecturer. More information can be found at: http://licos.ee.ucla.edu.